

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan pembahasan pada bab-bab sebelumnya, maka dapat ditarik beberapa kesimpulan yaitu :

1. Pembangunan perangkat lunak Sistem Pendukung Keputusan berdasarkan Siklus Hidup Produk telah menghasilkan suatu perangkat lunak yang handal yang dapat menentukan keberadaan suatu produk dalam siklus hidupnya dan strategi pemasaran yang dapat dilaksanakan berdasarkan Siklus Hidup Produk-nya tersebut serta dapat menentukan media periklanan yang sesuai dengan segmen target dari pemasaran.
2. Walaupun masih dalam taraf sederhana, namun aplikasi Sistem Pendukung Keputusan ini berhasil dibangun sesuai dengan tujuan semula yakni membangun sebuah perangkat lunak yang mampu membantu dalam melakukan pengelolaan bauran media periklanan, data promosi dan penawaran keputusan terhadap strategi pemasaran yang tepat.

5.2. Saran

Beberapa saran yang dapat diambil dari proses analisis sampai pada pembuatan tugas akhir ini adalah sebagai berikut :

1. Perangkat lunak diharapkan dapat dikembangkan lebih lanjut dengan mengintegrasikan Sistem Pendukung Keputusan dengan *data warehouse* yang memiliki informasi mengenai data *customer*,

sehingga strategi pemasaran dapat lebih efektif untuk membangun *brand loyalty* konsumen.

2. Perangkat lunak diharapkan dapat dikembangkan lebih lanjut dengan mengembangkan proses pemasaran secara terintegrasi yang mencakup proses pemasaran secara keseluruhan.
3. Perangkat lunak diharapkan dapat dikembangkan lebih lanjut dengan penambahan fungsionalitas dan interface yang lebih menarik untuk pelayanan agar dapat memberikan kenyamanan terikat dengan fasilitas yang diberikan.

DAFTAR PUSTAKA

- O'Brien, J., 2001, *Management Information Systems Fourth Edition*, Mc Graw Hill, New York.
- O'Brien, J., Marakas, G., 2005, *Management Information Systems Seventh Edition*, Mc Graw Hill, New York.
- Bennet, S., McRobb, S. et.al, *Object-Oriented System Analysis and Design Using UML*, Mc Graw Hill Companies, 2002.
- Ferrell et.al., 2004, *Marketing Strategy Second Edition*, South-Western, Ohio.
- Philip Kotler; alih bahasa, Hendra, T. Ronny, A. et.al, 2002, *Manajemen Pemasaran 2 Edisi 10*, Prenhallindo, Jakarta.
- Susanto, A., 2001, *Manajemen Pemasaran di Indonesia Jil. 2*, Salemba Empat, Jakarta.
- [http://en.wikipedia.org/wiki/Product life cycle](http://en.wikipedia.org/wiki/Product_life_cycle)
- [http://www.trumpuniversity.com/learn/Product life cycle](http://www.trumpuniversity.com/learn/Product_life_cycle)

serviens in lumine veritatis

LAMPIRAN




SKPL

**Sistem Pendukung Keputusan Promosi
Produk dan Layanan Telkom Berbasis
Web berdasarkan Siklus Hidup Produk**

Dipersiapkan oleh:

Adio Gustiasyah 03.07.03655

**Program Studi Teknik Informatika
Fakultas Teknologi Industri
Universitas Atma Jaya Yogyakarta**

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		SKPL-DSS		
				<i>Tgl: 15-10-2007</i>

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	Pembagian class diagram periklanan
B	
C	
D	

Indeks TGL	-	A	B	C	D
Ditulis oleh	Adio	Adio	Adio	Adio	
Diperiksa oleh		PM IW	PM IW	PM IW	
Disetujui oleh		PM IW	PM IW	PM IW	

Daftar Isi

Daftar Perubahan	2
Daftar Isi	3
1. Pendahuluan	7
1.1. Tujuan	7
1.2. Lingkup Masalah	8
1.3. Definisi Akronim dan Singkatan	10
1.4. Referensi	10
1.5. Deskripsi umum (Overview)	11
2. Deskripsi Kebutuhan	11
2.1. Perspektif Produk	11
2.2. Antarmuka Sistem	13
2.3. Kebutuhan Fungsionalitas Perangkat Lunak	14
2.3.1. Package Dependencies.....	14
2.3.1.1. Use Case for Package Account.....	14
2.3.1.2. Use Case for Package Elektronik.....	15
2.3.1.3. Use Case for Package Media Cetak Indoor.....	15
2.3.1.4. Use Case for Package Media Cetak Outdoor.....	16
2.3.1.5. Use Case for Package Produk.....	16
2.3.1.6. Use Case for Package Promosi	16
2.3.1.7. Use Case for Package DSS	16
2.3.2. Fungsi Produk	17
2.3.2.1. Use Case Login	17
2.3.2.2. Use Case Pengelolaan Data Operator	17
2.3.2.3. Use Case Ubah Password	17
2.3.2.4. Use Case Pengelolaan Periklanan Internet	17
2.3.2.5. Use Case Pengelolaan Periklanan Radio	18
2.3.2.6. Use Case Pengelolaan Periklanan Televisi	18
2.3.2.7. Use Case Pengelolaan Periklanan Banner	18
2.3.2.8. Use Case Pengelolaan Periklanan Buku	19
2.3.2.9. Use Case Pengelolaan Periklanan Brosur	19
2.3.2.10. Use Case Pengelolaan Periklanan Majalah	19
2.3.2.11. Use Case Pengelolaan Periklanan Koran	20
2.3.2.12. Use Case Pengelolaan Periklanan Poster	20
2.3.2.13. Use Case Pengelolaan Periklanan Billboard	21
2.3.2.14. Use Case Pengelolaan Periklanan Branding	21
2.3.2.15. Use Case Pengelolaan Periklanan Neon	21
2.3.2.16. Use Case Pengelolaan Periklanan Spanduk	22
2.3.2.17. Use Case Pengelolaan Periklanan Umbul	22
2.3.2.18. Use Case Pengelolaan Data Produk	23
2.3.2.19. Use Case Pengelolaan Data Promosi	23
2.3.2.20. Use Case SPK Promosi Periklanan	23
2.4. Karakteristik Pengguna	23
3. Deskripsi Rinci Kebutuhan	24
3.1. Spesifikasi Kebutuhan Fungsionalitas	24
3.1.1. Use Case Login	24
3.1.2. Use Case Pengelolaan Data Operator	25
3.1.3. Use Case Ubah Password	27
3.1.4. Use Case Pengelolaan Periklanan Internet	28
3.1.5. Use Case Pengelolaan Periklanan Radio	30
3.1.6. Use Case Pengelolaan Periklanan Televisi	31
3.1.7. Use Case Pengelolaan Periklanan Banner	33
3.1.8. Use Case Pengelolaan Periklanan Buku	35

3.1.9.	Use Case Pengelolaan Periklanan Brosur	37
3.1.10.	Use Case Pengelolaan Periklanan Majalah	39
3.1.11.	Use Case Pengelolaan Periklanan Koran	41
3.1.12.	Use Case Pengelolaan Periklanan Poster	43
3.1.13.	Use Case Pengelolaan Periklanan Billboard	46
3.1.14.	Use Case Pengelolaan Periklanan Branding	48
3.1.15.	Use Case Pengelolaan Periklanan Neon	50
3.1.16.	Use Case Pengelolaan Periklanan Spanduk	52
3.1.17.	Use Case Pengelolaan Periklanan Umbul	54
3.1.18.	Use Case Pengelolaan Data Produk	57
3.2.19.	Use Case Pengelolaan Data Promosi	58
3.2.20.	Use Case SPK Promosi Periklanan	59
3.2.	Spesifikasi Kebutuhan Non Fungsionalitas	61
3.2.1.	Antarmuka Eksternal	61
3.2.2.	Antarmuka Pemakai	61
3.2.3.	Antarmuka Perangkat Keras	61
3.2.4.	Antarmuka Perangkat Lunak	61
4.	Realisasi Use Case	62
4.1.	Static Structure Diagram	62
4.1.1.	Use Case Login	62
4.1.2.	Use Case Pengelolaan Data Operator	63
4.1.3.	Use Case Ubah Password	63
4.1.4.	Use Case Package Periklanan	64
4.1.5.	Use Case Package Elektronik	65
4.1.6.	Use Case Package MediaCetaIndoor	65
4.1.7.	Use Case Package MediaCetaOutdoor	66
4.1.8.	Use Case Pengelolaan Periklanan Internet	66
4.1.9.	Use Case Pengelolaan Periklanan Radio	67
4.1.10.	Use Case Pengelolaan Periklanan Televisi	67
4.1.11.	Use Case Pengelolaan Periklanan Banner	67
4.1.12.	Use Case Pengelolaan Periklanan Buku	68
4.1.13.	Use Case Pengelolaan Periklanan Brosur	68
4.1.14.	Use Case Pengelolaan Periklanan Majalah	68
4.1.15.	Use Case Pengelolaan Periklanan Koran	69
4.1.16.	Use Case Pengelolaan Periklanan Poster	69
4.1.17.	Use Case Pengelolaan Periklanan Billboard	69
4.1.18.	Use Case Pengelolaan Periklanan Branding	70
4.1.19.	Use Case Pengelolaan Periklanan Neon	70
4.1.20.	Use Case Pengelolaan Periklanan Spanduk	70
4.1.21.	Use Case Pengelolaan Periklanan Umbul	71
4.1.22.	Use Case Pengelolaan Data Produk	71
4.1.23.	Use Case Pengelolaan Data Promosi	71
4.1.24.	Use Case SPK Promosi Periklanan	72
4.2.	Interaction Diagram	72
4.2.1.	Use Case Pengelolaan Login	72
4.2.2.	Use Case Pengelolaan Data Operator	73
4.2.2.1.	Penambahan Data Operator	73
4.2.2.2.	Pencarian Data Operator	73
4.2.2.3.	Penghapusan Data Operator	74
4.2.3.	Use Case Ubah Password	74
4.2.4.	Use Case Pengelolaan Periklanan Internet	75
4.2.4.1.	Penambahan Data Internet	75
4.2.4.2.	Pengubahan Data Internet	75
4.2.4.3.	Pencarian Data Internet	76
4.2.4.4.	Penghapusan Data Operator	76

4.2.5. Use Case Pengelolaan Periklanan Radio.....	77
4.2.5.1. Penambahan Data Radio.....	77
4.2.5.2. Pengubahan Data Radio.....	77
4.2.5.3. Pencarian Data Radio.....	78
4.2.5.4. Penghapusan Data Radio.....	78
4.2.6. Use Case Pengelolaan Periklanan Televisi.....	79
4.2.6.1. Penambahan Data Televisi.....	79
4.2.6.2. Pengubahan Data Televisi.....	79
4.2.6.3. Pencarian Data Televisi.....	80
4.2.6.4. Penghapusan Data Televisi.....	80
4.2.7. Use Case Pengelolaan Periklanan Banner.....	81
4.2.7.1. Penambahan Data Banner.....	81
4.2.7.2. Pengubahan Data Banner.....	81
4.2.7.3. Pencarian Data Banner.....	82
4.2.7.4. Penghapusan Data Banner.....	82
4.2.8. Use Case Pengelolaan Periklanan Buku.....	83
4.2.8.1. Penambahan Data Buku.....	83
4.2.8.2. Pengubahan Data Buku.....	83
4.2.8.3. Pencarian Data Buku.....	84
4.2.8.4. Penghapusan Data Buku.....	84
4.2.9. Use Case Pengelolaan Periklanan Brosur.....	85
4.2.9.1. Penambahan Data Brosur.....	85
4.2.9.2. Pengubahan Data Brosur.....	85
4.2.9.3. Pencarian Data Brosur.....	86
4.2.9.4. Penghapusan Data Brosur.....	86
4.2.10. Use Case Pengelolaan Periklanan Majalah.....	87
4.2.10.1. Penambahan Data Majalah.....	87
4.2.10.2. Pengubahan Data Majalah.....	87
4.2.10.3. Pencarian Data Majalah.....	88
4.2.10.4. Penghapusan Data Majalah.....	88
4.2.11. Use Case Pengelolaan Periklanan Koran.....	89
4.2.11.1. Penambahan Data Koran.....	89
4.2.11.2. Pengubahan Data Koran.....	89
4.2.11.3. Pencarian Data Koran.....	90
4.2.11.4. Penghapusan Data Koran.....	90
4.2.12. Use Case Pengelolaan Periklanan Poster.....	91
4.2.12.1. Penambahan Data Poster.....	91
4.2.12.2. Pengubahan Data Poster.....	91
4.2.12.3. Pencarian Data Poster.....	92
4.2.12.4. Penghapusan Data Poster.....	92
4.2.13. Use Case Pengelolaan Periklanan Billboard.....	93
4.2.13.1. Penambahan Data Billboard.....	93
4.2.13.2. Pengubahan Data Billboard.....	93
4.2.13.3. Pencarian Data Billboard.....	94
4.2.13.4. Penghapusan Data Billboard.....	94
4.2.14. Use Case Pengelolaan Periklanan Branding.....	95
4.2.14.1. Penambahan Data Branding.....	95
4.2.14.2. Pengubahan Data Branding.....	95
4.2.14.3. Pencarian Data Branding.....	96
4.2.14.4. Penghapusan Data Branding.....	96
4.2.15. Use Case Pengelolaan Periklanan Neon.....	97
4.2.15.1. Penambahan Data Neon.....	97
4.2.15.2. Pengubahan Data Neon.....	97
4.2.15.3. Pencarian Data Neon.....	98
4.2.15.4. Penghapusan Data Neon.....	98

4.2.16. Use Case Pengelolaan Periklanan Spanduk.....	99
4.2.16.1. Penambahan Data Spanduk.....	99
4.2.16.2. Pengubahan Data Spanduk.....	99
4.2.16.3. Pencarian Data Spanduk.....	100
4.2.16.4. Penghapusan Data Spanduk.....	100
4.2.17. Use Case Pengelolaan Periklanan Umbul.....	101
4.2.17.1. Penambahan Data Umbul.....	101
4.2.17.2. Pengubahan Data Umbul.....	101
4.2.17.3. Pencarian Data Umbul.....	102
4.2.17.4. Penghapusan Data Umbul.....	102
4.2.18. Use Case Pengelolaan Data Produk.....	103
4.2.18.1. Penambahan Data Produk.....	103
4.2.18.2. Pengubahan Data Produk.....	103
4.2.18.3. Pencarian Data Produk.....	104
4.2.19. Use Case Pengelolaan Data Promosi.....	104
4.2.19.1. Penambahan Data Promosi.....	104
4.2.19.2. Pengubahan Data Promosi.....	105
4.2.19.3. Pencarian Data Promosi.....	105
4.2.20. Use Case SPK Promosi Periklanan.....	106
4.3. Persistence Data	107

Sistem Pendukung Keputusan Promosi Produk dan Layanan Telkom berbasis Web berdasarkan Siklus Hidup Produk

1 Pendahuluan

1.1 Tujuan

Dokumen ini berisi tentang penjelasan Spesifikasi Kebutuhan Perangkat Lunak (SKPL). Dokumen SKPL ini merupakan dokumen spesifikasi kebutuhan perangkat lunak untuk Sistem Pendukung Keputusan yang akan dibangun.

Dokumen ini digunakan oleh pengembang perangkat lunak sebagai acuan teknis untuk pengembangan perangkat lunak. Dalam SKPL ini akan dijelaskan mengenai kebutuhan-kebutuhan yang harus tersedia agar perangkat lunak yang diharapkan dapat terwujud.

1.2 Lingkup Masalah

Sejak tahun 2004, TELKOM telah menyediakan layanan akses pita lebar berbasis telepon "Speedy" yang merupakan teknologi dengan layanan internet access end to end dengan basis teknologi *Asymmetric Digital Subscriber Line (ADSL)*, yang dapat menyalurkan data dan suara secara simultan melalui satu saluran telepon biasa dengan kecepatan maksimal 384 kbps yang dijamin dari modem sampai BRAS (Broadband Remote Access Server). Dengan teknologi Speedy ini, saluran telepon dapat digunakan secara bersamaan dengan fasilitas akses Internet (Speedy) tanpa saling

mengganggu dengan kecepatan upstream/downstream 64K/384K disisi akses TELKOM ADSL.

Sampai dengan 31 Desember 2006, terdapat lebih dari 93.200 pelanggan di daerah tempat layanan tersebut tersedia, Speedy tersedia di seluruh Divisi I sampai VII. Pelanggan Speedy biasanya adalah pengguna dial-up rumahan dengan penggunaan bulanan lebih dari Rp 250.000, perusahaan skala kecil - menengah, agen perjalanan, kafe internet dan sekolah-sekolah.

Hadirnya teknologi Speedy ini memberikan respon yang sangat baik dari masyarakat, hal ini ditunjukkan dengan dengan semakin meningkatnya jumlah peminat Speedy untuk menggunakan layanan ini. Hal ini tentunya memberikan keuntungan tersendiri bagi pihak Telkom, teknologi Speedy yang ditawarkan dapat menjadi salah satu alternatif untuk mereka yang bertujuan untuk dapat menikmati akses internet dengan kehandalan dan kemudahan yang dimiliki Speedy.

Kemajuan tersebut hendaknya harus dapat dibarengi dengan efektifitas dari promosi jenis-jenis layanan yang disediakan. Sehingga masyarakat dapat menggunakan jenis layanan yang sering disebut dengan paket Speedy ini dengan tepat. Seperti contoh untuk penggunaan layanan pribadi dengan kebutuhan akan akses internet tidak terlalu signifikan, lebih efisien apabila menggunakan paket layanan personal. Maka dari itu promosi dari paket layanan yang disediakan akan sangat membantu baik itu dari sisi pengguna maupun penyedia. Selain itu, hal ini dapat meningkatkan efektifitas penyebarluasan produk layanan yang disediakan, yang tentunya akan memberikan keuntungan bagi pihak Telkom.

Untuk mendukung hal diatas, diperlukan manajemen yang baik terhadap promosi yang telah dilakukan serta adanya suatu sistem yang memberikan fasilitas untuk mendukung keputusan yang akan diambil dalam rangka menentukan strategi pemasaran maupun pemilihan media periklanan yang tepat untuk melakukan promosi. Selama ini, promosi hanya dilakukan menggunakan berkas dari berbagai catatan media promosi yang ada, berdasarkan berkas-berkas itulah pemilihan media promosi dilakukan. Dan apabila telah dilakukan kesepakatan promosi terhadap suatu media tertentu, maka hal ini akan dicatat lagi ke dalam suatu berkas promosi, yang menyebabkan tidak efektifnya manajemen promosi untuk produk Speedy ini. Selain itu, hal ini tentunya akan menyulitkan apabila kita ingin mengetahui promosi apa saja yang telah dilakukan dan kapan promosi itu berakhir, karena kita harus mencari berkas-berkas tersebut serta penyesuaian terhadap segmen target dari paket layanan yang akan dipromosikan akan menjadi lebih rumit. Sementara itu banyak pekerjaan lain yang harus dilakukan.

Maka dari itu diperlukan suatu sistem yang baik yang dapat menangani hal ini. Sistem Pendukung Keputusan mengenai proses promosi yang akan dikembangkan ini merupakan suatu solusi atas permasalahan yang dihadapi. Dengan adanya sistem ini, akan memberikan efisiensi dan efektifitas serta memberikan kemudahan karena semuanya dilakukan secara komputerisasi. Disamping itu, dengan dikembangkannya Sistem Pendukung Keputusan ini akan memeberikan keputusan yang baik dalam hal strategi promosi yang

akan dilaksanakan. Dengan adanya sistem ini, promotor dapat mengetahui sejauh mana tahap kesiapan pembeli terhadap produk yang akan ditawarkan, sehingga promotor dapat melakukan promosi sesuai dengan masing-masing tahap kesiapan pembeli tersebut. Selain itu informasi dari promosi paket layanan Speedy ini dapat diakses oleh banyak pengguna yang terhubung dengan jaringan lokal di PT. Telekomunikasi Indonesia Tbk. Kandatel Yogyakarta. Sehingga bagi mereka yang berkepentingan dapat mengetahui informasi ini tanpa harus mencari lembaran-lembaran media promosi yang ada dan promosi yang telah dijalankan.

1.3 Definisi Akronim dan Singkatan

Daftar definisi dan akronim yang digunakan :

Keyword atau Phrase	<i>Definisi</i>
SKPL	Dokumen yang berisi tentang spesifikasi kebutuhan pengembangan perangkat lunak.
ADSL	Asymmetric Digital Subscriber Line
DBMS	DataBase Management System atau pengelola manajemen data base
DSS	Decision Support System (Sistem Pendukung Keputusan)

TBL_SKPL_01. Tabel definisi akronim dan singkatan

1.4 Referensi

Dokumen yang digunakan sebagai acuan dalam rencana pengembangan perangkat lunak ini adalah :

1. *Software Engineering*, 1997, Roger S. Pressman, Mc Graw-Hill International Edition.
2. *Software Requirements*, 1993, Alan M. Davis, Prentice Hall, International Edition..
3. *Spesifikasi Kebutuhan Perangkat Lunak*, GLO1, Jurusan Teknik Informatika - UAJY
4. *Rekayasa Perangkat Lunak*, 1997, Presman Roger S., McGraw-Hill Book Co., Andi Yogyakarta.
5. *PRODUCT-KNOWLEDGE Speedy*, 2007, PT. Telekomunikasi Indonesia TBK.

1.5 Deskripsi umum (Overview)

Dokumen SKPL ini dibagi menjadi empat bab. Bab pertama adalah **Pendahuluan**, yang berisi tentang deskripsi dokumen. Bab kedua adalah **Deskripsi Keseluruhan**, yang berisi penjelasan secara umum mengenai system yang akan dikembangkan meliputi fungsi-fungsi dari sistem, karakteristik pengguna, batasan dan asumsi yang diambil dalam pengembangan perangkat lunak. Bab ketiga adalah **Spesifikasi Rinci Kebutuhan**, yang berisi penjelasan Tentang kebutuhan sistem yang akan dikembangkan secara lebih rinci. Bab keempat adalah **Realisasi Use Case**, yang berisi realisasi use case dalam tahap analisis (konseptual), yang akan digunakan sebagai dasar realisasi use case pada tahap desain.

2. Deskripsi Kebutuhan

2.1 Perspektif Produk

DSS Promosi Speedy adalah perangkat lunak yang dikembangkan untuk membantu berbagai proses yang berkaitan dengan pengelolaan promosi paket layanan

Speedy pada PT. Telekomunikasi Indonesia Tbk. Kandatel Yogyakarta. Perangkat lunak yang dikembangkan ini diharapkan mampu untuk menangani manajemen promosi baik itu untuk penyebaran media promosi maupun pengelolaan terhadap data-data promosi yang telah dilakukan serta mampu memberikan solusi terhadap strategi promosi apa yang efisien untuk dilaksanakan sesuai dengan tahap kesiapan pembeli dari produk yang ditawarkan. Selain itu juga, perangkat lunak ini diharapkan dapat membantu kinerja staff serta Memfasilitasi dengan *Information On-line*.

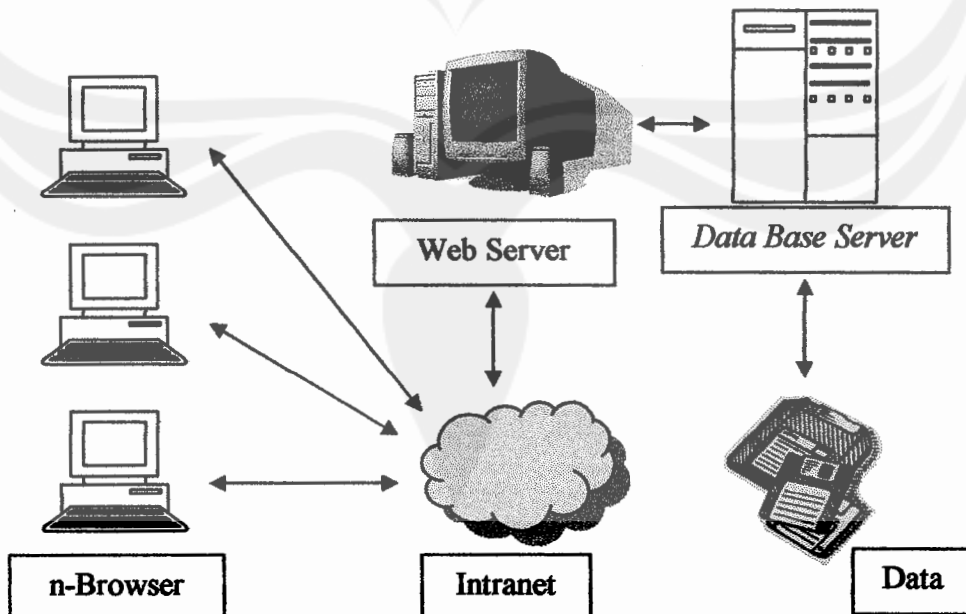
Information On-line ini dapat dilakukan oleh karyawan dengan berinteraksi terhadap sistem yang ditampilkan dalam bentuk halaman web. Secara garis besar, proses diawali dengan adanya request halaman web dari Client, yang diteruskan dari browser ke Web Server, kemudian membuat HTTP_request yang dikirimkan kepada web server tersebut. Web server mencari file yang berisi kumpulan instruksi tersebut. Web server mengenali tipe file yang di-request kemudian mengirimkannya kepada handler yang bertugas menangani tipe file tersebut, yang selanjutnya dikompilasi, kemudian hasil eksekusinya dikirimkan kembali pada browser sebagai response. Dimana dalam hal ini akan diberikan suatu layanan terhadap system akan informasi tentang penyebaran media promosi dan data-data promosi.

Perangkat lunak DSS Promosi Speedy akan dibangun menggunakan bahasa pemrograman ASP .Net(C#). Pengguna akan berinteraksi dengan perangkat lunak melalui

antarmuka berbasis GUI, sehingga piranti masukan yang digunakan adalah *mouse* dan *keyboard*.

2.2 Antarmuka Sistem

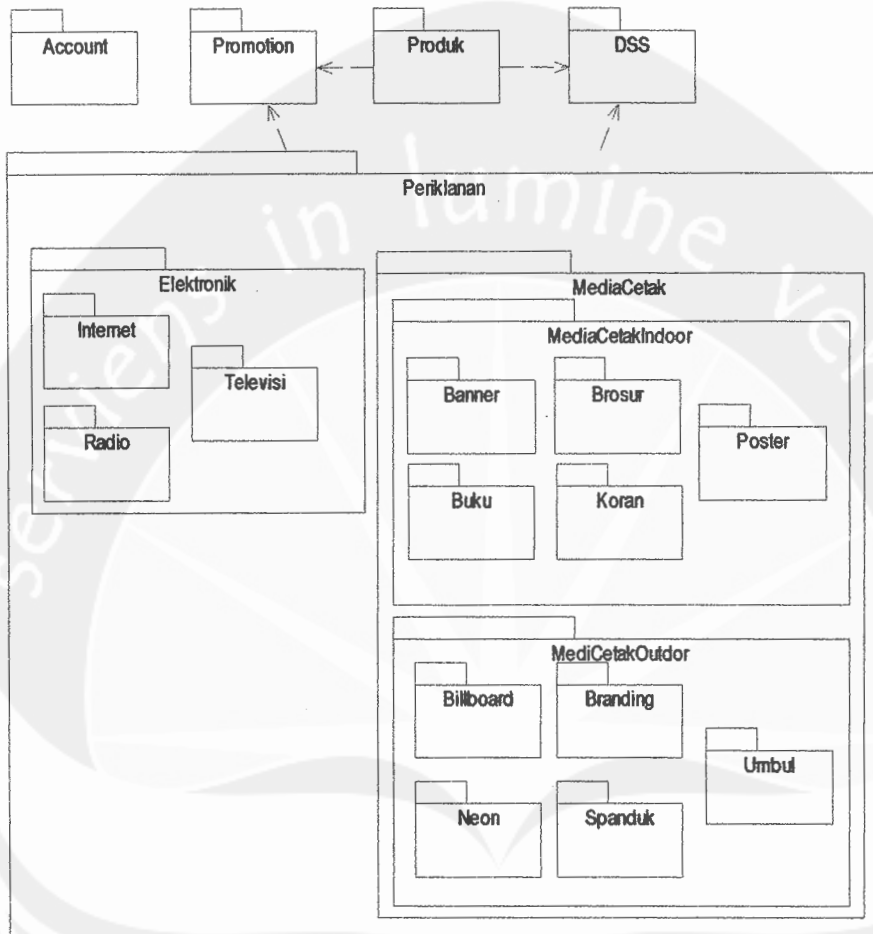
Perangkat lunak DSS Promosi Speedy ini memiliki antarmuka sistem yang berinteraksi berbasis *web*, yang sifatnya adalah menggunakan sebuah *web server* dan sebuah *database server* (*server* yang digunakan untuk penyimpanan data utama) yang dapat diakses oleh banyak *browser* menggunakan jaringan internet. Komputer-komputer *browser* nantinya dapat digunakan oleh user. Dengan demikian, proses *insert*, *update*, dan *delete* ataupun langkah-langkah yang diambil untuk melakukan pengelolaan data-data yang bersangkutan dari komputer *browser* akan berpengaruh terhadap data dalam *server database*, yang menggunakan konsep *object-relational* tersebut sesuai dengan pengelolaan yang dilakukan oleh user sistem.



Gambar 2.1. Arsitektur Aplikasi

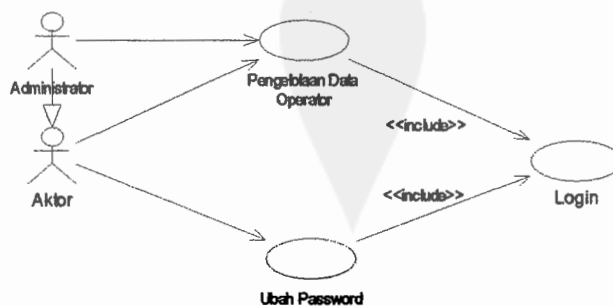
2.3 Kebutuhan Fungsionalitas Perangkat Lunak

2.3.1. Package Dependencies



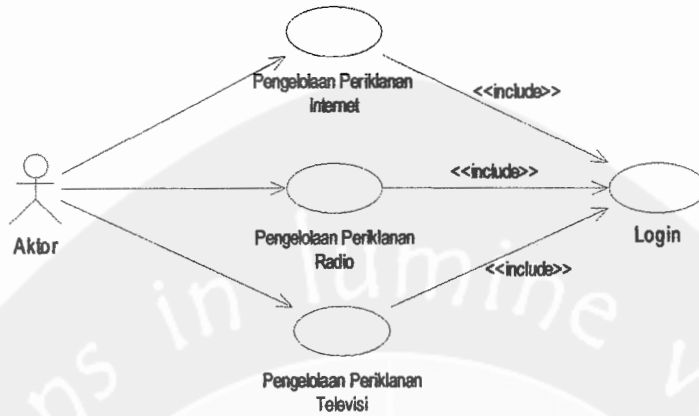
Gambar 2.2 Package Dependencies

2.3.1.1. Use Case for Package Account



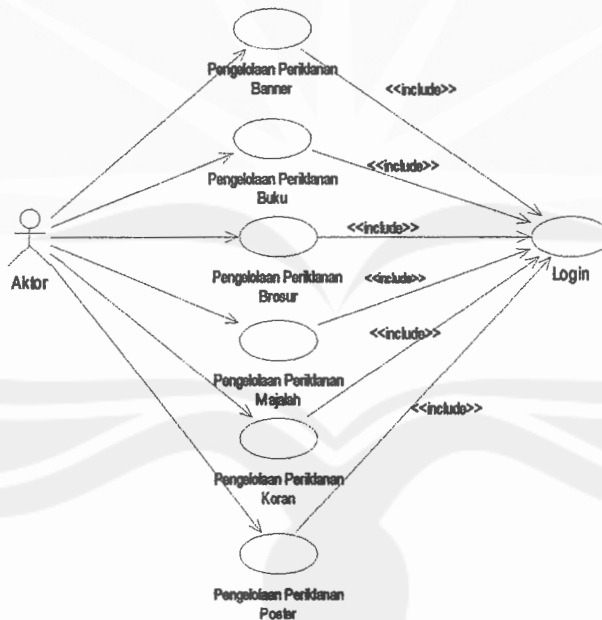
Gambar 2.3 Use Case Package Account

2.3.1.2. Use Case for Package Elektronik



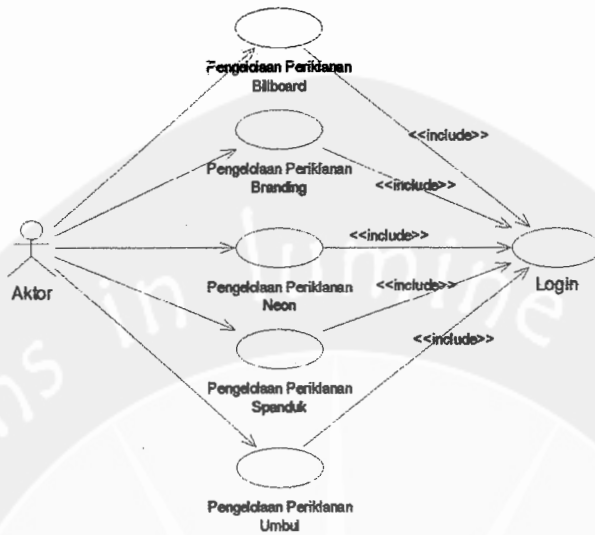
Gambar 2.4 Use Case Package Elektronik

2.3.1.3. Use Case for Package Media Cetak Indoor



Gambar 2.5 Use Case Package Media Cetak Indoor

2.3.1.4. Use Case for Package Media Cetak Outdoor



Gambar 2.6 Use Case Package Media Cetak Outdoor

2.3.1.5. Use Case for Package Produk



Gambar 2.7 Use Case Package Produk

2.3.1.6. Use Case for Package Promosi



Gambar 2.8 Use Case Package Promosi

2.3.1.7. Use Case for Package DSS



Gambar 2.9 Use Case Package DSS

2.3.2. Fungsi Produk

2.3.2.1. Use Case : Login

Use case ini digunakan oleh aktor baik administrator maupun operator untuk memperoleh akses ke sistem. Login didasarkan pada sebuah username, password dan role masing-masing user.

Lihat : **Spesifikasi Use Case : Login**

2.3.2.2 Use Case : Account.Pengelolaan Data Operator

Use case ini digunakan oleh aktor (administrator) untuk melakukan pengelolaan data operator dari sistem, yang meliputi penambahan operator, pencarian data dan penghapusan operator. Sistem hanya memiliki satu account administrator yang tidak dapat diubah maupun dihapus.

Lihat : **Spesifikasi Use Case : Account.Pengelolaan Data Operator**

2.3.2.3. Use Case : Account.Ubah Password

Use case ini digunakan oleh aktor untuk melakukan pengubahan password dari user aktif.

Lihat : **Spesifikasi Use Case : Account.Ubah Password**

2.3.2.4. Use Case : Periklanan.Elektronik.Pengelolaan Periklanan Internet

Use case ini digunakan untuk melakukan pengelolaan data periklanan internet yang meliputi penambahan data periklanan internet, pengubahan periklanan internet ,pencarian data periklanan internet dan penghapusan data periklanan internet.

Lihat : **Spesifikasi Use Case : Periklanan.
Elektronik.Pengelolaan Periklanan Internet**

**2.3.2.5. Use Case : Periklanan.Elektronik.Pengelolaan
Periklanan Radio**

Use case ini digunakan untuk melakukan pengelolaan data periklanan radio yang meliputi penambahan data periklanan radio, pengubahan data periklanan radio, pencarian data periklanan radio dan penghapusan data periklanan radio.

Lihat : **Spesifikasi Use Case : Periklanan.
Elektronik.Pengelolaan Periklanan Radio**

**2.3.2.6. Use Case : Periklanan.Elektronik.Pengelolaan
Periklanan Televisi**

Use case ini digunakan untuk melakukan pengelolaan data periklanan televisi yang meliputi penambahan data periklanan televisi, pengubahan data periklanan televisi , pencarian data periklanan televisi dan penghapusan data periklanan televisi.

Lihat : **Spesifikasi Use Case : Periklanan.
Elektronik.Pengelolaan Periklanan Televisi**

2.3.2.7. Use Case : Periklanan.MediaCetak.

MediaCetakIndoor.Pengelolaan Periklanan Banner

Use case ini digunakan untuk melakukan pengelolaan data periklanan media cetak indoor banner yang meliputi penambahan data periklanan banner, pengubahan data periklanan banner ,pencarian data periklanan banner dan penghapusan data periklanan banner.

Lihat : **Spesifikasi Use Case : Periklanan.**

MediaCetak.MediaCetakIndoor.Pengelolaan Periklanan Banner

2.3.2.8. Use Case : Periklanan.MediaCetak.

MediaCetakIndoor.Pengelolaan Periklanan Buku

Use case ini digunakan untuk melakukan pengelolaan data periklanan media cetak indoor buku yang meliputi penambahan data periklanan buku, pengubahan data periklanan buku ,pencarian data periklanan buku dan penghapusan data periklanan buku.

Lihat : **Spesifikasi Use Case : Periklanan.**

MediaCetak.MediaCetakIndoor.Pengelolaan Periklanan Buku

2.3.2.9. Use Case : Periklanan.MediaCetak.

MediaCetakIndoor.Pengelolaan Periklanan Brosur

Use case ini digunakan untuk melakukan pengelolaan data periklanan media cetak indoor brosur yang meliputi penambahan data periklanan brosur, pengubahan data periklanan brosur ,pencarian data periklanan brosur dan penghapusan data periklanan brosur.

Lihat : **Spesifikasi Use Case : Periklanan.**

MediaCetak.MediaCetakIndoor.Pengelolaan Periklanan Brosur

2.3.2.10. Use Case : Periklanan.MediaCetak.

MediaCetakIndoor.Pengelolaan Periklanan Majalah

Use case ini digunakan untuk melakukan pengelolaan data periklanan media cetak indoor majalah yang meliputi penambahan data periklanan majalah, pengubahan data periklanan majalah ,pencarian data periklanan majalah dan penghapusan data periklanan majalah.

Lihat : **Spesifikasi Use Case : Periklanan.**

MediaCetak.MediaCetakIndoor.Pengelolaan Periklanan Majalah

2.3.2.11. Use Case : Periklanan.MediaCetak.

MediaCetakIndoor.Pengelolaan Periklanan Koran

Use case ini digunakan untuk melakukan pengelolaan data periklanan media cetak indoor koran yang meliputi penambahan data periklanan koran, pengubahan data periklanan koran ,pencarian data periklanan koran dan penghapusan data periklanan koran.

Lihat : **Spesifikasi Use Case : Periklanan.**

MediaCetak.MediaCetakIndoor.Pengelolaan Periklanan Koran

2.3.2.12. Use Case : Periklanan.MediaCetak.

MediaCetakIndoor.Pengelolaan Periklanan Poster

Use case ini digunakan untuk melakukan pengelolaan data periklanan media cetak indoor poster yang meliputi penambahan data periklanan poster, pengubahan data periklanan poster ,pencarian data periklanan poster dan penghapusan data periklanan poster.

Lihat : **Spesifikasi Use Case : Periklanan.
MediaCetak.MediaCetakIndoor.Pengelolaan Periklanan
Poster**

2.3.2.13. Use Case : Periklanan.MediaCetak.

MediaCetakOutdoor.Pengelolaan Periklanan Billboard
Use case ini digunakan untuk melakukan pengelolaan data periklanan media cetak outdoor billboard yang meliputi penambahan data periklanan billboard, pengubahan data periklanan billboard , pencarian data periklanan billboard dan penghapusan data periklanan billboard.

Lihat : **Spesifikasi Use Case : Periklanan.
MediaCetak.MediaCetakOutdoor.Pengelolaan
Periklanan Billboard**

2.3.2.14. Use Case : Periklanan.MediaCetak.

MediaCetakOutdoor.Pengelolaan Periklanan Branding
Use case ini digunakan untuk melakukan pengelolaan data periklanan media cetak outdoor branding yang meliputi penambahan data periklanan branding, pengubahan data periklanan branding , pencarian data periklanan branding dan penghapusan data periklanan branding.

Lihat : **Spesifikasi Use Case : Periklanan.
MediaCetak.MediaCetakOutdoor.Pengelolaan
Periklanan Branding**

2.3.2.15. Use Case : Periklanan.MediaCetak.

MediaCetakOutdoor.Pengelolaan Periklanan Neon

Use case ini digunakan untuk melakukan pengelolaan data periklanan media cetak outdoor Neon yang meliputi penambahan data periklanan neon, pengubahan data periklanan neon, pencarian data periklanan neon dan penghapusan data periklanan neon.

Lihat : **Spesifikasi Use Case : Periklanan.**

MediaCetak.MediaCetakOutdoor.Pengelolaan Periklanan Neon

2.3.2.16. Use Case : Periklanan.MediaCetak.

MediaCetakOutdoor.Pengelolaan Periklanan Spanduk

Use case ini digunakan untuk melakukan pengelolaan data periklanan media cetak outdoor spanduk yang meliputi penambahan data periklanan spanduk, pengubahan data periklanan spanduk, pencarian data periklanan spanduk dan penghapusan data periklanan spanduk.

Lihat : **Spesifikasi Use Case : Periklanan.**

MediaCetak.MediaCetakOutdoor.Pengelolaan Periklanan Spanduk

2.3.2.17. Use Case : Periklanan.MediaCetak.

MediaCetakOutdoor.Pengelolaan Periklanan Umbul

Use case ini digunakan untuk melakukan pengelolaan data periklanan media cetak outdoor umbul-umbul yang meliputi penambahan data periklanan umbul-umbul, pengubahan data periklanan umbul-umbul, pencarian data periklanan umbul-umbul dan penghapusan data periklanan umbul-umbul.

Lihat : **Spesifikasi Use Case : Periklanan.
MediaCetak.MediaCetakOutdoor.Pengelolaan
Periklanan Umbul**

2.3.2.18. Use Case : Pengelolaan Data Produk

Use case ini digunakan untuk melakukan pengelolaan data produk yang meliputi penambahan data produk, pencarian data produk dan pengubahan data produk.

Lihat : **Spesifikasi Use Case : Pengelolaan Data Produk**

2.3.2.19. Use Case : Pengelolaan Data Promosi

Use case ini digunakan untuk melakukan pengelolaan data promosi yang meliputi penambahan, pencarian data promosi, pengubahan data promosi.

Lihat : **Spesifikasi Use Case : Pengelolaan Data Promosi**

2.3.2.20. Use Case : SPK Promosi Periklanan

Use case ini merupakan sistem pendukung keputusan terhadap strategi pemasaran yang harus dilakukan berdasarkan tahap kesiapan pembeli terhadap produk yang akan ditawarkan.

Lihat : **Spesifikasi Use Case : SPK Promosi Periklanan**

2.4. Karakteristik Pengguna

Karakteristik pengguna yang menggunakan perangkat lunak DSSPROMOSI yaitu :

2.3.1 Administrator

1. Memahami pengoperasian komputer secara aktif

2. Memahami sistem komputer tempat perangkat lunak dijalankan

3. Mengerti tentang internet

2.3.2 Operator

1. Memahami pengoperasian komputer

2. Mengerti dan memahami perangkat lunak yang digunakan

3. Deskripsi Rinci Kebutuhan

3.1 Spesifikasi Kebutuhan Fungsionalitas

3.1.1 Spesifikasi Use Case : Login

Tabel 3.1 Spesifikasi Use Case : Login

Use Case ID	UC-DSSPROMOSI-01
Use Case Name	Login
Use Case Type	Essential
Priority	High
Actors	Administrator, Operator
Description	Use case ini digunakan oleh aktor untuk memperoleh akses ke sistem. Login didasarkan pada sebuah username unik dan password yang berupa rangkaian karakter.
Preconditions	-
Basic Path	1. Sistem menampilkan antarmuka untuk login. 2. Aktor memasukkan username dan password. 3. Sistem memeriksa username dan password Aktor. 4. Sistem memberikan akses ke user sesuai statusnya (administrator, operator).
Alternative Paths	-

Postconditions	Aktor memasuki sistem dan dapat menggunakan fungsi-fungsi pada sistem.
Exception Paths	Username atau password yang dimasukkan tidak sesuai (setelah Tabel 3.1 Basic Path 2) a. Sistem menampilkan peringatan bahwa password tidak sesuai. b. Kembali ke Tabel 3.1 Basic Path 2.
Extends	-
Includes	-
Business Rules	-

3.1.2 Spesifikasi Use Case : Account.Pengelolaan Data Operator

Tabel 3.2 Spesifikasi Use Case : Pengelolaan Data Operator

Use Case ID	UC-DSSPROMOSI-02
Use Case Name	Pengelolaan Data Operator
Use Case Type	Essential
Priority	High
Actors	Administrator
Description	Use case ini digunakan oleh aktor untuk melakukan pengelolaan data operator, dimana pengelolaan ini meliputi penambahan data operator, pencarian data operator dan penghapusan data operator.
Preconditions	Use Case : Login sudah dilaksanakan dan user sudah berhasil memasuki sistem.
Basic Path	Tambah Data Operator 1. Sistem menampilkan antarmuka untuk penambahan data operator. 2. Aktor memasukkan username operator. 3. Aktor memasukkan password dan konfirmasi password operator. 4. Sistem menambahkan data operator ke dalam basis data.

Alternative	<p>A-1. Penghapusan Data Operator</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penghapusan data operator. 2. Aktor akan memilih data operator yang akan dihapus. 3. Sistem akan menghapus data operator tersebut dari basis data. <p>A-2. Pencarian Data Operator</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pencarian data operator. 2. Aktor akan memasukkan username operator yang akan dicari pada tempat yang telah tersedia. 3. Sistem akan menampilkan data operator yang akan dicari.
Postconditions	Data operator di basis data bertambah atau berkurang.
Exception Paths	<ol style="list-style-type: none"> 1. Username yang dimasukkan sudah ada (setelah tabel 3.2 Basic Path 3) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa username sudah terdaftar. b. Kembali ke Tabel 3.2 Basic Path 2. 2. Password dan konfirmasi password tidak sama (setelah Tabel 3.2 Basic Path 3) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa password dan konfirmasinya tidak sama. b. Kembali ke Tabel 3.2 Basic Path 3. 3. Password aktor tidak cocok (setelah Tabel 3.2 Alternative Paths A-1.3) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa password aktor tidak cocok. b. Kembali ke Tabel 3.2 Alternative Paths A-1.3
Extends	-
Includes	Use Case : Login

Extends	-
Includes	Use Case : Login
Business Rules	-

3.1.4 Use Case : Periklanan.Elektronik.Pengelolaan Periklanan Internet

Tabel 3.4 Spesifikasi Use Case : Periklanan Internet

Use Case ID	UC-DSSPROMOSI-04
Use Case Name	Pengelolaan Periklanan Internet
Use Case Type	Essential
Priority	High
Actors	Administrator, Operator
Description	Use case ini digunakan oleh aktor untuk melakukan pengelolaan periklanan internet, dimana pengelolaan ini meliputi penambahan data periklanan internet, pengubahan data periklanan internet, pencarian data periklanan internet dan penghapusan data periklanan internet.
Preconditions	Use Case : Login sudah dilaksanakan dan user sudah berhasil memasuki sistem.
Basic Path	Tambah Data Periklanan Internet <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penambahan data periklanan internet. 2. Aktor memasukkan detail data periklanan internet. 3. Sistem menyimpan data periklanan internet ke dalam basis data.
Alternative	A-1. Pengubahan Data Periklanan Internet <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pengubahan data periklanan internet. 2. Aktor akan memilih data periklanan internet yang akan diubah.

	<p>3. Aktor mengubah data periklanan internet pada tempat yang disediakan.</p> <p>4. Sistem menyimpan data periklanan internet ke basis data.</p> <p>A-2. Penghapusan Data Periklanan Internet</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penghapusan data periklanan internet. 2. Aktor akan memilih data periklanan internet yang akan dihapus. 3. Sistem menghapus data periklanan internet dari basis data. <p>A-3. Pencarian Data Periklanan Internet</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pencarian data periklanan internet. 2. Aktor akan memasukkan situs yang akan dicari pada tempat yang telah tersedia. 3. Sistem akan menampilkan data periklanan internet yang akan dicari.
Postconditions	Data periklanan internet di basis data ter-update.
Exception Paths	<ol style="list-style-type: none"> 1. Data periklanan internet baru yang diinputkan salah (setelah Tabel 3.4 Basic Path 2 atau setelah Tabel 3.4 Alternative Paths A-1.3) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa data periklanan internet tidak valid. b. Kembali ke Tabel 3.4 Basic Path 2 atau ke Tabel 3.4 Alternative Paths A-1.3.
Extends	-
Includes	Use Case : Login
Business Rules	-

3.1.5 Spesifikasi Use Case : Periklanan.Elektronik.

Pengelolaan Periklanan Radio

Tabel 3.5 Spesifikasi Use Case : Periklanan Radio

Use Case ID	UC-DSSPROMOSI-05
Use Case Name	Pengelolaan Periklanan Radio
Use Case Type	Essential
Priority	High
Actors	Administrator, Operator
Description	Use case ini digunakan oleh aktor untuk melakukan pengelolaan data radio, dimana pengelolaan ini meliputi penambahan data radio, pengubahan data radio, pencarian dan penghapusan data radio.
Preconditions	Use Case : Login sudah dilaksanakan dan user sudah berhasil memasuki sistem.
Basic Path	<p>Tambah Data Radio</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penambahan data radio. 2. Aktor memasukkan detail data radio. 3. Sistem menyimpan data radio ke dalam basis data.
Alternative	<p>A-1. Pengubahan Data Radio</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pengubahan data radio. 2. Aktor akan memilih data radio yang akan diubah. 3. Aktor mengubah data radio pada tempat yang disediakan. 4. Sistem menyimpan data radio ke basis data. <p>A-2. Penghapusan Data Radio</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penghapusan data radio. 2. Aktor akan memilih data radio yang akan dihapus.

	<p>3. Sistem menghapus data radio dari basis data.</p> <p>A-3. Pencarian Data Periklanan Radio</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pencarian data periklanan radio. 2. Aktor akan memasukkan nama radio yang akan dicari pada tempat yang telah tersedia. 3. Sistem akan menampilkan data periklanan radio yang akan dicari.
Postconditions	Data periklanan radio di basis data ter-update.
Exception Paths	<ol style="list-style-type: none"> 1. Data periklanan radio baru yang diinputkan salah (setelah Tabel 3.5 Basic Path 2 atau setelah Tabel 3.5 Alternative Paths A-1.2) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa data periklanan radio tidak valid. b. Kembali ke Tabel 3.5 Basic Path 2 atau ke Tabel 3.5 Alternative Paths A-1.2.
Extends	-
Includes	Use Case : Login
Business Rules	-

3.1.6 Spesifikasi Use Case : Periklanan.Elektronik.

Pengelolaan Periklanan Televisi

Tabel 3.6 Spesifikasi Use Case : Periklanan Televisi

Use Case ID	UC-DSSPROMOSI-06
Use Case Name	Pengelolaan Periklanan Televisi
Use Case Type	Essential
Priority	High
Actors	Administrator, Operator
Description	Use case ini digunakan oleh aktor untuk melakukan pengelolaan data televisi,

	dimana pengelolaan ini meliputi penambahan data televisi, perubahan data televisi dan pencarian data televisi dan pencarian data televisi.
Preconditions	Use Case : Login sudah dilaksanakan dan user sudah berhasil memasuki sistem.
Basic Path	<p>Tambah Data Televisi</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penambahan data televisi. 2. Aktor memasukkan detail data televisi. 3. Sistem menambahkan data televisi ke dalam basis data.
Alternative	<p>A-1. Perubahan Data Televisi</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk perubahan data televisi. 2. Aktor akan memilih data televisi yang akan diubah. 3. Aktor mengubah data televisi pada tempat yang disediakan. 4. Sistem akan menyimpan data televisi tersebut ke basis data. <p>A-2. Penghapusan Data Televisi</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penghapusan data televisi. 2. Aktor akan memilih data televisi yang akan dihapus. 3. Sistem menghapus data televisi dari basis data. <p>A-3. Pencarian Data Periklanan Televisi</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pencarian data periklanan televisi. 2. Aktor akan memasukkan key televisi yang akan dicari pada tempat yang telah tersedia. 3. Sistem akan menampilkan data

	periklanan televisi yang akan dicari.
Postconditions	Data televisi di basis data bertambah atau ter-update.
Exception Paths	1 Data periklanan televisi baru yang diinputkan salah (setelah Tabel 3.6 Basic Path 2 atau setelah Alternatif Alternative Paths A-1.2) a. Sistem menampilkan peringatan bahwa data televisi tidak valid. b. Kembali ke Tabel 3.6 Basic Path 2.
Extends	-
Includes	Use Case : Login
Business Rules	-

3.1.7 Spesifikasi Use Case : Periklanan.MediaCetak. MediaCetakIndoor.Pengelolaan Periklanan Banner

Tabel 3.7 Spesifikasi Use Case : Pengelolaan Periklanan Banner

Use Case ID	UC-DSSPROMOSI-07
Use Case Name	Pengelolaan Periklanan Banner
Use Case Type	Essential
Priority	High
Actors	Administrator, Operator
Description	Use case ini digunakan oleh aktor untuk melakukan pengelolaan data media periklanan banner, dimana pengelolaan ini meliputi penambahan data media periklanan banner, pengubahan data media periklanan banner, pencarian data media periklanan banner dan penghapusan data media periklanan banner.
Preconditions	Use Case : Login sudah dilaksanakan dan user sudah berhasil memasuki sistem.
Basic Path	Tambah Data Media Periklanan Banner

	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penambahan data media periklanan banner. 2. Aktor memasukkan detail data media periklanan banner. 3. Sistem menambahkan data media periklanan banner ke dalam basis data.
Alternative	<p>A-1. Pengubahan Data Media Periklanan Banner</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pengubahan data media periklanan banner. 2. Aktor akan memilih data media periklanan banner yang akan diubah. 3. Aktor mengubah data media periklanan banner pada tempat yang disediakan. 4. Sistem akan menyimpan data media periklanan banner tersebut ke basis data.
	<p>A-2. Penghapusan Data Media Periklanan Banner</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penghapusan data media periklanan banner. 2. Aktor akan memilih data media periklanan banner yang akan dihapus. 3. Sistem akan menghapus data media periklanan banner tersebut dari basis data.
	<p>A-3. Pencarian Data Periklanan Banner</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pencarian data periklanan banner. 2. Aktor akan memasukkan key banner

	<p>yang akan dicari pada tempat yang telah tersedia.</p> <p>3. Sistem akan menampilkan data periklanan banner yang akan dicari.</p>
Postconditions	Data media periklanan banner di basis data ter-update.
Exception Paths	<p>1. Data periklanan banner baru yang diinputkan salah (setelah Tabel 3.7 Basic Path 2 atau setelah Alternatif Alternative Paths A-1.2)</p> <p>a. Sistem menampilkan peringatan bahwa data banner tidak valid.</p> <p>b. Kembali ke Tabel 3.7 Basic Path 2.</p>
Extends	-
Includes	Use Case : Login
Business Rules	-

3.1.8 Spesifikasi Use Case : Periklanan.MediaCetak. MediaCetakIndoor.Pengelolaan Periklanan Buku

Tabel 3.8 Spesifikasi Use Case :Pengelolaan Periklanan Buku

Use Case ID	UC-DSSPROMOSI-08
Use Case Name	Pengelolaan Periklanan Buku
Use Case Type	Essential
Priority	High
Actors	Administrator, Operator
Description	Use case ini digunakan oleh aktor untuk melakukan pengelolaan data media periklanan cetak, yakni buku dimana pengelolaan ini meliputi penambahan data media periklanan buku, pengubahan data media periklanan cetak buku, pencarian data media periklanan cetak buku dan penghapusan data media periklanan cetak buku.
Preconditions	Use Case : Login sudah dilaksanakan dan

	user sudah berhasil memasuki sistem.
Basic Path	<p>Tambah Data Media Periklanan Cetak Buku</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penambahan data media periklanan cetak buku. 2. Aktor memasukkan detail data media periklanan cetak buku. 3. Sistem menambahkan data media periklanan cetak buku ke dalam basis data.
Alternative	<p>A-1. Pengubahan Data Media Periklanan Cetak Buku</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pengubahan data media periklanan cetak buku. 2. Aktor akan memilih data media periklanan cetak buku yang akan diubah. 3. Aktor mengubah data media periklanan cetak buku pada tempat yang disediakan. 4. Sistem akan menyimpan data media periklanan cetak buku tersebut ke basis data. <p>A-2. Penghapusan Data Media Periklanan Cetak Buku</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penghapusan data media periklanan cetak buku. 2. Aktor akan memilih data media periklanan cetak buku yang akan dihapus. 3. Sistem akan menghapus data media periklanan cetak buku tersebut dari basis data. <p>A-3. Pencarian Data Periklanan Buku</p>

	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pencarian data periklanan buku. 2. Aktor akan memasukkan key buku yang akan dicari pada tempat yang telah tersedia. 3. Sistem akan menampilkan data periklanan buku yang akan dicari.
Postconditions	Data media periklanan cetak buku di basis data ter-update.
Exception Paths	<ol style="list-style-type: none"> 1. Data periklanan buku baru yang diinputkan salah (setelah Tabel 3.8 Basic Path 2 atau setelah Alternatif Alternative Paths A-1.2) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa data buku tidak valid. b. Kembali ke Tabel 3.8 Basic Path 2.
Extends	-
Includes	Use Case : Login
Business Rules	-

3.1.9 Spesifikasi Use Case : Periklanan.MediaCetak. MediaCetakIndoor.Pengelolaan Periklanan Brosur

Tabel 3.9 Spesifikasi Use Case : Pengelolaan Periklanan Brosur

Use Case ID	UC-DSSPROMOSI-09
Use Case Name	Pengelolaan Periklanan Brosur
Use Case Type	Essential
Priority	High
Actors	Administrator, Operator
Description	Use case ini digunakan oleh aktor untuk melakukan pengelolaan data media periklanan cetak, yakni brosur dimana pengelolaan ini meliputi penambahan data media periklanan brosur, pengubahan data media periklanan cetak brosur, pencarian

	data media periklanan cetak brosur dan penghapusan data media periklanan cetak brosur.
Preconditions	Use Case : Login sudah dilaksanakan dan user sudah berhasil memasuki sistem.
Basic Path	<p>Tambah Data Media Periklanan Cetak Brosur</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penambahan data media periklanan cetak brosur. 2. Aktor memasukkan detail data media periklanan cetak brosur. 3. Sistem menambahkan data media periklanan cetak brosur ke dalam basis data.
Alternative	<p>A-1. Pengubahan Data Media Periklanan Cetak Brosur</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pengubahan data media periklanan cetak brosur. 2. Aktor akan memilih data media periklanan cetak brosur yang akan diubah. 3. Aktor mengubah data media periklanan cetak brosur pada tempat yang disediakan. 4. Sistem akan menyimpan data media periklanan cetak brosur tersebut ke basis data. <p>A-2. Penghapusan Data Media Periklanan Cetak Brosur</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penghapusan data media periklanan cetak brosur. 2. Aktor akan memilih data media periklanan cetak brosur yang akan dihapus.

	<p>3. Sistem akan menghapus data media periklanan cetak brosur tersebut dari basis data.</p> <p>A-3. Pencarian Data Periklanan Brosur</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pencarian data periklanan brosur. 2. Aktor akan memasukkan key brosur yang akan dicari pada tempat yang telah tersedia. 3. Sistem akan menampilkan data periklanan brosur yang akan dicari.
Postconditions	Data media periklanan cetak brosur di basis data ter-update.
Exception Paths	<ol style="list-style-type: none"> 1. Data periklanan brosur baru yang diinputkan salah (setelah Tabel 3.9 Basic Path 2 atau setelah Alternatif Alternative Paths A-1.2) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa data brosur tidak valid. b. Kembali ke Tabel 3.8 Basic Path 2.
Extends	-
Includes	Use Case : Login
Business Rules	-

3.1.10 Spesifikasi Use Case : Periklanan.MediaCetak. MediaCetakIndoor.Pengelolaan Periklanan Majalah

Tabel 3.10 Spesifikasi Use Case : Pengelolaan Periklanan Majalah

Use Case ID	UC-DSSPROMOSI-10
Use Case Name	Pengelolaan Periklanan Majalah
Use Case Type	Essential
Priority	High
Actors	Administrator, Operator
Description	Use case ini digunakan oleh aktor untuk melakukan pengelolaan data media

	<p>periklanan cetak indoor, yakni majalah. Dimana pengelolaan ini meliputi penambahan data media periklanan majalah, pengubahan data media periklanan majalah, pencarian data media periklanan majalah dan penghapusan data media periklanan majalah.</p>
Preconditions	<p>Use Case : Login sudah dilaksanakan dan user sudah berhasil memasuki sistem.</p>
Basic Path	<p>Tambah Data Media Periklanan Cetak Majalah</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penambahan data media periklanan cetak majalah. 2. Aktor memasukkan detail data media periklanan cetak majalah. 3. Sistem menambahkan data media periklanan cetak majalah ke dalam basis data.
Alternative	<p>A-1. Pengubahan Data Media Periklanan Cetak Majalah</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pengubahan data media periklanan cetak majalah. 2. Aktor akan memilih data media periklanan cetak majalah yang akan diubah. 3. Aktor mengubah data media periklanan cetak majalah pada tempat yang disediakan. 4. Sistem akan menyimpan data media periklanan cetak majalah tersebut ke basis data. <p>A-2. Penghapusan Data Media Periklanan Cetak Majalah</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk

	<p>penghapusan data media periklanan cetak majalah.</p> <ol style="list-style-type: none"> 2. Aktor akan memilih data media periklanan cetak majalah yang akan dihapus. 3. Sistem akan menghapus data media periklanan cetak majalah tersebut dari basis data. <p>A-3. Pencarian Data Periklanan Majalah</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pencarian data periklanan majalah. 2. Aktor akan memasukkan key majalah yang akan dicari pada tempat yang telah tersedia. 3. Sistem akan menampilkan data periklanan majalah yang akan dicari.
Postconditions	Data media periklanan cetak majalah di basis data ter-update.
Exception Paths	<ol style="list-style-type: none"> 1. Data periklanan majalah baru yang diinputkan salah (setelah Tabel 3.10 Basic Path 2 atau setelah Alternatif Alternative Paths A-1.2) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa data majalah tidak valid. b. Kembali ke Tabel 3.10 Basic Path 2.
Extends	-
Includes	Use Case : Login
Business Rules	-

3.1.11 Spesifikasi Use Case : Periklanan.MediaCetak.

MediaCetakIndoor.Pengelolaan Periklanan Koran

Tabel 3.11 Spesifikasi Use Case : Pengelolaan Periklanan Koran

Use Case ID	UC-DSSPROMOSI-11
-------------	------------------

Use Case Name	Pengelolaan Periklanan Koran
Use Case Type	Essential
Priority	High
Actors	Administrator, Operator
Description	Use case ini digunakan oleh aktor untuk melakukan pengelolaan data media periklanan cetak outdoor, yaitu majalah dimana pengelolaan ini meliputi penambahan data media periklanan koran, pengubahan data media periklanan koran, pencarian data media periklanan koran dan penghapusan data media periklanan koran.
Preconditions	Use Case : Login sudah dilaksanakan dan user sudah berhasil memasuki sistem.
Basic Path	Tambah Data Media Periklanan Cetak Koran <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penambahan data media periklanan cetak koran. 2. Aktor memasukkan detail data media periklanan cetak koran. 3. Sistem menambahkan data media periklanan cetak koran ke dalam basis data.
Alternative	A-1. Pengubahan Data Media Periklanan Cetak Koran <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pengubahan data media periklanan cetak koran. 2. Aktor akan memilih data media periklanan cetak koran yang akan diubah. 3. Aktor mengubah data media periklanan cetak koran pada tempat yang disediakan. 4. Sistem akan menyimpan data media periklanan cetak koran tersebut ke

	<p>basis data.</p> <p>A-2. Penghapusan Data Media Periklanan Cetak Koran</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penghapusan data media periklanan cetak koran. 2. Aktor akan memilih data media periklanan cetak koran yang akan dihapus. 3. Sistem akan menghapus data media periklanan cetak koran tersebut dari basis data. <p>A-3. Pencarian Data Periklanan Koran</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pencarian data periklanan koran. 2. Aktor akan memasukkan key koran yang akan dicari pada tempat yang telah tersedia. 3. Sistem akan menampilkan data periklanan koran yang akan dicari.
Postconditions	Data media periklanan cetak koran di basis data ter-update.
Exception Paths	<ol style="list-style-type: none"> 1. Data periklanan koran baru yang diinputkan salah (setelah Tabel 3.11 Basic Path 2 atau setelah Alternatif Alternative Paths A-1.2) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa data koran tidak valid. b. Kembali ke Tabel 3.11 Basic Path 2.
Extends	-
Includes	Use Case : Login
Business Rules	-

3.1.12 Spesifikasi Use Case : Periklanan.MediaCetak. MediaCetakIndoor.Pengelolaan Periklanan Poster

Tabel 3.12 Spesifikasi Use Case : Pengelolaan Periklanan Poster

Use Case ID	UC-DSSPROMOSI-12
Use Case Name	Pengelolaan Periklanan Poster
Use Case Type	Essential
Priority	High
Actors	Administrator, Operator
Description	Use case ini digunakan oleh aktor untuk melakukan pengelolaan data media periklanan cetak outdoor, yaitu poster dimana pengelolaan ini meliputi penambahan data media periklanan poster, pengubahan data media periklanan poster, pencarian data media periklanan poster dan penghapusan data media periklanan poster.
Preconditions	Use Case : Login sudah dilaksanakan dan user sudah berhasil memasuki sistem.
Basic Path	Tambah Data Media Periklanan Cetak Poster <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penambahan data media periklanan cetak poster. 2. Aktor memasukkan detail data media periklanan cetak poster. 3. Sistem menambahkan data media periklanan cetak poster ke dalam basis data.
Alternative	A-1. Pengubahan Data Media Periklanan Cetak Poster <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pengubahan data media periklanan cetak poster. 2. Aktor akan memilih data media periklanan cetak poster yang akan diubah. 3. Aktor mengubah data media

	<p>periklanan cetak poster pada tempat yang disediakan.</p> <p>4. Sistem akan menyimpan data media periklanan cetak poster tersebut ke basis data.</p> <p>A-2. Penghapusan Data Media Periklanan Cetak Poster</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penghapusan data media periklanan cetak poster. 2. Aktor akan memilih data media periklanan cetak poster yang akan dihapus. 3. Sistem akan menghapus data media periklanan cetak poster tersebut dari basis data. <p>A-3. Pencarian Data Media Periklanan Poster</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pencarian data periklanan poster. 2. Aktor akan memasukkan key poster yang akan dicari pada tempat yang telah tersedia. 3. Sistem akan menampilkan data periklanan poster yang akan dicari.
Postconditions	Data media periklanan cetak poster di basis data ter-update.
Exception Paths	<ol style="list-style-type: none"> 1. Data periklanan poster baru yang diinputkan salah (setelah Tabel 3.12 Basic Path 2 atau setelah Alternatif Alternative Paths A-1.2) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa data poster tidak valid. b. Kembali ke Tabel 3.12 Basic Path 2.
Extends	-
Includes	Use Case : Login

Business Rules	-
----------------	---

3.1.13 Spesifikasi Use Case : Periklanan.MediaCetak. MediaCetakIndoor.Pengelolaan Periklanan Billboard

Tabel 3.13 Spesifikasi Use Case : Pengelolaan Periklanan Billboard

Use Case ID	UC-DSSPROMOSI-13
Use Case Name	Pengelolaan Periklanan Billboard
Use Case Type	Essential
Priority	High
Actors	Administrator, Operator
Description	Use case ini digunakan oleh aktor untuk melakukan pengelolaan data media periklanan cetak outdoor, yaitu billboard dimana pengelolaan ini meliputi penambahan data media periklanan billboard, pengubahan data media periklanan billboard, pencarian data media periklanan billboard dan penghapusan data media periklanan billboard.
Preconditions	Use Case : Login sudah dilaksanakan dan user sudah berhasil memasuki sistem.
Basic Path	Tambah Data Media Periklanan Cetak Billboard <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penambahan data media periklanan cetak billboard. 2. Aktor memasukkan detail data media periklanan cetak billboard. 3. Sistem menambahkan data media periklanan cetak billboard ke dalam basis data.
Alternative	A-1. Pengubahan Data Media Periklanan

	<p>Cetak Billboard</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk perubahan data media periklanan cetak billboard. 2. Aktor akan memilih data media periklanan cetak billboard yang akan diubah. 3. Aktor mengubah data media periklanan cetak billboard pada tempat yang disediakan. 4. Sistem akan menyimpan data media periklanan cetak billboard tersebut ke basis data. <p>A-2. Penghapusan Data Media Periklanan Cetak Billboard</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penghapusan data media periklanan cetak billboard. 2. Aktor akan memilih data media periklanan cetak billboard yang akan dihapus. 3. Sistem akan menghapus data media periklanan cetak billboard tersebut dari basis data. <p>A-3. Pencarian Data Media Periklanan Billboard</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pencarian data periklanan billboard. 2. Aktor akan memasukkan key billboard yang akan dicari pada tempat yang telah tersedia. 3. Sistem akan menampilkan data periklanan billboard yang akan dicari.
Postconditions	Data media periklanan cetak billboard di

	basis data ter-update.
Exception Paths	<ol style="list-style-type: none"> 1. Data periklanan billboard baru yang diinputkan salah (setelah Tabel 3.13 Basic Path 2 atau setelah Alternatif Alternative Paths A-1.2) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa data billboard tidak valid. b. Kembali ke Tabel 3.13 Basic Path 2.
Extends	-
Includes	Use Case : Login
Business Rules	-

3.1.14 Spesifikasi Use Case : Periklanan.MediaCetak. MediaCetakIndoor.Pengelolaan Periklanan Branding

Tabel 3.14 Spesifikasi Use Case : Pengelolaan Periklanan Branding

Use Case ID	UC-DSSPROMOSI-14
Use Case Name	Pengelolaan Periklanan Branding
Use Case Type	Essential
Priority	High
Actors	Administrator, Operator
Description	Use case ini digunakan oleh aktor untuk melakukan pengelolaan data media periklanan cetak outdoor,yaitu branding dimana pengelolaan ini meliputi penambahan data media periklanan branding, pengubahan data media periklanan branding, pencarian data media periklanan branding dan penghapusan data media periklanan branding.
Preconditions	Use Case : Login sudah dilaksanakan dan user sudah berhasil memasuki sistem.
Basic Path	Tambah Data Media Periklanan Cetak Branding

	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penambahan data media periklanan cetak branding. 2. Aktor memasukkan detail data media periklanan cetak branding. 3. Sistem menambahkan data media periklanan cetak branding ke dalam basis data.
Alternative	<p>A-1. Perubahan Data Media Periklanan Cetak Branding</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk perubahan data media periklanan cetak branding. 2. Aktor akan memilih data media periklanan cetak branding yang akan diubah. 3. Aktor mengubah data media periklanan cetak branding pada tempat yang disediakan. 4. Sistem akan menyimpan data media periklanan cetak branding tersebut ke basis data. <p>A-2. Penghapusan Data Media Periklanan Cetak Branding</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penghapusan data media periklanan cetak branding. 2. Aktor akan memilih data media periklanan cetak branding yang akan dihapus. 3. Sistem akan menghapus data media periklanan cetak branding tersebut dari basis data. <p>A-3. Pencarian Data Media Periklanan Branding</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk

	<p>pencarian data periklanan branding.</p> <p>2. Aktor akan memasukkan key branding yang akan dicari pada tempat yang telah tersedia.</p> <p>3. Sistem akan menampilkan data periklanan branding yang akan dicari.</p>
Postconditions	Data media periklanan cetak branding di basis data ter-update.
Exception Paths	<p>1. Data periklanan branding baru yang diinputkan salah (setelah Tabel 3.14 Basic Path 2 atau setelah Alternatif Alternative Paths A-1.2)</p> <p>a. Sistem menampilkan peringatan bahwa data branding tidak valid.</p> <p>b. Kembali ke Tabel 3.14 Basic Path 2.</p>
Extends	-
Includes	Use Case : Login
Business Rules	-

3.1.15 Spesifikasi Use Case : Periklanan.MediaCetak.

MediaCetakIndoor.Pengelolaan Periklanan Neon

Tabel 3.12 Spesifikasi Use Case : Pengelolaan Periklanan Neon

Use Case ID	UC-DSSPROMOSI-15
Use Case Name	Pengelolaan Periklanan Neon
Use Case Type	Essential
Priority	High
Actors	Administrator, Operator
Description	Use case ini digunakan oleh aktor untuk melakukan pengelolaan data media periklanan cetak outdoor, yaitu neon dimana pengelolaan ini meliputi penambahan data media periklanan neon, pengubahan data media periklanan neon, pencarian data media periklanan neon dan

	penghapusan data media periklanan neon.
Preconditions	Use Case : Login sudah dilaksanakan dan user sudah berhasil memasuki sistem.
Basic Path	<p>Tambah Data Media Periklanan Cetak Neon</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penambahan data media periklanan cetak neon. 2. Aktor memasukkan detail data media periklanan cetak neon. 3. Sistem menambahkan data media periklanan cetak neon ke dalam basis data.
Alternative	<p>A-1. Pengubahan Data Media Periklanan Cetak Neon</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pengubahan data media periklanan cetak neon. 2. Aktor akan memilih data media periklanan cetak neon yang akan diubah. 3. Aktor mengubah data media periklanan cetak neon pada tempat yang disediakan. 4. Sistem akan menyimpan data media periklanan cetak neon tersebut ke basis data. <p>A-2. Penghapusan Data Media Periklanan Cetak Neon</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penghapusan data media periklanan cetak neon. 2. Aktor akan memilih data media periklanan cetak neon yang akan dihapus. 3. Sistem akan menghapus data media periklanan cetak neon tersebut dari

	<p>basis data.</p> <p>A-3. Pencarian Data Media Periklanan Neon</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pencarian data periklanan neon. 2. Aktor akan memasukkan key neon yang akan dicari pada tempat yang telah tersedia. 3. Sistem akan menampilkan data periklanan neon yang akan dicari.
Postconditions	Data media periklanan cetak neon di basis data ter-update.
Exception Paths	<ol style="list-style-type: none"> 1. Data periklanan neon baru yang diinputkan salah (setelah Tabel 3.15 Basic Path 2 atau setelah Alternatif Alternative Paths A-1.2) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa data neon tidak valid. b. Kembali ke Tabel 3.15 Basic Path 2.
Extends	-
Includes	Use Case : Login
Business Rules	-

3.1.16 Spesifikasi Use Case : Periklanan.MediaCetak. MediaCetakIndoor.Pengelolaan Periklanan Spanduk

Tabel 3.16 Spesifikasi Use Case : Pengelolaan Periklanan Spanduk

Use Case ID	UC-DSSPROMOSI-16
Use Case Name	Pengelolaan Periklanan Spanduk
Use Case Type	Essential
Priority	High
Actors	Administrator, Operator
Description	Use case ini digunakan oleh aktor untuk melakukan pengelolaan data media periklanan cetak outdoor, yaitu spanduk dimana pengelolaan ini meliputi

	penambahan data media periklanan spanduk, pengubahan data media periklanan spanduk, pencarian data media periklanan spanduk dan penghapusan data media periklanan spanduk.
Preconditions	Use Case : Login sudah dilaksanakan dan user sudah berhasil memasuki sistem.
Basic Path	<p>Tambah Data Media Periklanan Cetak Spanduk</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penambahan data media periklanan cetak spanduk. 2. Aktor memasukkan detail data media periklanan cetak spanduk. 3. Sistem menambahkan data media periklanan cetak spanduk ke dalam basis data.
Alternative	<p>A-1. Pengubahan Data Media Periklanan Cetak Spanduk</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pengubahan data media periklanan cetak spanduk. 2. Aktor akan memilih data media periklanan cetak spanduk yang akan diubah. 3. Aktor mengubah data media periklanan cetak spanduk pada tempat yang disediakan. 4. Sistem akan menyimpan data media periklanan cetak spanduk tersebut ke basis data. <p>A-2. Penghapusan Data Media Periklanan Cetak Spanduk</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penghapusan data media periklanan cetak spanduk.

	<p>2. Aktor akan memilih data media periklanan cetak spanduk yang akan dihapus.</p> <p>3. Sistem akan menghapus data media periklanan cetak spanduk tersebut dari basis data.</p> <p>A-3. Pencarian Data Media Periklanan Spanduk</p> <p>1. Sistem menampilkan antarmuka untuk pencarian data periklanan spanduk.</p> <p>2. Aktor akan memasukkan key spanduk yang akan dicari pada tempat yang telah tersedia.</p> <p>3. Sistem akan menampilkan data periklanan spanduk yang akan dicari.</p>
Postconditions	Data media periklanan cetak spanduk di basis data ter-update.
Exception Paths	<p>1. Data periklanan spanduk baru yang diinputkan salah (setelah Tabel 3.16 Basic Path 2 atau setelah Alternatif Alternative Paths A-1.2)</p> <p>a. Sistem menampilkan peringatan bahwa data spanduk tidak valid.</p> <p>b. Kembali ke Tabel 3.16 Basic Path 2.</p>
Extends	-
Includes	Use Case : Login
Business Rules	-

3.1.17 Spesifikasi Use Case : Periklanan.MediaCetak.

MediaCetakIndoor.Pengelolaan Periklanan Umbul

Tabel 3.17 Spesifikasi Use Case : Pengelolaan Periklanan Umbul

Use Case ID	UC-DSSPROMOSI-17
Use Case Name	Pengelolaan Periklanan Umbul

Use Case Type	Essential
Priority	High
Actors	Administrator, Operator
Description	Use case ini digunakan oleh aktor untuk melakukan pengelolaan data media periklanan cetak outdoor, yaitu umbul-umbul dimana pengelolaan ini meliputi penambahan data media periklanan umbul-umbul, pengubahan data media periklanan umbul-umbul, pencarian data media periklanan umbul-umbul dan penghapusan data media periklanan umbul-umbul.
Preconditions	Use Case : Login sudah dilaksanakan dan user sudah berhasil memasuki sistem.
Basic Path	Tambah Data Media Periklanan Cetak Umbul <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penambahan data media periklanan cetak umbul-umbul. 2. Aktor memasukkan detail data media periklanan cetak umbul-umbul. 3. Sistem menambahkan data media periklanan cetak umbul-umbul ke dalam basis data.
Alternative	A-1. Pengubahan Data Media Periklanan Cetak Umbul <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pengubahan data media periklanan cetak umbul-umbul. 2. Aktor akan memilih data media periklanan cetak umbul-umbul yang akan diubah. 3. Aktor mengubah data media periklanan cetak umbul-umbul pada tempat yang disediakan. 4. Sistem akan menyimpan data media periklanan cetak umbul-umbul

	<p>tersebut ke basis data.</p> <p>A-2. Penghapusan Data Media Periklanan Cetak Umbul</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penghapusan data media periklanan cetak umbul-umbul. 2. Aktor akan memilih data media periklanan cetak umbul-umbul yang akan dihapus. 3. Sistem akan menghapus data media periklanan cetak umbul-umbul tersebut dari basis data. <p>A-3. Pencarian Data Media Periklanan Umbul</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pencarian data periklanan umbul-umbul. 2. Aktor akan memasukkan key umbul-umbul yang akan dicari pada tempat yang telah tersedia. 3. Sistem akan menampilkan data periklanan umbul-umbul yang akan dicari.
Postconditions	Data media periklanan cetak umbul-umbul di basis data ter-update.
Exception Paths	<ol style="list-style-type: none"> 1. Data periklanan umbul-umbul baru yang diinputkan salah (setelah Tabel 3.17 Basic Path 2 atau setelah Alternatif Alternative Paths A-1.2) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa data umbul-umbul tidak valid. b. Kembali ke Tabel 3.17 Basic Path 2.
Extends	-
Includes	Use Case : Login
Business Rules	-

3.1.18 Spesifikasi Use Case : Pengelolaan Data Produk

Tabel 3.18 Spesifikasi Use Case : Pengelolaan Data Produk

Use Case ID	UC-DSSPROMOSI-18
Use Case Name	Pengelolaan Data Produk
Use Case Type	Essential
Priority	High
Actors	Administrator, Operator
Description	Use case ini digunakan oleh aktor untuk melakukan pengelolaan data produk, dimana pengelolaan ini meliputi penambahan data produk, pengubahan data produk, pencarian data produk dan penghapusan data produk.
Preconditions	Use Case : Login sudah dilaksanakan dan user sudah berhasil memasuki sistem.
Basic Path	<p>Tambah Data Produk</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk penambahan data produk. 2. Aktor memasukkan detail data produk. 3. Sistem menyimpan data produk ke dalam basis data.
Alternative	<p>A-1. Pengubahan Data Produk</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pengubahan data produk. 2. Aktor akan memilih data produk yang akan diubah. 3. Aktor mengubah data produk pada tempat yang disediakan. 4. Sistem menyimpan data produk ke basis data. <p>A-2. Pencarian Data Produk</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pencarian data produk. 2. Aktor akan memasukkan key produk yang akan dicari pada tempat yang

	<p>telah tersedia.</p> <p>3. Sistem akan menampilkan data produk yang akan dicari.</p>
Postconditions	Data produk di basis data ter-update.
Exception Paths	<p>1. Data produk baru yang diinputkan salah (setelah Tabel 3.18 Basic Path 2 atau setelah Alternatif Alternative Paths A-1.2)</p> <p>a. Sistem menampilkan peringatan bahwa data produk tidak valid.</p> <p>b. Kembali ke Tabel 3.18 Basic Path 2.</p>
Extends	-
Includes	Use Case : Login
Business Rules	-

3.1.19 Spesifikasi Use Case : Pengelolaan Data Promosi

Tabel 3.19 Spesifikasi Use Case : Pengelolaan Data Promosi

Use Case ID	UC-DSSPROMOSI-19
Use Case Name	Pengelolaan Data Promosi
Use Case Type	Essential
Priority	High
Actors	Administrator, Operator
Description	Use case ini digunakan oleh aktor untuk melakukan pengelolaan data promosi, dimana pengelolaan ini meliputi penambahan data promosi dan pengubahan data promosi dan pencarian data promosi
Preconditions	Use Case : Login sudah dilaksanakan dan user sudah berhasil memasuki sistem.
Basic Path	<p>Tambah Data Promosi</p> <p>1. Sistem menampilkan antarmuka untuk penambahan data promosi.</p> <p>2. Aktor memasukkan detail data promosi.</p>

	3. Sistem menambahkan data promosi ke dalam basis data.
Alternative	<p>A-1. Perubahan Data Promosi</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk perubahan data promosi. 2. Aktor akan memilih data promosi yang akan diubah. 3. Aktor mengubah data promosi pada tempat yang disediakan. 4. Sistem akan menyimpan data promosi tersebut ke basis data. <p>A-2. Pencarian Data Periklanan Promosi</p> <ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pencarian data periklanan promosi. 2. Aktor akan memasukkan key promosi yang akan dicari pada tempat yang telah tersedia. 3. Sistem akan menampilkan data periklanan promosi yang akan dicari.
Postconditions	Data promosi di basis data bertambah atau ter-update.
Exception Paths	<ol style="list-style-type: none"> 1. Data promosi baru yang diinputkan salah (setelah Tabel 3.19 Basic Path 2 atau setelah Alternatif Alternative Paths A-1.2) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa data promosi tidak valid. b. Kembali ke Tabel 3.19 Basic Path 2.
Extends	-
Includes	Use Case : Login
Business Rules	-

3.1.20 Spesifikasi Use Case : SPK Promosi Periklanan

Tabel 3.20 Spesifikasi Use Case : SPK Promosi Periklanan

Use Case ID	UC-DSSPROMOSI-20
-------------	------------------

Use Case Name	SPK Promosi Periklanan
Use Case Type	Essential
Priority	High
Actors	Administrator, Operator
Description	Use case ini digunakan oleh aktor untuk menentukan strategi pemasaran yang akan dijalankan serta media periklanan yang tepat sesuai dengan produk yang akan ditawarkan.
Preconditions	Use Case : Login sudah dilaksanakan dan user sudah berhasil memasuki sistem.
Basic Path	<p>Penentuan Keputusan</p> <ol style="list-style-type: none"> 1. Sistem akan menampilkan antarmuka untuk penentuan strategi pemasaran dan penentuan media periklanan. 2. Aktor memasukkan detail faktor penentuan keputusan pada tempat yang telah tersedia. 3. Sistem akan memberikan keputusan terhadap strategi pemasaran yang harus dilakukan beserta bauran media periklanan yang sesuai
Alternative	-
Postconditions	Strategi pemasaran dan media periklanan akan ditampilkan
Exception Paths	<ol style="list-style-type: none"> 1. Data kriteria keputusan yang diinputkan salah (setelah Tabel 3.20 Basic Path 2) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa data tidak valid. b. Kembali ke Tabel 3.20 Basic Path 2.
Extends	-
Includes	Use Case : Login
Business Rules	-

3.2. Spesifikasi Kebutuhan Non-Fungsionalitas

3.2.1. Kebutuhan Antarmuka Eksternal

Kebutuhan antar muka eksternal pada sistem DSSPROMOSI mencakup kebutuhan antarmuka pemakai, antarmuka perangkat keras dan antarmuka perangkat lunak.

3.2.2. Antarmuka Pemakai

Pemakai berinteraksi langsung dengan sistem DSSPROMOSI dengan antarmuka berbasis GUI. Piranti masukan yang digunakan untuk memasukkan data masukan adalah *keyboard* dan *mouse*. Sedangkan keluaran dari sistem berupa data-data yang disimpan dalam basis data dan dalam bentuk file serta akan ditampilkan langsung ke layar monitor.

3.2.3. Antarmuka Perangkat Keras

Antarmuka perangkat keras yang dibutuhkan dalam penggunaan perangkat lunak DSSPROMOSI adalah:

- *PC Compatible*
- *Keyboard*
- *Mouse*

3.2.4. Antarmuka Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk mendukung berjalannya perangkat lunak DSSPROMOSI adalah:

- Nama : *Windows Me/NT/2000/XP*
 Sumber : *Microsoft*
 Fungsi : *Sistem Operasi Komputer*

- Nama : *ASP .NET 2003*
 (Bahasa Pemrograman C#)
 Sumber : *Microsoft*
 Fungsi : *Tool perancang antarmuka aplikasi*

- Nama : *Internet Information Server (IIS)*
 Sumber : *Microsoft*
 Fungsi : *Web Server*

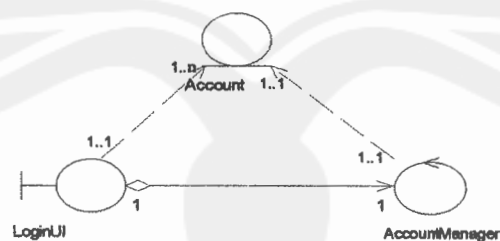
- Nama : *Internet Explorer*
 Sumber : *Microsoft*
 Fungsi : *Internet Browser*

- Nama : *MySql 5.0*
 Sumber : *Open Source*
 Fungsi : *Database Management System*

4. Realisasi Use Case

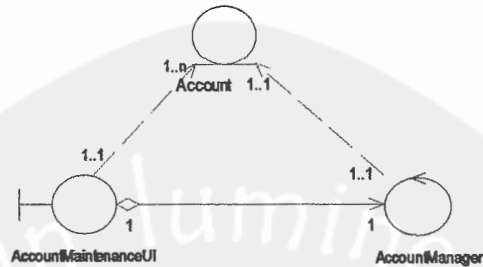
4.1. Static Structure Diagram

4.1.1. Analysis Class Diagram : Use Case Login



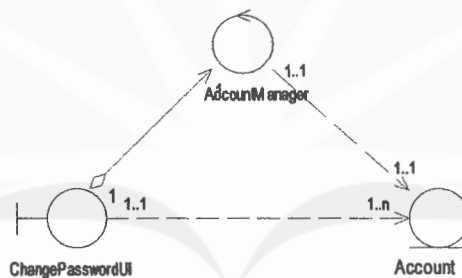
Gambar 4.1 Analysis Class Diagram : Use Case Login

4.1.2. Analysis Class Diagram : Use Case Account.Pengelolaan Data Operator



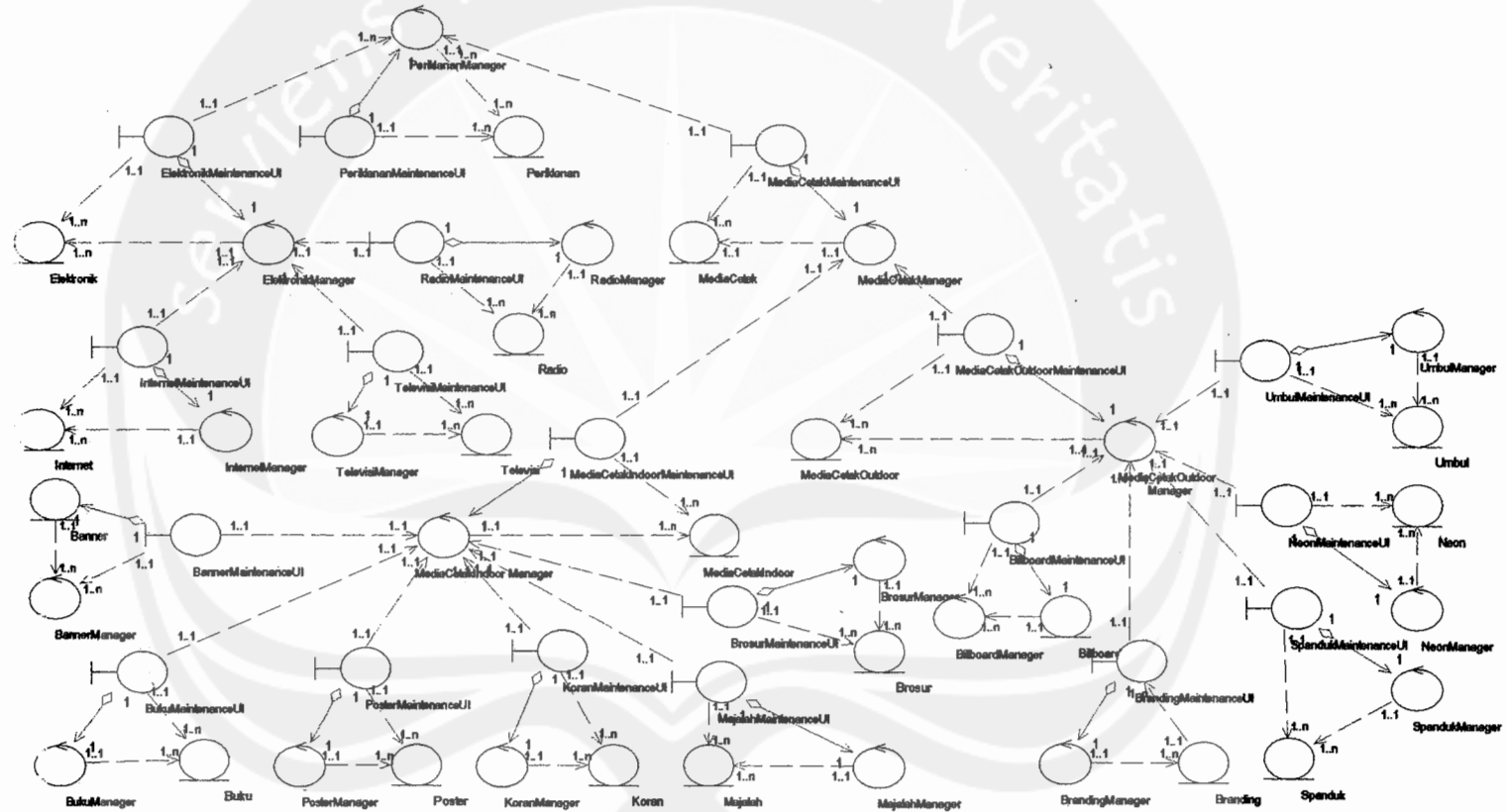
Gambar 4.2 Analysis Class Diagram : Use Case Pengelolaan Data Operator

4.1.3. Analysis Class Diagram : Use Case Account.Ubah Password



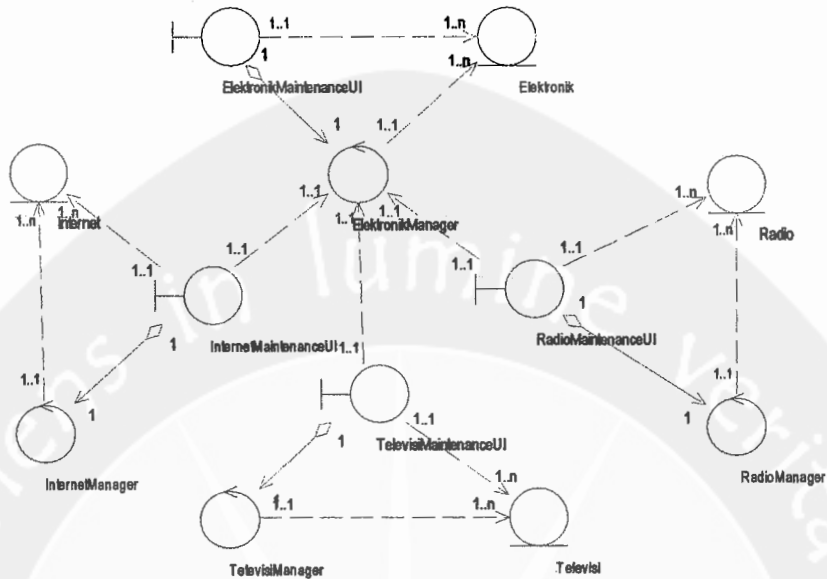
Gambar 4.3 Analysis Class Diagram : Use Case Ubah Password

4.1.4. Analysis Class Diagram : Package Periklanan



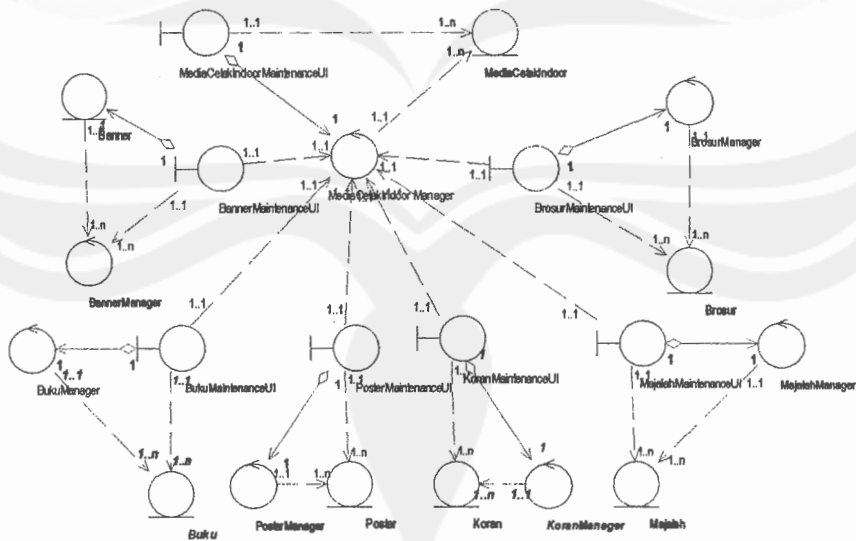
Gambar 4.4 Analysis Class Diagram : Package Periklanan

4.1.5. Analysis Class Diagram Package Elektronik



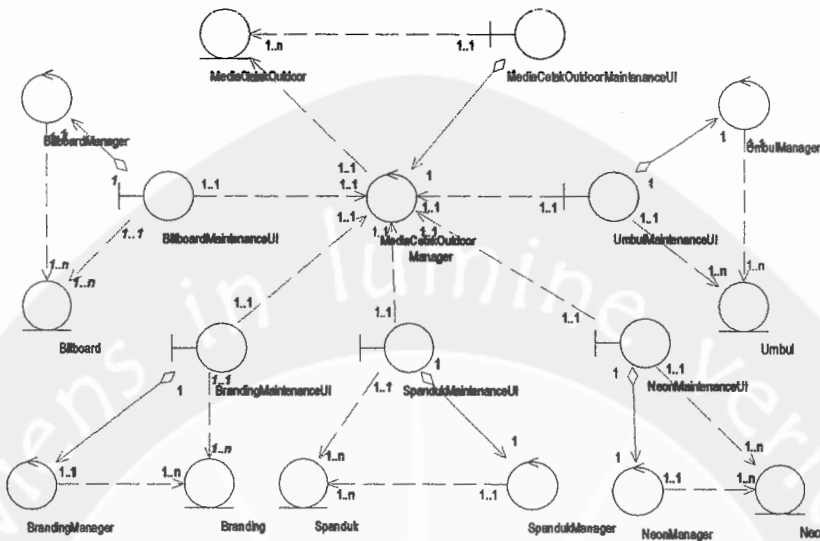
Gambar 4.5 Analysis Class Diagram Package Elektronik

4.1.6. Analysis Class Diagram Package MediaCetakIndoor



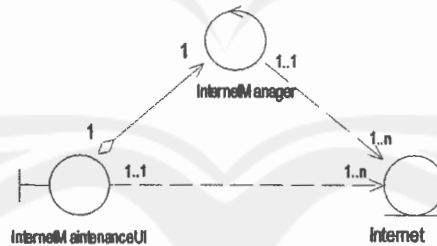
Gambar 4.6 Analysis Class Diagram Package MediaCetakIndoor

4.1.7. Analysis Class Diagram Package MediaCetakOutdoor



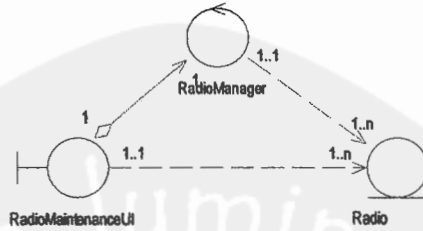
Gambar 4.7 Analysis Class Diagram Package MediaCetakOutdoor

4.1.8. Analysis Class Diagram : Use Case Periklanan.Elektronik.Pengelolaan Periklanan Internet



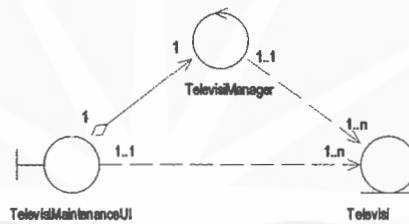
Gambar 4.5 Analysis Class Diagram : Use Case Pengelolaan Periklanan Internet

**4.1.9. Analysis Class Diagram : Use Case Periklanan.
Elektronik.Pengelolaan Periklanan Radio**



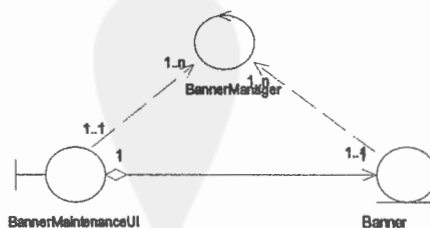
Gambar 4.9 Analysis Class Diagram : Use Case Pengelolaan Periklanan Radio

4.1.10. Analysis Class Diagram : Use Case Periklanan.Elektronik.Pengelolaan Periklanan Televisi



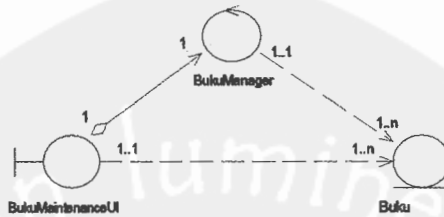
Gambar 4.10 Analysis Class Diagram : Use Case Pengelolaan Periklanan Televisi

**4.1.11. Analysis Class Diagram : Use Case Periklanan.
MediaCetak.MediaCetakIndoor.Pengelolaan Periklanan Banner**



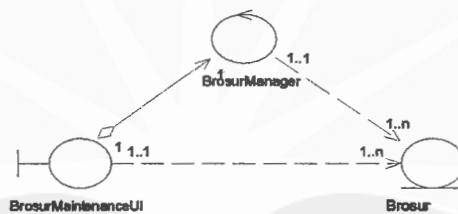
Gambar 4.11 Analysis Class Diagram : Use Case Pengelolaan Periklanan Banner

**4.1.12. Analysis Class Diagram : Use Case
Periklanan.MediaCetak.MediaCetakIndoor.Pengelolaan
Periklanan Buku**



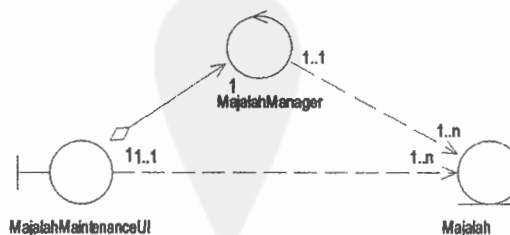
Gambar 4.12 Analysis Class Diagram : Use Case Pengelolaan Periklanan Buku

**4.1.13. Analysis Class Diagram : Use Case
Periklanan.MediaCetak.MediaCetakIndoor.
Pengelolaan Periklanan Brosur**



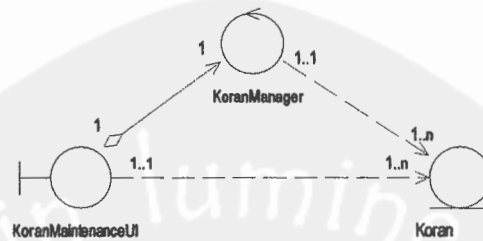
Gambar 4.13 Analysis Class Diagram : Use Case Pengelolaan Periklanan Brosur

**4.1.14. Analysis Class Diagram : Use Case Periklanan.
MediaCetak.MediaCetakIndoor.Pengelolaan Periklanan
Majalah**



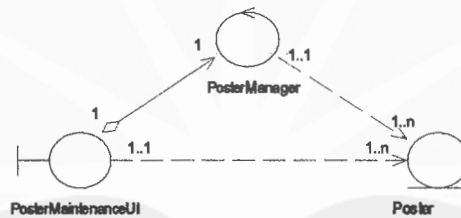
Gambar 4.14 Analysis Class Diagram : Use Case Pengelolaan Periklanan Majalah

**4.1.15. Analysis Class Diagram : Use Case
Periklanan.MediaCetak. MediaCetakIndoor.Pengelolaan
Periklanan Koran**



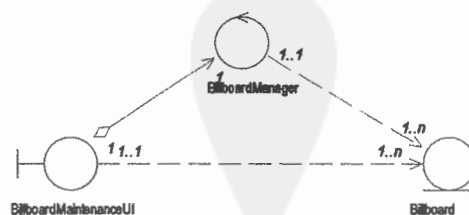
Gambar 4.15 Analysis Class Diagram : Use Case Pengelolaan Periklanan Koran

**4.1.16. Analysis Class Diagram : Use Case
Periklanan.MediaCetak. MediaCetakIndoor.Pengelolaan
Periklanan Poster**



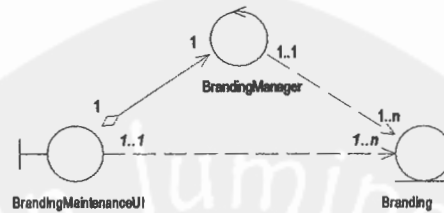
Gambar 4.16 Analysis Class Diagram : Use Case Pengelolaan Periklanan Poster

**4.1.17. Analysis Class Diagram : Use Case
Periklanan.MediaCetak.MediaCetakOutdoor.
Pengelolaan Periklanan Billboard**



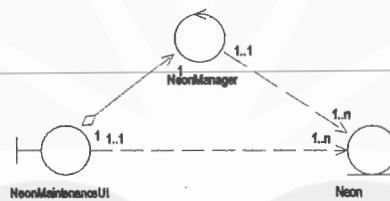
Gambar 4.17 Analysis Class Diagram : Use Case Pengelolaan Periklanan Billboard

4.1.18. Analysis Class Diagram : Use Case Periklanan.MediaCetak.MediaCetakOutdoor.Pengelolaan Periklanan Branding



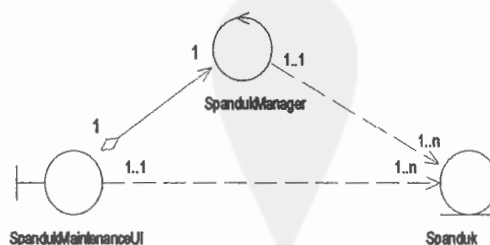
Gambar 4.18 Analysis Class Diagram : Use Case Pengelolaan Periklanan Branding

4.1.19. Analysis Class Diagram : Use Case Periklanan.MediaCetak.MediaCetakOutdoor.Pengelolaan Periklanan Neon



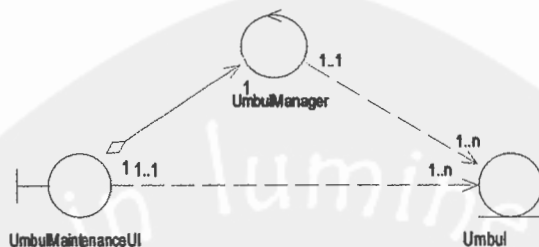
Gambar 4.19 Analysis Class Diagram : Use Case Pengelolaan Periklanan Neon

4.1.20. Analysis Class Diagram : Use Case Periklanan.MediaCetak.MediaCetakOutdoor.Pengelolaan Periklanan Spanduk



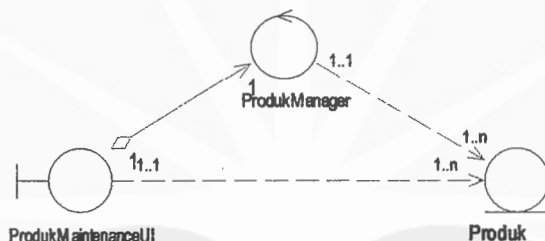
Gambar 4.20 Analysis Class Diagram : Use Case Pengelolaan Periklanan Spanduk

4.1.21. Analysis Class Diagram : Use Case Periklanan.MediaCetak.MediaCetakOutdoor.Pengelolaan Periklanan Umbul



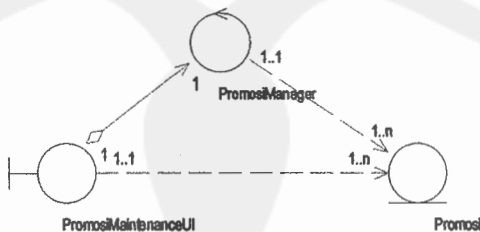
Gambar 4.21 Analysis Class Diagram : Use Case Pengelolaan Periklanan Umbul

4.1.22. Analysis Class Diagram : Use Case Pengelolaan Data Produk



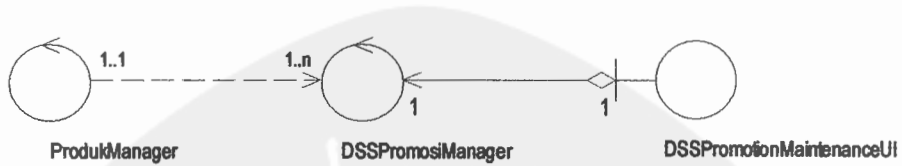
Gambar 4.22 Analysis Class Diagram : Use Case Pengelolaan Produk

4.1.23. Analysis Class Diagram : Use Case Pengelolaan Data Promosi



Gambar 4.23 Analysis Class Diagram : Use Case Pengelolaan Promosi

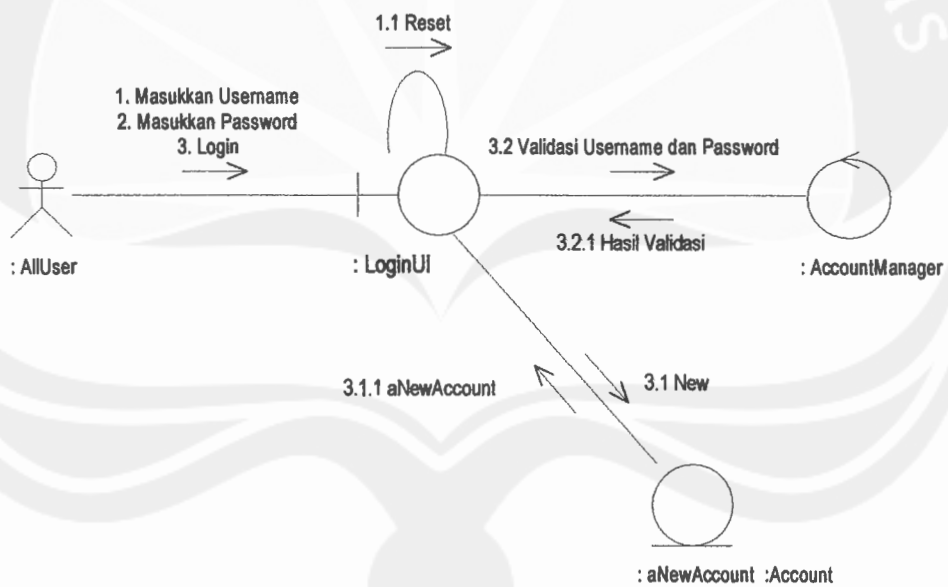
4.1.24. Analysis Class Diagram : Use Case SPK Promosi Periklanan



Gambar 4.24 Analysis Class Diagram : Use Case SPK Promosi Periklanan

4.2. Interaction Diagram

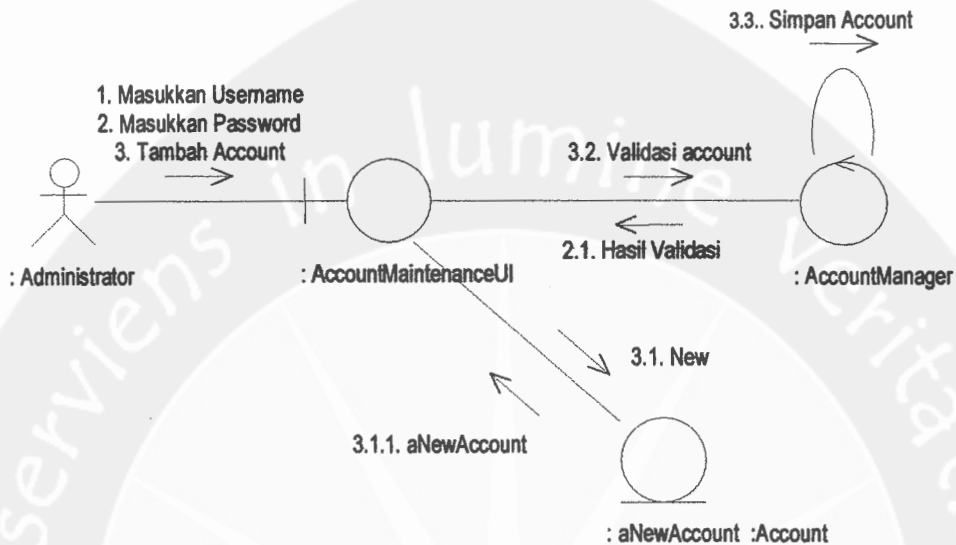
4.2.1. Analisis Collaboration Diagram : Use Case Login



Gambar 4.25 Analysis Collaboration Diagram : Use Case Pengolahan Login

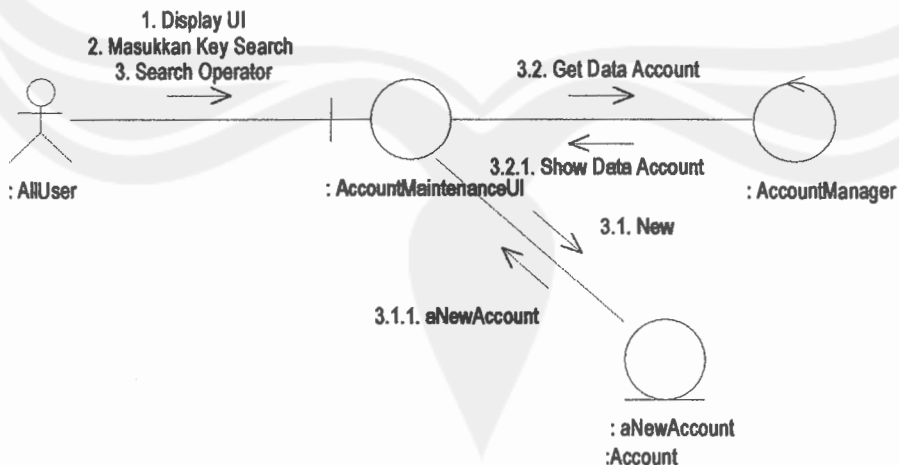
4.2.2. Analisis Collaboration Diagram : Use Case Pengelolaan Data Operator

4.2.2.1. Penambahan Data Operator



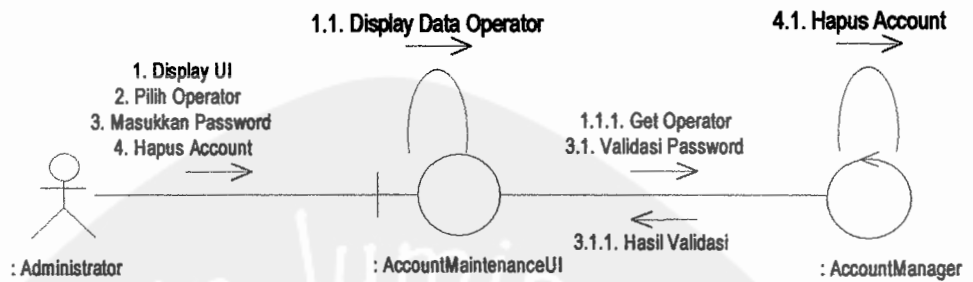
Gambar 4.26 Analisis Collaboration Diagram : Use Case Pengolahan Data Operator - Penambahan Data Operator

4.2.2.2. Pencarian Data Operator



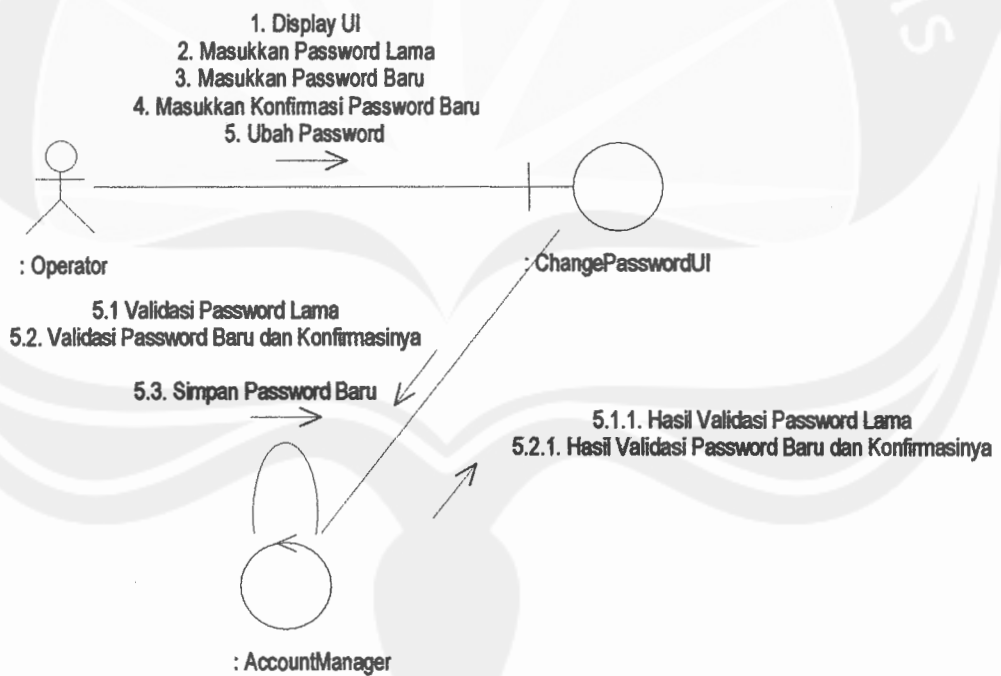
Gambar 4.27 Analisis Collaboration Diagram : Use Case Pengolahan Data Operator - Pencarian Data Operator

4.2.2.3. Penghapusan Data Operator



Gambar 4.28 Analysis Collaboration Diagram : Use Case Pengolahan Data Operator - Penghapusan Data Operator

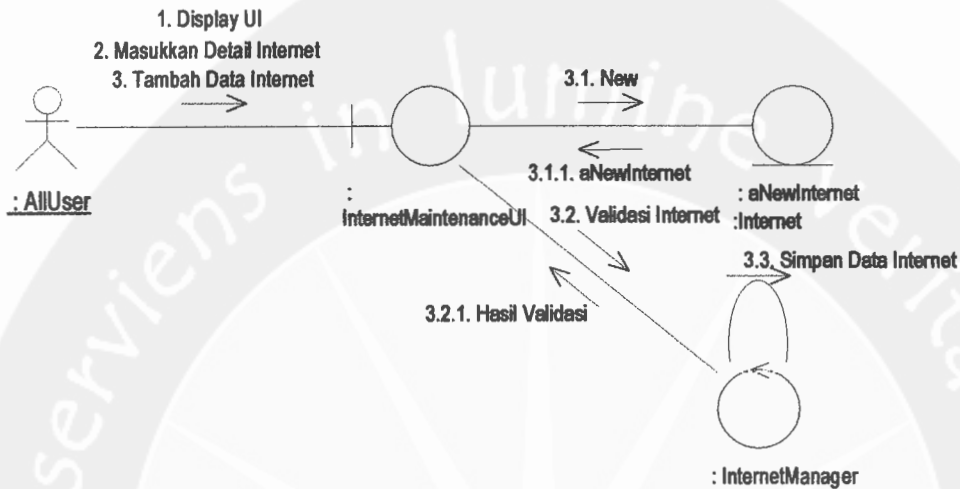
4.2.3. Analisis Collaboration Diagram : Use Case Ubah Password



Gambar 4.29 Analysis Collaboration Diagram : Use Case Ubah Password

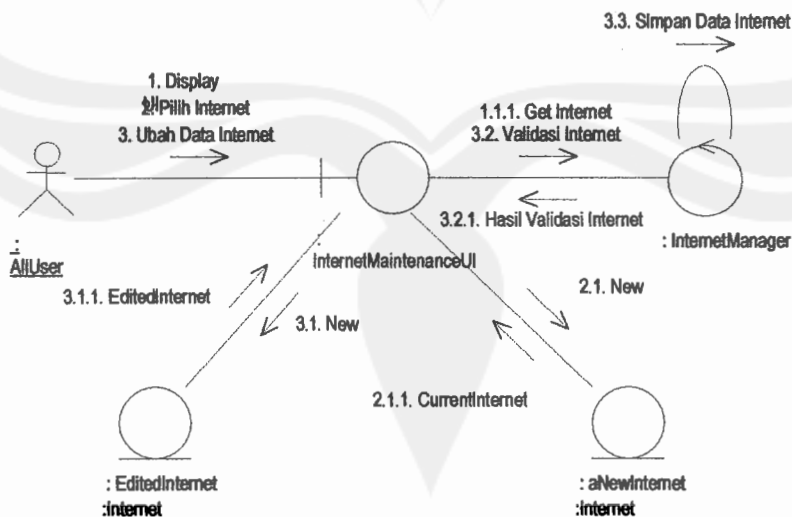
4.2.4. Analisis Collaboration Diagram : Use Case Periklanan.Elektronik.Pengelolaan Periklanan Internet

4.2.4.1. Penambahan Data Periklanan Internet



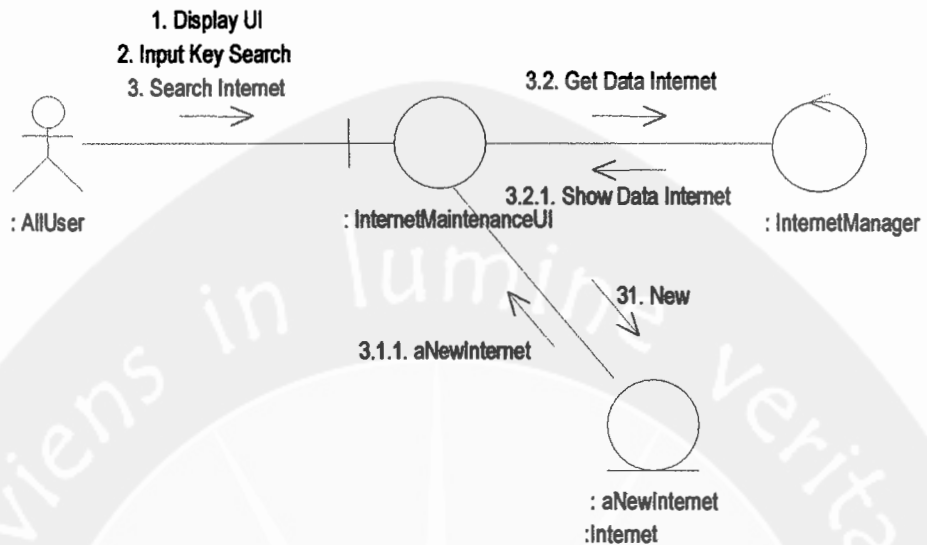
Gambar 4.30 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Internet - Penambahan Data Periklanan Internet

4.2.4.2. Pengubahan Data Periklanan Internet



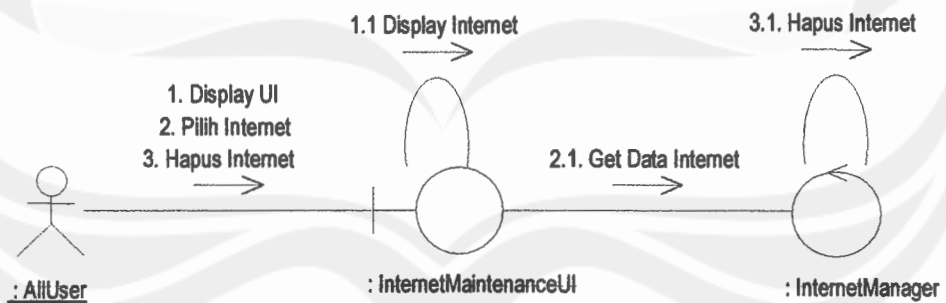
Gambar 4.31 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Internet - Pengubahan Data Periklanan Internet

4.2.4.3. Pencarian Data Periklanan Internet



Gambar 4.32 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Internet - Pencarian Data Periklanan Internet

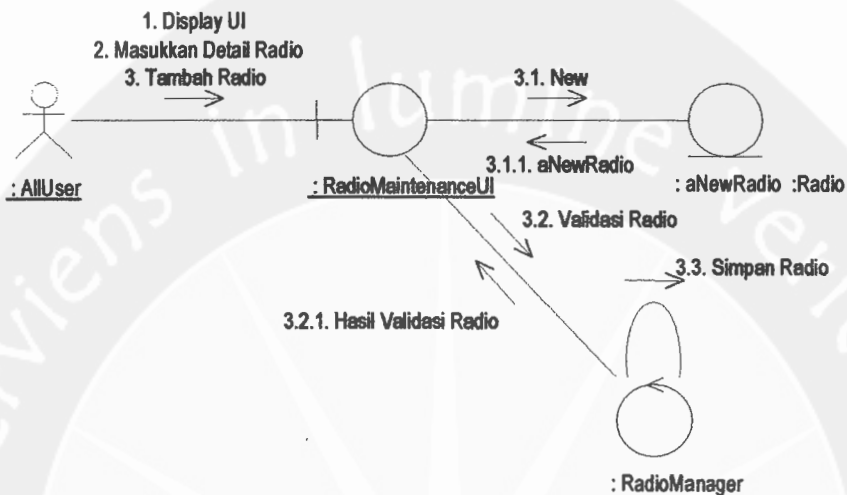
4.2.4.4. Penghapusan Data Periklanan Internet



Gambar 4.33 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Internet - Penghapusan Data Periklanan Internet

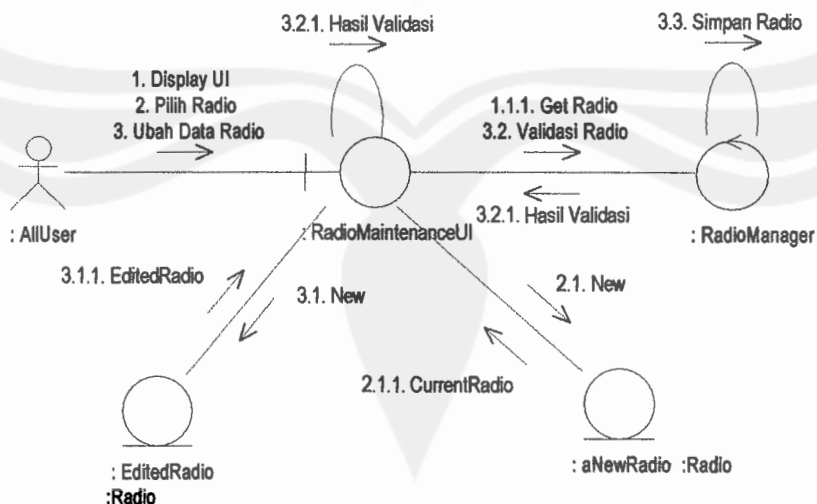
4.2.5. Analisis Collaboration Diagram : Use Case Periklanan.Elektronik.Pengelolaan Periklanan Radio

4.2.5.1. Penambahan Data Periklanan Radio



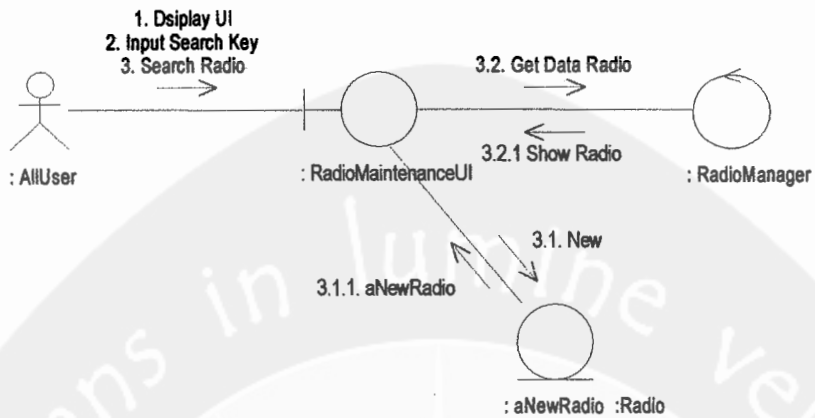
Gambar 4.34 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Radio - Penambahan Data Periklanan Radio

4.2.5.2. Pengubahan Data Periklanan Radio



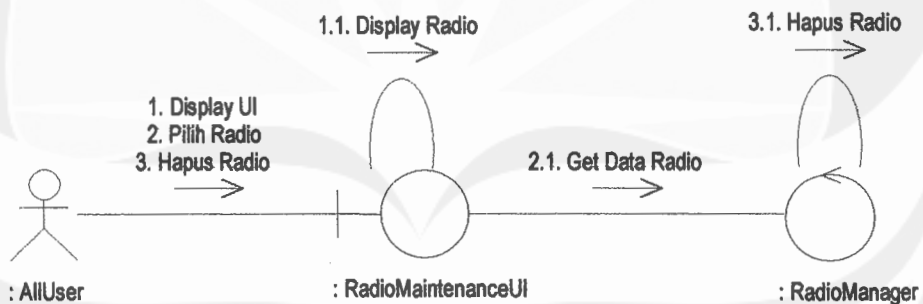
Gambar 4.35 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Radio - Pengubahan Data Periklanan Radio

4.2.5.3. Pencarian Data Periklanan Radio



Gambar 4.36 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Radio - Pencarian Data Periklanan Radio

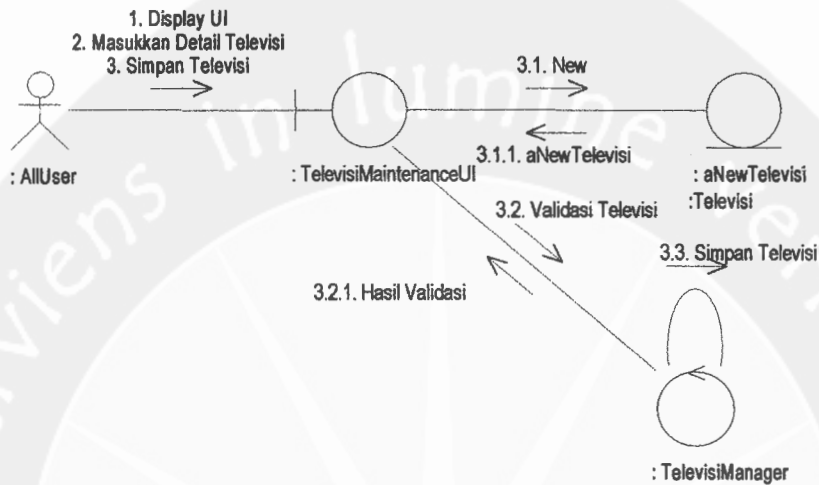
4.2.5.4. Penghapusan Data Periklanan Radio



Gambar 4.37 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Radio - Penghapusan Data Periklanan Radio

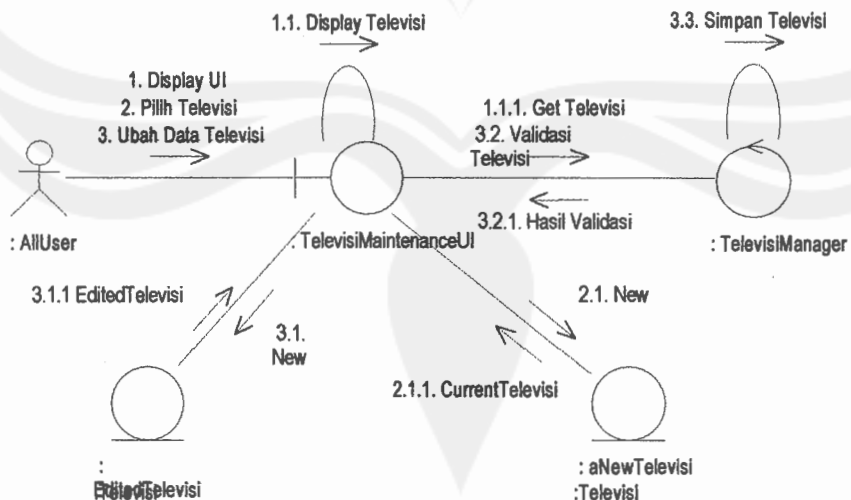
4.2.6. Analisis Collaboration Diagram : Use Case Periklanan.Elektronik.Pengelolaan Periklanan Televisi

4.2.6.1. Penambahan Data Periklanan Televisi



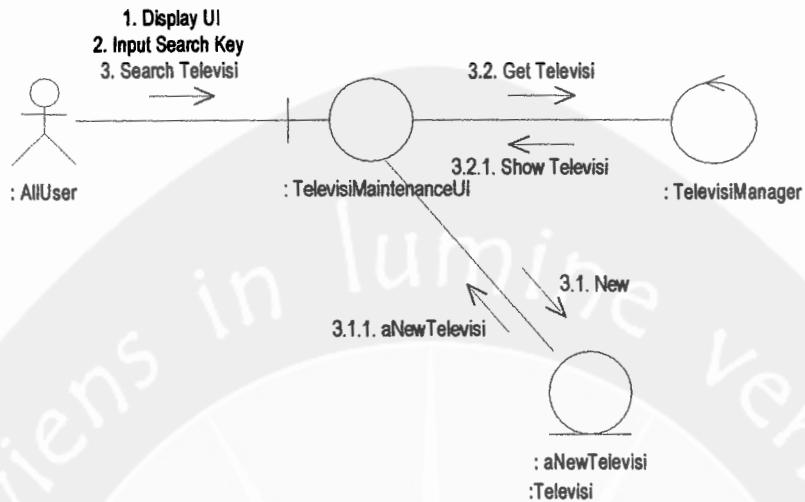
Gambar 4.38 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Televisi - Penambahan Data Periklanan Televisi

4.2.6.2. Pengubahan Data Periklanan Televisi



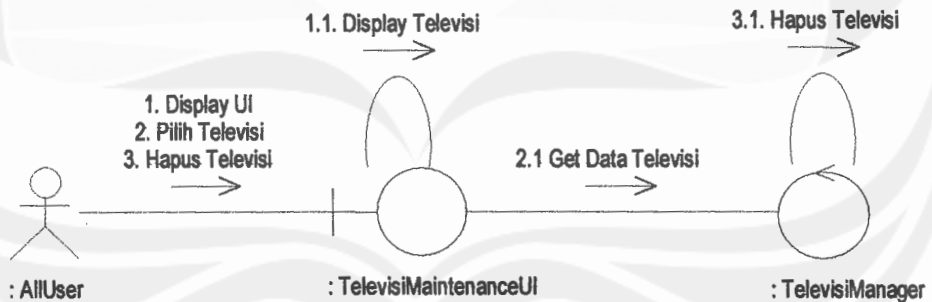
Gambar 4.39 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Televisi - Pengubahan Data Periklanan Televisi

4.2.6.3. Pencarian Data Periklanan Televisi



Gambar 4.40 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Televisi - Pencarian Data Periklanan Televisi

4.2.6.4. Penghapusan Data Periklanan Televisi

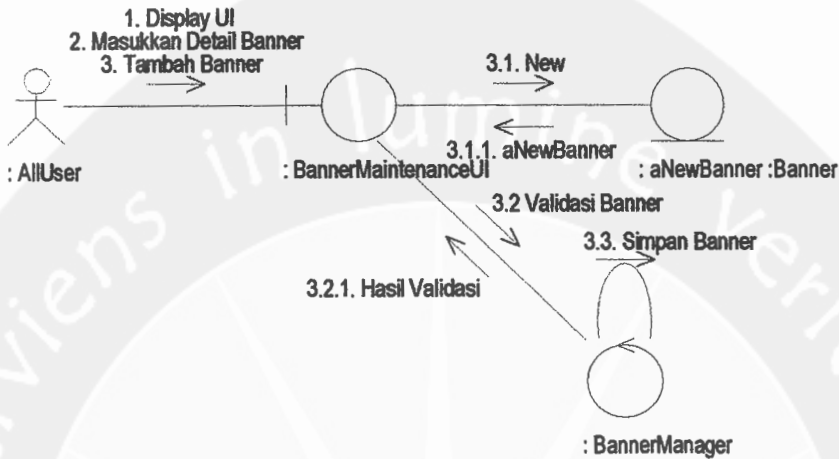


Gambar 4.41 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Televisi - Penghapusan Data Periklanan Televisi

4.2.7. Analisis Collaboration Diagram : Use Case Periklanan.MediaCetak.MediaCetakIndoor.

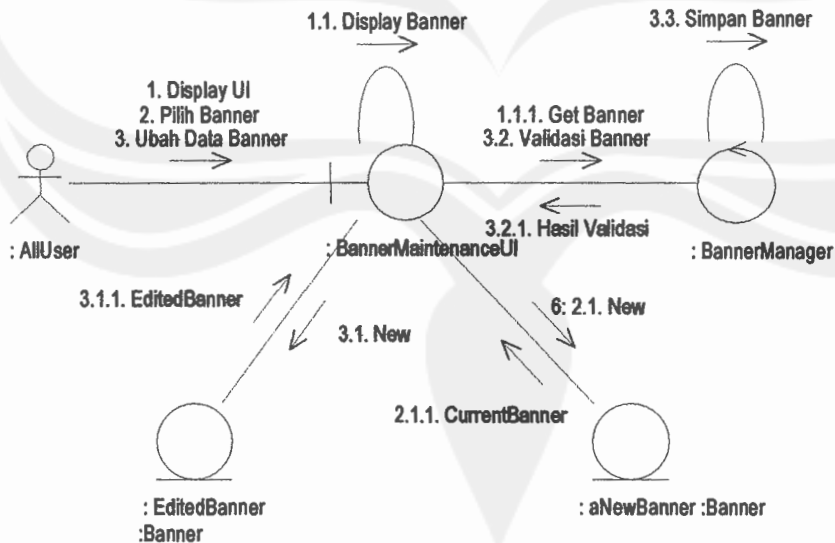
Pengelolaan Periklanan Banner

4.2.7.1. Penambahan Data Periklanan Banner



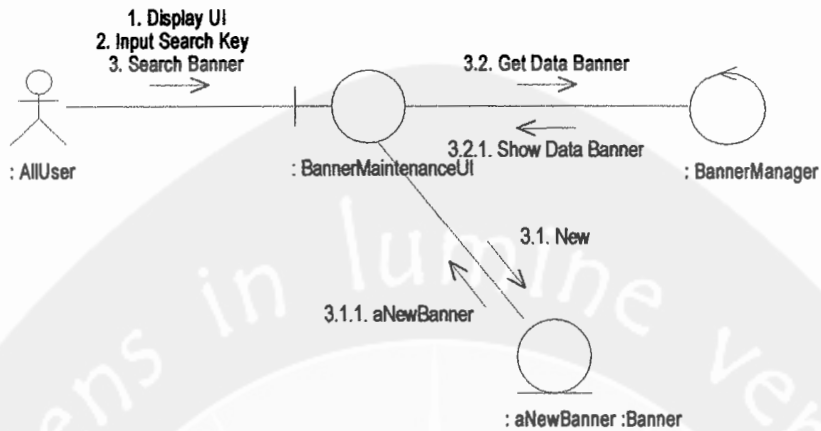
Gambar 4.42 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Banner - Penambahan Data Periklanan Banner

4.2.7.2. Pengubahan Data Periklanan Banner



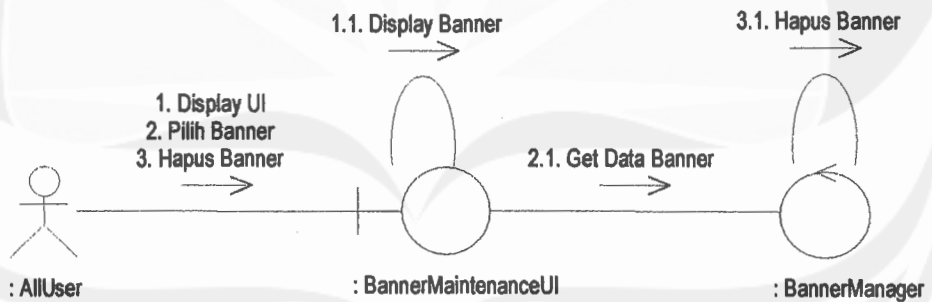
Gambar 4.43 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Banner - Pengubahan Data Periklanan Banner

4.2.7.3. Pencarian Data Periklanan Banner



Gambar 4.44 Analisis Collaboration Diagram : Use Case Pengelolaan Periklanan Banner - PencarianData Periklanan Banner

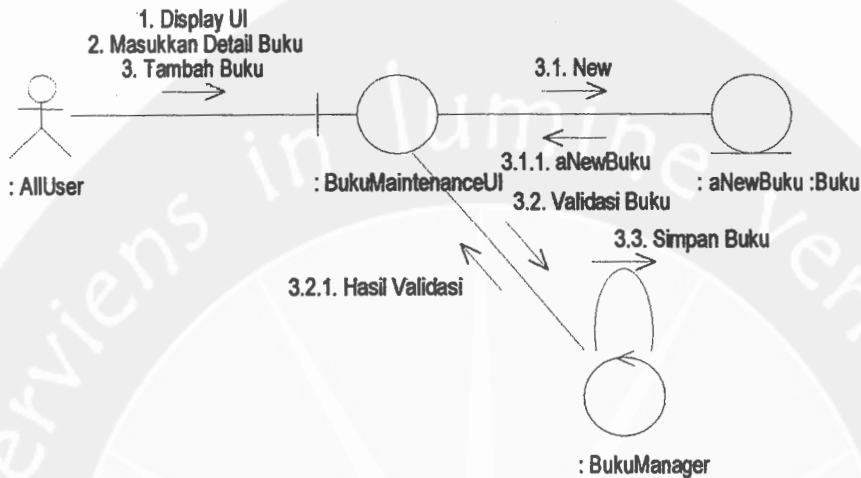
4.2.7.4. Penghapusan Data Periklanan Banner



Gambar 4.45 Analisis Collaboration Diagram : Use Case Pengelolaan Periklanan Banner - Penghapusan Data Periklanan Banner

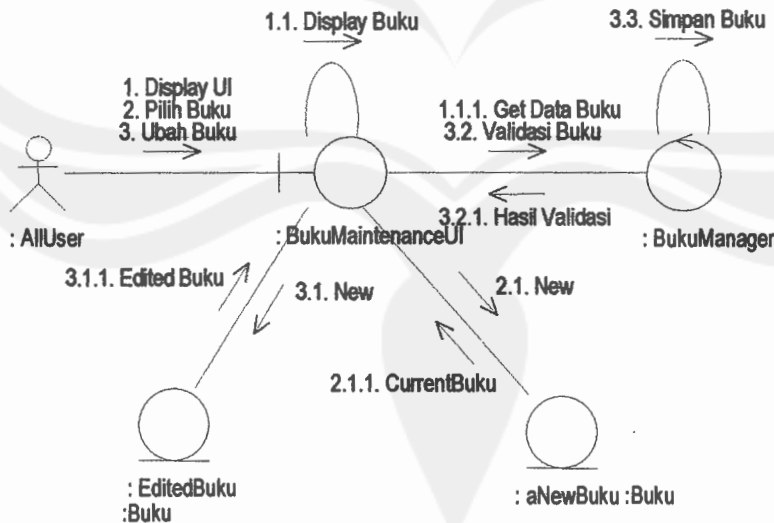
4.2.8. Analisis Collaboration Diagram : Use Case Periklanan.MediaCetak.MediaCetakIndoor. Pengelolaan Periklanan Buku

4.2.8.1. Penambahan Data Periklanan Buku



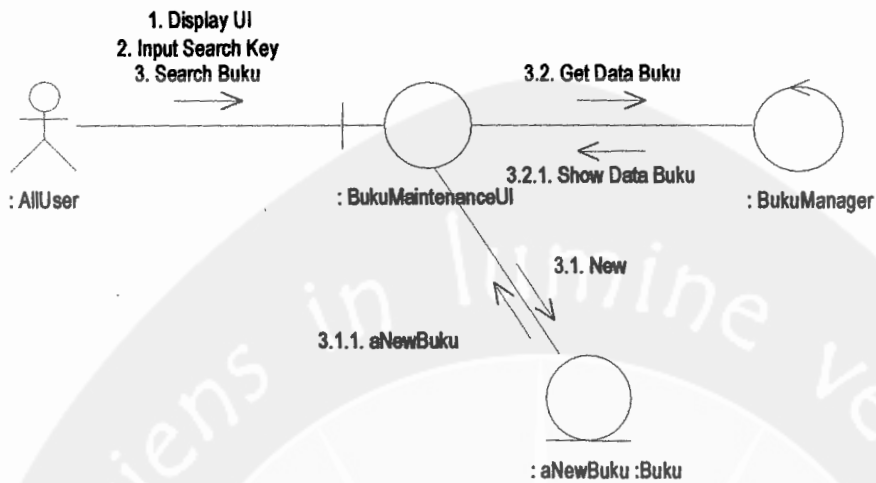
Gambar 4.46 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Buku - Penambahan Data Periklanan Buku

4.2.8.2. Pengubahan Data Periklanan Buku



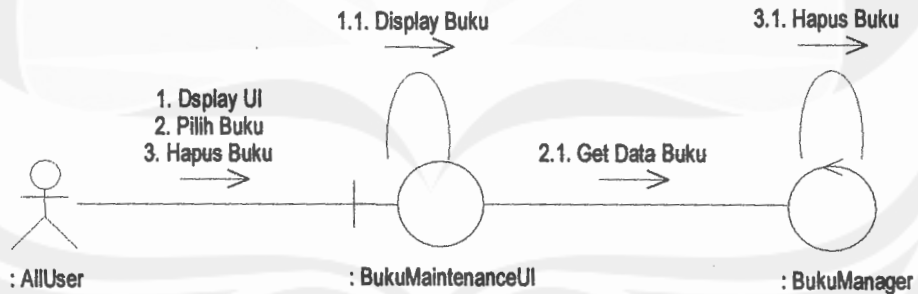
Gambar 4.47 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Banner - Pengubahan Data Periklanan Banner

4.2.8.3. Pencarian Data Periklanan Buku



Gambar 4.48 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Buku - PencarianData Periklanan Buku

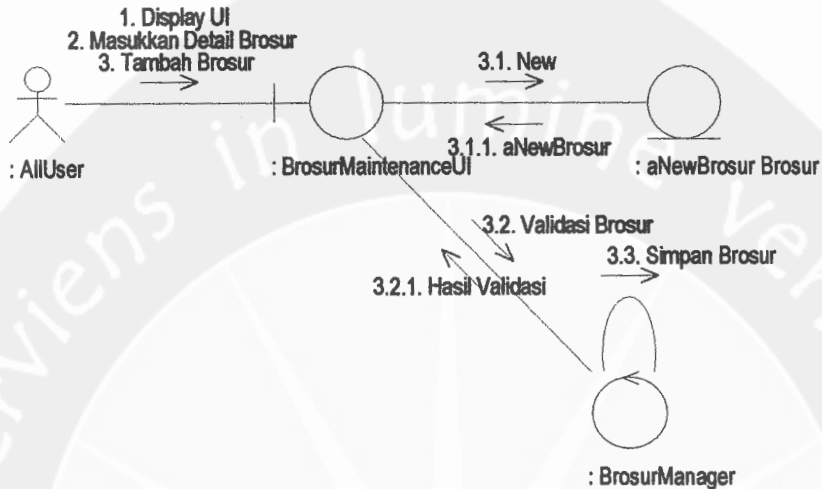
4.2.8.4. Penghapusan Data Periklanan Buku



Gambar 4.49 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Buku - Penghapusan Data Periklanan Buku

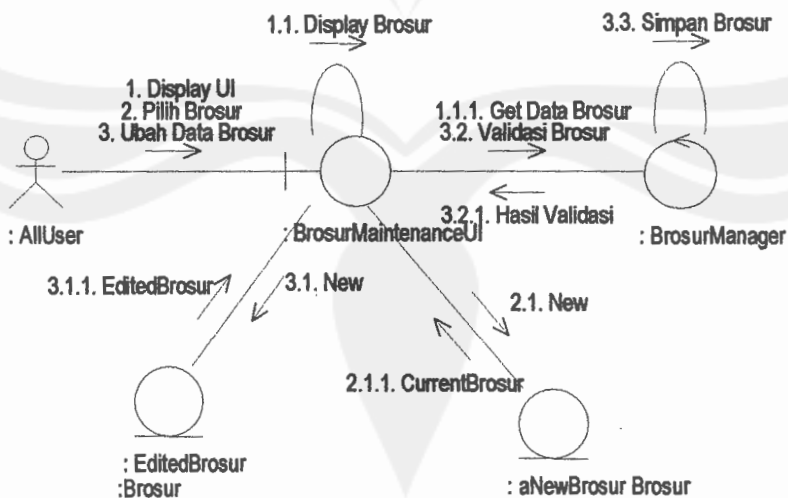
4.2.9. Analisis Collaboration Diagram : Use Case Periklanan.MediaCetak.MediaCetakIndoor. Pengelolaan Periklanan Brosur

4.2.9.1. Penambahan Data Periklanan Brosur



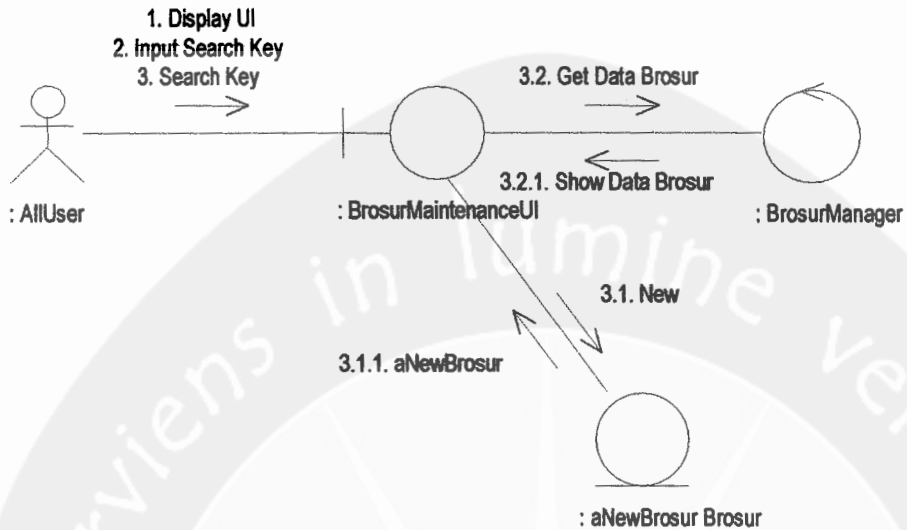
Gambar 4.50 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Brosur - Penambahan Data Periklanan Brosur

4.2.9.2. Pengubahan Data Periklanan Brosur



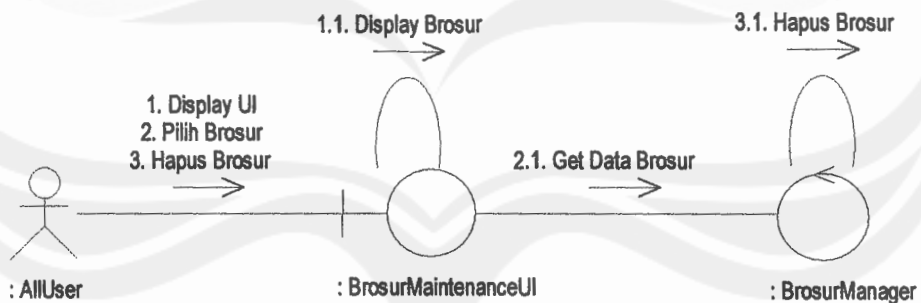
Gambar 4.51 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Brosur - Pengubahan Data Periklanan Brosur

4.2.9.3. Pencarian Data Periklanan Brosur



Gambar 4.52 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Brosur - PencarianData Periklanan Brosur

4.2.9.4. Penghapusan Data Periklanan Brosur

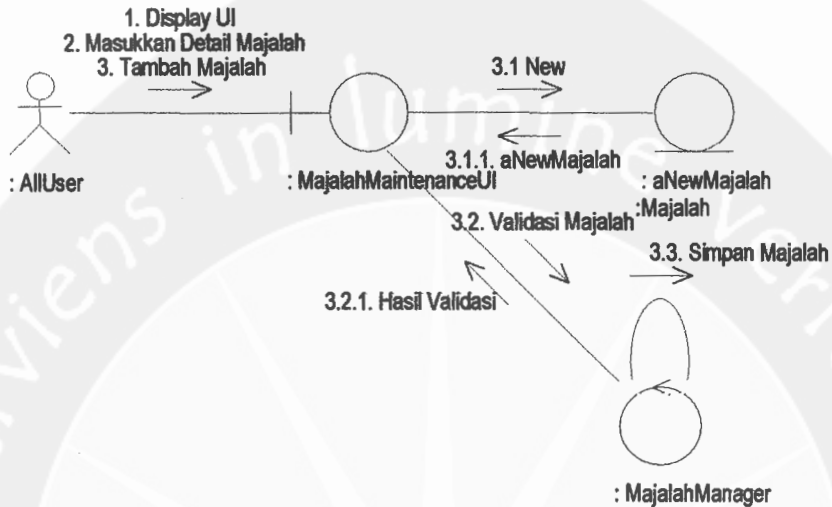


Gambar 4.53 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Brosur - Penghapusan Data Periklanan Brosur

4.2.10. Analisis Collaboration Diagram : Use Case Periklanan.MediaCetak.MediaCetakIndoor.

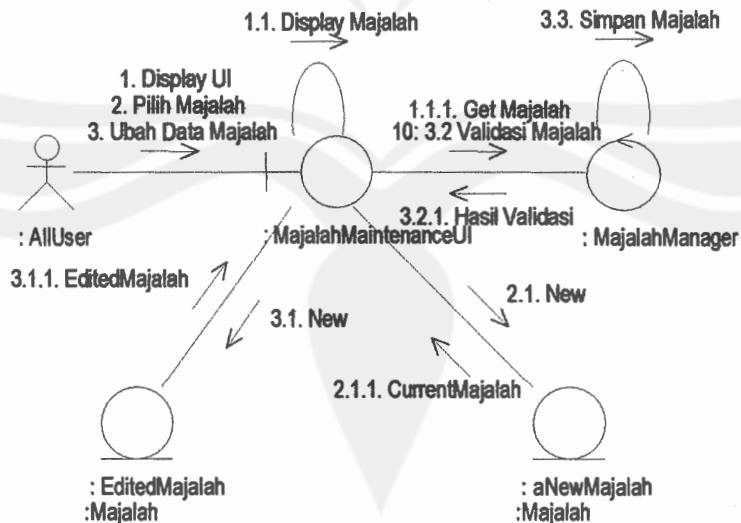
Pengelolaan Periklanan Majalah

4.2.10.1. Penambahan Data Periklanan Majalah



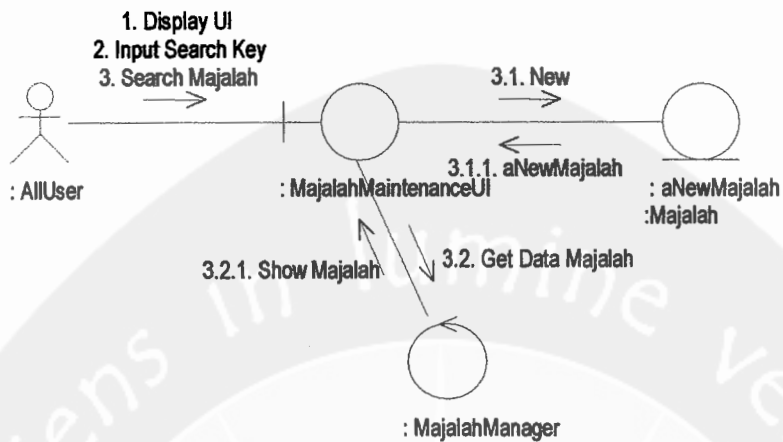
Gambar 4.54 Analisis Collaboration Diagram : Use Case Pengelolaan Periklanan Majalah - Penambahan Data Periklanan Majalah

4.2.10.2. Perubahan Data Periklanan Majalah



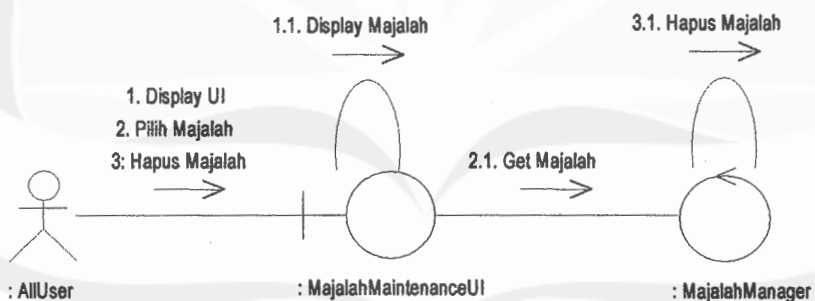
Gambar 4.55 Analisis Collaboration Diagram : Use Case Pengelolaan Periklanan Majalah - Perubahan Data Periklanan Majalah

4.2.10.3. Pencarian Data Periklanan Majalah



Gambar 4.56 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Majalah - PencarianData Periklanan Majalah

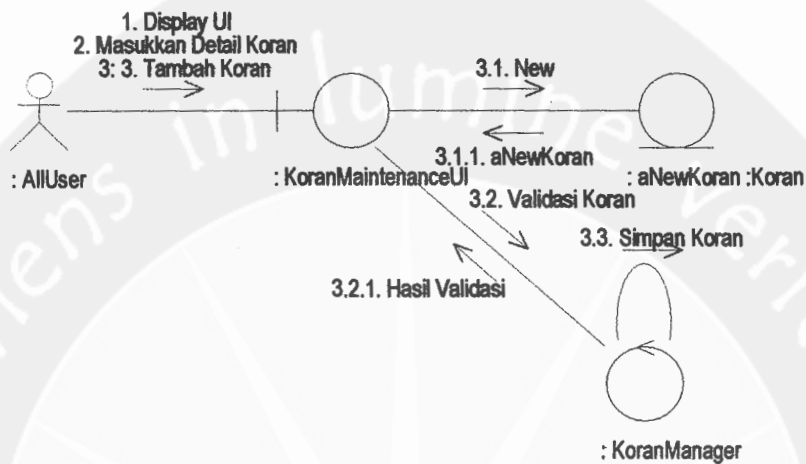
4.2.10.4. Penghapusan Data Periklanan Majalah



Gambar 4.57 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Majalah - Penghapusan Data Periklanan Majalah

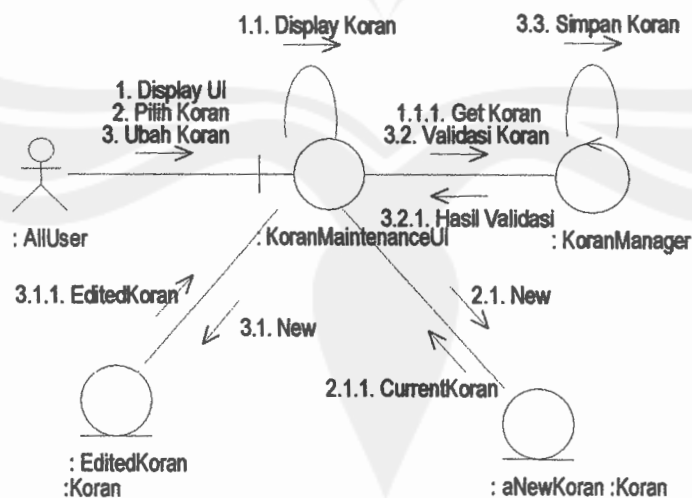
4.2.11. Analisis Collaboration Diagram : Use Case Periklanan.MediaCetak.MediaCetakIndoor. Pengelolaan Periklanan Koran

4.2.11.1. Penambahan Data Periklanan Koran



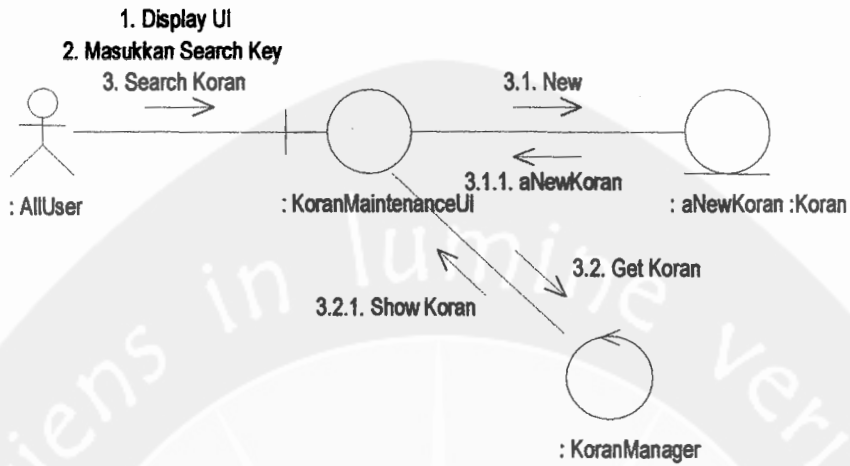
Gambar 4.58 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Koran - Penambahan Data Periklanan Koran

4.2.11.2. Pengubahan Data Periklanan Koran



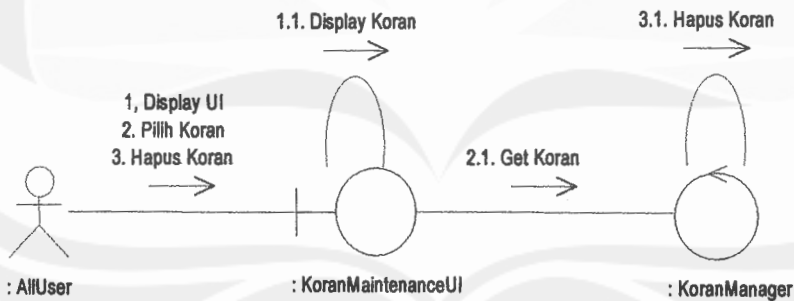
Gambar 4.59 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Koran - Pengubahan Data Periklanan Koran

4.2.11.3. Pencarian Data Periklanan Koran



Gambar 4.60 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Koran - PencarianData Periklanan Koran

4.2.11.4. Penghapusan Data Periklanan Koran

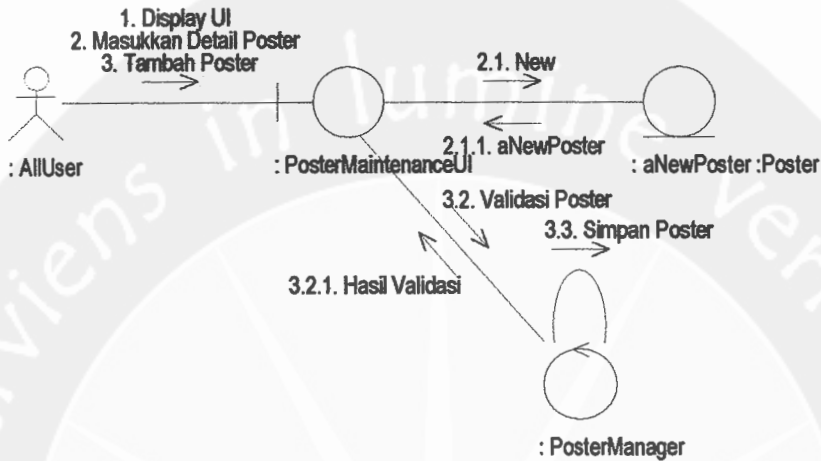


Gambar 4.61 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Koran - Penghapusan Data Periklanan Koran

4.2.12. Analisis Collaboration Diagram : Use Case Periklanan.MediaCetak.MediaCetakIndoor.

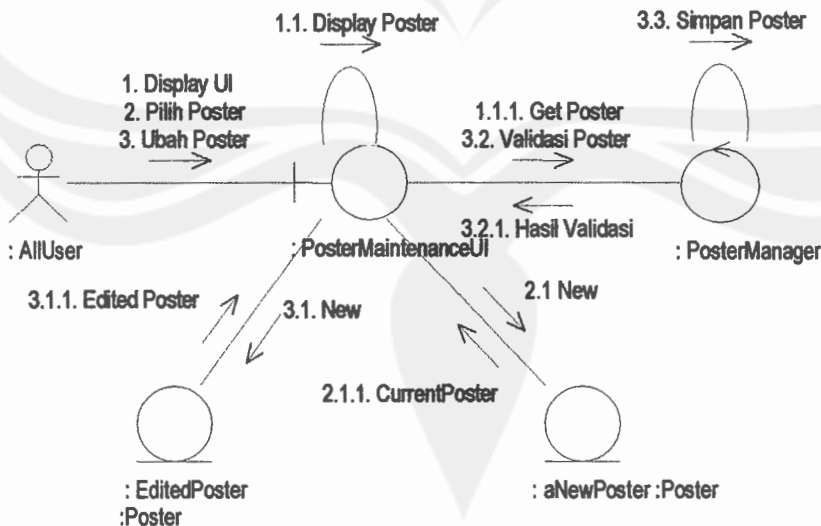
Pengelolaan Periklanan Poster

4.2.12.1. Penambahan Data Periklanan Poster



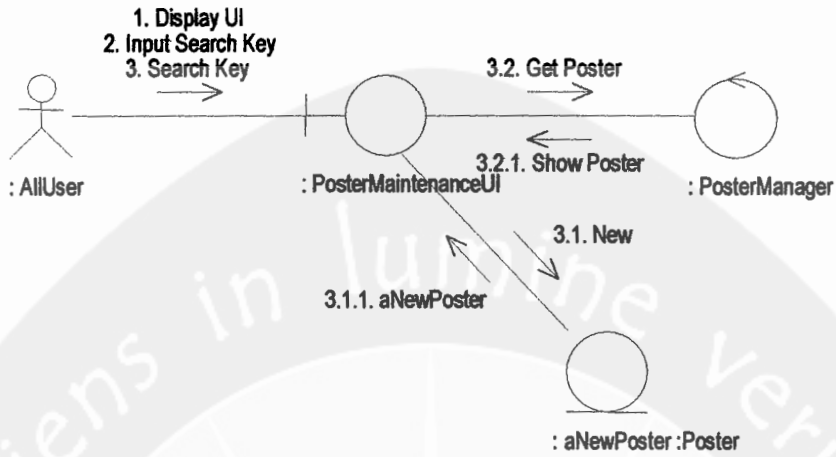
Gambar 4.62 Analisis Collaboration Diagram : Use Case Pengelolaan Periklanan Poster - Penambahan Data Periklanan Poster

4.2.12.2. Pengubahan Data Periklanan Poster



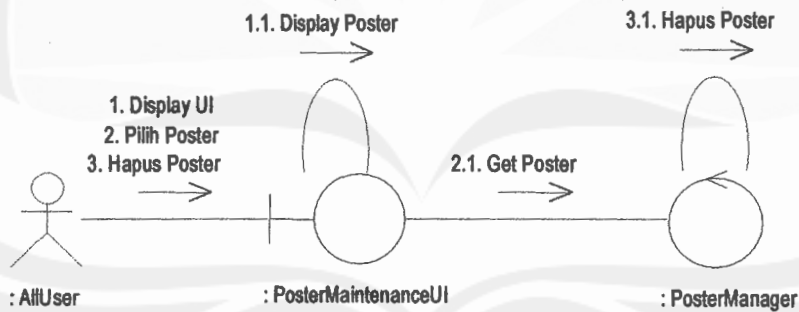
Gambar 4.63 Analisis Collaboration Diagram : Use Case Pengelolaan Periklanan Poster - Pengubahan Data Periklanan Poster

4.2.12.3. Pencarian Data Periklanan Poster



Gambar 4.64 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Poster - PencarianData Periklanan Poster

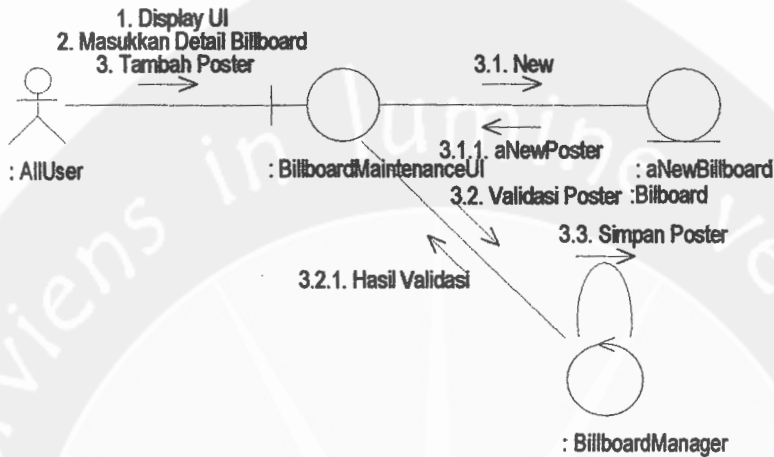
4.2.12.4. Penghapusan Data Periklanan Poster



Gambar 4.65 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Poster - Penghapusan Data Periklanan Poster

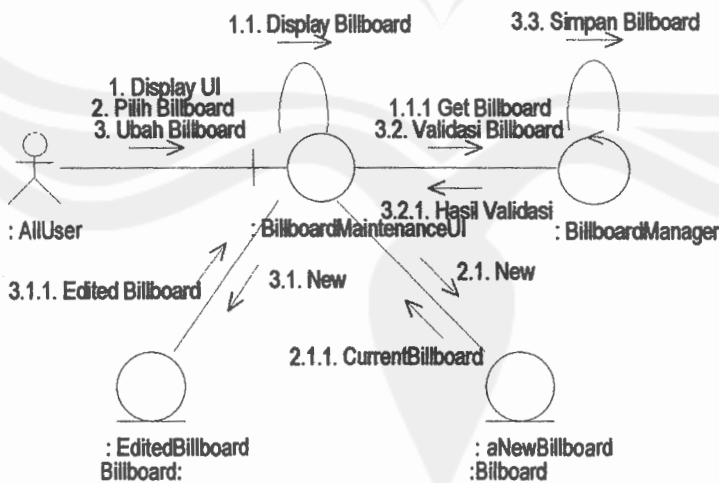
4.2.13. Analisis Collaboration Diagram : Use Case Periklanan.MediaCetak.MediaCetakOutdoor. Pengelolaan Periklanan Billboard

4.2.13.1. Penambahan Data Periklanan Billboard



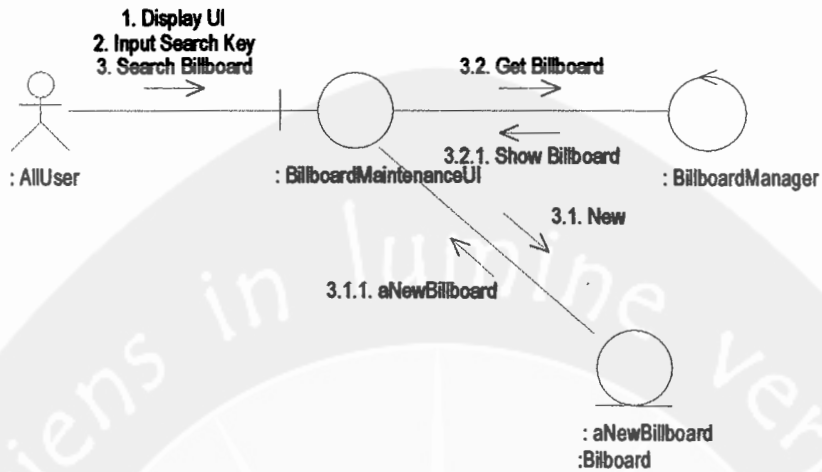
Gambar 4.66 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Billboard - Penambahan Data Periklanan Billboard

4.2.13.2. Pengubahan Data Periklanan Billboard



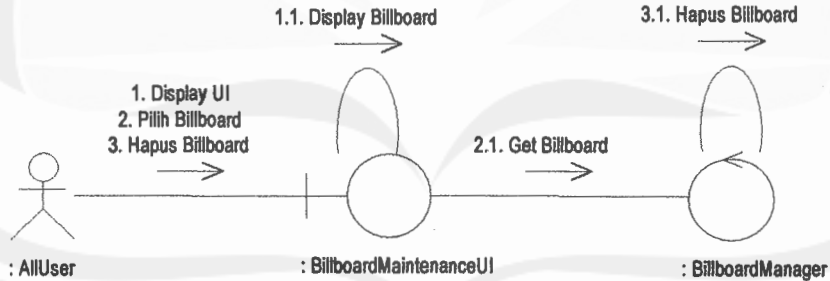
Gambar 4.67 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Billboard-Pengubahan Data Periklanan Billboard

4.2.13.3. Pencarian Data Periklanan Billboard



Gambar 4.68 Analisis Collaboration Diagram : Use Case Pengelolaan Periklanan Billboard - PencarianData Periklanan Billboard

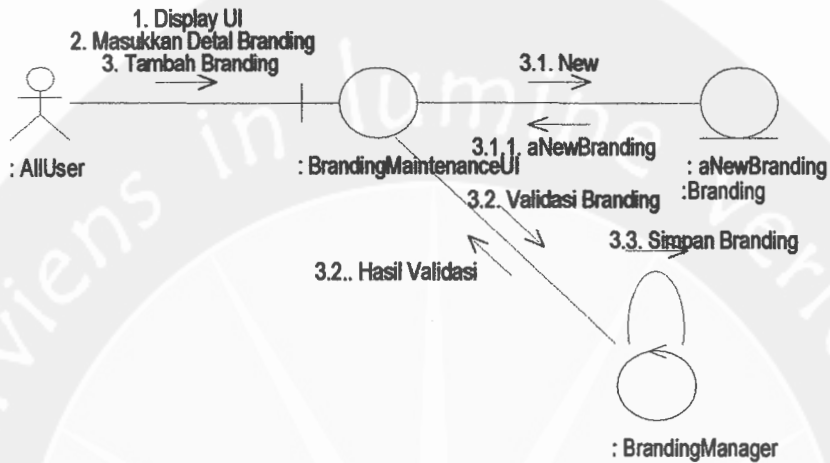
4.2.13.4. Penghapusan Data Periklanan Billboard



Gambar 4.69 Analisis Collaboration Diagram : Use Case Pengelolaan Periklanan Billboard - Penghapusan Data Periklanan Billboard

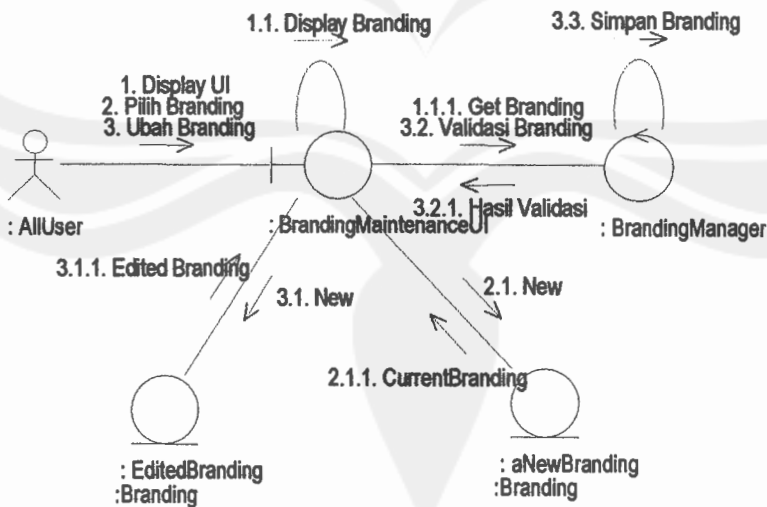
4.2.14. Analisis Collaboration Diagram : Use Case Periklanan.MediaCetak.MediaCetakOutdoor. Pengelolaan Periklanan Branding

4.2.14.1. Penambahan Data Periklanan Branding



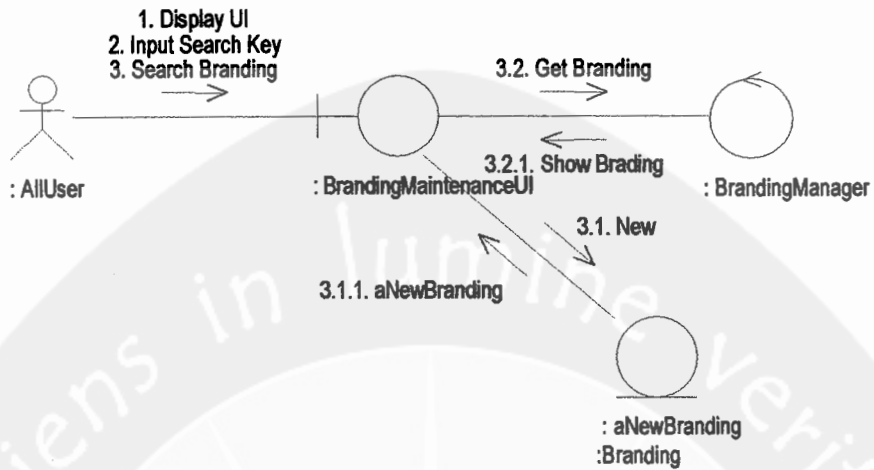
Gambar 4.70 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Branding - Penambahan Data Periklanan Branding

4.2.14.2. Pengubahan Data Periklanan Branding



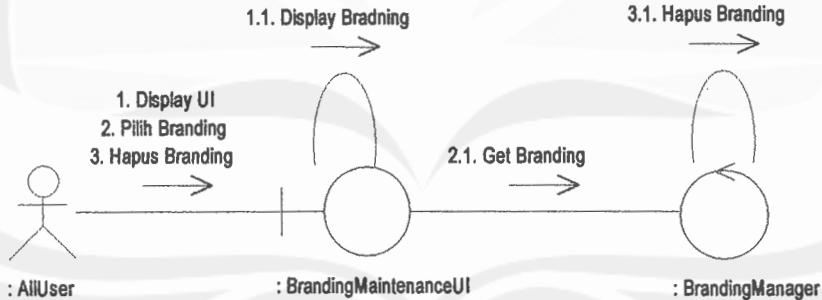
Gambar 4.71 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Branding - Pengubahan Data Periklanan Branding

4.2.14.3. Pencarian Data Periklanan Branding



Gambar 4.72 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Branding - PencarianData Periklanan Branding

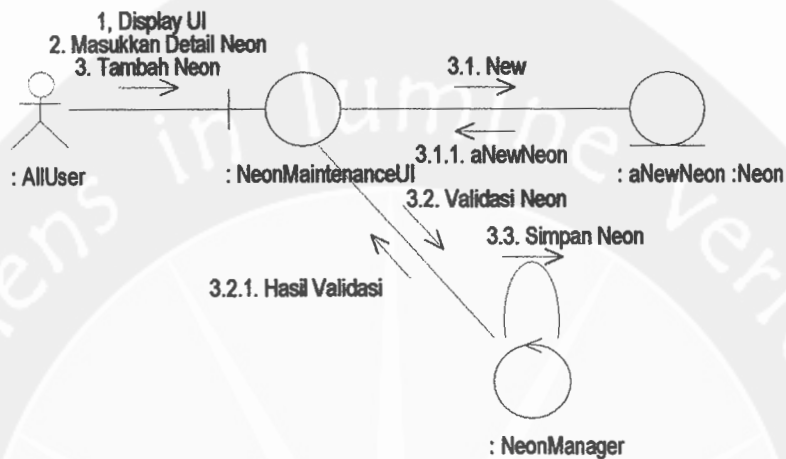
4.2.14.4. Penghapusan Data Periklanan Branding



Gambar 4.73 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Branding - Penghapusan Data Periklanan Branding

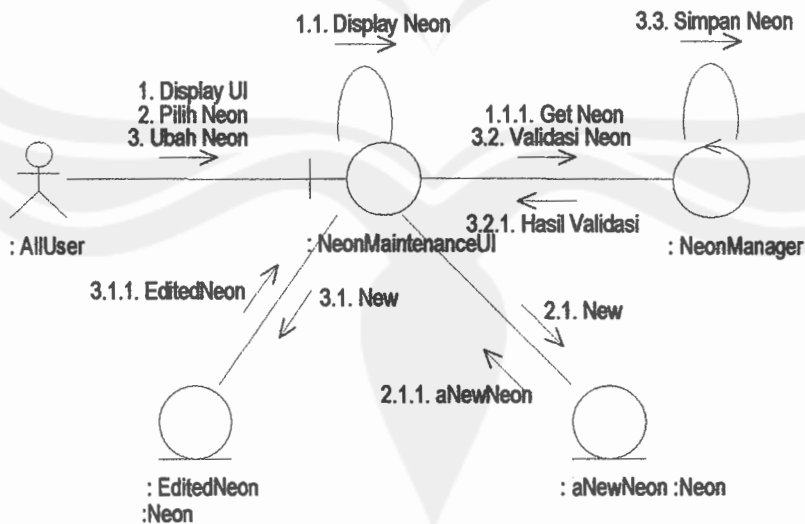
4.2.15. Analisis Collaboration Diagram : Use Case Periklanan.MediaCetak.MediaCetakOutdoor. Pengelolaan Periklanan Neon

4.2.15.1. Penambahan Data Periklanan Neon



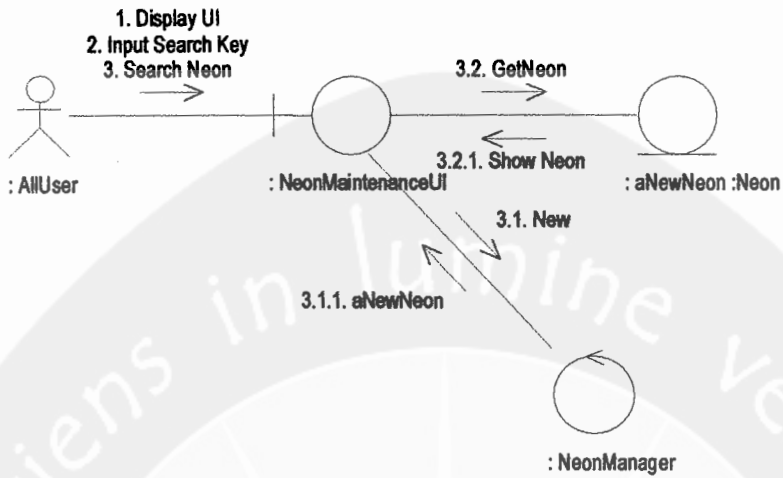
Gambar 4.74 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Neon - Penambahan Data Periklanan Neon

4.2.15.2. Pengubahan Data Periklanan Neon



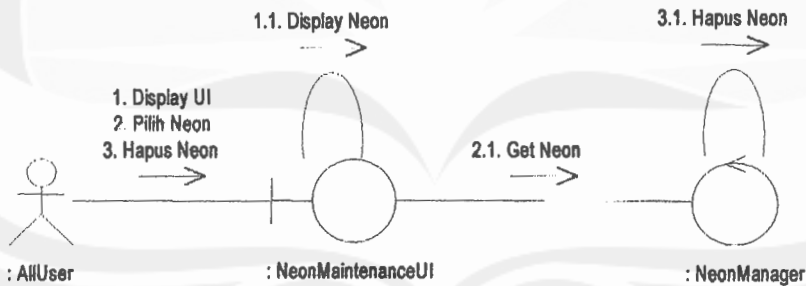
Gambar 4.75 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Neon - Pengubahan Data Periklanan Neon

4.2.15.3. Pencarian Data Periklanan Neon



Gambar 4.76 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Neon - PencarianData Periklanan Neon

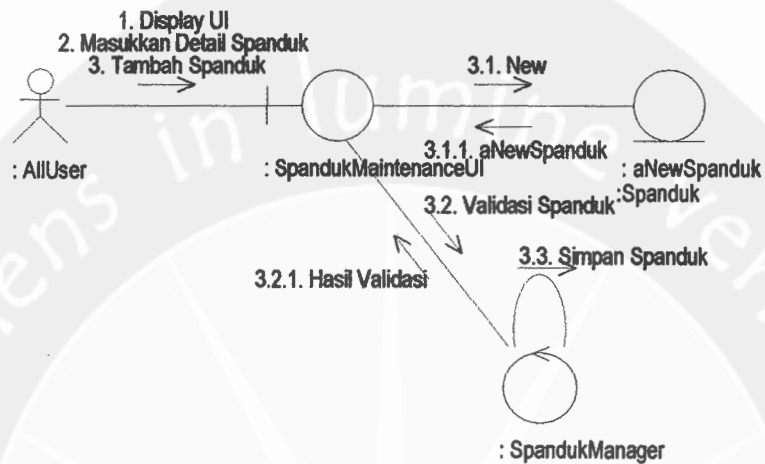
4.2.15.4. Penghapusan Data Periklanan Neon



Gambar 4.77 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Neon - Penghapusan Data Periklanan Neon

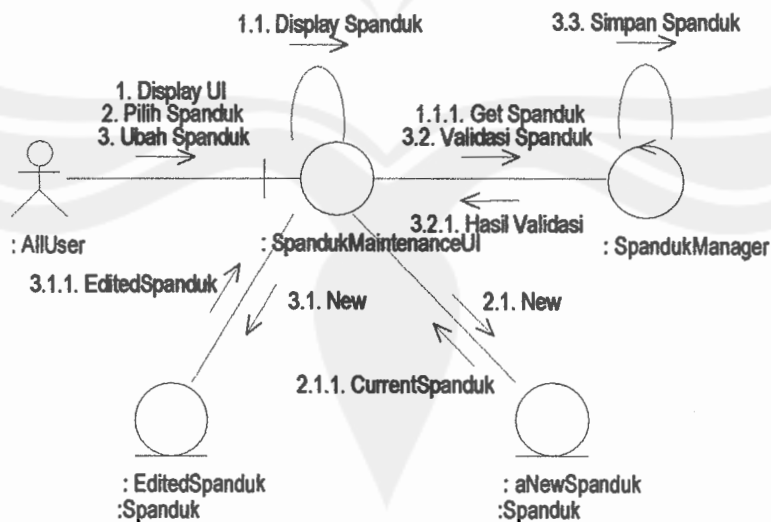
4.2.16. Analisis Collaboration Diagram : Use Case Periklanan.MediaCetak.MediaCetakOutdoor. Pengelolaan Periklanan Spanduk

4.2.16.1. Penambahan Data Periklanan Spanduk



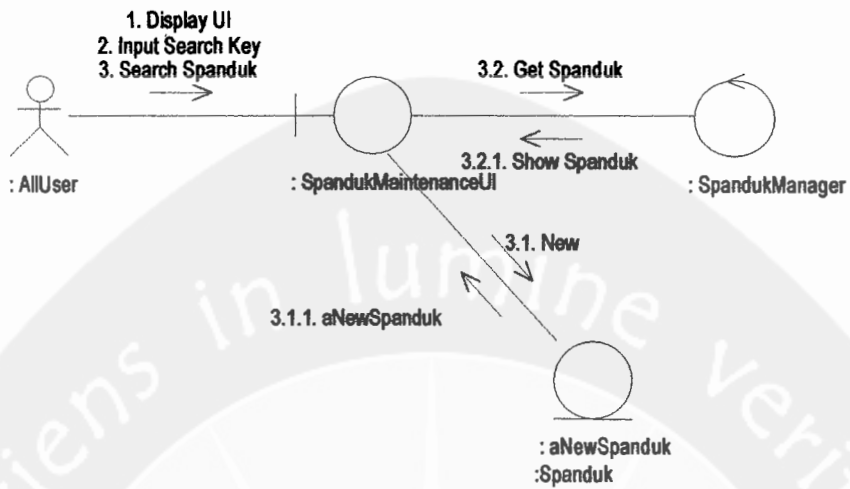
Gambar 4.78 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Spanduk - Penambahan Data Periklanan Spanduk

4.2.16.2. Pengubahan Data Periklanan Spanduk



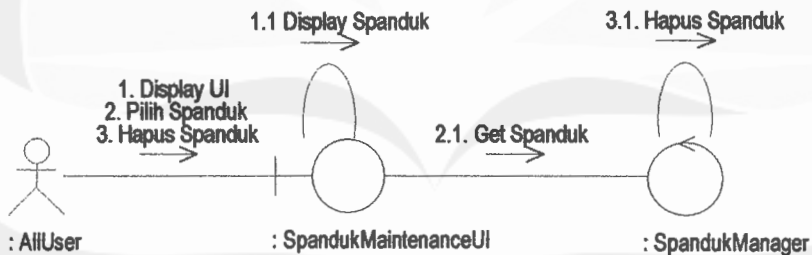
Gambar 4.79 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Spanduk - Pengubahan Data Periklanan Spanduk

4.2.16.3. Pencarian Data Periklanan Spanduk



Gambar 4.80 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Spanduk - PencarianData Periklanan Spanduk

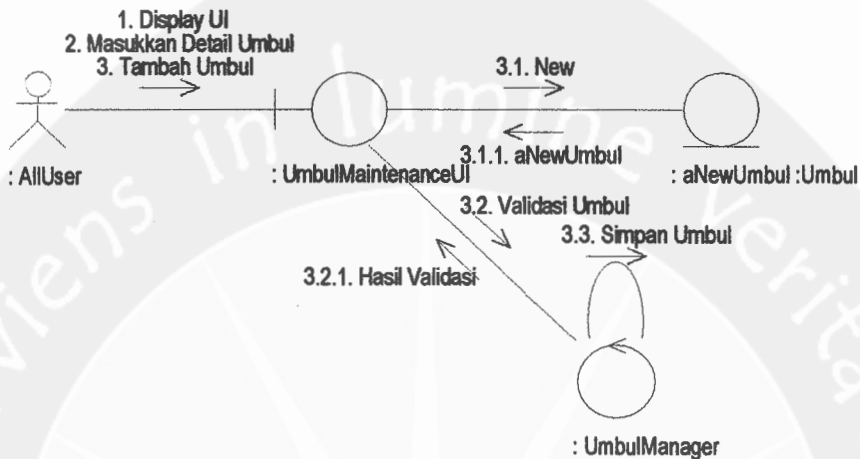
4.2.16.4. Penghapusan Data Periklanan Spanduk



Gambar 4.81 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Spanduk - Penghapusan Data Periklanan Spanduk

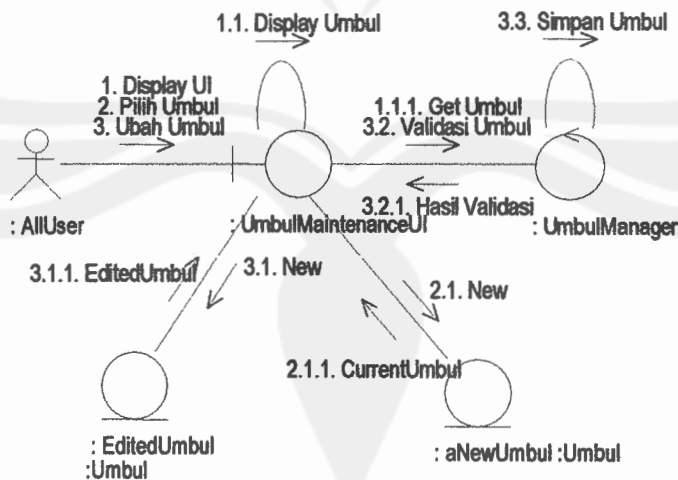
4.2.17. Analisis Collaboration Diagram : Use Case Periklanan.MediaCetak.MediaCetakOutdoor. Pengelolaan Periklanan Umbul

4.2.17.1. Penambahan Data Periklanan Umbul



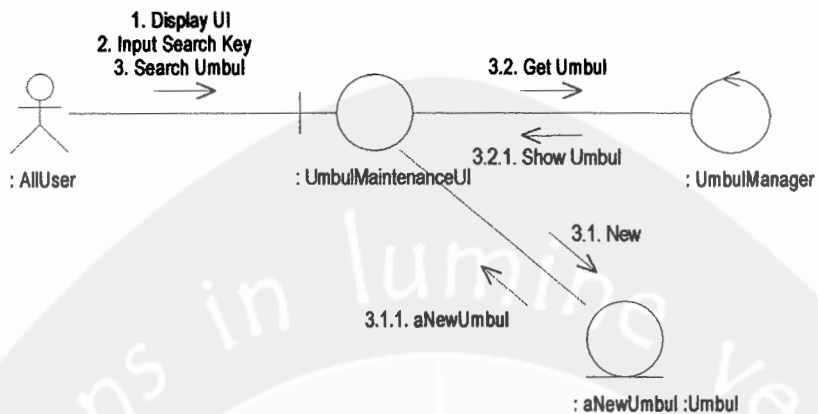
Gambar 4.82 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Umbul - Penambahan Data Periklanan Umbul

4.2.17.2. Pengubahan Data Periklanan Umbul



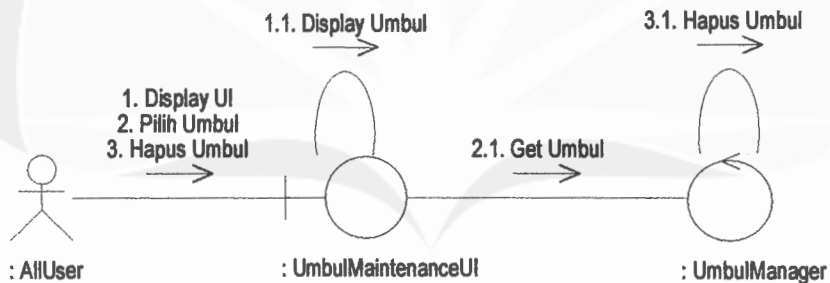
Gambar 4.83 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Umbul - Pengubahan Data Periklanan Umbul

4.2.17.3. Pencarian Data Periklanan Umbul



Gambar 4.84 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Umbul - PencarianData Periklanan Umbul

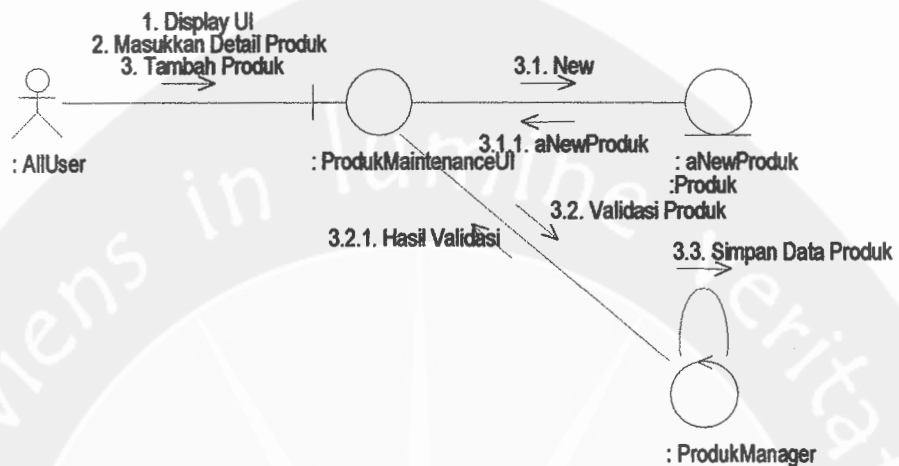
4.2.17.4. Penghapusan Data Periklanan Umbul



Gambar 4.85 Analysis Collaboration Diagram : Use Case Pengelolaan Periklanan Umbul - Penghapusan Data Periklanan Umbul

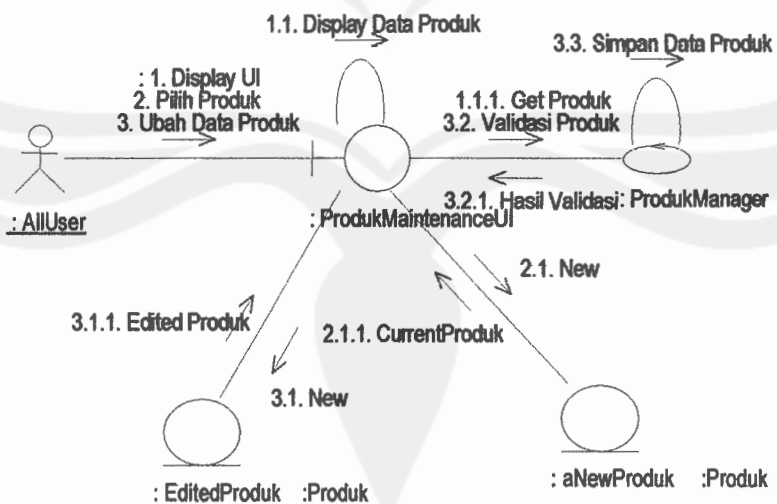
4.2.18. Analisis Collaboration Diagram : Use Case Pengelolaan Data Produk

4.2.18.1. Penambahan Data Produk



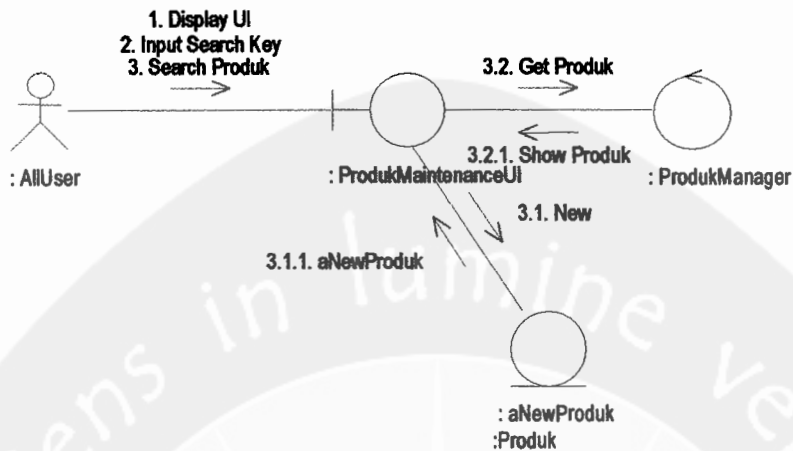
Gambar 4.86 Analysis Collaboration Diagram : Use Case Pengelolaan Data Produk - Penambahan Data Produk

4.2.18.2. Pengubahan Data Produk



Gambar 4.87 Analysis Collaboration Diagram : Use Case Pengelolaan Data Produk - Pengubahan Data Produk

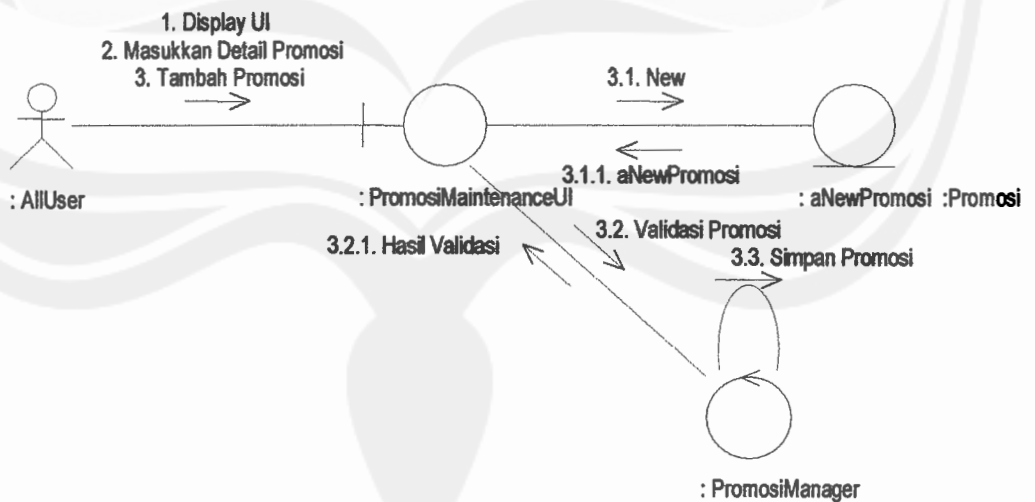
4.2.18.3. Pencarian Data Produk



Gambar 4.88 Analisis Collaboration Diagram : Use Case Pengelolaan Data Produk - Pencarian Data Produk

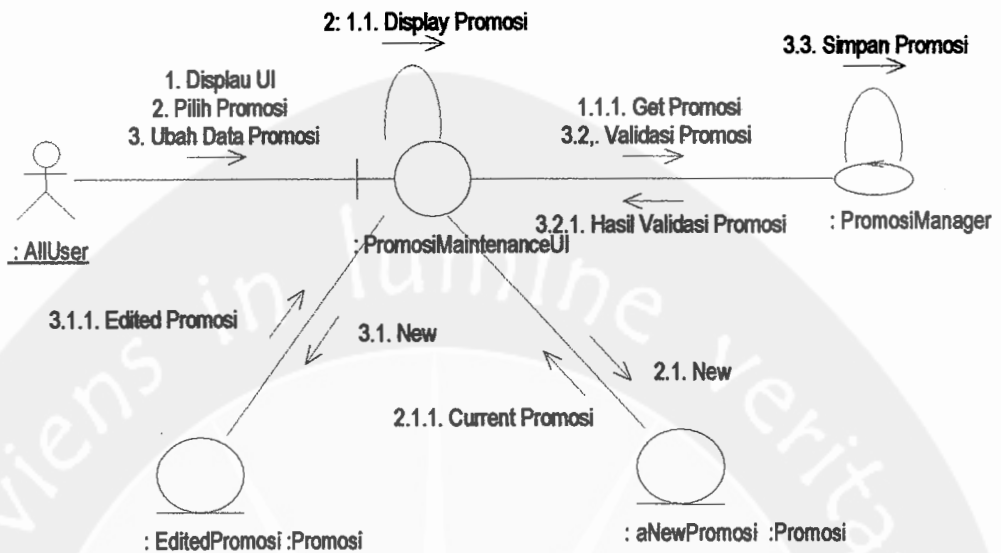
4.2.19. Analisis Collaboration Diagram : Use Case Pengelolaan Data Promosi

4.2.19.1. Penambahan Data Promosi



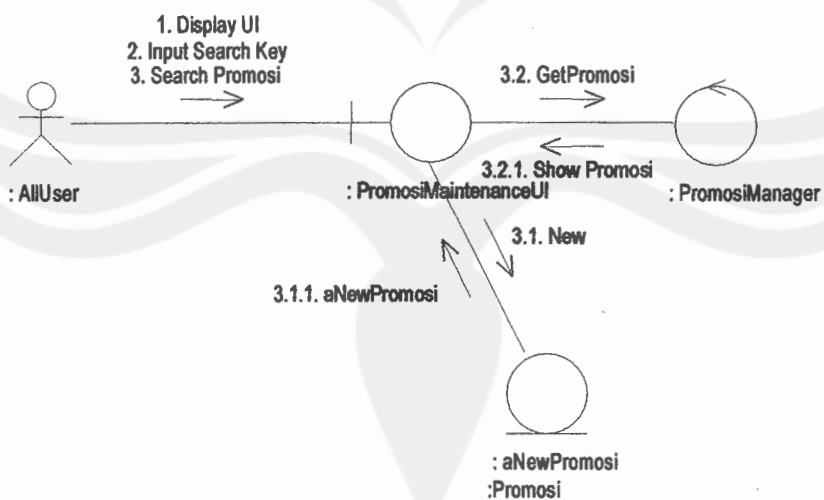
Gambar 4.89 Analisis Collaboration Diagram : Use Case Pengelolaan Data Promosi - Penambahan Data Promosi

4.2.19.2. Perubahan Data Promosi



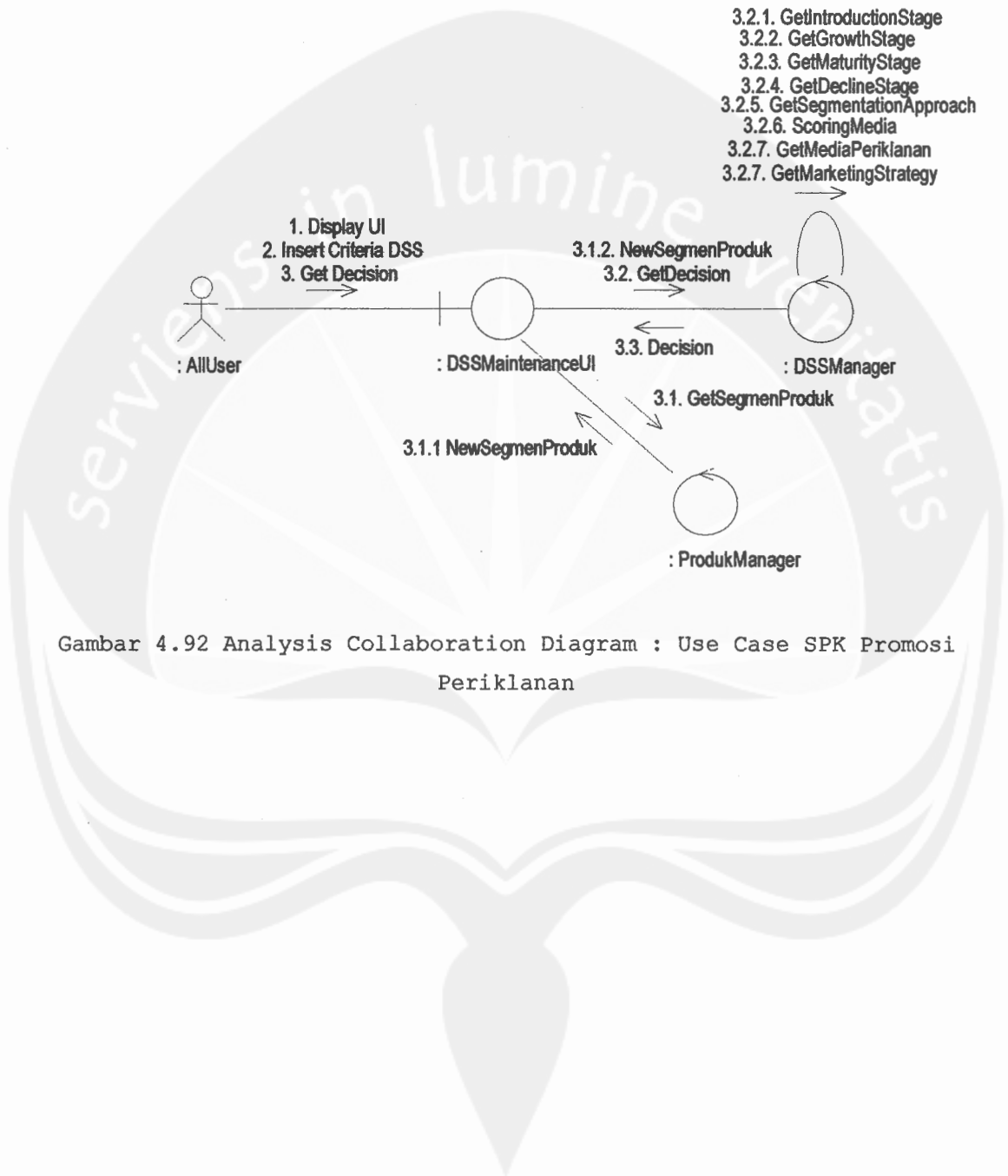
Gambar 4.90 Analysis Collaboration Diagram : Use Case Pengelolaan Data Promosi - Perubahan Data Promosi

4.2.19.3. Pencarian Data Promosi



Gambar 4.91 Analysis Collaboration Diagram : Use Case Pengelolaan Data Promosi - Pencarian Data Promosi

4.2.20. Analisis Collaboration Diagram : Use Case SPK Promosi Periklanan



Gambar 4.92 Analisis Collaboration Diagram : Use Case SPK Promosi Periklanan

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

Sistem Pendukung Keputusan Promosi Produk dan Layanan Telkom Berbasis Web berdasarkan Siklus Hidup Produk


untuk:

**PT. Telekomunikasi Indonesia Tbk.
Kandatel Yogyakarta**

Dipersiapkan oleh:

Adio Gustiasyah 03.07.03655

**Program Studi Teknik Informatika
Fakultas Teknologi Industri
Universitas Atma Jaya Yogyakarta**

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
		DPPL-DSSPromosi		
		Revisi		1/09/2007

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	

Indeks TGL	-	A	B	C	D
Ditulis oleh	Adio	Adio	Adio	Adio	Adio
Diperiksa oleh		PM IW	PM IW	PM IW	PM IW
Disetujui oleh					

Daftar Isi

1	Pendahuluan	1
1.1	Tujuan	1
1.2	Lingkup Masalah	1
1.3	Definisi, Akronim dan Singkatan	1
1.4	Referensi	2
1.5	Deskripsi umum (Overview)	2
2	Deskripsi Kebutuhan	3
2.1	Deployment Diagram	3
2.1.1	Node Client	3
2.1.2	Node Web Server	4
2.1.3	Node Database Server	4
2.2	Design Class	4
2.2.1	Pengantar	4
2.2.2	Package Depedencies	5
2.2.3	Package Account Management	6
2.2.3.1	Class Diagram Package DSSPROMOSI.account	6
2.2.3.2	Class.DSSPROMOSI.account.Account	6
2.2.3.3	Class.DSSPROMOSI.account.AccountManager	7
2.2.3.4	Class.DSSPROMOSI.account.AccountMaintenanceUI	11
2.2.3.5	Class.DSSPROMOSI.account.ChangePasswordUI	11
2.2.3.6	Class.DSSPROMOSI.account.LoginUI	12
2.2.4	Package Periklanan	12
2.2.4.1	Class DSSPROMOSI.periklanan	13
2.2.4.2	Class DSSPROMOSI.periklanan.Elektronik	14
2.2.4.2.1	Class DSSPROMOSI.periklanan.Elektronk.Internet	14
2.2.4.2.2	Class DSSPROMOSI.periklanan.Elektronk.InternetManager	16
2.2.4.2.3	Class DSSPROMOSI.periklanan.Elektronk.InternetMaintenanceUI	18
2.2.4.2.4	Class DSSPROMOSI.periklanan.Elektronk.Radio	19
2.2.4.2.5	Class DSSPROMOSI.periklanan.Elektronk.RadioManager	22
2.2.4.2.6	Class DSSPROMOSI.periklanan.Elektronk.RadioMaintenanceUI	25
2.2.4.2.7	Class DSSPROMOSI.periklanan.Elektronk.Televisi	26
2.2.4.2.8	Class DSSPROMOSI.periklanan.Elektronk.TelevisiManager	28
2.2.4.2.9	Class DSSPROMOSI.periklanan.Elektronk.TelevisiMaintenanceUI	31
2.2.4.3	Class DSSPROMOSI.periklanan.MediaCetakIndoor	32
2.2.4.3.1	Class DSSPROMOSI.periklanan.MediaCetakIndoor.Banner	33
2.2.4.3.2	Class DSSPROMOSI.periklanan.MediaCetakIndoor.BannerManager ..	34
2.2.4.3.3	Class DSSPROMOSI.periklanan.MediaCetakIndoor. BannerMaintenanceUI	36
2.2.4.3.4	Class DSSPROMOSI.periklanan.MediaCetakIndoor.Brosur	37
2.2.4.3.5	Class DSSPROMOSI.periklanan.MediaCetakIndoor.BrosurManager ..	38
2.2.4.3.6	Class DSSPROMOSI.periklanan.MediaCetakIndoor. BrosurMaintenanceUI	41
2.2.4.3.7	Class DSSPROMOSI.periklanan.MediaCetakIndoor.Buku	42
2.2.4.3.8	Class DSSPROMOSI.periklanan.MediaCetakIndoor.BukuManager	43
2.2.4.3.9	Class DSSPROMOSI.periklanan.MediaCetakIndoor. BukuMaintenanceUI	45
2.2.4.3.10	Class DSSPROMOSI.periklanan.MediaCetakIndoor.Koran	46
2.2.4.3.11	Class DSSPROMOSI.periklanan.MediaCetakIndoor.KoranManager ..	49
2.2.4.3.12	Class DSSPROMOSI.periklanan.MediaCetakIndoor. KoranMaintenanceUI	52
2.2.4.3.13	Class DSSPROMOSI.periklanan.MediaCetakIndoor.Majalah	53

2.2.4.3.14	Class DSSPROMOSI.periklanan.MediaCetakIndoor.MajalahManager	55
2.2.4.3.15	Class DSSPROMOSI.periklanan.MediaCetakIndoor.MajalahMaintenanceUI	58
2.2.4.3.16	Class DSSPROMOSI.periklanan.MediaCetakIndoor.Poster	59
2.2.4.3.17	Class DSSPROMOSI.periklanan.MediaCetakIndoor.PosterManager	61
2.2.4.4	Class DSSPROMOSI.periklanan.MediaCetakOutdoor	63
2.2.4.4.1	Class DSSPROMOSI.periklanan.MediaCetakOutdoor.Billboard	64
2.2.4.4.2	Class DSSPROMOSI.periklanan.MediaCetakOutdoor.BillboardManager	66
2.2.4.4.3	Class DSSPROMOSI.periklanan.MediaCetakOutdoor.BillboardMaintenanceUI	69
2.2.4.4.4	Class DSSPROMOSI.periklanan.MediaCetakOutdoor.Branding	70
2.2.4.4.5	Class DSSPROMOSI.periklanan.MediaCetakOutdoor.BrandingManager	72
2.2.4.4.6	Class DSSPROMOSI.periklanan.MediaCetakOutdoor.BrandingMaintenanceUI	74
2.2.4.4.7	Class DSSPROMOSI.periklanan.MediaCetakOutdoor.Neon	75
2.2.4.4.8	Class DSSPROMOSI.periklanan.MediaCetakOutdoor.NeonManager	77
2.2.4.4.9	Class DSSPROMOSI.periklanan.MediaCetakOutdoor.NeonMaintenanceUI	80
2.2.4.4.10	Class DSSPROMOSI.periklanan.MediaCetakOutdoor.Spanduk	81
2.2.4.4.11	Class DSSPROMOSI.periklanan.MediaCetakOutdoor.SpandukManager	83
2.2.4.4.12	Class DSSPROMOSI.periklanan.MediaCetakOutdoor.SpandukMaintenanceUI	86
2.2.4.4.13	Class DSSPROMOSI.periklanan.MediaCetakOutdoor.Umbul	87
2.2.4.4.14	Class DSSPROMOSI.periklanan.MediaCetakOutdoor.UmbulManager	89
2.2.4.4.15	Class DSSPROMOSI.periklanan.MediaCetakOutdoor.UmbulMaintenanceUI	92
2.2.4.5	Class Diagram Package DSSPROMOSI.produk	93
2.2.4.5.1	Class DSSPROMOSI.Produk.Produk	93
2.2.4.5.2	Class DSSPROMOSI.Produk.ProdukManager	94
2.2.4.5.3	Class DSSPROMOSI.Produk.ProdukMaintenanceUI	96
2.2.4.6	Class Diagram Package DSSPROMOSI.Promosi	97
2.2.4.6.1	Class DSSPROMOSI.Promosi.Promosi	97
2.2.4.6.2	Class DSSPROMOSI.Promosi.PromosiManager	100
2.2.4.6.3	Class DSSPROMOSI.Promosi.PromosiMaintenanceUI	103
2.2.4.7	Class Diagram Package DSSPROMOSI.DSS	103
2.2.4.7.1	Class DSSPROMOSI.DSS.DSSManager	104
2.2.4.7.2	Class DSSPROMOSI.DSS.DSSMaintenanceUI	110
2.3	Realisasi Use Case	111
2.3.1	Use Case Login	111
2.3.2	Use Case Pengelolaan Data Operator	112
2.3.2.1	Penambahan Data Operator	112
2.3.2.1	Penghapusan Data Operator	113
2.3.3	Use Case Ubah Password	114
2.3.4	Use Case Penglolaan Data Periklanan	115
2.3.4.1	Use Case Pengelolaan Data Internet	115
2.3.4.1.1	Penambahan Data Internet	115
2.3.4.1.2	Pengubahan Data Internet	116
2.3.4.1.3	Pencarian Data Internet	117
2.3.4.1.4	Penghapusan Data Internet	118
2.3.4.2	Use Case Pengelolaan Data Radio	118
2.3.4.2.1	Penambahan Data Radio	118
2.3.4.2.2	Pengubahan Data Radio	119

2.3.4.2.3	Pencarian Data Radio.....	120
2.3.4.2.4	Penghapusan Data Radio.....	121
2.3.4.3	Use Case Pengelolaan Data Televisi	122
2.3.4.3.1	Penambahan Data Televisi.....	122
2.3.4.3.2	Pengubahan Data Televisi.....	123
2.3.4.3.3	Pencarian Data Televisi.....	124
2.3.4.3.4	Penghapusan Data Televisi.....	124
2.3.4.4	Use Case Pengelolaan Data Banner	125
2.3.4.4.1	Penambahan Data Banner.....	125
2.3.4.4.2	Pengubahan Data Banner.....	126
2.3.4.4.3	Pencarian Data Banner.....	127
2.3.4.4.4	Penghapusan Data Banner.....	128
2.3.4.5	Use Case Pengelolaan Data Buku	128
2.3.4.5.1	Penambahan Data Buku.....	128
2.3.4.5.2	Pengubahan Data Buku.....	129
2.3.4.5.3	Pencarian Data Buku.....	130
2.3.4.5.4	Penghapusan Data Buku.....	131
2.3.4.6	Use Case Pengelolaan Data Brosur	132
2.3.4.6.1	Penambahan Data Brosur.....	132
2.3.4.6.2	Pengubahan Data Brosur.....	133
2.3.4.6.3	Pencarian Data Brosur.....	134
2.3.4.6.4	Penghapusan Data Brosur.....	134
2.3.4.7	Use Case Pengelolaan Data Majalah.....	135
2.3.4.7.1	Penambahan Data Majalah.....	135
2.3.4.7.2	Pengubahan Data Majalah.....	136
2.3.4.7.3	Pencarian Data Majalah.....	137
2.3.4.7.4	Penghapusan Data Majalah.....	138
2.3.4.8	Use Case Pengelolaan Data Koran	138
2.3.4.8.1	Penambahan Data Koran.....	138
2.3.4.8.2	Pengubahan Data Koran.....	139
2.3.4.8.3	Pencarian Data Koran.....	140
2.3.4.8.4	Penghapusan Data Koran.....	141
2.3.4.9	Use Case Pengelolaan Data Poster.....	141
2.3.4.9.1	Penambahan Data Poster.....	141
2.3.4.9.2	Pengubahan Data Poster.....	142
2.3.4.9.3	Pencarian Data Poster.....	143
2.3.4.9.4	Penghapusan Data Poster.....	144
2.3.4.10	Use Case Pengelolaan Data Billboard.....	145
2.3.4.10.1	Penambahan Data Billboard.....	145
2.3.4.10.2	Pengubahan Data Billboard.....	146
2.3.4.10.3	Pencarian Data Billboard.....	147
2.3.4.10.4	Penghapusan Data Billboard.....	147
2.3.4.11	Use Case Pengelolaan Data Branding.....	148
2.3.4.11.1	Penambahan Data Branding.....	148
2.3.4.11.2	Pengubahan Data Branding.....	149
2.3.4.11.3	Pencarian Data Branding.....	150
2.3.4.11.4	Penghapusan Data Branding.....	151
2.3.4.12	Use Case Pengelolaan Data Neon.....	151
2.3.4.12.1	Penambahan Data Neon.....	151
2.3.4.12.2	Pengubahan Data Neon.....	152
2.3.4.12.3	Pencarian Data Neon.....	153
2.3.4.12.4	Penghapusan Data Neon.....	154
2.3.4.13	Use Case Pengelolaan Data Spanduk.....	155
2.3.4.13.1	Penambahan Data Spanduk.....	155
2.3.4.13.2	Pengubahan Data Spanduk.....	156
2.3.4.13.3	Pencarian Data Spanduk.....	157
2.3.4.13.4	Penghapusan Data Spanduk.....	157

2.3.4.14	<i>Use Case</i> Pengelolaan Data Umbul	158
2.3.4.14.1	Penambahan Data Umbul	158
2.3.4.14.2	Pengubahan Data Umbul	159
2.3.4.14.3	Pencarian Data Umbul	160
2.3.4.14.4	Penghapusan Data Umbul	160
2.3.4.15	<i>Use Case</i> Pengelolaan Data Produk	161
2.3.4.15.1	Penambahan Data Produk	161
2.3.4.15.2	Pengubahan Data Produk	162
2.3.4.15.3	Pencarian Data Produk	163
2.3.4.16	<i>Use Case</i> Pengelolaan Data Promosi	164
2.3.4.16.1	Penambahan Data Promosi	164
2.3.4.16.2	Pengubahan Data Promosi	165
2.3.4.16.3	Pencarian Data Promosi	166
2.3.4.17	<i>Use Case</i> SPK Promosi Periklanan	167
3	Deskripsi Perancangan Persistenst Data	168
3.1	Basis Data	168
3.1.1	User_web	169
3.1.2	Periklanan	169
3.1.3	Elektronik	169
3.1.4	Internet	169
3.1.5	Televisi	170
3.1.6	Radio	170
3.1.7	Media_Cetak	171
3.1.8	MCetak_Area	171
3.1.9	Banner	172
3.1.10	Buku_Petunjuk	172
3.1.11	Brosur	172
3.1.12	Koran	173
3.1.13	Majalah	174
3.1.14	Poster	174
3.1.15	Billboard	175
3.1.16	Branding	175
3.1.17	Neon_Sign	176
3.1.18	Spanduk	176
3.1.19	Umbul_Umbul	177
3.1.20	Divre	177
3.1.21	Kandatel	178
3.1.22	Pulau	178
3.1.23	Propinsi	178
3.1.24	Data_Promosi	178
3.1.25	Produk	179
4	Deskripsi Perancangan Antarmuka	180
4.1	<i>Use Case</i> Login	180
4.2	<i>Use Case</i> Pengelolaan Data Operator	181
4.3	<i>Use Case</i> Ubah Password	183
4.4	<i>Use Case</i> Pengelolaan Data Internet	184
4.5	<i>Use Case</i> Pengelolaan Data Radio	187
4.6	<i>Use Case</i> Pengelolaan Data Televisi	190
4.7	<i>Use Case</i> Pengelolaan Data Banner	193
4.8	<i>Use Case</i> Pengelolaan Data Buku	196
4.9	<i>Use Case</i> Pengelolaan Data Brosur	198
4.10	<i>Use Case</i> Pengelolaan Data Majalah	201
4.11	<i>Use Case</i> Pengelolaan Data Koran	204
4.12	<i>Use Case</i> Pengelolaan Data Poster	207
4.13	<i>Use Case</i> Pengelolaan Data Billboard	210
4.14	<i>Use Case</i> Pengelolaan Data Branding	213
4.15	<i>Use Case</i> Pengelolaan Data Neon	216

4.16	<i>Use Case</i> Pengelolaan Data Spanduk	219
4.17	<i>Use Case</i> Pengelolaan Data Umbul	222
4.18	<i>Use Case</i> Pengelolaan Data Produk	225
4.19	<i>Use Case</i> Pengelolaan Data Promosi	228
4.20	<i>Use Case</i> SPK Promosi Periklanan	231



Sistem Pendukung Keputusan Promosi Produk dan Layanan Telkom Berbasis Web berdasarkan Siklus Hidup Produk

1 Pendahuluan

1.1 Tujuan

Dokumen ini menjelaskan tentang Deskripsi Perancangan Perangkat Lunak. Untuk penamaan dokumen ini selanjutnya akan digunakan istilah DPPL. Dokumen ini merupakan dokumen deskripsi perancangan perangkat lunak untuk Sistem Pendukung Keputusan Promosi Produk dan Layanan Telkom Berbasis Web berdasarkan Siklus Hidup Produk yang akan dibangun. Dokumen ini digunakan oleh pengembang perangkat lunak sebagai acuan teknis untuk pembangunan perangkat lunak.

1.2 Lingkup Masalah

Dokumen DPPL ini menyediakan deskripsi lengkap perancangan perangkat lunak. Perancangan ini merupakan arsitektur sistem yang dijelaskan melalui perancangan class/modul, detail operasi apa yang akan dilakukan oleh masing-masing class/modul, dan layout basis data. Metodologi yang digunakan dalam perancangan adalah metode USDP (*Unified Software Development Process*) dari *Rational Software*.

1.3 Definisi Akronim dan Singkatan

Untuk definisi istilah dan singkatan yang digunakan dalam dokumen dapat mengacu pada tabel berikut :

Keyword atau Phrase	Definisi
DSS	Decision Support System (Sistem Pendukung Keputusan)

DPPL	Deskripsi Perancangan Perangkat Lunak
Role	Hak akses User
DataBase	Merupakan kelompok data (arsip) yang saling berhubungan dan diorganisir sedemikianrupa agar dapat menghasilkan informasi dan dapat dimanfaatkan kembali dengan cepat dan mudah.

1.4 Referensi

Dokumen yang digunakan sebagai acuan dalam rencana pengembangan perangkat lunak ini adalah :

- *Software Engineering*, 1997, Roger S. Pressman, Mc Graw-Hill International Edition.
- *Rekayasa Perangkat Lunak*, 1997, Presman Roger S., McGraw-Hill Book Co., Andi Yogyakarta..
- Ivar Jacobson, Grady Booch, James Rumbaugh. *The Unified Software Development Process*. Addison-Wesley. 1998.
- Martin Fowler, Kendall Scott. *UML Distilled - Second Edition*. Adisson Wesley. 1999.
- Stephen R Schach. *An Introduction to Object-Oriented Systems*
- *Analysis and Design with UML and the Unified Process*. Mc Graw-Hill. 2004.
- Wendy Boqqs, Michael Boqqs. *Mastering UML With Rational Rose*. 2002.
- <http://ilmukomputer.com:81/umum/vanti-uml.php>

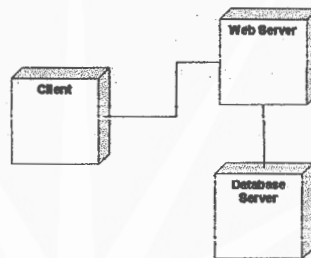
1.5 Deskripsi umum (Overview)

Dokumen ini terdiri dari empat bab. Bab pertama adalah **Pendahuluan**, yang berisi deskripsi dokumen. Bab kedua adalah **Deskripsi Perancangan Arsitektural**, yang berisi deskripsi

arsitektur sistem. Bab ketiga adalah **Deskripsi Perancangan Persistent Data**, yang berisi deskripsi data-data yang akan disimpan pada persistent storage. Bab keempat adalah **Deskripsi Perancangan Antarmuka**, yang berisi deskripsi rancangan GUI yang digunakan sistem untuk berinteraksi dengan user.

2. Deskripsi Kebutuhan

2.1. Deployment Diagram



Gambar 2.1 Deployment Diagram DssPromosi

Deployment diagram ini dibuat untuk menunjukkan semua node pada sistem, hubungan di antara mereka, dan proses yang akan dijalankan di masing-masing node.

2.1.1 Node : Client

Client merupakan komputer yang digunakan oleh user untuk melakukan pengelolaan terhadap dokumen. Proses yang ada didalamnya adalah *Web browser*, digunakan untuk melakukan permintaan proses yang dijalankan.

2.1.2 Node : *Web Server*

Web Server merupakan komputer yang menyediakan layanan ke web bagi *client* yang mengakses Internet, dimana Web server ini akan mengakses basis data untuk operasi yang melibatkan data-data pada basis data.

2.1.3 Node : *Database Server*

Database Server merupakan komputer yang dipakai untuk menyediakan koneksi ke basis data dan mengautentikasi *Web server* dan tidak akan mengizinkan melihat informasi atau menjalankan prosedur kecuali jika ia memiliki hak yang sesuai.

2.2. Design Class

2.2.1. Pengantar

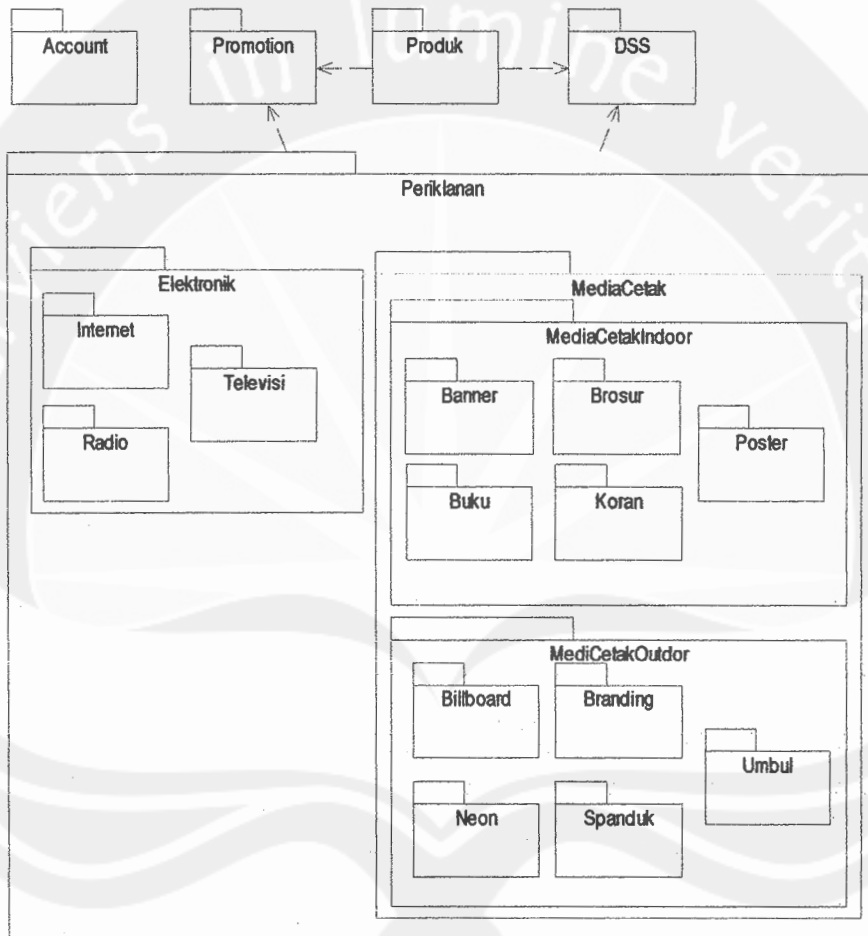
Nama *class* yang digunakan dalam *design class* adalah nama *class* yang valid, termasuk nama *package*-nya. Untuk class-class yang berasal dari *framework .Net* juga digunakan nama *class* dengan *package* lengkap, misalnya *System.IO.File*. Untuk penjelasan tipe data yang utuh dapat dilihat pada bagian deskripsi *class*, sedangkan gambar *design class* tidak akan menggunakan nama *package* yang lengkap. *Stereotype* yang digunakan dalam *design class* adalah :

- << boundary >>
Boundary class merupakan *class* yang berfungsi untuk menghubungkan sistem dengan user di luar sistem.
- << control >>
Control class adalah suatu *class* yang objek-nya melakukan interaksi antar sekelompok objek lain. *Control class* biasanya memiliki karakteristik yang spesifik untuk satu *use case*, dan objek *class* ini biasanya hanya aktif pada realisasi *use case*.

- << entity >>

Entity class adalah class yang bersifat pasif, dalam arti class tersebut tidak memulai interaksi dengan class lain. Entity class ini biasanya merepresentasikan suatu objek yang disimpan dalam persistent storage.

2.2.2. Package Dependencies

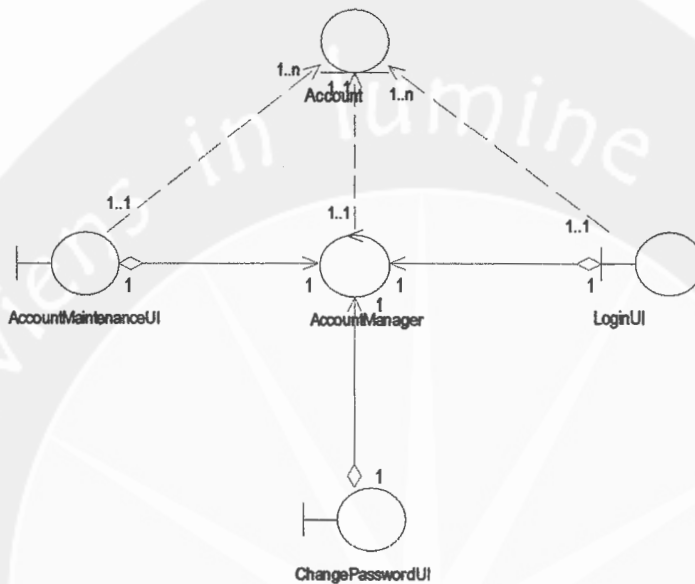


Gambar 2.2 Package Dependencies DSSPROMOSI

2.2.3. Package Account Management (account)

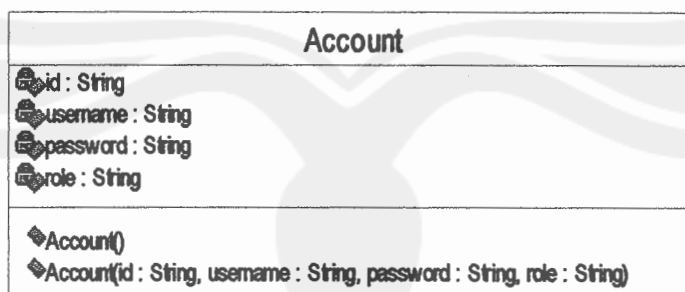
Package ini menyediakan class-class yang digunakan oleh program untuk mengelola hal-hal yang berhubungan dengan account user.

2.2.3.1. Class Diagram Package account



Gambar 2.3 Class Diagram Package DSSPROMOSI.account

2.2.3.1.1. Class.DSSPROMOSI.account.Account



Gambar 2.4 Class DSSPROMOSI.account.Account

Deskripsi

Class ini merepresentasikan account pengguna sistem DSSPROMOSI.

Atribut

- - id : string

Merepresentasikan id user

- - username : string

Merepresentasikan username user.

- - password : string

Merepresentasikan password user.

- - role : string

Merepresentasikan role user, apakah administrator atau operator.

Method

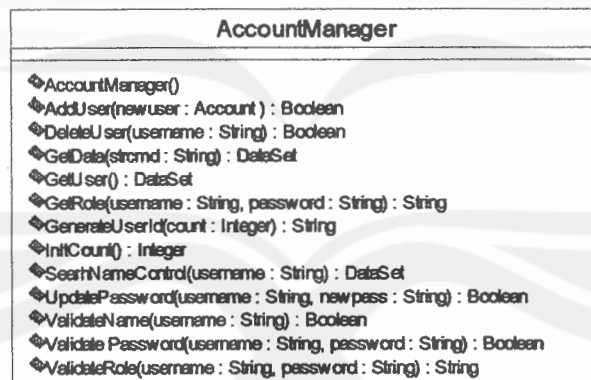
- + Account()

Konstruktor class DSSPROMOSI.account.Account. Buat instance baru DSSPROMOSI.account.Account tanpa atribut terdefinisi.

- + Account (id : string, username : string, password : string, role : string)

Konstruktor class DSSPROMOSI.account.Account. Buat instance baru DSSPROMOSI.account.Account dengan atribut terdefinisi.

2.2.3.1.2. Class.DSSPROMOSI.account.AccountManager



Gambar 2.5 Class DSSPROMOSI.account.AccountManager

Deskripsi

Class yang berperan sebagai control class untuk aksi yang berhubungan dengan account user DSSPROMOSI.

Atribut

-

Method

- + AccountManager()

Konstruktor class DSSPROMOSI.account.AccountManager. Buat instance baru DSSPROMOSI.account.AccountManager.

• + addUser(account : DSSPROMOSI.account.Account) :
boolean

Digunakan untuk menambahkan account baru.

Parameters:

account - DSSPROMOSI.account.Account yang akan ditambahkan.

Returns:

true jika account berhasil ditambahkan.

• + DeleteUser(username : string) : Boolean
Menghapus user dari database.

Parameters:

username - username pengguna.

Returns:

true jika penghapusan user berhasil dilakukan.

• + GetData(strcmd : string) : DataSet

Ambil data account pada basis data sesuai atribut yang diinginkan.

Parameters:

Strcmd - query untuk mengambil data user pada field tertentu.

Returns:

Set data yang berisi data account yang tersimpan pada basis data.

• + GetUser() : DataSet

Ambil semua data account pada basis data.

Parameters:

-

Returns:

Set data yang berisi semua data account yang tersimpan pada basis data.

• + GetRole(username : string,password : string) : String

Ambil role account pada basis.

Parameters:

username - username pengguna

password - password pengguna

Returns:

String role.

- + GetUser() : DataSet

Ambil semua data account pada basis data.

Parameters:

-

Returns:

Set data yang berisi semua data account yang tersimpan pada basis data.

- + GenerateUserId (count : integer) : String

Mendapatkan user id yang telah ter-generate.

Parameters:

Count - id maximum dari user yang telah di counter yang tersimpan di database.

Returns:

String id.

- + InItCount () : integer

Mendapatkan maximum user id dan di tambah 1.

Parameters:

-

Returns:

String id yang telah di counter.

- + SearchNameControl (username : string) : DataSet

Mendapatkan class DssPromosi.account.Account untuk data user tertentu.

Parameters:

Username - username dari data user.

Returns:

Set data user untuk user tertentu.

- + updatePassword(name : string, newPass : string) : boolean

Mengupdate *password* user dalam *database*. Untuk mengupdate password., harus dipenuhi tiga syarat berikut :

- o password lama harus sesuai dengan password system yang ada.
- o password baru (argument newPass) harus sesuai dengan konfirmasi password baru.

Perlu diingat bahwa password bersifat *case-sensitive*.

Parameters:

username - username yang akan meng-update password (user yang sedang login).

newPass - password baru dari user yang bersifat *case-sensitive*.

Returns:

true jika dan hanya jika password berhasil diubah untuk user yang sedang aktif.

• + ValidateName(username : string) : boolean
Validasi username apakah ada dalam basis data.

Parameters:

username - username yang akan divalidasi.

Returns:

true jika dan hanya jika username ada dalam database.

• ValidatePassword(username : string, password : string) : boolean

Validasi password apakah ada dalam basis data.

Parameters:

username - username yang akan divalidasi.

Password - password dari user yang akan divalidasi.

Returns:

true jika dan hanya jika password yang divalidasi sesuai dengan username ada dalam database.

• ValidateRole(username : string, password : string) : boolean

Validasi role apakah ada dalam basis data dari username dan password yang terdefinisi.

Parameters:

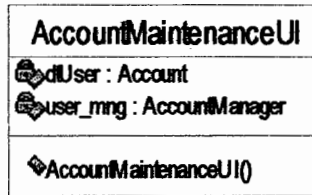
username - username yang akan divalidasi.

Password - password dari user yang akan divalidasi.

Returns:

true jika role sesuai dengan username dan password yang terdefinisi yang ada dalam database.

2.2.3.1.3. Class.DSSPROMOSI.account.AccountMaintenanceUI



Gambar 2.6 Class DSSPROMOSI.account.AccountMaintenanceUI

Deskripsi

Class ini merupakan GUI untuk melakukan pengelolaan account user yang meliputi penambahan dan penghapusan account.

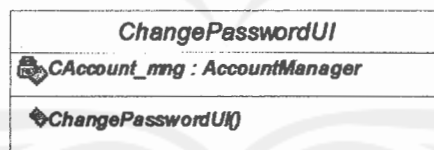
Atribut

- - dtUser : DSSPROMOSI.account.Account
Merepresentasikan *object control class* untuk pengelolaan user.
- - user_mng : DSSPROMOSI.account.AccountManager
Merupakan *object control class* untuk melakukan perubahan terhadap data user.

Method

- + AccountMaintenanceUI()
Buat instance baru DSSPROMOSI.account.AccountMaintenanceUI.

2.2.3.1.4. Class.DSSPROMOSI.account.ChangePasswordUI



Gambar 2.7 Class DSSPROMOSI.account.ChangePasswordUI

Deskripsi

Class ini merupakan GUI yang digunakan untuk melakukan penggantian password bagi username yang sedang aktif.

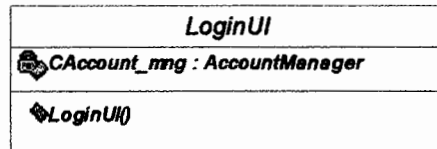
Atribut

- - Caccount_mng : DSSPROMOSI.account.AccountManager
Merupakan *object control class* untuk melakukan penggantian password.

Method

- + ChangePasswordUI()
Buat instance baru DSSPROMOSI.account.ChangePasswordUI.

2.2.3.1.5. Class.DSSPROMOSI.account.LoginUI



Gambar 2.8 Class DSSPROMOSI.account.LoginUI

Deskripsi

Class ini merupakan GUI yang digunakan untuk validasi username, password serta role untuk login.

Atribut

- - CAccount_mng : DSSPROMOSI.account.AccountManager
Merupakan *object control class* untuk melakukan validasi username, password dan role.

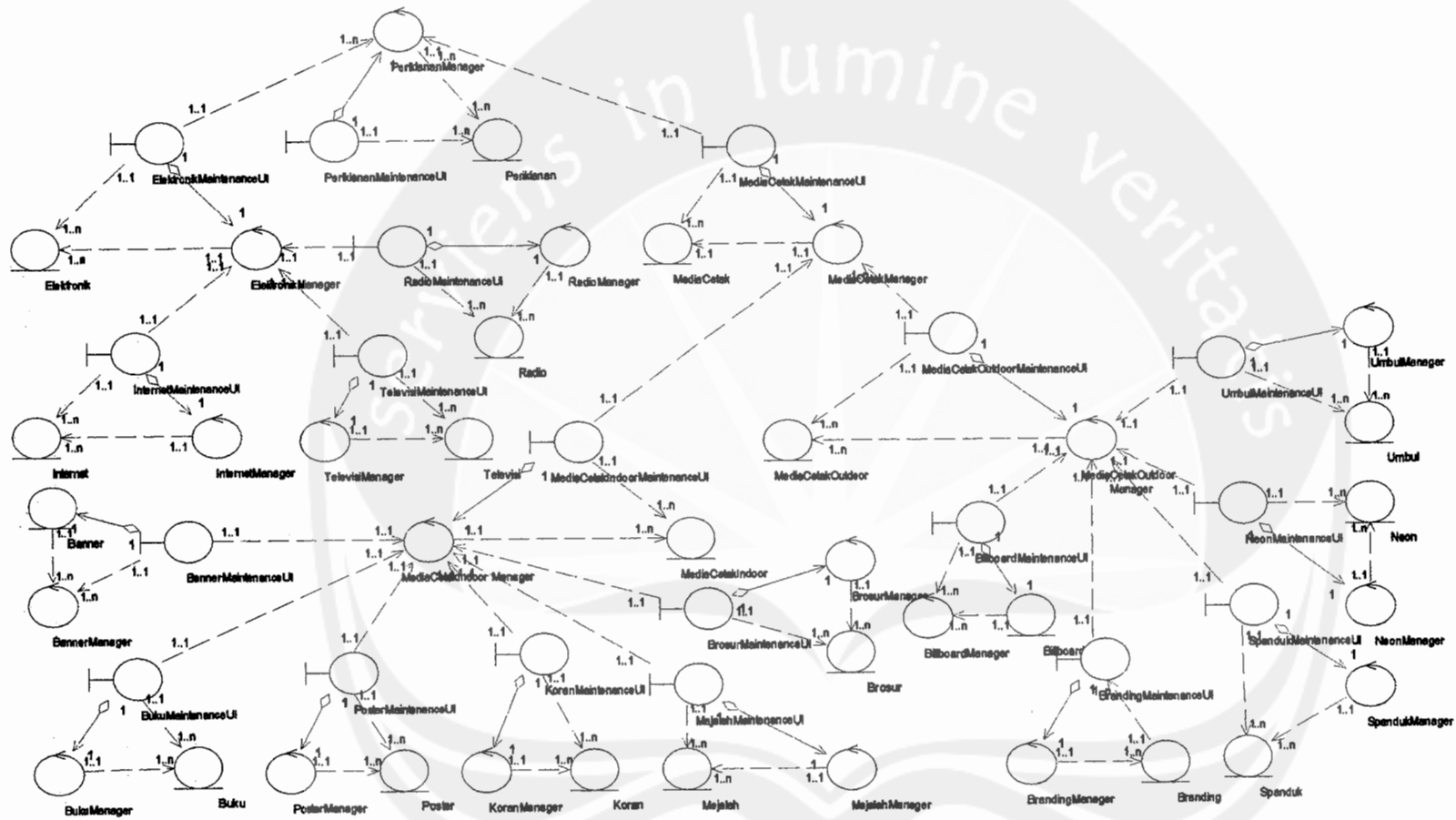
Method

- + LoginUI()
Buat instance baru DSSPROMOSI.account.LoginUI.

2.2.4. Package Periklanan Mintenance (DSSPROMOSI.periklanan)

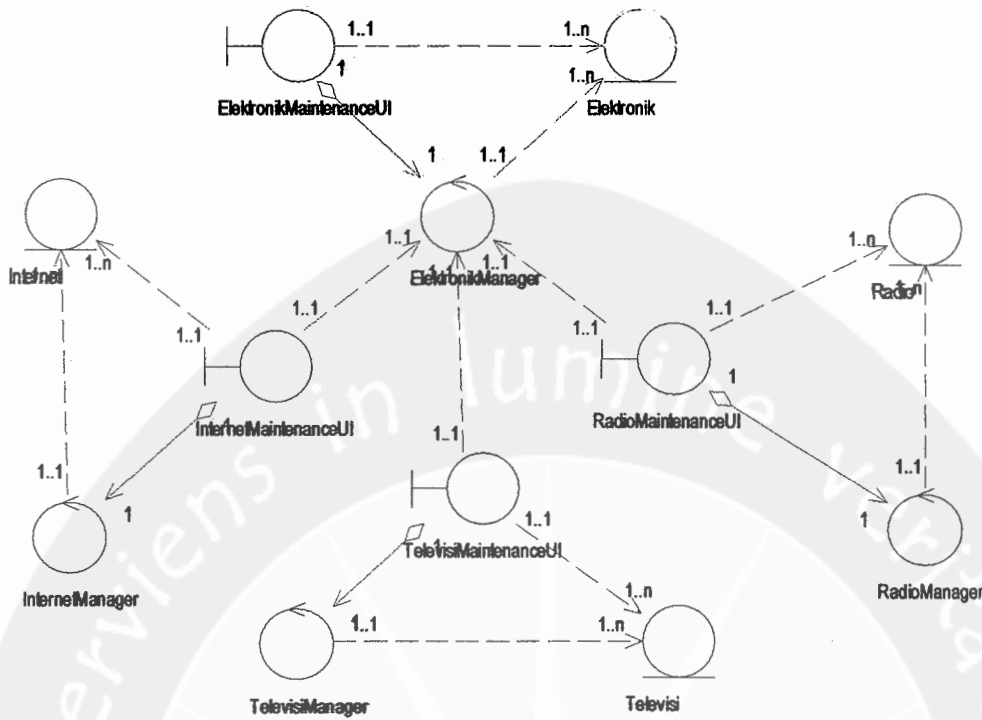
Package ini menyediakan class-class yang diperlukan untuk melakukan pengelolaan data periklanan yakni pengelolaan periklanan elektronik dan media cetak. Sub-package dari package elektronik yakni pengelolaan data internet, pengelolaan data radio dan pengelolaan data televisi. Sedangkan untuk sub-package media cetak ada media cetak indoor dan media cetak outdoor. Dimana sub-package media cetak indoor terdiri dari sub-package untuk pengelolaan data banner, pengelolaan data brosur, pengelolaan data buku, pengelolaan data koran, pengelolaan data majalah dan pengelolaan data poster. Untuk sub-package media cetak outdoor terdiri dari sub-package pengelolaan data billboard, pengelolaan data branding, pengelolaan data neon, pengelolaan data spanduk dan pengelolaan data umbul-umbul.

2.2.4.1. Class Diagram Package DSSPROMOSI.periklanan



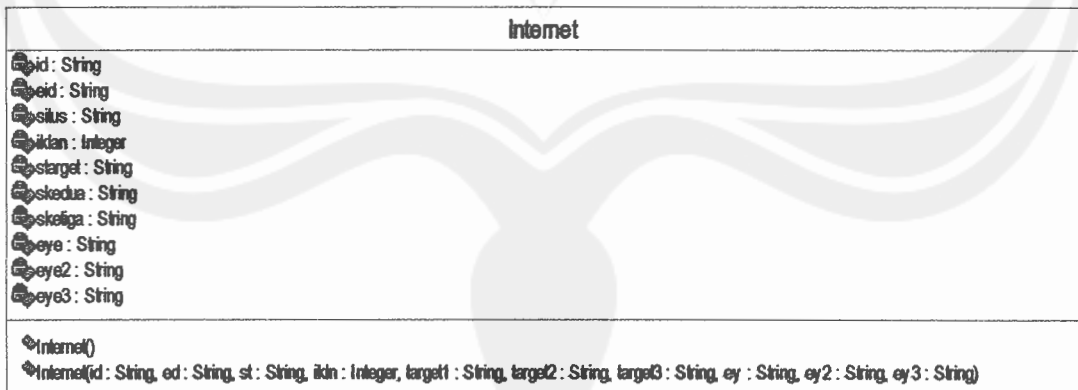
Gambar 2.9 Class Diagram Package DSSPROMOSI.periklanan

2.2.4.2. Class Diagram Package DSSPROMOSI.periklanan. Elektronik



Gambar 2.10 Class DSSPROMOSI.account.periklanan.Elektronik

2.2.4.3. Class Diagram Package DSSPROMOSI.periklanan. elektronik.internet. Internet



Gambar 2.11 Class DSSPROMOSI.periklanan.elektronik.internet. Internet

Deskripsi

Class ini merepresentasikan internet di dalam DssPromosi.

Atribut

- - Id : string

Merepresentasikan id dari internet

- - eid : string

Merepresentasikan elektronik id.

- - situs : string

Merepresentasikan nama situs internet

- - iklan : integer

Merepresentasikan tarif iklan di situs tersebut

- - starget : string

Merepresentasikan segmen target utama dari situs.

- - skedua : string

Merepresentasikan segmen target kedua dari situs.

- - sketiga : string

Merepresentasikan segmen target ketiga dari situs.

- - eye : string

Merepresentasikan eye catching score dari situs.

- - eye2 : string

Merepresentasikan eye catching score kedua dari situs.

- - eye3 : string

Merepresentasikan eye catching score ketiga dari situs.

Method

- + Internet()

Merupakan konstruktor class, tanpa atribut terdefinisi.

- + Internet(id : String, ed : String, st : String, ikln : Integer, target1 : String, target2 : String, target3 : String, ey : String, ey2 : String, ey3 : String)

Merupakan konstruktor class, dengan atribut terdefinisi.

Parameters

id : id internet.

ed : elektronik id.

st : nama situs.

ikln : tarif iklan.

target1 : segmen target utama

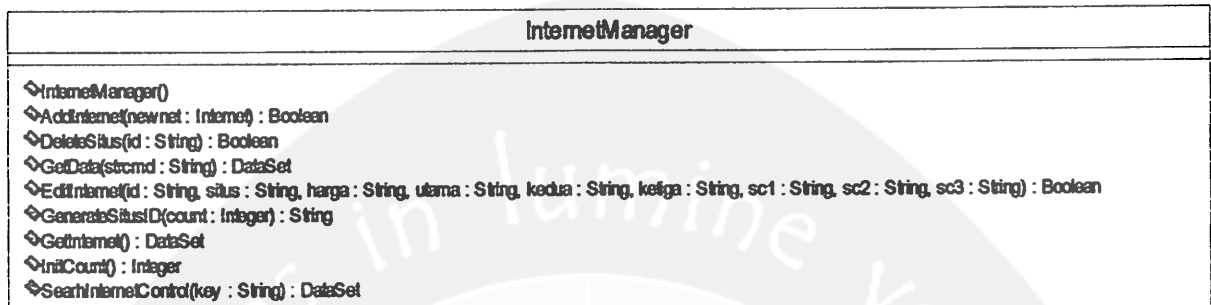
target2 : segmen target kedua

target3 : segmen target ketiga

ey : eye catching score pertama

ey2 : eye catching score kedua
 ey3 : eye catching score ketiga

2.2.4.4. Class Diagram Package DSSPROMOSI.periklanan.elektronik.internet.InternetManager



Gambar 2.12 Class DSSPROMOSI.periklanan.elektronik.internet.InternetManager

Deskripsi

Class yang berperan sebagai *object control class* untuk aksi yang berhubungan dengan data internet.

Atribut

-

Method

- + InternetManager()

Konstruktor class DSSPROMOSI.periklanan.elektronik.internet.InternetManager. Buat instance baru DSSPROMOSI.periklanan.elektronik.internet.InternetManager.

- + addInternet(newnet : DSSPROMOSI.periklanan.elektronik.internet.InternetManager) :

boolean

Digunakan untuk menambahkan data internet baru.

Parameters:

newnet - DSSPROMOSI.periklanan.elektronik.internet.Internet yang akan ditambahkan.

Returns:

true jika internet berhasil ditambahkan.

- + DeleteSitus(id : string) : Boolean

Menghapus internet dari database.

Parameters:

id - id internet.

Returns:

true jika penghapusan internet berhasil dilakukan.

• + EditInternet(id : String, situs : String, harga : String, utama : String, kedua : String, ketiga : String, sc1 : String, sc2 : String, sc3 : String) : Boolean

Parameters:

id : id internet.

situs : nama situs.

harga : tarif iklan.

utama : segmen target utama

kedua : segmen target kedua

ketiga : segmen target ketiga

sc1 : eye catching score pertama

sc2 : eye catching score kedua

sc3 : eye catching score ketiga

Returns:

True jika data internet berhasil diubah.

• + GetData(strcmd : string) : DataSet

Ambil data internet pada basis data sesuai field yang diinginkan.

Parameters:

Strcmd - query untuk mengambil data internet pada field tertentu.

Returns:

Set data yang berisi data internet yang tersimpan pada basis data.

• + GetInternet() : DataSet

Ambil semua data internet pada basis data.

Parameters:

-

Returns:

Set data yang berisi semua data internet yang tersimpan pada basis data.

• + GenerateInternetId (count : integer) : String

Mendapatkan internet id yang telah ter-generate.

Parameters:

Count - id maximum dari internet yang telah di counter yang tersimpan di database.

Returns:

String id.

- + InItCount () : integer

Mendapatkan maximum internet id dan di tambah 1.

Parameters:

-

Returns:

id yang telah di counter.

- + SearchInternetControl (key : string) : DataSet

Mendapatkan class DSSPROMOSI. periklanan. elektronik. internet.Internet untuk data internet tertentu.

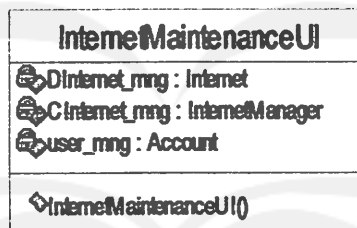
Parameters:

key - situs internet yang akan dicari.

Returns:

Set data internet untuk internet tertentu.

2.2.4.5. Class Diagram Package DSSPROMOSI.periklanan. elektronik.internet.InternetMaintenanceUI



Gambar 2.13 Class DSSPROMOSI.periklanan.elektronik.internet. InternetMaintenanceUI

Deskripsi

Class ini merupakan GUI yang berperan sebagai *boundary class* yang digunakan pengelolaan data Internet.

Atribut

- - DIInternet_mng : DSSPROMOSI.periklanan.elektronik. internet.Internet

Merepresentasikan *object entity class* untuk pengelolaan internet saat melakukan perubahan terhadap data internet.

- - CInternet_mng : DSSPROMOSI.periklanan.elektronik.internet.InternetManager

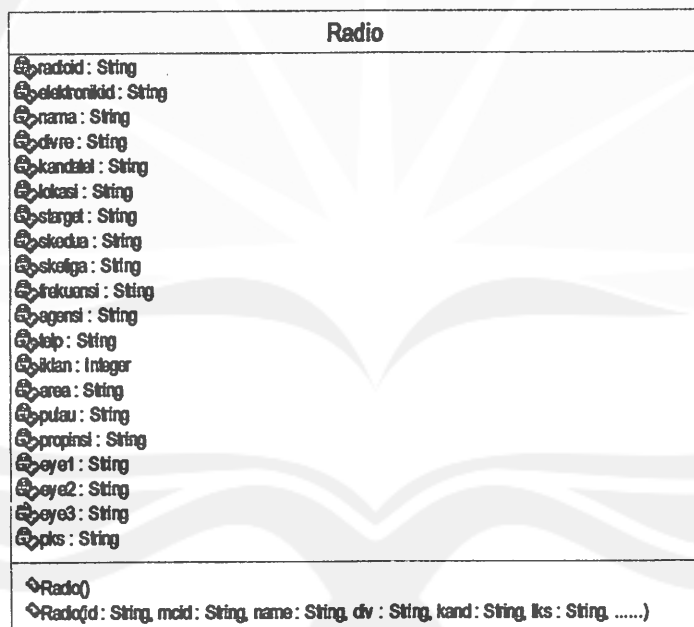
Merepresentasikan *object control class* untuk pengelolaan data internet saat melakukan perubahan terhadap data internet.

- - user_mng : DSSPROMOSI.account.Account
- Merepresentasikan *object control class* untuk pengelolaan session user.

Method

- + InternetMaintenanceUI()
- Merupakan konstruktor class, tanpa atribut terdefinisi.

2.2.4.6. Class Diagram Package DSSPROMOSI.periklanan.elektronik.radio.Radio



Gambar 2.14 Class DSSPROMOSI.periklanan.elektronik.radio.Radio

Deskripsi

Class ini merepresentasikan radio di dalam DssPromosi.

Atribut

- - radioid : string
- Merepresentasikan id dari internet
- - elektronikid: string

Merepresentasikan elektronik id.

- - nama : string

Merepresentasikan nama radio.

- - divre : string

Merepresentasikan lokasi divre dari radio.

- - kandatel: string

Merepresentasikan lokasi kandatel dari radio.

- - lokasi : string

Merepresentasikan lokasi dari radio.

- - starget : string

Merepresentasikan segmen target utama dari radio.

- - skedua : string

Merepresentasikan segmen target kedua dari radio.

- - sketiga : string

Merepresentasikan segmen target ketiga dari radio.

- - frekuensi: string

Merepresentasikan frekuensi radio.

- - agensi : string

Merepresentasikan agensi atau penanggung jawab radio.

- - telp : string

Merepresentasikan no. Telp radio

- - iklan : integer

Merepresentasikan tarif iklan di radio tersebut

- - area : string

Merepresentasikan jangkauan radio, nasional atau internasional.

- - pulau : string

Merepresentasikan jangkauan pulau dari radio.

- - propinsi: string

Merepresentasikan jangkauan propinsi dari radio.

- - eye1 : string

Merepresentasikan eye catching score dari radio.

- - eye2 : string

Merepresentasikan eye catching score kedua dari radio.

- - eye3 : string

Merepresentasikan eye catching score ketiga dari radio.

- - pks : string

Merepresentasikan perjanjian kerja sama dari radio, apakah sudah pernah dilakukan perjanjian kerja sama atau belum.

Method

- + Radio()

Merupakan konstruktor class, tanpa atribut terdefinisi.

- + Radio(id : String, mcid : String, name : String, div : String, kand : String, lks : String, target1 : String, target2 : String, target3 : String, frek : String, agen : String, telp : String, iklan : integer, area : String, pulau : String, prop : String, eyl : String, ey2 : String, ey3 : String, pks : String)

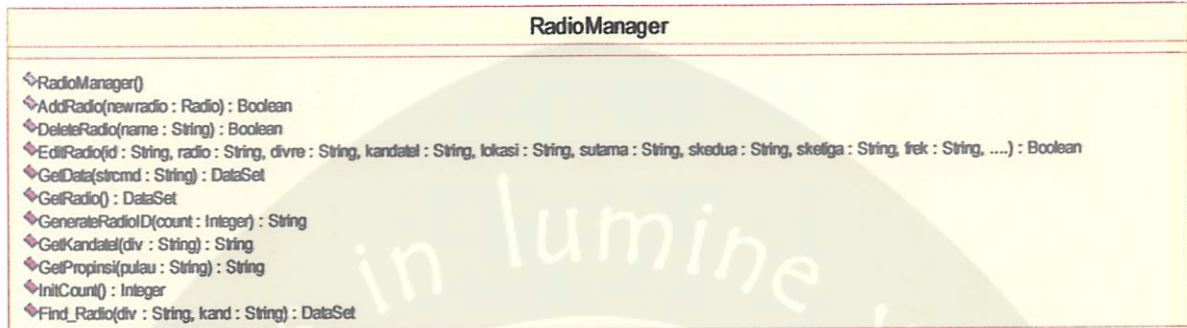
Merupakan konstruktor class, dengan atribut terdefinisi.

Parameters

- radioid : id radio
- elektronikid : elektronik id.
- nama : nama radio.
- divre : divre radio.
- kandatel : kandatel radio.
- lokasi : lokasi radio.
- starget : segmen target utama dari radio.
- skedua : segmen target kedua dari radio.
- sketiga : segmen target ketiga dari radio.
- frekuensi : frekuensi radio.
- agensi : agensi atau penanggung jawab radio.
- telp : no. Telp radio
- iklan : tarif iklan radio
- area : jangkauan radio.
- pulau : jangkauan pulau radio.
- propinsi : jangkauan dari radio.
- eyl : eye catching score radio.
- ey2 : eye catching score kedua radio.
- eye3 : eye catching score ketiga radio.

- pks : perjanjian kerja sama dengan radio.

2.2.4.7. Class Diagram Package DSSPROMOSI.periklanan. elektronik.radio.RadioManager



Gambar 2.15 Class DSSPROMOSI.periklanan.elektronik.radio. RadioManager

Deskripsi

Class yang berperan sebagai *object control class* untuk aksi yang berhubungan dengan data radio.

Atribut

-

Method

• + RadioManager()

Konstruktor class DSSPROMOSI.periklanan.elektronik.radio.RadioManager. Buat instance baru DSSPROMOSI.periklanan.elektronik.radio.RadioManager.

• + addRadio(newradio : DSSPROMOSI.periklanan.elektronik.radio.Radio) : boolean

Digunakan untuk menambahkan data radio baru.

Parameters:

newradio - DSSPROMOSI.periklanan.elektronik.radio. Radio yang akan ditambahkan.

Returns:

true jika radio berhasil ditambahkan.

• + Delete Radio (name : string) : Boolean

Menghapus radio dari database.

Parameters:

name - nama radio.

Returns:

true jika penghapusan radio berhasil dilakukan.

• + EditRadio(id : String, radio : String, divre : String, kandatel : String, lokasi : String, sutama : String, skedua : String, sketiga : String, frek : String, agen : String, telp : String, iklan : String, covarea : String, pulau : String, prop : String, scl : String, sc2 : String, sc3 : String, pks : String) : Boolean

Parameters:

id : id radio
radio : nama radio
divre : divre radio.
kandatel: kandatel radio.
lokasi : lokasi radio.
starget: segmen target utama dari radio.
skedua : segmen target kedua dari radio.
sketiga: segmen target ketiga dari radio.
frekuensi: frekuensi radio.
agensi : agensi atau penanggung jawab radio.
telp : no. Telp radio
iklan : tarif iklan radio
area : jangkauan radio.
pulau : jangkauan pulau radio.
propinsi: jangkauan dari radio.
eyel : eye catching score radio.
eye2 : eye catching score kedua radio.
eye3 : eye catching score ketiga radio.
pks : perjanjian kerja sama dengan radio.

Returns:

True jika data radio berhasil diubah.

• + GetData(strcmd : string) : DataSet

Ambil data radio pada basis data sesuai field yang diinginkan.

Parameters:

Strcmd - query untuk mengambil data radio pada field tertentu.

Returns:

Set data yang berisi data radio yang tersimpan pada basis data.

• + GetRadio() : DataSet

Ambil semua data radio pada basis data.

Parameters:

-

Returns:

Set data yang berisi semua data radio yang tersimpan pada basis data.

• + GenerateRadioId (count : integer) : String

Mendapatkan radio id yang telah ter-generate.

Parameters:

Count - id maximum dari radio yang telah di counter yang tersimpan di database.

Returns:

String id.

• + GetKandatel(div : string) : DataSet

Ambil data kandatel pada basis data sesuai dengan divre tertentu.

Parameters:

div - divre dari data kandatel.

Returns:

Set data yang berisi data kandatel yang tersimpan pada basis data.

• + GetPropinsi(pulau : string) : DataSet

Ambil data propinsi pada basis data sesuai dengan pulau tertentu.

Parameters:

pulau - pulau dari propinsi.

Returns:

Set data yang berisi data propinsi yang tersimpan pada basis data.

• + InItCount () : integer

Mendapatkan maximum radio id dan di tambah 1.

Parameters:

-

Returns:

id yang telah di counter.

• + Find_Radio (div : string, kand : string) : DataSet
Mendapatkan class DSSPROMOSI. periklanan. elektronik.
radio.Radio untuk data radio tertentu.

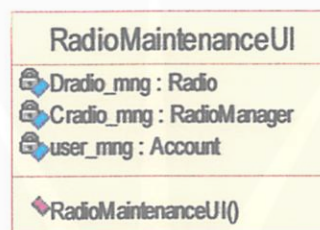
Parameters:

div - divre radio yang akan dicari.
kand - kandatel radio yang akan dicari.

Returns:

Set data radio untuk radio tertentu.

**2.2.4.8. Class Diagram Package DSSPROMOSI.periklanan.
Elektronik.radio.RadioMaintenanceUI**



Gambar 2.16 Class DSSPROMOSI.periklanan.elektronik.radio.
RadioMaintenanceUI

Deskripsi

Class ini merupakan GUI yang berperan sebagai *boundary class* yang digunakan pengelolaan data radio.

Atribut

- - DRadio_mng : DSSPROMOSI.periklanan.elektronik.
radio.Radio
Merepresentasikan *object entity class* untuk pengelolaan radio saat melakukan perubahan terhadap data radio.
- - CRadio_mng : DSSPROMOSI.periklanan.elektronik.
radio.RadioManager
Merepresentasikan *object control class* untuk pengelolaan data radio saat melakukan perubahan terhadap data radio.
- - user_mng : DSSPROMOSI.account.Account
Merepresentasikan *object control class* untuk pengelolaan session user.

Method

- + RadioMaintenanceUI ()

Merupakan konstruktor class, tanpa atribut terdefinisi.

2.2.4.9. Class Diagram Package DSSPROMOSI.periklanan. elektronik.televisi.Televisi



Gambar 2.17 Class DSSPROMOSI.periklanan.elektronik.televisi. Televisi

Deskripsi

Class ini merepresentasikan televisi di dalam DssPromosi.

Atribut

- - televisiid : string
Merepresentasikan id dari televisi
- - elektronikid : string
Merepresentasikan elektronik id.
- - nama : string
Merepresentasikan nama televisi.
- - divre : string
Merepresentasikan lokasi divre dari televisi.
- - kandatel: string
Merepresentasikan lokasi kandatel dari televisi.
- - lokasi : string

Merepresentasikan lokasi dari televisi.

- - starget : string

Merepresentasikan segmen target utama dari televisi.

- - skedua : string

Merepresentasikan segmen target kedua dari televisi.

- - sketiga : string

Merepresentasikan segmen target ketiga dari televisi.

- - agensi : string

Merepresentasikan agensi atau penanggung jawab televisi.

- - telp : string

Merepresentasikan no. Telp televisi

- - iklan : integer

Merepresentasikan tarif iklan di televisi tersebut

- - area : string

Merepresentasikan jangkauan televisi, nasional atau internasional.

- - eye1 : string

Merepresentasikan eye catching score dari televisi.

- - eye2 : string

Merepresentasikan eye catching score kedua dari televisi.

- - eye3 : string

Merepresentasikan eye catching score ketiga dari televisi.

Method

- + Televisi ()

Merupakan konstruktor class, tanpa atribut terdefinisi.

- + Televisi(id : String, mcid : String, name : String, div : String, kand : String, lks : String, target1 : String, target2 : String, target3 : String, frek : String, agen : String, notelp : String, Hiklan : Integer, covarea : String, ey1 : String, ey2 : String, ey3 : String)

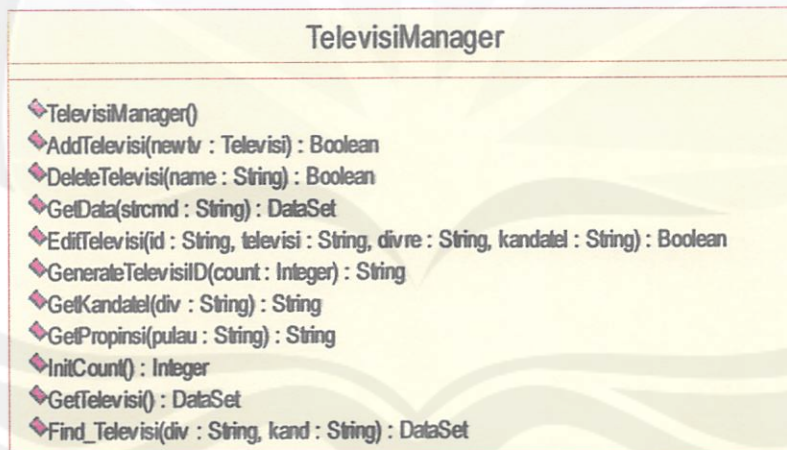
Merupakan konstruktor class, dengan atribut terdefinisi.

Parameters

- id : id televisi
- elektronikid : elektronik id.

- nama : nama televisi.
- divre : divre televisi.
- kandatel : kandatel televisi.
- lokasi : lokasi televisi.
- starget : segmen target utama dari televisi.
- skedua : segmen target kedua dari televisi.
- sketiga : segmen target ketiga dari televisi.
- agensi : agensi atau penanggung jawab televisi.
- telp : no. Telp televisi
- iklan : tarif iklan televisi
- area : jangkauan televisi.
- eye1 : eye catching score televisi.
- eye2 : eye catching score kedua televisi.
- eye3 : eye catching score ketiga televisi.

2.2.4.10. Class Diagram Package DSSPROMOSI.periklanan.elektronik.televisi.TelevisiManager



Gambar 2.18 Class DSSPROMOSI.periklanan.elektronik.televisi.
TelevisiManager

Deskripsi

Class yang berperan sebagai *object control class* untuk aksi yang berhubungan dengan data televisi.

Atribut

-

Method

- + TelevisiManager()

Konstruktor class DSSPROMOSI.periklanan. elektronik.
televisi.TelevisiManager. Buat instance baru DSSPROMOSI.
periklanan.elektronik.televisi.TelevisiManager

• + addTelevisi (newtelevisi : DSSPROMOSI.periklanan.
elektronik.televisi.Televisi) : boolean

Digunakan untuk menambahkan data televisi baru.

Parameters:

newTelevisi - DSSPROMOSI.periklanan.elektronik.
televisi.Televisi yang akan ditambahkan.

Returns:

true jika televisi berhasil ditambahkan.

• + DeleteTelevisi (name : string) : Boolean
Menghapus televisi dari database.

Parameters:

name - nama televisi.

Returns:

true jika penghapusan televisi berhasil dilakukan.

• + EditTelevisi(id : String,televisi : String, divre :
String, kandatel : String, lokasi : String, sutama :
String, skedua : String, sketiga : String, agen : String,
telp : String, iklan : String, covarea : String, sc1 :
String, sc2 : String, sc3 : String, id : String, tv :
String) : Boolean

Parameters:

id : id televisi

televisi : nama televisi

divre : divre televisi.

Kandatel : kandatel televisi.

lokasi : lokasi televisi.

starget : segmen target utama dari televisi.

skedua : segmen target kedua dari televisi.

sketiga : segmen target ketiga dari televisi.

agensi : agensi atau penanggung jawab televisi.

telp : no. Telp televisi

iklan : tarif iklan televisi

area : jangkauan televisi.
eye1 : eye catching score televisi.
eye2 : eye catching score kedua televisi.
eye3 : eye catching score ketiga televisi.

Returns:

True jika data televisi berhasil diubah.

• + GetData(strcmd : string) : DataSet

Ambil data televisi pada basis data sesuai field yang diinginkan.

Parameters:

Strcmd - query untuk mengambil data televisi pada field tertentu.

Returns:

Set data yang berisi data televisi yang tersimpan pada basis data.

• + GetTelevisi() : DataSet

Ambil semua data televisi pada basis data.

Parameters:

-

Returns:

Set data yang berisi semua data televisi yang tersimpan pada basis data.

• + GenerateTelevisiId (count : integer) : String

Mendapatkan televisi id yang telah ter-generate.

Parameters:

Count - id maximum dari televisi yang telah di counter yang tersimpan di database.

Returns:

String id.

• + GetKandatel(div : string) : DataSet

Ambil data kandatel pada basis data sesuai dengan divre tertentu.

Parameters:

div - divre dari data kandatel.

Returns:

Set data yang berisi data kandatel yang tersimpan pada basis data.

- + GetPropinsi(pulau : string) : DataSet

Ambil data propinsi pada basis data sesuai dengan pulau tertentu.

Parameters:

pulau - pulau dari propinsi.

Returns:

Set data yang berisi data propinsi yang tersimpan pada basis data.

- + InItCount () : integer

Mendapatkan maximum televisi id dan di tambah 1.

Parameters:

-

Returns:

id yang telah di counter.

- + Find_Televisi (div : string, kand : string) : DataSet

Mendapatkan class DSSPROMOSI.periklanan.elektronik.televisi.Televisi untuk data radio tertentu.

Parameters:

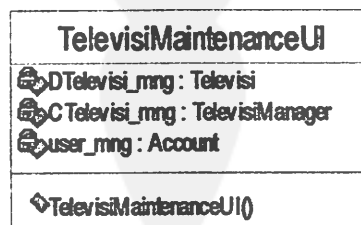
div - divre televisi yang akan dicari.

kand - kandatel televisi yang akan dicari.

Returns:

Set data televisi untuk televisi tertentu.

2.2.4.11. Class Diagram Package DSSPROMOSI.periklanan. Elektronik.televisi.TelevisiMaintenanceUI



Gambar 2.19 Class DSSPROMOSI.periklanan.elektronik.televisi. TelevisiMaintenanceUI

Deskripsi

Class ini merupakan GUI yang berperan sebagai *boundary class* yang digunakan pengelolaan data televisi.

Atribut

- - DTelevisi_mng : DSSPROMOSI.periklanan.elektronik.televisi.Televisi

Merepresentasikan *object entity class* untuk pengelolaan televisi saat melakukan perubahan terhadap data televisi.

- - CTelevisi_mng : DSSPROMOSI.periklanan.elektronik.televisi.TelevisiManager

Merepresentasikan *object control class* untuk pengelolaan data televisi saat melakukan perubahan terhadap data televisi.

- - user_mng : DSSPROMOSI.account.Account

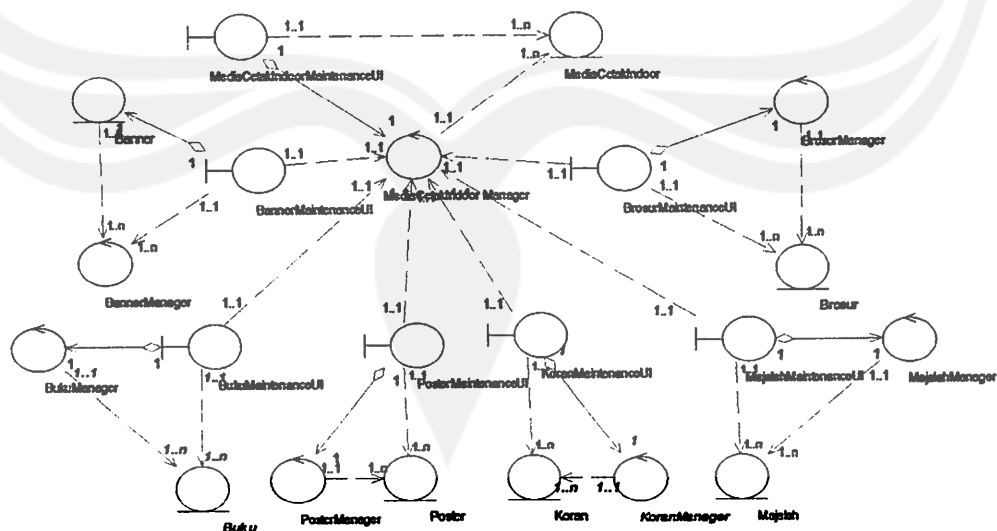
Merepresentasikan *object control class* untuk pengelolaan session user.

Method

- + TelevisiMaintenanceUI()

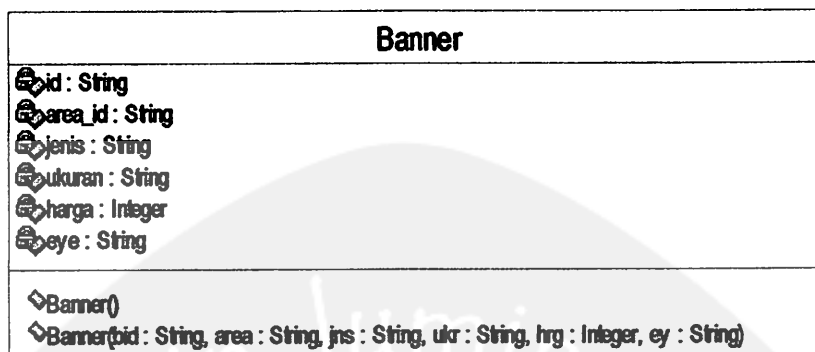
Merupakan konstruktor class, tanpa atribut terdefinisi.

2.2.4.3. Class Diagram Package DSSPROMOSI.periklanan. MediaCetak.MediaCetakIndoor



Gambar 2.20 Class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor

**2.2.4.3.1. Class Diagram Package DSSPROMOSI.periklanan.
mediacetak.mediacetakindoor.banner.Banner**



Gambar 2.21 Class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.
banner.Banner

Deskripsi

Class ini merepresentasikan banner di dalam DssPromosi.

Atribut

- - id : string
Merepresentasikan id dari banner
- - areaid : string
Merepresentasikan area id, yakni id media cetak indoor.
- - jenis : string
Merepresentasikan jenis banner.
- - ukuran : string
Merepresentasikan ukuran banner.
- - iklan : integer
Merepresentasikan tarif iklan banner.
- - eye : string
Merepresentasikan eye catching score dari banner.

Method

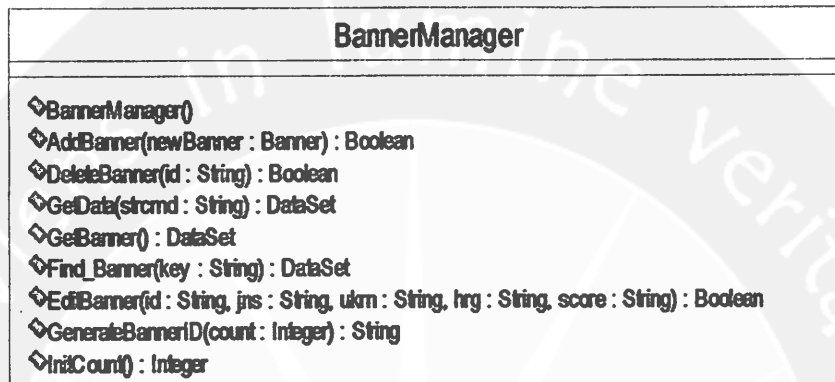
- + Banner ()
Merupakan konstruktor class, tanpa atribut terdefinisi.
- + Banner (bid : String, area : String, jns : String, ukr : String, hrg : Integer, ey : String)
Merupakan konstruktor class, dengan atribut terdefinisi.

Parameters

- bid : id banner.

- area : id area media cetak indoor.
- jns : jenis banner.
- ukr : ukuran banner.
- hrg : tarif iklan banner
- ey : eye catching score banner.

**2.2.4.3.2. Class Diagram Package DSSPROMOSI.periklanan.
mediacetak.mediacetakindoor.banner.BannerManager**



Gambar 2.22 Class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.
banner.BannerManager

Deskripsi

Class yang berperan sebagai *object control class* untuk aksi yang berhubungan dengan data banner.

Atribut

-

Method

- + BannerManager()

Konstruktor class DSSPROMOSI.periklanan. mediacetak. mediacetakindoor.banner.BannerManager. Buat instance baru DSSPROMOSI.periklanan. mediacetak. mediacetakindoor.banner.BannerManager

- + addBanner (newbanner : DSSPROMOSI.periklanan. mediacetak.mediacetakindoor. banner.Banner) : boolean
Digunakan untuk menambahkan data televisi baru.

Parameters:

newbanner - DSSPROMOSI.periklanan.mediacetak. mediacetakindoor.banner.Banner yang akan ditambahkan.

Returns:

true jika banner berhasil ditambahkan.

• + DeleteBanner (id : string) : Boolean

Menghapus banner dari database.

Parameters:

id - id banner.

Returns:

true jika penghapusan banner berhasil dilakukan.

• + EditBanner(id : String, jns : String, ukrn : String, hrg : String, score : String) : Boolean

Parameters:

- id : id banner.
- jns : id jenis banner.
- ukr : ukuran banner.
- hrg : tarif iklan banner.
- ey : eye catching score banner.

Returns:

True jika data banner berhasil diubah.

• + GetData(strcmd : string) : DataSet

Ambil data banner pada basis data sesuai field yang diinginkan.

Parameters:

Strcmd - query untuk mengambil data banner pada field tertentu.

Returns:

Set data yang berisi data banner yang tersimpan pada basis data.

• + GetBanner () : DataSet

Ambil semua data banner pada basis data.

Parameters:

-

Returns:

Set data yang berisi semua data banner yang tersimpan pada basis data.

• + GenerateBannerId (count : integer) : String

Mendapatkan banner id yang telah ter-generate.

Parameters:

Count - id maximum dari banner yang telah di counter yang tersimpan di database.

Returns:

String id.

- + InItCount () : integer

Mendapatkan maximum banner id dan di tambah 1.

Parameters:

-

Returns:

id yang telah di counter.

- + Find_Banner (key : string) : DataSet

Mendapatkan class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.Banner untuk data banner tertentu.

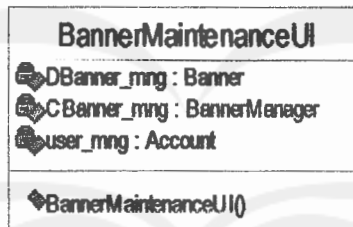
Parameters:

key - nama banner yang akan dicari.

Returns:

Set data banner untuk banner tertentu.

2.2.4.3.3. Class Diagram Package DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.banner. BannerMaintenanceUI



Gambar 2.23 Class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.
banner.BannerMaintenanceUI

Deskripsi

Class ini merupakan GUI yang berperan sebagai *boundary class* yang digunakan pengelolaan data banner.

Atribut

- DBanner_mng : DSSPROMOSI.periklanan. mediacetak.mediacetakindoor.banner.Banner

Merepresentasikan *object entity class* untuk pengelolaan banner saat melakukan perubahan terhadap data banner.

- - `CBanner_mng` : `DSSPROMOSI.periklanan. mediacetak. mediacetakindoor.banner.BannerManager`

Merepresentasikan *object control class* untuk pengelolaan data banner saat melakukan perubahan terhadap data banner.

- - `user_mng` : `DSSPROMOSI.account.Account`

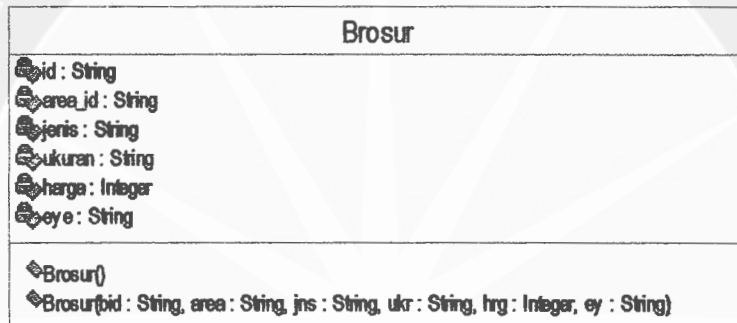
Merepresentasikan *object control class* untuk pengelolaan session user.

Method

- + `BannerMaintenanceUI()`

Merupakan konstruktor class, tanpa atribut terdefinisi.

2.2.4.3.4. Class Diagram Package `DSSPROMOSI.periklanan. mediacetak. mediacetakindoor.brosur.Brosur`



Gambar 2.24 Class `DSSPROMOSI.periklanan. mediacetak. mediacetakindoor.brosur.Brosur`

Deskripsi

Class ini merepresentasikan brosur di dalam `DssPromosi`.

Atribut

- - `id` : string

Merepresentasikan id dari brosur

- - `areaid` : string

Merepresentasikan area id, yakni id media cetak indoor.

- - `jenis` : string

Merepresentasikan jenis brosur.

- - `ukuran` : string

Merepresentasikan ukuran brosur.

- - `iklan` : integer

Merepresentasikan tarif iklan brosur.

- - eye : string

Merepresentasikan eye catching score dari brosur.

Method

- + Brosur ()

Merupakan konstruktor class, tanpa atribut terdefinisi.

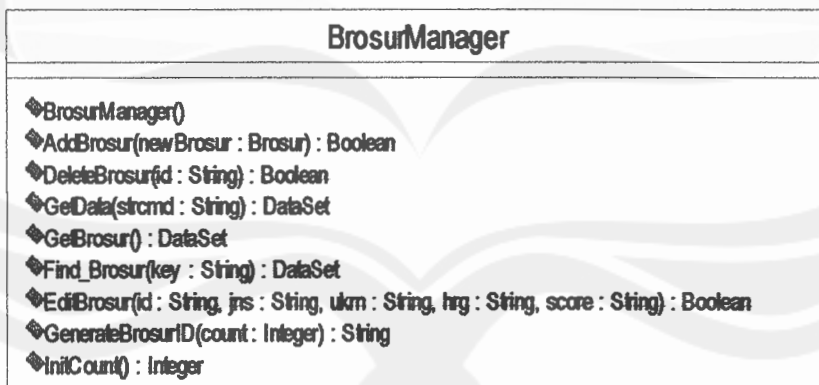
- + Brosur (bid : String, area : String, jns : String, ukr : String, hrg : Integer, ey : String)

Merupakan konstruktor class, dengan atribut terdefinisi.

Parameters

- bid : id brosur.
- area : id area media cetak indoor.
- jns : jenis brosur.
- ukr : ukuran brosur.
- hrg : tarif iklan brosur
- ey : eye catching score brosur.

2.2.4.3.5. Class Diagram Package DSSPROMOSI.periklanan. mediacetak.mediacetakindoor.brosur.BrosurManager



Gambar 2.25 Class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor. brosur.BrosurManager

Deskripsi

Class yang berperan sebagai *object control class* untuk aksi yang berhubungan dengan data brosur.

Atribut

-

Method

• + BrosurManager()

Konstruktor class DSSPROMOSI.periklanan. mediacetak. mediacetakindoor.brosur.BrosurManager. Buat instance baru DSSPROMOSI.periklanan. mediacetak. mediacetakindoor. brosur.BrosurManager

• + addBrosur (newbrosur : DSSPROMOSI.periklanan. mediacetak. mediacetakindoor.brosur.Brosur) : boolean
Digunakan untuk menambahkan data brosur baru.

Parameters:

newbrosur - DSSPROMOSI.periklanan. mediacetak. mediacetakindoor.brosur.Brosur yang akan ditambahkan.

Returns:

true jika brosur berhasil ditambahkan.

• + DeleteBrosurr (id : string) : Boolean
Menghapus brosur dari database.

Parameters:

id - id brosur.

Returns:

true jika penghapusan brosur berhasil dilakukan.

• + EditBrosur (id : String, jns : String, uknr : String, hrg : String, score : String) : Boolean

Parameters:

- id : id brosur.
- jns : id jenis brosur.
- ukr : ukuran brosur.
- hrg : tarif iklan brosur.
- ey : eye catching score brosur.

Returns:

True jika data banner berhasil diubah.

• + GetData(strcmd : string) : DataSet

Ambil data brosur pada basis data sesuai field yang diinginkan.

Parameters:

Strcmd - query untuk mengambil data brosur pada field tertentu.

Returns:

Set data yang berisi data brosur yang tersimpan pada basis data.

- + GetBrosur () : DataSet

Ambil semua data brosur pada basis data.

Parameters:

-

Returns:

Set data yang berisi semua data brosur yang tersimpan pada basis data.

- + GenerateBrosurId (count : integer) : String

Mendapatkan brosur id yang telah ter-generate.

Parameters:

Count - id maximum dari brosur yang telah di counter yang tersimpan di database.

Returns:

String id.

- + InItCount () : integer

Mendapatkan maximum brosur id dan di tambah 1.

Parameters:

-

Returns:

id yang telah di counter.

- + Find_Brosur (key : string) : DataSet

Mendapatkan class DSSPROMOSI.periklanan.mediapetak.mediapetakindoor.brosur.Brosur untuk data brosur tertentu.

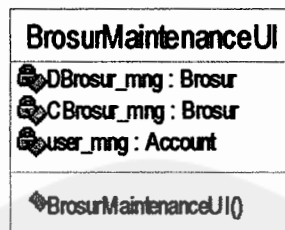
Parameters:

key - nama brosur yang akan dicari.

Returns:

Set data brosur untuk brosur tertentu.

**2.2.4.3.6. Class Diagram Package DSSPROMOSI.periklanan.
mediacetak.mediacetakindoor.brosur.BrosurMaintenanceUI**



Gambar 2.26 Class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.
brosur.BrosurMaintenanceUI

Deskripsi

Class ini merupakan GUI yang berperan sebagai *boundary class* yang digunakan pengelolaan data brosur.

Atribut

- **DBrosur_mng** : DSSPROMOSI.periklanan. mediacetak. mediacetakindoor.brosur.Brosur

Merepresentasikan *object entity class* untuk pengelolaan brosur saat melakukan perubahan terhadap data brosur.

- - **CBrosur_mng** : DSSPROMOSI.periklanan. mediacetak. mediacetakindoor.brosur.BrosurManager

Merepresentasikan *object control class* untuk pengelolaan data brosur saat melakukan perubahan terhadap data brosur.

- - **user_mng** : DSSPROMOSI.account.Account

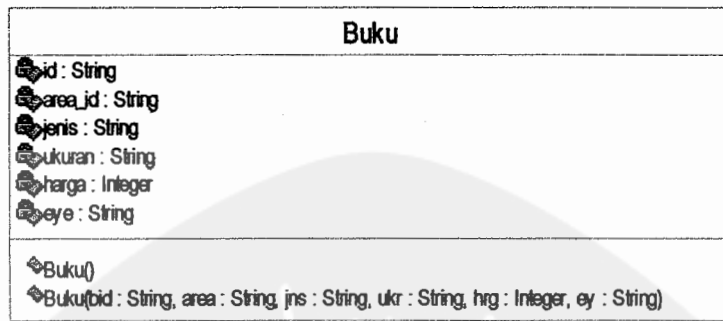
Merepresentasikan *object control class* untuk pengelolaan session user.

Method

- + **BrosurMaintenanceUI()**

Merupakan konstruktor class, tanpa atribut terdefinisi.

**2.2.4.3.7. Class Diagram Package DSSPROMOSI.periklanan.
mediacetak.mediacetakindoor.buku.Buku**



Gambar 2.27 Class DSSPROMOSI.periklanan.
mediacetak.mediacetakindoor.buku.Buku

Deskripsi

Class ini merepresentasikan buku di dalam DssPromosi.

Atribut

- - id : string
Merepresentasikan id dari buku
- - areaid : string
Merepresentasikan area id, yakni id media cetak indoor.
- - jenis : string
Merepresentasikan jenis buku.
- - ukuran : string
Merepresentasikan ukuran buku.
- - iklan : integer
Merepresentasikan tarif iklan buku.
- - eye : string
Merepresentasikan eye catching score dari buku.

Method

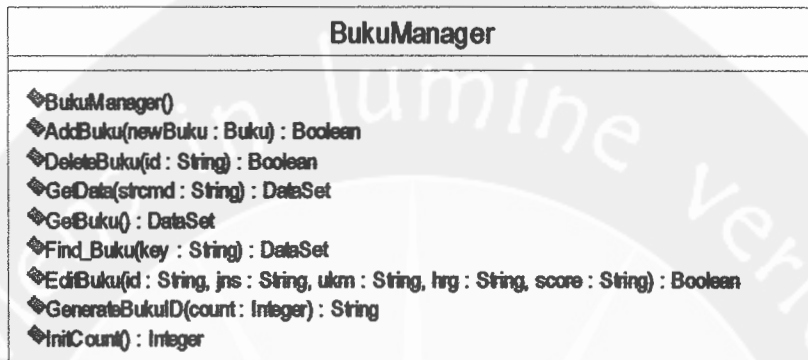
- + Buku()
Merupakan konstruktor class, tanpa atribut terdefinisi.
- + Buku (bid : String, area : String, jns : String, ukr : String, hrg : Integer, ey : String)
Merupakan konstruktor class, dengan atribut terdefinisi.

Parameters

- bid : id buku.
- area : id area media cetak indoor.

- jns : jenis buku.
- ukr : ukuran buku.
- hrg : tarif iklan buku
- ey : eye catching score buku.

2.2.4.3.8. Class Diagram Package DSSPROMOSI.periklanan. mediacetak.mediacetakindoor.buku.BukuManager



Gambar 2.28 Class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor. buku.BukuManager

Deskripsi

Class yang berperan sebagai *object control class* untuk aksi yang berhubungan dengan data buku.

Atribut

-

Method

- + BukuManager()

Konstruktor class DSSPROMOSI.periklanan. mediacetak. mediacetak.mediacetakindoor.buku.BukuManager. Buat instance baru DSSPROMOSI.periklanan.mediacetak.mediacetakindoor. buku.BukuManager

- + addBuku (newbuku : DSSPROMOSI.periklanan. mediacetak.mediacetakindoor.buku.Buku) : boolean
Digunakan untuk menambahkan data buku baru.

Parameters:

newbuku - DSSPROMOSI.periklanan.mediacetak. mediacetakindoor.buku.Buku yang akan ditambahkan.

Returns:

true jika buku berhasil ditambahkan.

• + DeleteBuku (id : string) : Boolean

Menghapus buku dari database.

Parameters:

id - id buku.

Returns:

true jika penghapusan buku berhasil dilakukan.

• + EditBuku (id : String, jns : String, ukrn : String, hrg : String, score : String) : Boolean

Parameters:

- id : id buku.
- jns : jenis buku.
- ukr : ukuran buku.
- hrg : tarif iklan buku.
- ey : eye catching score buku.

Returns:

True jika data buku berhasil diubah.

• + GetData(strcmd : string) : DataSet

Ambil data buku pada basis data sesuai field yang diinginkan.

Parameters:

Strcmd - query untuk mengambil data buku pada field tertentu.

Returns:

Set data yang berisi data buku yang tersimpan pada basis data.

• + GetBuku () : DataSet

Ambil semua data buku pada basis data.

Parameters:

-

Returns:

Set data yang berisi semua data buku yang tersimpan pada basis data.

• + GenerateBukuId (count : integer) : String

Mendapatkan buku id yang telah ter-generate.

Parameters:

Count - id maximum dari buku yang telah di counter yang tersimpan di database.

Returns:

String id.

- + InItCount () : integer

Mendapatkan maximum buku id dan di tambah 1.

Parameters:

-

Returns:

id yang telah di counter.

- + Find_Buku (key : string) : DataSet

Mendapatkan class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.buku.Buku untuk data buku tertentu.

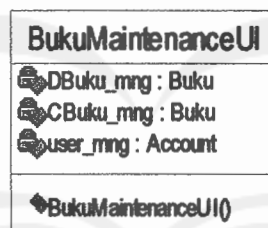
Parameters:

key - nama buku yang akan dicari.

Returns:

Set data buku untuk buku tertentu.

2.2.4.3.9. Class Diagram Package DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.buku.BukuMaintenanceUI



Gambar 2.29 Class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.buku.BukuMaintenanceUI

Deskripsi

Class ini merupakan GUI yang berperan sebagai *boundary class* yang digunakan pengelolaan data banner.

Atribut

- - DBuku_mng : DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.buku.Buku

Merepresentasikan *object entity class* untuk pengelolaan buku saat melakukan perubahan terhadap data buku.

- - CBuku_mng : DSSPROMOSI.periklanan. mediacetak. mediacetakindoor.buku.BukuManager

Merepresentasikan *object control class* untuk pengelolaan data buku saat melakukan perubahan terhadap data buku.

- - user_mng : DSSPROMOSI.account.Account

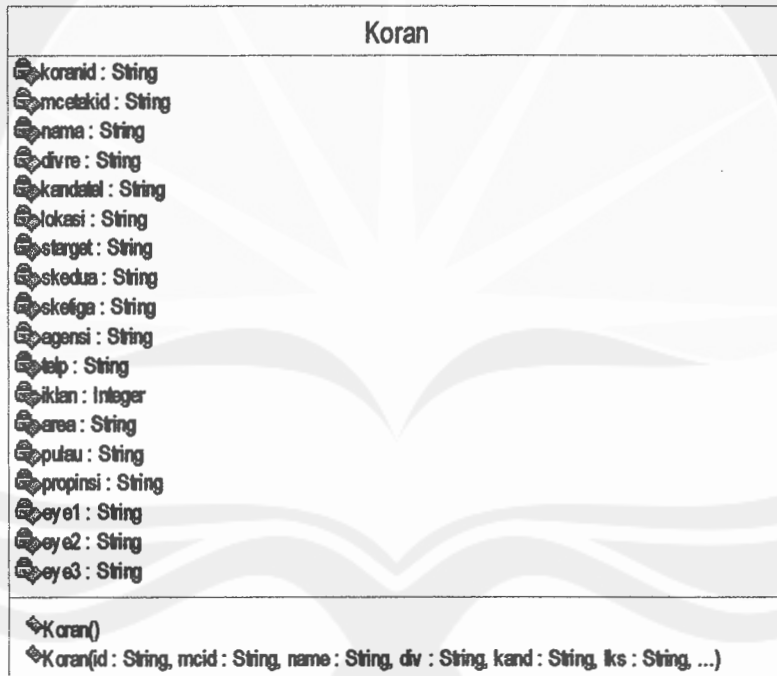
Merepresentasikan *object control class* untuk pengelolaan session user.

Method

- + BukuMaintenanceUI()

Merupakan konstruktor class, tanpa atribut terdefinisi.

2.2.4.3.10. Class Diagram Package DSSPROMOSI.periklanan. mediacetak.mediacetakindoor.koran.Koran



Gambar 2.30 Class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor. koran.Koran

Deskripsi

Class ini merepresentasikan koran di dalam DssPromosi.

Atribut

- - koranid : string

Merepresentasikan id dari koran

- - mceatamid: string

Merepresentasikan media cetak id.

- - nama : string

Merepresentasikan nama koran.

- - divre : string

Merepresentasikan lokasi divre dari koran.

- - kandatel: string

Merepresentasikan lokasi kandatel dari koran.

- - lokasi : string

Merepresentasikan lokasi dari koran.

- - starget : string

Merepresentasikan segmen target utama dari koran.

- - skedua : string

Merepresentasikan segmen target kedua dari koran.

- - sketiga : string

Merepresentasikan segmen target ketiga dari koran.

- - agensi : string

Merepresentasikan agensi atau penanggung jawab koran.

- - telp : string

Merepresentasikan no. Telp koran

- - iklan : integer

Merepresentasikan tarif iklan di koran tersebut

- - area : string

Merepresentasikan jangkauan koran, nasional atau internasional.

- - pulau : string

Merepresentasikan jangkauan pulau koran.

- - propinsi: string

Merepresentasikan jangkauan propinsi koran.

- - eye1 : string

Merepresentasikan eye catching score dari koran.

- - eye2 : string

Merepresentasikan eye catching score kedua dari koran.

- - eye3 : string

Merepresentasikan eye catching score ketiga dari koran.

Method

- + Koran()

Merupakan konstruktor class, tanpa atribut terdefinisi.

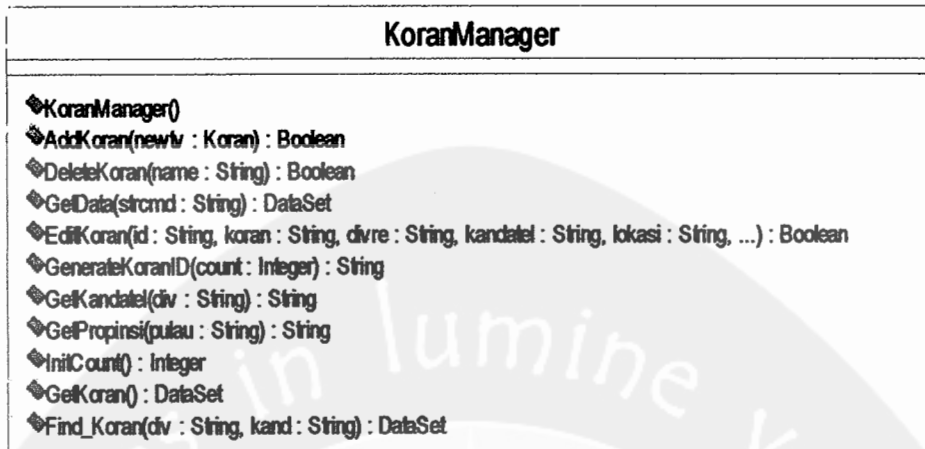
- + Koran(id : String, mcid : String, name : String, div : String, kand : String, lks : String, target1 : String, target2 : String, target3 : String, agen : String, notelp : String, Hiklan : Integer, covarea : String, pl : String, prop : String, eyl : String, ey2 : String, ey3 : String)

Merupakan konstruktor class, dengan atribut terdefinisi.

Parameters

- id : id koran
- mcid : media cetak indoor id.
- nama : nama koran.
- divre : divre koran.
- kandatel : kandatel koran.
- lokasi : lokasi koran.
- starget : segmen target utama dari koran.
- skedua : segmen target kedua dari koran.
- sketiga : segmen target ketiga dari koran.
- agensi : agensi atau penanggung jawab koran.
- telp : no. Telp koran.
- iklan : tarif iklan koran.
- area : jangkauan koran.
- pulau : jangkauan pulau koran.
- propinsi : jangkauan propinsi koran.
- eyl : eye catching score koran.
- ey2 : eye catching score kedua koran.
- ey3 : eye catching score ketiga koran.

2.2.4.3.11. Class Diagram Package DSSPROMOSI.periklanan.
mediacetak.mediacetakindoor.koran.KoranManager



Gambar 2.31 Class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.
buku.BukuManager

Deskripsi

Class yang berperan sebagai *object control class* untuk aksi yang berhubungan dengan data buku.

Atribut

-

Method

- + KoranManager()

Konstruktor class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.koran.KoranManager. Buat instance baru DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.koran.KoranManager

- + addKoran (newkoran : DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.koran.Koran) : boolean
Digunakan untuk menambahkan data koran baru.

Parameters:

newkoran - DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.koran.Koran yang akan ditambahkan.

Returns:

true jika koran berhasil ditambahkan.

- + DeleteKoran (nama : string) : Boolean
Menghapus koran dari database.

Parameters:

nama - nama koran.

Returns:

true jika penghapusan koran berhasil dilakukan.

• + EditKoran(id : String, koran : String, divre : String, kandatel : String, lokasi : String, sutama : String, skedua : String, sketiga : String, agen : String, telp : String, iklan : String, covarea : String, pulau : String, prop : String, sc1 : String, sc2 : String, sc3 : String) : Boolean

Parameters:

- id : id koran
- koran : nama koran.
- divre : divre koran.
- kandatel : kandatel koran.
- lokasi : lokasi koran.
- sutama : segmen target utama dari koran.
- skedua : segmen target kedua dari koran.
- sketiga : segmen target ketiga dari koran.
- agensi : agensi atau penanggung jawab koran.
- telp : no. Telp koran.
- iklan : tarif iklan koran.
- area : jangkauan koran.
- pulau : jangkauan pulau koran.
- propinsi : jangkauan propinsi koran.
- sc1 : eye catching score koran.
- sc2 : eye catching score kedua koran.
- sc3 : eye catching score ketiga koran.

Returns:

True jika data koran berhasil diubah.

• + GetData(strcmd : string) : DataSet

Ambil data koran pada basis data sesuai field yang diinginkan.

Parameters:

Strcmd - query untuk mengambil data koran pada field tertentu.

Returns:

Set data yang berisi data koran yang tersimpan pada basis data.

- + GetKoran () : DataSet

Ambil semua data koran pada basis data.

Parameters:

-

Returns:

Set data yang berisi semua data koran yang tersimpan pada basis data.

- + GenerateKoranId (count : integer) : String

Mendapatkan koran id yang telah ter-generate.

Parameters:

Count - id maximum dari koran yang telah di counter yang tersimpan di database.

Returns:

String id.

- + InItCount () : integer

Mendapatkan maximum koran id dan di tambah 1.

Parameters:

-

Returns:

id yang telah di counter.

- + Find_Koran (div : string, kand : string) : DataSet

Mendapatkan class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.koran.Koran untuk data koran tertentu.

Parameters:

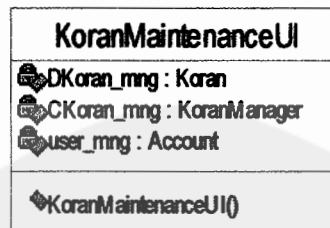
div - divre koran yang akan dicari.

kand - kandatel koran yang akan dicari.

Returns:

Set data koran untuk koran tertentu.

2.2.4.3.12. Class Diagram Package DSSPROMOSI.periklanan.
mediacetak.mediacetakindoor.koran.
KoranMaintenanceUI



Gambar 2.32 Class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.
koran.KoranMaintenanceUI

Deskripsi

Class ini merupakan GUI yang berperan sebagai *boundary class* yang digunakan pengelolaan data koran.

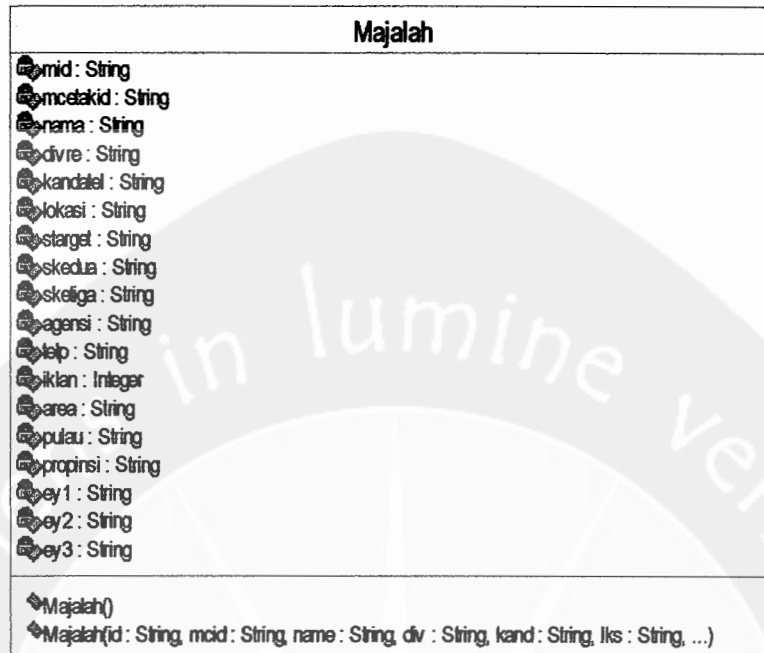
Atribut

- - DKoran_mng : DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.koran.Koran
Merepresentasikan *object entity class* untuk pengelolaan koran saat melakukan perubahan terhadap data koran.
- - CKoran_mng : DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.koran.KoranManager
Merepresentasikan *object control class* untuk pengelolaan data koran saat melakukan perubahan terhadap data koran.
- - user_mng : DSSPROMOSI.account.Account
Merepresentasikan *object control class* untuk pengelolaan session user.

Method

- + KoranMaintenanceUI()
Merupakan konstruktor class, tanpa atribut terdefinisi.

2.2.4.3.13. Class Diagram Package DSSPROMOSI.periklanan.
mediacetak.mediacetakindoor.majalah.Majalah



Gambar 2.33 Class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.
majalah.Majalah

Deskripsi

Class ini merepresentasikan majalah di dalam DssPromosi.

Atribut

- - mid : string
Merepresentasikan id dari majalah.
- - mctetakid: string
Merepresentasikan media cetak id.
- - nama : string
Merepresentasikan nama majalah.
- - divre : string
Merepresentasikan lokasi divre dari majalah.
- - kandatel: string
Merepresentasikan lokasi kandatel dari majalah.
- - lokasi : string
Merepresentasikan lokasi dari majalah.
- - starget : string

Merepresentasikan segmen target utama dari majalah.

- - skedua : string

Merepresentasikan segmen target kedua dari majalah.

- - sketiga : string

Merepresentasikan segmen target ketiga dari majalah.

- - agensi : string

Merepresentasikan agensi atau penanggung jawab majalah.

- - telp : string

Merepresentasikan no. Telp majalah

- - iklan : integer

Merepresentasikan tarif iklan di majalah tersebut

- - area : string

Merepresentasikan jangkauan majalah, nasional atau internasional.

- - pulau : string

Merepresentasikan jangkauan pulau majalah.

- - propinsi : string

Merepresentasikan jangkauan propinsi majalah.

- - ey1 : string

Merepresentasikan eye catching score dari majalah.

- - ey2 : string

Merepresentasikan eye catching score kedua dari majalah.

- - ey3 : string

Merepresentasikan eye catching score ketiga dari majalah.

Method

- + Majalah()

Merupakan konstruktor class, tanpa atribut terdefinisi.

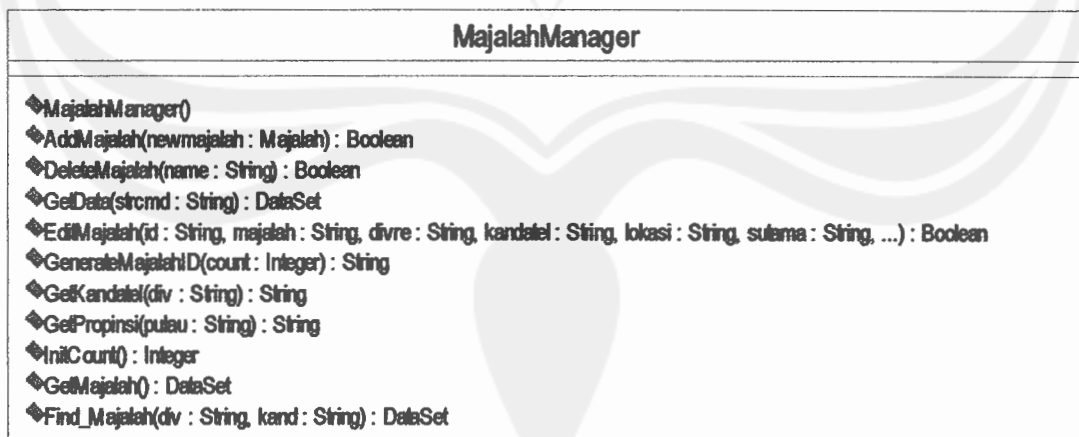
- + Majalah(id : String, mcid : String, name : String, div : String, kand : String, lks : String, target1 : String, target2 : String, target3 : String, agen : String, notelp : String, Hiklan : Integer, covarea : String, pl : String, prop : String, ey1 : String, ey2 : String, ey3 : String)

Merupakan konstruktor class, dengan atribut terdefinisi.

Parameters

- id : id majalah.
- mcid : media cetak id.
- name : nama majalah.
- div : divre majalah.
- kand : kandatel majalah.
- lks : lokasi majalah.
- target : segmen target utama dari majalah.
- target2 : segmen target kedua dari majalah.
- target3 : segmen target ketiga dari majalah.
- agen : agensi atau penanggung jawab majalah.
- notelp : no. Telp majalah.
- Hiklan : tarif iklan majalah.
- covarea: jangkauan majalah.
- pl : jangkauan majalah.
- prop : jangkauan majalah.
- ey1 : eye catching score majalah.
- ey2 : eye catching score kedua majalah.
- ey3 : eye catching score ketiga majalah.

2.2.4.3.14. Class Diagram Package DSSPROMOSI.periklanan. mediacetak.mediacetakindoor.majalah.MajalahManager



Gambar 2.34 Class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.
majalah.MajalahManager

Deskripsi

Class yang berperan sebagai *object control class* untuk aksi yang berhubungan dengan data majalah.

Atribut

-

Method

- + MajalahManager()

Konstruktor class DSSPROMOSI.periklanan.mediacetak.mediacetak.mediacetakindoor.majalah.MajalahManager. Buat instance baru DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.majalah.MajalahManager

- + addMajalah (newmajalah : DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.majalah.MajalahManager) : boolean

Digunakan untuk menambahkan data majalah baru.

Parameters:

newmajalah - DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.majalah.Majalah yang akan ditambahkan

Returns:

true jika majalah berhasil ditambahkan.

- + DeleteMajalah (name : string) : Boolean

Menghapus majalah dari database.

Parameters:

nama - nama majalah.

Returns:

true jika penghapusan majalah berhasil dilakukan.

- + EditMajalah(id : String, majalah : String, divre : String, kandatel : String, lokasi : String, sutama : String, skedua : String, sketiga : String, agen : String, telp : String, iklan : String, covarea : String, pulau : String, prop : String, sc1 : String, sc2 : String, sc3 : String) : Boolean

Parameters:

- id : id majalah.
- majalah : nama majalah.
- divre : divre majalah.

- kandatel: kandatel majalah.
- lokasi : lokasi majalah.
- sutama : segmen target utama dari majalah.
- skedua : segmen target kedua dari majalah.
- sketiga : segmen target ketiga dari majalah.
- agen : agensi atau penanggung jawab majalah.
- telp : no. Telp majalah.
- iklan : tarif iklan majalah.
- covarea : jangkauan majalah.
- pulau : jangkauan majalah.
- prop : jangkauan majalah.
- sc1 : eye catching score majalah.
- sc2 : eye catching score kedua majalah.
- sc3 : eye catching score ketiga majalah.

Returns:

True jika data buku berhasil diubah.

- + GetData(strcmd : string) : DataSet

Ambil data majalah pada basis data sesuai field yang diinginkan.

Parameters:

Strcmd - query untuk mengambil data majalah pada field tertentu.

Returns:

Set data yang berisi data majalah yang tersimpan pada basis data.

- + GetMajalah () : DataSet

Ambil semua data majalah pada basis data.

Parameters:

-

Returns:

Set data yang berisi semua data majalah yang tersimpan pada basis data.

- + GenerateMajalahId (count : integer) : String

Mendapatkan majalah id yang telah ter-generate.

Parameters:

Count - id maximum dari majalah yang telah di counter yang tersimpan di database.

Returns:

String id.

- + InItCount () : integer

Mendapatkan maximum majalah id dan di tambah 1.

Parameters:

-

Returns:

id yang telah di counter.

- + Find_Majalah (div : string, kand : string) : DataSet

Mendapatkan class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.majalah.Majalah untuk data majalah tertentu.

Parameters:

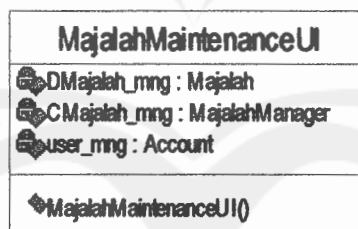
div - nama majalah yang akan dicari.

kand - nama majalah yang akan dicari.

Returns:

Set data majalah untuk majalah tertentu.

2.2.4.3.15. Class Diagram Package DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.majalah.MajalahMaintenanceUI



Gambar 2.35 Class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.majalah.MajalahMaintenanceUI

Deskripsi

Class ini merupakan GUI yang berperan sebagai *boundary class* yang digunakan pengelolaan data majalah.

Atribut

- - DMajalah_mng : DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.majalah.Majalah

Merepresentasikan *object entity class* untuk pengelolaan majalah saat melakukan perubahan terhadap data majalah.

- - CMajalah_mng : DSSPROMOSI.periklanan. mediacetak. mediacetakindoor.majalah.MajalahManager

Merepresentasikan *object control class* untuk pengelolaan data majalah saat melakukan perubahan terhadap data majalah.

- - user_mng : DSSPROMOSI.account.Account

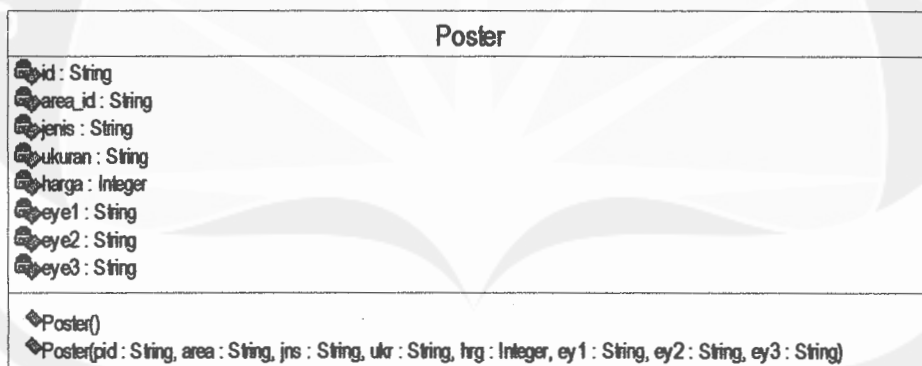
Merepresentasikan *object control class* untuk pengelolaan session user.

Method

- + MajalahMaintenanceUI()

Merupakan konstruktor class, tanpa atribut terdefinisi.

2.2.4.3.16. Class Diagram Package DSSPROMOSI.periklanan. mediacetak.mediacetakindoor.poster.Poster



Gambar 2.36 Class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor. poster.Poster

Deskripsi

Class ini merepresentasikan poster di dalam DssPromosi.

Atribut

- - id : string

Merepresentasikan id dari poster

- - areaid : string

Merepresentasikan area id, yakni id media cetak indoor.

- - jenis : string

Merepresentasikan jenis poster.

- - ukuran : string

Merepresentasikan ukuran poster.

- - harga : integer

Merepresentasikan harga poster.

- - eye1 : string

Merepresentasikan eye catching score dari poster.

- - eye2 : string

Merepresentasikan eye catching score dari poster.

- - eye3 : string

Merepresentasikan eye catching score dari poster.

Method

- + Poster()

Merupakan konstruktor class, tanpa atribut terdefinisi.

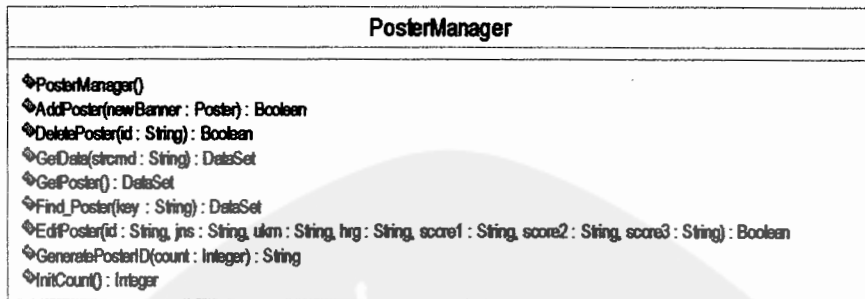
- + Poster(pid : String, area : String, jns : String, ukr : String, hrg : Integer, ey1 : String, ey2 : String, ey3 : String)

Merupakan konstruktor class, dengan atribut terdefinisi.

Parameters

- pid : id poster.
- area : id area media cetak indoor.
- jns : jenis poster.
- ukr : ukuran poster.
- hrg : tarif iklan poster
- ey1 : eye catching score poster.
- ey2 : eye catching score kedua poster.
- ey3 : eye catching score ketiga poster.

**2.2.4.3.17. Class Diagram Package DSSPROMOSI.periklanan.
mediacetak.mediacetakindoor.poster.PosterManager**



Gambar 2.37 Class DSSPROMOSI.periklanan.
mediacetak.mediacetakindoor.poster.PosterManager

Deskripsi

Class yang berperan sebagai *object control class* untuk aksi yang berhubungan dengan data poster.

Atribut

-

Method

- + PosterManager()

Konstruktor class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.poster.PosterManager. Buat instance baru DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.poster.PosterManager

- + addPoster (newposter : DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.poster.Poster) : boolean

Digunakan untuk menambahkan data poster baru.

Parameters:

newposter - DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.poster.Poster yang akan ditambahkan.

Returns:

true jika poster berhasil ditambahkan.

- + DeletePoster (id : string) : Boolean

Menghapus poster dari database.

Parameters:

id - id poster.

Returns:

true jika penghapusan poster berhasil dilakukan.

• + EditPoster(id : String, jns : String, ukrn : String, hrg : String, score1 : String, score2 : String, score3 : String) : Boolean

Parameters:

- id : id poster.
- jns : jenis poster.
- ukrn : ukuran poster.
- hrg : tarif iklan poster.
- score1 : eye catching score poster.
- score2 : eye catching score kedua poster.
- score3 : eye catching score ketiga poster.

Returns:

True jika data poster berhasil diubah.

• + GetData(strcmd : string) : DataSet

Ambil data poster pada basis data sesuai field yang diinginkan.

Parameters:

Strcmd - query untuk mengambil data poster pada field tertentu.

Returns:

Set data yang berisi data poster yang tersimpan pada basis data.

• + GetPoster () : DataSet

Ambil semua data poster pada basis data.

Parameters:

-

Returns:

Set data yang berisi semua data poster yang tersimpan pada basis data.

• + GeneratePosterId (count : integer) : String

Mendapatkan poster id yang telah ter-generate.

Parameters:

Count - id maximum dari poster yang telah di counter yang tersimpan di database.

Returns:

String id.

- + InItCount () : integer

Mendapatkan maximum poster id dan di tambah 1.

Parameters:

-

Returns:

id yang telah di counter.

- + Find_Poster (key : string) : DataSet

Mendapatkan class DSSPROMOSI.periklanan.mediacetak.mediacetakindoor.poster.Poster untuk data buku tertentu.

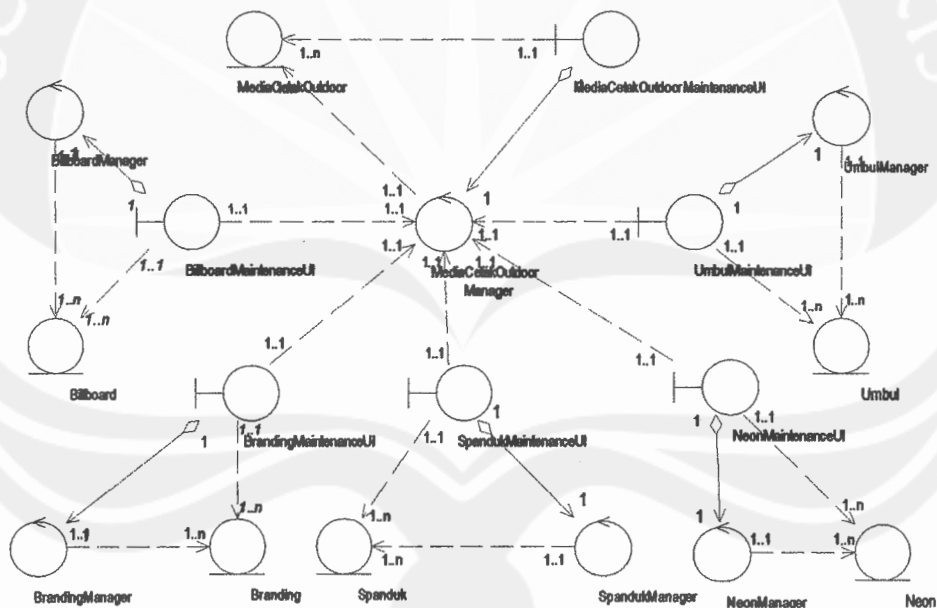
Parameters:

key - nama poster yang akan dicari.

Returns:

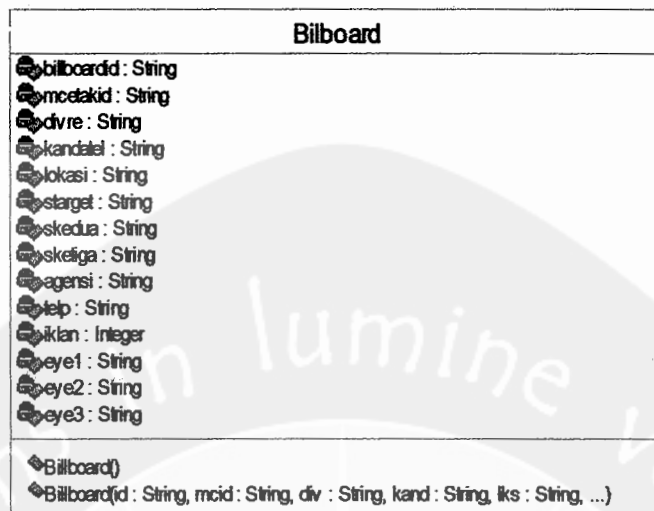
Set data poster untuk poster tertentu.

2.2.4.4. Class Diagram Package DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor



Gambar 2.38 Class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.

2.2.4.4.1. Class Diagram Package DSSPROMOSI.periklanan.
mediacetak.mediacetakoutdoor.billboard.Billboard



Gambar 2.39 Class DSSPROMOSI.periklanan.mediacetak.
mediacetakoutdoor.billboard.Billboard

Deskripsi

Class ini merepresentasikan billboard di dalam DssPromosi.

Atribut

- - billboardid : string
Merepresentasikan id dari billboard
- - mcetakid: string
Merepresentasikan media cetak id.
- - divre : string
Merepresentasikan lokasi divre dari billboard.
- - kandatel: string
Merepresentasikan lokasi kandatel dari billboard.
- - lokasi : string
Merepresentasikan lokasi dari billboard.
- - starget : string
Merepresentasikan segmen target utama dari billboard.
- - skedua : string
Merepresentasikan segmen target kedua dari billboard.
- - sketiga : string
Merepresentasikan segmen target ketiga dari billboard.

- - `agensi` : `string`

Merepresentasikan agensi atau penanggung jawab billboard.

- - `telp` : `string`

Merepresentasikan no. Telp billboard

- - `iklan` : `integer`

Merepresentasikan tarif iklan di billboard tersebut

- - `eyel` : `string`

Merepresentasikan eye catching score dari billboard.

- - `eye2` : `string`

Merepresentasikan eye catching score kedua dari billboard.

- - `eye3` : `string`

Merepresentasikan eye catching score ketiga dari billboard.

Method

- + `Billboard()`

Merupakan konstruktor class, tanpa atribut terdefinisi.

- + `Billboard(id : String, mcid : String, div : String, kand : String, lks : String, target1 : String, target2 : String, target3 : String, agen : String, notelp : String, Hiklan : Integer, eyl : String, ey2 : String, ey3 : String)`

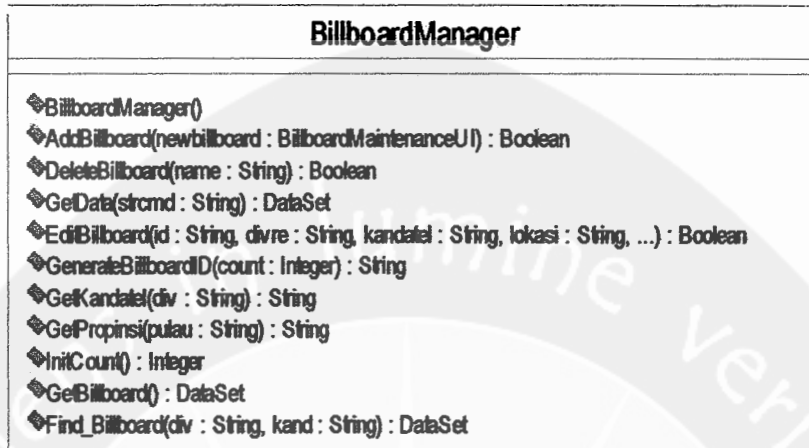
Merupakan konstruktor class, dengan atribut terdefinisi.

Parameters

- `id` : id billboard
- `mcid` : media cetak indoor id.
- `divre` : divre billboard.
- `kandatel` : kandatel billboard.
- `lokasi` : lokasi billboard.
- `starget` : segmen target utama dari billboard.
- `skedua` : segmen target kedua dari billboard.
- `sketiga` : segmen target ketiga dari billboard.
- `agensi` : agensi atau penanggung jawab billboard.
- `telp` : no. Telp billboard.
- `iklan` : tarif iklan billboard.
- `eyel` : eye catching score billboard.
- `eye2` : eye catching score kedua billboard.

- eye3 : eye catching score ketiga billboard.

2.2.4.4.2. Class Diagram Package DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.billboard.BillboardManager



Gambar 2.40 Class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.billboard.BillboardManager

Deskripsi

Class yang berperan sebagai *object control class* untuk aksi yang berhubungan dengan data billboard.

Atribut

-

Method

• + BillboardManager()

Konstruktor class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.billboard.BillboardManager. Buat instance baru DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.billboard.BillboardManager

• + addBillboard (newbillboard : DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.billboard.Billboard) : boolean

Digunakan untuk menambahkan data billboard baru.

Parameters:

newbillboard - DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.billboard.Billboard yang akan ditambahkan.

Returns:

true jika billboard berhasil ditambahkan.

• + DeleteBillboard (nama : string) : Boolean

Menghapus billboard dari database.

Parameters:

nama - nama billboard.

Returns:

true jika penghapusan billboard berhasil dilakukan.

• + EditBillboard(id : String, divre : String, kandatel : String, lokasi : String, sutama : String, skedua : String, sketiga : String, agen : String, telp : String, iklan : String, scl : String, sc2 : String, sc3 : String) : Boolean

Parameters:

- id : id billboard
- divre : divre billboard.
- kandatel : kandatel billboard.
- lokasi : lokasi billboard.
- sutama : segmen target utama dari billboard.
- skedua : segmen target kedua dari billboard.
- sketiga : segmen target ketiga dari billboard.
- agensi : agensi atau penanggung jawab billboard.
- telp : no. Telp periklanan billboard.
- iklan : tarif iklan billboard.
- scl : eye catching score billboard.
- sc2 : eye catching score kedua billboard.
- sc3 : eye catching score ketiga billboard.

Returns:

True jika data billboard berhasil diubah.

• + GetData(strcmd : string) : DataSet

Ambil data billboard pada basis data sesuai field yang diinginkan.

Parameters:

Strcmd - query untuk mengambil data billboard pada field tertentu.

Returns:

Set data yang berisi data billboard yang tersimpan pada basis data.

- + GetBillboard () : DataSet

Ambil semua data billboard pada basis data.

Parameters:

-

Returns:

Set data yang berisi semua data billboard yang tersimpan pada basis data.

- + GenerateBillboardId (count : integer) : String

Mendapatkan billboard id yang telah ter-generate.

Parameters:

Count - id maximum dari billboard yang telah di counter yang tersimpan di database.

Returns:

String id.

- + InItCount () : integer

Mendapatkan maximum billboard id dan di tambah 1.

Parameters:

-

Returns:

id yang telah di counter.

- + Find_Billboard (div : string, kand : string) : DataSet

Mendapatkan class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.billboard.Billboard untuk data billboard tertentu.

Parameters:

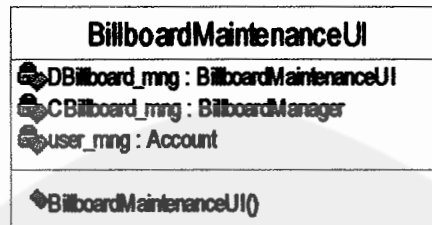
div - divre billboard yang akan dicari.

kand - kandatel billboard yang akan dicari.

Returns:

Set data billboard untuk billboard tertentu.

**2.2.4.4.3. Class Diagram Package DSSPROMOSI.periklanan.
mediacetak.mediacetakoutdoor.billboard.
BillboardMaintenanceUI**



Gambar 2.41 Class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.billboard.BillboardMaintenanceUI

Deskripsi

Class ini merupakan GUI yang berperan sebagai *boundary class* yang digunakan pengelolaan data billboard.

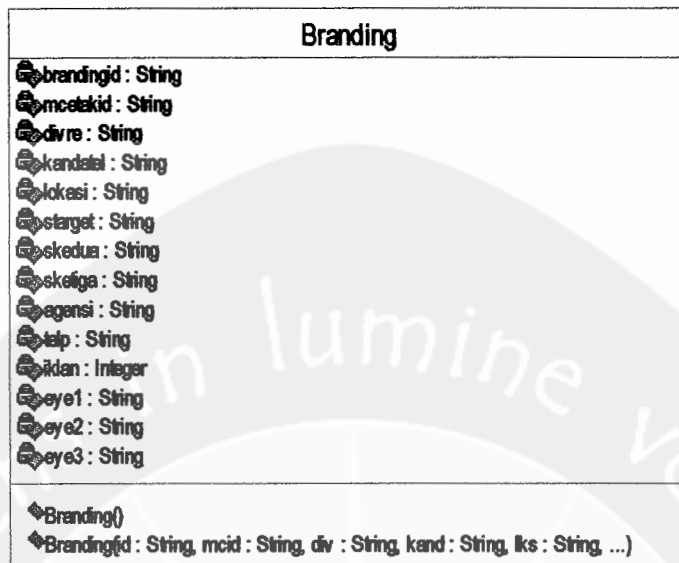
Atribut

- - **DBillboard_mng** : DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.billboard.Billboard
Merepresentasikan *object entity class* untuk pengelolaan billboard saat melakukan perubahan terhadap data billboard.
- - **CBillboard_mng** : DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.billboard.BillboardManager
Merepresentasikan *object control class* untuk pengelolaan data billboard saat melakukan perubahan terhadap data billboard.
- - **user_mng** : DSSPROMOSI.account.Account
Merepresentasikan *object control class* untuk pengelolaan session user.

Method

- + **BillboardMaintenanceUI()**
Merupakan konstruktor class, tanpa atribut terdefinisi.

2.2.4.4.4. Class Diagram Package DSSPROMOSI.periklanan.
mediacetak.mediacetakoutdoor.brandig.Branding



Gambar 2.42 Class DSSPROMOSI.periklanan.mediacetak.
mediacetakoutdoor.brandig.Branding

Deskripsi

Class ini merepresentasikan branding di dalam DssPromosi.

Atribut

- - brandingid : string
Merepresentasikan id dari branding
- - mcetakid : string
Merepresentasikan media cetak id.
- - divre : string
Merepresentasikan lokasi divre dari branding.
- - kandatel: string
Merepresentasikan lokasi kandatel dari branding.
- - lokasi : string
Merepresentasikan lokasi dari branding.
- - starget : string
Merepresentasikan segmen target utama dari branding.
- - skedua : string
Merepresentasikan segmen target kedua dari branding.
- - sketiga : string

Merepresentasikan segmen target ketiga dari branding.

- - agensi : string

Merepresentasikan agensi atau penanggung jawab branding.

- - telp : string

Merepresentasikan no. Telp branding

- - iklan : integer

Merepresentasikan tarif iklan di branding tersebut

- - eye1 : string

Merepresentasikan eye catching score dari branding.

- - eye2 : string

Merepresentasikan eye catching score kedua dari branding.

- - eye3 : string

Merepresentasikan eye catching score ketiga dari branding.

Method

- + Branding()

Merupakan konstruktor class, tanpa atribut terdefinisi.

- + Branding(id : String, mcid : String, div : String, kand : String, lks : String, target1 : String, target2 : String, target3 : String, agen : String, notelp : String, Hiklan : Integer, ey1 : String, ey2 : String, ey3 : String)

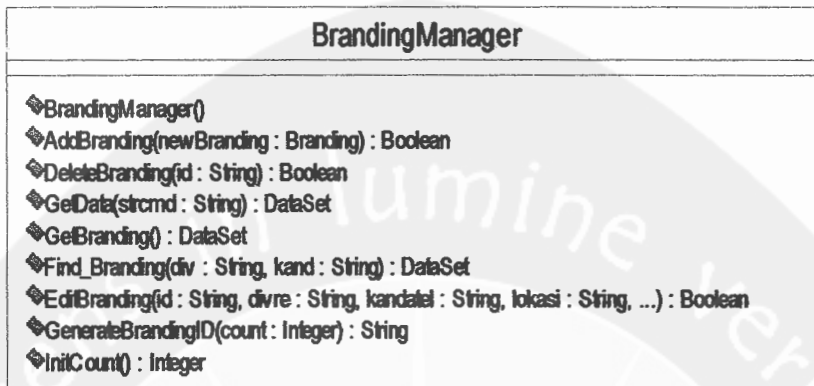
Merupakan konstruktor class, dengan atribut terdefinisi.

Parameters

- id : id branding
- mcid : media cetak indoor id.
- divre : divre branding.
- kandatel : kandatel branding.
- lokasi : lokasi branding.
- target1 : segmen target utama dari branding.
- target2 : segmen target kedua dari branding.
- target3 : segmen target ketiga dari branding.
- agensi : agensi atau penanggung jawab branding.
- telp : no. Telp periklanan branding.
- Hiklan : tarif iklan branding.
- eye1 : eye catching score branding.

- eye2 : eye catching score kedua branding.
- eye3 : eye catching score ketiga branding.

2.2.4.4.5. Class Diagram Package DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.branding.BrandingManager



Gambar 2.43 Class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.branding.BrandingManager

Deskripsi

Class yang berperan sebagai *object control class* untuk aksi yang berhubungan dengan data branding.

Atribut

-

Method

- + BrandingManager()

Konstruktor class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.branding.BrandingManager. Buat instance baru DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.branding.BrandingManager

mediacetakoutdoor.branding.BrandingManager

- + addBranding (newbranding : DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.branding.Branding) : boolean

Digunakan untuk menambahkan data branding baru.

Parameters:

newbranding - DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.branding.Branding yang akan ditambahkan.

mediacetakoutdoor.branding.Branding yang akan ditambahkan.

Returns:

true jika branding berhasil ditambahkan.

- + DeleteBranding (id : string) : Boolean

Menghapus branding dari database.

Parameters:

id - id branding.

Returns:

true jika penghapusan branding berhasil dilakukan.

• + EditBranding(id : String, divre : String, kandatel : String, lokasi : String, sutama : String, skedua : String, sketiga : String, agen : String, telp : String, iklan : String, sc1 : String, sc2 : String, sc3 : String) : Boolean

Parameters:

- id : id branding
- divre : divre branding.
- kandatel : kandatel branding.
- lokasi : lokasi branding.
- sutama : segmen target utama dari branding.
- skedua : segmen target kedua dari branding.
- sketiga : segmen target ketiga dari branding.
- agensi : agensi atau penanggung jawab branding.
- telp : no. Telp branding.
- iklan : tarif iklan branding.
- sc1 : eye catching score branding.
- sc2 : eye catching score kedua branding.
- sc3 : eye catching score ketiga branding.

Returns:

True jika data branding berhasil diubah.

• + GetData(strcmd : string) : DataSet

Ambil data branding pada basis data sesuai field yang diinginkan.

Parameters:

Strcmd - query untuk mengambil data branding pada field tertentu.

Returns:

Set data yang berisi data branding yang tersimpan pada basis data.

• + GetBranding () : DataSet

Ambil semua data branding pada basis data.

Parameters:

-

Returns:

Set data yang berisi semua data branding yang tersimpan pada basis data.

• + GenerateBrandingId (count : integer) : String

Mendapatkan branding id yang telah ter-generate.

Parameters:

Count - id maximum dari branding yang telah di counter yang tersimpan di database.

Returns:

String id.

• + InItCount () : integer

Mendapatkan maximum branding id dan di tambah 1.

Parameters:

-

Returns:

id yang telah di counter.

• + Find_Branding (div : string,kand : string) : DataSet

Mendapatkan class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.branding.Branding untuk data branding tertentu.

Parameters:

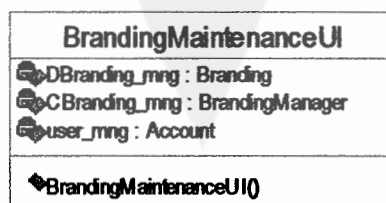
div - divre branding yang akan dicari.

kand - kandatel branding yang akan dicari.

Returns:

Set data branding untuk branding tertentu.

2.2.4.4.6. Class Diagram Package DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.billboard.BrandingMaintenanceUI



Gambar 2.44 Class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.branding.BrandingMaintenanceUI

Deskripsi

Class ini merupakan GUI yang berperan sebagai *boundary class* yang digunakan pengelolaan data branding.

Atribut

- - DBranding_mng : DSSPROMOSI.periklanan.mediacetak.

mediacetakoutdoor.branding.Branding

Merepresentasikan *object entity class* untuk pengelolaan branding saat melakukan perubahan terhadap data branding.

- - CBranding_mng : DSSPROMOSI.periklanan. mediacetak. mediacetakoutdoor.branding.BrandingManager

Merepresentasikan *object control class* untuk pengelolaan data branding saat melakukan perubahan terhadap data branding.

- - user_mng : DSSPROMOSI.account.Account

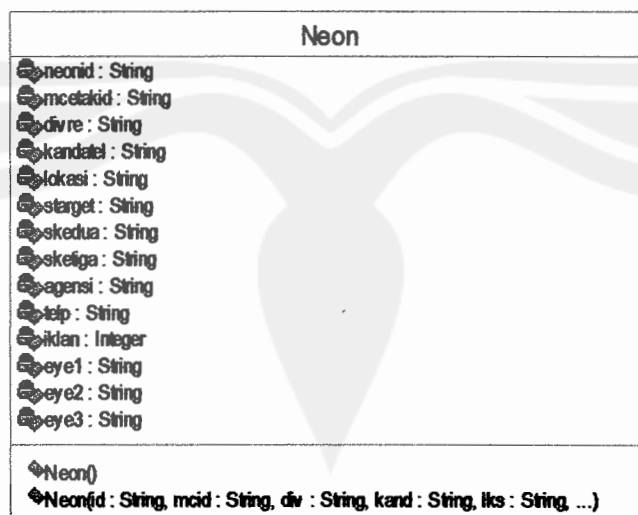
Merepresentasikan *object control class* untuk pengelolaan session user.

Method

- + BrandingMaintenanceUI()

Merupakan konstruktor class, tanpa atribut terdefinisi.

2.2.4.4.7. Class Diagram Package DSSPROMOSI.periklanan. mediacetak.mediacetakoutdoor.neon.Neon



Gambar 2.45 Class DSSPROMOSI.periklanan.mediacetak. mediacetakoutdoor.neon.Neon

Deskripsi

Class ini merepresentasikan neon di dalam DssPromosi.

Atribut

- - neonid : string

Merepresentasikan id dari neon.

- - mcetakid : string

Merepresentasikan media cetak id.

- - divre : string

Merepresentasikan lokasi divre dari neon.

- - kandatel: string

Merepresentasikan lokasi kandatel dari neon.

- - lokasi : string

Merepresentasikan lokasi dari neon.

- - starget : string

Merepresentasikan segmen target utama dari neon.

- - skedua : string

Merepresentasikan segmen target kedua dari neon.

- - sketiga : string

Merepresentasikan segmen target ketiga dari neon.

- - agensi : string

Merepresentasikan agensi atau penanggung jawab neon.

- - telp : string

Merepresentasikan no. Telp neon.

- - iklan : integer

Merepresentasikan tarif iklan di neon tersebut

- - eye1 : string

Merepresentasikan eye catching score dari neon.

- - eye2 : string

Merepresentasikan eye catching score kedua dari neon.

- - eye3 : string

Merepresentasikan eye catching score ketiga dari neon.

Method

- + Neon ()

Merupakan konstruktor class, tanpa atribut terdefinisi.

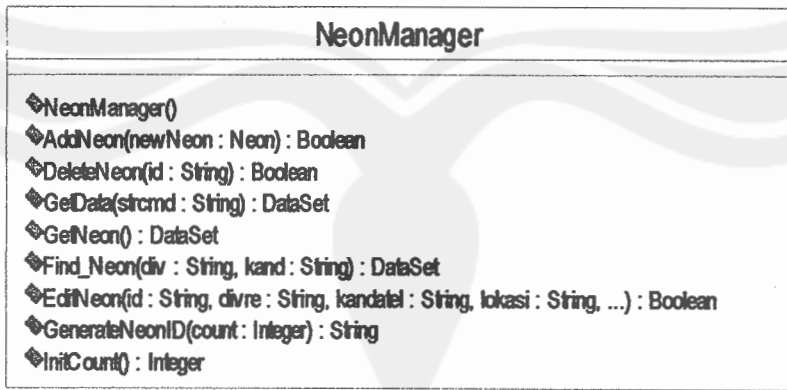
- + Neon(id : String, mcid : String, div : String, kand : String, lks : String, target1 : String, target2 : String, target3 : String, agen : String, notelp : String, Hiklan : Integer, eyl : String, ey2 : String, ey3 : String)

Merupakan konstruktor class, dengan atribut terdefinisi.

Parameters

- id : id neon
- mcid : media cetak indoor id.
- divre : divre neon.
- kandatel : kandatel neon.
- lokasi : lokasi neon.
- target1 : segmen target utama dari neon.
- target2 : segmen target kedua dari neon.
- target3 : segmen target ketiga dari neon.
- agensi : agensi atau penanggung jawab neon.
- telp : no. Telp neon.
- Hiklan : tarif iklan neon.
- eyl : eye catching score neon.
- ey2 : eye catching score kedua neon.
- ey3 : eye catching score ketiga neon.

2.2.4.4.8. Class Diagram Package DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.neon.NeonManager



Gambar 2.46 Class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.neon.Neonmanager

Deskripsi

Class yang berperan sebagai *object control class* untuk aksi yang berhubungan dengan data neon.

Atribut

-

Method

- + NeonManager()

Konstruktor class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.neon.NeonManager. Buat instance baru DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.neon.

NeonManager

- + addNeon (newneon : DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.neon.Neon) : boolean

Digunakan untuk menambahkan data neon baru.

Parameters:

newneon - DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.neon.Neon yang akan ditambahkan.

Returns:

true jika neon berhasil ditambahkan.

- + DeleteNeon (id : string) : Boolean

Menghapus neon dari database.

Parameters:

id - id neon.

Returns:

true jika penghapusan neon berhasil dilakukan.

- + EditNeon(id : String, divre : String, kandatel : String, lokasi : String, sutama : String, skedua : String, sketiga : String, agen : String, telp : String, iklan : String, sc1 : String, sc2 : String, sc3 : String) : Boolean

Parameters:

- id : id neon
- divre : divre neon.
- kandatel : kandatel neon.
- lokasi : lokasi neon.
- sutama : segmen target utama dari neon.

- skedua : segmen target kedua dari neon.
- sketiga : segmen target ketiga dari neon.
- agensi : agensi atau penanggung jawab neon.
- telp : no. Telp neon.
- iklan : tarif iklan neon.
- scl : eye catching score neon.
- sc2 : eye catching score kedua neon.
- sc3 : eye catching score ketiga neon.

Returns:

True jika data neon berhasil diubah.

- + GetData(strcmd : string) : DataSet

Ambil data neon pada basis data sesuai field yang diinginkan.

Parameters:

Strcmd - query untuk mengambil data neon pada field tertentu.

Returns:

Set data yang berisi data neon yang tersimpan pada basis data.

- + GetNeon () : DataSet

Ambil semua data neon pada basis data.

Parameters:

-

Returns:

Set data yang berisi semua data neon yang tersimpan pada basis data.

- + GenerateNeonId (count : integer) : String

Mendapatkan neon id yang telah ter-generate.

Parameters:

Count - id maximum dari neon yang telah di counter yang tersimpan di database.

Returns:

String id.

- + InItCount () : integer

Mendapatkan maximum neon id dan di tambah 1.

Parameters:

-
Returns:

id yang telah di counter.

• + Find_Neon (div : string,kand : string) : DataSet

Mendapatkan class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.neon.Neon untuk data neon tertentu.

Parameters:

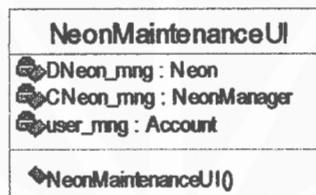
div - divre neon yang akan dicari.

kand - kandatel neon yang akan dicari.

Returns:

Set data neon untuk neon tertentu.

2.2.4.4.9. Class Diagram Package DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.neon.NeonMaintenanceUI



Gambar 2.47 Class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.neon.NeonMaintenanceUI

Deskripsi

Class ini merupakan GUI yang berperan sebagai *boundary class* yang digunakan pengelolaan data neon.

Atribut

- - DNeon_mng : DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.neon.Neon
Merepresentasikan *object entity class* untuk pengelolaan neon saat melakukan perubahan terhadap data neon.
- - CNeon_mng : DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.branding.BrandingManager
Merepresentasikan *object control class* untuk pengelolaan data neon saat melakukan perubahan terhadap data neon.
- - user_mng : DSSPROMOSI.account.Account

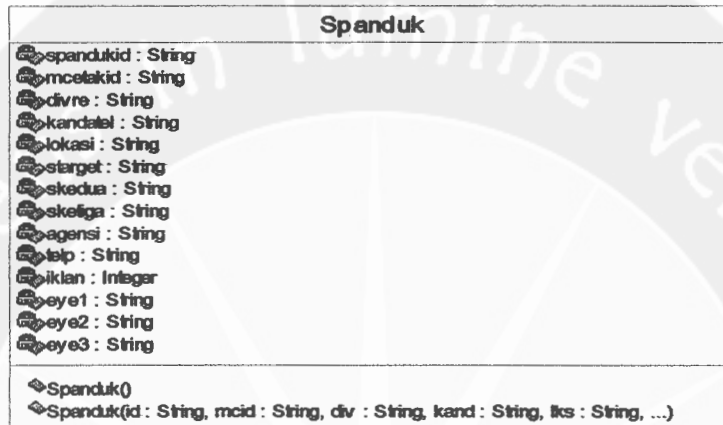
Merepresentasikan *object control class* untuk pengelolaan session user.

Method

- + NeonMaintenanceUI()

Merupakan konstruktor class, tanpa atribut terdefinisi.

2.2.4.4.10. Class Diagram Package DSSPROMOSI.periklanan. mediacetak.mediacetakoutdoor.spanduk.Spanduk



Gambar 2.48 Class DSSPROMOSI.periklanan.mediacetak.
mediacetakoutdoor.spanduk.Spanduk

Deskripsi

Class ini merepresentasikan spanduk di dalam DssPromosi.

Atribut

- - spandukid : string
Merepresentasikan id dari spanduk.
- - mcetakid : string
Merepresentasikan media cetak id.
- - divre : string
Merepresentasikan lokasi divre dari spanduk.
- - kandatel: string
Merepresentasikan lokasi kandatel dari spanduk.
- - lokasi : string
Merepresentasikan lokasi dari spanduk.
- - starget : string
Merepresentasikan segmen target utama dari spanduk.

- - skedua : string

Merepresentasikan segmen target kedua dari spanduk.

- - sketiga : string

Merepresentasikan segmen target ketiga dari spanduk.

- - agensi : string

Merepresentasikan agensi atau penanggung jawab spanduk.

- - telp : string

Merepresentasikan no. Telp spanduk.

- - iklan : integer

Merepresentasikan tarif iklan di spanduk tersebut

- - eye1 : string

Merepresentasikan eye catching score dari spanduk.

- - eye2 : string

Merepresentasikan eye catching score kedua dari spanduk.

- - eye3 : string

Merepresentasikan eye catching score ketiga dari spanduk.

Method

- + Spanduk ()

Merupakan konstruktor class, tanpa atribut terdefinisi.

- + Spanduk(id : String, mcid : String, div : String, kand : String, lks : String, target1 : String, target2 : String, target3 : String, agen : String, notelp : String, Hiklan : Integer, ey1 : String, ey2 : String, ey3 : String)

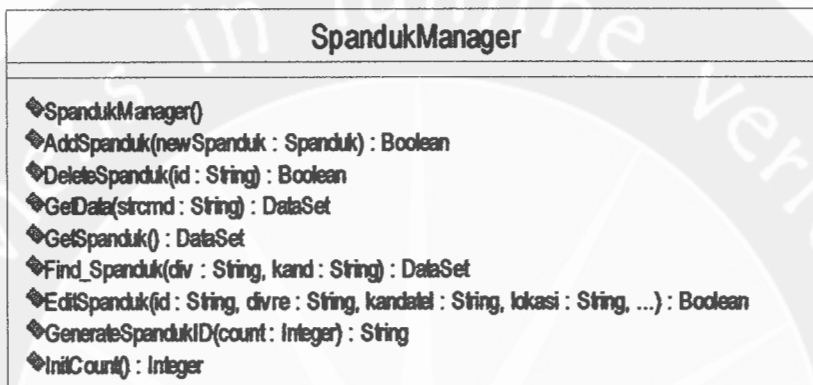
Merupakan konstruktor class, dengan atribut terdefinisi.

Parameters

- id : id spanduk
- mcid : media cetak indoor id.
- divre : divre spanduk.
- kandatel : kandatel spanduk.
- lokasi : lokasi spanduk.
- target1 : segmen target utama dari spanduk.
- target2 : segmen target kedua dari spanduk.
- target3 : segmen target ketiga dari spanduk.

- agensi : agensi atau penanggung jawab spanduk.
- telp : no. Telp spanduk.
- Hiklan : tarif iklan spanduk.
- eye1 : eye catching score spanduk.
- eye2 : eye catching score kedua spanduk.
- eye3 : eye catching score ketiga spanduk.

**2.2.4.4.11. Class Diagram Package DSSPROMOSI.periklanan.
mediacetak.mediacetakoutdoor.spanduk. SpandukManager**



Gambar 2.49 Class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.spanduk.SpandukManager

Deskripsi

Class yang berperan sebagai *object control class* untuk aksi yang berhubungan dengan data spanduk.

Atribut

-

Method

- + NeonManager()

Konstruktor class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.spanduk.SpandukManager. Buat instance baru DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.neon.NeonManager

- + addSpanduk (newspanduk : DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.spanduk.Spanduk) : boolean

Digunakan untuk menambahkan data spanduk baru.

Parameters:

newspanduk - DSSPROMOSI.periklanan.mediacetak.
mediacetakoutdoor.spanduk.Spanduk yang akan ditambahkan.

Returns:

true jika spanduk berhasil ditambahkan.

• + DeleteSpanduk (id : string) : Boolean

Menghapus spanduk dari database.

Parameters:

id - id spanduk.

Returns:

true jika penghapusan spanduk berhasil dilakukan.

• + EditSpanduk(id : String, divre : String, kandatel : String, lokasi : String, sutama : String, skedua : String, sketiga : String, agen : String, telp : String, iklan : String, sc1 : String, sc2 : String, sc3 : String) : Boolean

Parameters:

- id : id spanduk
- divre : divre spanduk.
- kandatel : kandatel spanduk.
- lokasi : lokasi spanduk.
- sutama : segmen target utama dari spanduk.
- skedua : segmen target kedua dari spanduk.
- sketiga : segmen target ketiga dari spanduk.
- agensi : agensi atau penanggung jawab spanduk.
- telp : no. Telp spanduk.
- iklan : tarif iklan spanduk.
- sc1 : eye catching score spanduk.
- sc2 : eye catching score kedua spanduk.
- sc3 : eye catching score ketiga spanduk.

Returns:

True jika data spanduk berhasil diubah.

• + GetData(strcmd : string) : DataSet

Ambil data spanduk pada basis data sesuai field yang diinginkan.

Parameters:

Strcmd - query untuk mengambil data spanduk pada field tertentu.

Returns:

Set data yang berisi data spanduk yang tersimpan pada basis data.

• + GetNeon () : DataSet

Ambil semua data spanduk pada basis data.

Parameters:

-

Returns:

Set data yang berisi semua data spanduk yang tersimpan pada basis data.

• + GenerateSpandukId (count : integer) : String

Mendapatkan spanduk id yang telah ter-generate.

Parameters:

Count - id maximum dari spanduk yang telah di counter yang tersimpan di database.

Returns:

String id.

• + InItCount () : integer

Mendapatkan maximum spanduk id dan di tambah 1.

Parameters:

-

Returns:

id yang telah di counter.

• + Find_Spanduk (div : string, kand : string) : DataSet

Mendapatkan class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.spanduk.Spanduk untuk data spanduk tertentu.

Parameters:

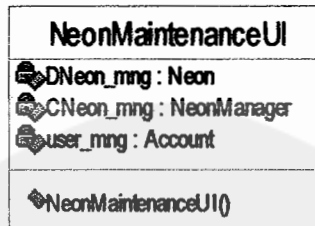
div - divre spanduk yang akan dicari.

kand - kandatel spanduk yang akan dicari.

Returns:

Set data spanduk untuk spanduk tertentu.

2.2.4.4.12. Class Diagram Package DSSPROMOSI.periklanan.
mediacetak.mediacetakoutdoor.spanduk.
SpandukMaintenanceUI



Gambar 2.50 Class DSSPROMOSI.periklanan.mediacetak.
mediacetakoutdoor.spanduk.SpandukMaintenanceUI

Deskripsi

Class ini merupakan GUI yang berperan sebagai *boundary class* yang digunakan pengelolaan data spanduk.

Atribut

- - DSpanduk_mng : DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.spanduk.Spanduk

Merepresentasikan *object entity class* untuk pengelolaan spanduk saat melakukan perubahan terhadap data spanduk.

- - CSpanduk_mng : DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.spanduk.SpandukManager

Merepresentasikan *object control class* untuk pengelolaan data spanduk saat melakukan perubahan terhadap data spanduk.

- - user_mng : DSSPROMOSI.account.Account

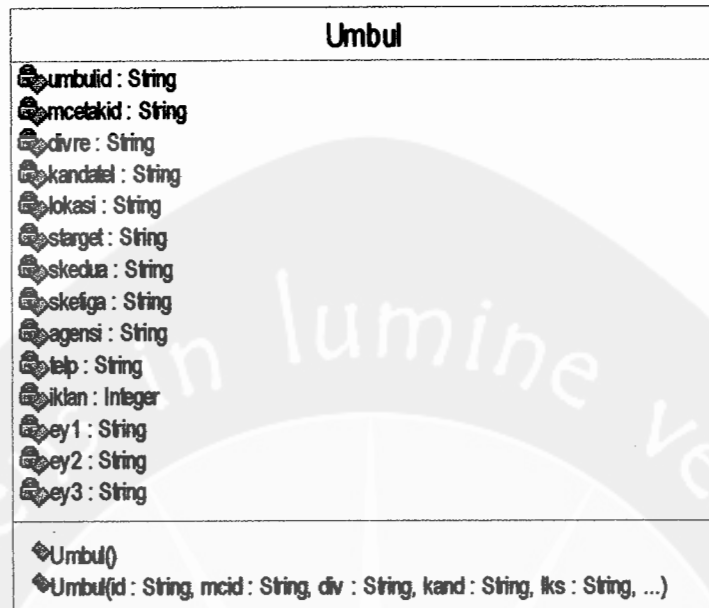
Merepresentasikan *object control class* untuk pengelolaan session user.

Method

- + SpandukMaintenanceUI()

Merupakan konstruktor class, tanpa atribut terdefinisi.

2.2.4.4.13. Class Diagram Package DSSPROMOSI.periklanan.
mediacetak.mediacetakoutdoor.umbul.Umbul



Gambar 2.51 Class DSSPROMOSI.periklanan.mediacetak.
mediacetakoutdoor.umbul.Umbul

Deskripsi

Class ini merepresentasikan umbul-umbul di dalam DssPromosi.

Atribut

- - umbulid : string
Merepresentasikan id dari umbul-umbul.
- - mceakid: string
Merepresentasikan media cetak id.
- - divre : string
Merepresentasikan lokasi divre dari umbul-umbul.
- - kandatel: string
Merepresentasikan lokasi kandatel dari umbul-umbul.
- - lokasi : string
Merepresentasikan lokasi dari umbul-umbul.
- - starget : string
Merepresentasikan segmen target utama dari umbul-umbul.
- - skedua : string

Merepresentasikan segmen target kedua dari umbul-umbul.

- - sketiga : string

Merepresentasikan segmen target ketiga dari umbul-umbul.

- - agensi : string

Merepresentasikan agensi atau penanggung jawab umbul-umbul.

- - telp : string

Merepresentasikan no. Telp umbul-umbul.

- - iklan : integer

Merepresentasikan tarif iklan di umbul-umbul tersebut

- - eyel : string

Merepresentasikan eye catching score dari umbul-umbul.

- - eye2 : string

Merepresentasikan eye catching score kedua dari umbul-umbul.

- - eye3 : string

Merepresentasikan eye catching score ketiga dari umbul-umbul.

Method

- + Umbul ()

Merupakan konstruktor class, tanpa atribut terdefinisi.

- + Umbul(id : String, mcid : String, div : String, kand : String, lks : String, target1 : String, target2 : String, target3 : String, agen : String, notelp : String, Hiklan : Integer, eyl : String, ey2 : String, ey3 : String)

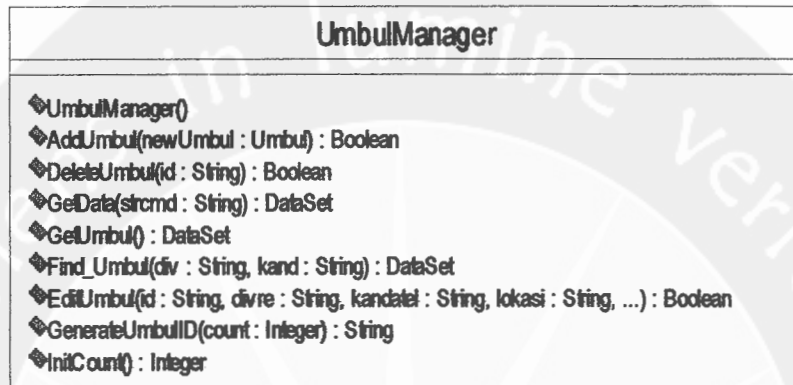
Merupakan konstruktor class, dengan atribut terdefinisi.

Parameters

- id : id umbul-umbul
- mcid : media cetak indoor id.
- divre : divre umbul-umbul.
- kandatel : kandatel umbul-umbul.
- lokasi : lokasi umbul-umbul.
- target1 : segmen target utama dari umbul-umbul.
- target2 : segmen target kedua dari umbul-umbul.
- target3 : segmen target ketiga dari umbul-umbul.
- agensi : agensi atau penanggung jawab umbul-umbul.

- telp : no. Telp umbul-umbul.
- Hiklan : tarif iklan umbul-umbul.
- eye1 : eye catching score umbul-umbul.
- eye2 : eye catching score kedua umbul-umbul.
- eye3 : eye catching score ketiga umbul-umbul.

2.2.4.4.14. Class Diagram Package DSSPROMOSI.periklanan. mediacetak.mediacetakoutdoor.umbul.UmbulManager



Gambar 2.52 Class DSSPROMOSI.periklanan.mediacetak. mediacetakoutdoor. umbul.UmbulManager

Deskripsi

Class yang berperan sebagai *object control class* untuk aksi yang berhubungan dengan data umbul-umbul.

Atribut

-

Method

- + UmbulManager()

Konstruktor class DSSPROMOSI.periklanan.mediacetak. mediacetakoutdoor. umbul.UmbulManager. Buat instance baru DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor. umbul.UmbulManager

- + addUmbul (newumbul : DSSPROMOSI. periklanan.mediacetak.mediacetakoutdoor.umbul.Umbul) : boolean

Digunakan untuk menambahkan data umbul-umbul baru.

Parameters:

newumbul - DSSPROMOSI.periklanan.mediacetak. mediacetakoutdoor.umbul.Umbul yang akan ditambahkan.

Returns:

true jika umbul-umbul berhasil ditambahkan.

• + DeleteUmbul (id : string) : Boolean

Menghapus umbul-umbul dari database.

Parameters:

id - id umbul-umbul.

Returns:

true jika penghapusan umbul-umbul berhasil dilakukan.

• + EditUmbul(id : String, divre : String, kandatel : String, lokasi : String, sutama : String, skedua : String, sketiga : String, agen : String, telp : String, iklan : String, sc1 : String, sc2 : String, sc3 : String) : Boolean

Parameters:

- id : id umbul-umbul
- divre : divre umbul-umbul.
- kandatel : kandatel umbul-umbul.
- lokasi : lokasi umbul-umbul.
- sutama : segmen target utama dari umbul-umbul.
- skedua : segmen target kedua dari umbul-umbul.
- sketiga : segmen target ketiga dari umbul-umbul.
- agensi : agensi atau penanggung jawab umbul-umbul.
- telp : no. Telp umbul-umbul.
- iklan : tarif iklan umbul-umbul.
- sc1 : eye catching score umbul-umbul.
- sc2 : eye catching score kedua umbul-umbul.
- sc3 : eye catching score ketiga umbul-umbul.

Returns:

True jika data umbul-umbul berhasil diubah.

• + GetData(strcmd : string) : DataSet

Ambil data umbul-umbul pada basis data sesuai field yang diinginkan.

Parameters:

Strcmd - query untuk mengambil data umbul-umbul pada field tertentu.

Returns:

Set data yang berisi data umbul-umbul yang tersimpan pada basis data.

- + GetUmbul () : DataSet

Ambil semua data umbul-umbul pada basis data.

Parameters:

-

Returns:

Set data yang berisi semua data umbul-umbul yang tersimpan pada basis data.

- + GenerateUmbulId (count : integer) : String

Mendapatkan umbul-umbul id yang telah ter-generate.

Parameters:

Count - id maximum dari umbul-umbul yang telah di counter yang tersimpan di database.

Returns:

String id.

- + InItCount () : integer

Mendapatkan maximum umbul-umbul id dan di tambah 1.

Parameters:

-

Returns:

id yang telah di counter.

- + Find_Umbul (div : string,kand : string) : DataSet

Mendapatkan class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.umbul.Umbul untuk data umbul-umbul tertentu.

Parameters:

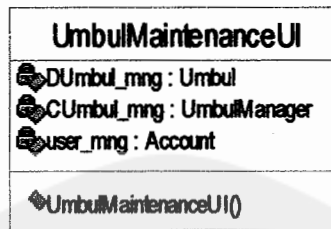
div - divre umbul-umbul yang akan dicari.

kand - kandatel umbul-umbul yang akan dicari.

Returns:

Set data umbul-umbul untuk umbul-umbul tertentu.

2.2.4.4.15. Class Diagram Package DSSPROMOSI.periklanan.
mediacetak.mediacetakoutdoor.umbul. UmbulMaintenanceUI



Gambar 2.53 Class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.
umbul.UmbulMaintenanceUI

Deskripsi

Class ini merupakan GUI yang berperan sebagai *boundary class* yang digunakan pengelolaan data umbul-umbul.

Atribut

- - **DUmbul_mng** : DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.umbul.Umbul

Merepresentasikan *object entity class* untuk pengelolaan umbul-umbul saat melakukan perubahan terhadap data umbul-umbul.

- - **CUmbul_mng** : DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.umbul.UmbulManager

Merepresentasikan *object control class* untuk pengelolaan data umbul-umbul saat melakukan perubahan terhadap data umbul-umbul.

- - **user_mng** : DSSPROMOSI.account.Account

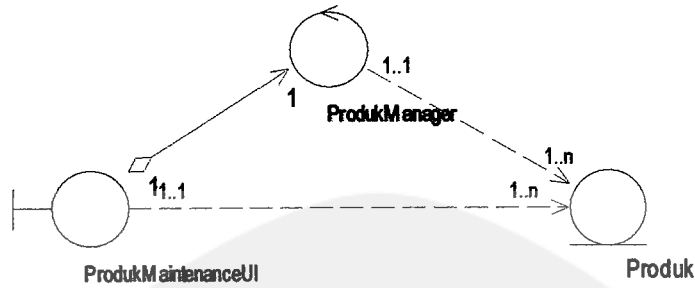
Merepresentasikan *object control class* untuk pengelolaan session user.

Method

- + **UmbulMaintenanceUI()**

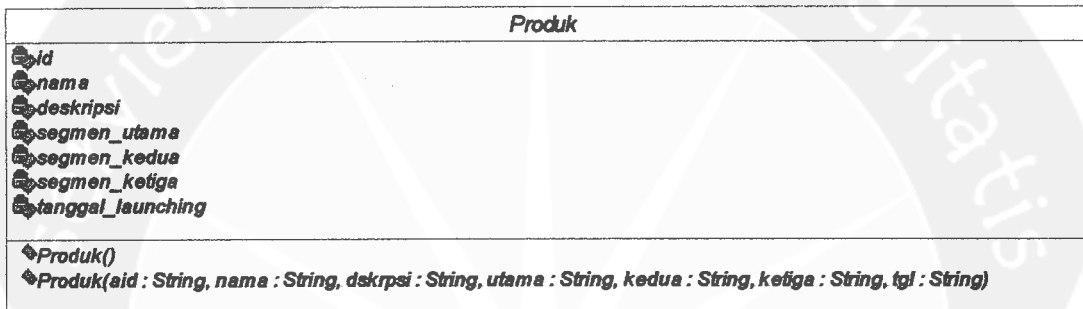
Merupakan konstruktor class, tanpa atribut terdefinisi.

2.2.4.5. Class Diagram Package DSSPROMOSI.produk



Gambar 2.54 Class package DSSPROMOSI.produk

2.2.4.5.1. Class Diagram Package DSSPROMOSI.produk.Produk



Gambar 2.55 Class DSSPROMOSI.produk.Produk

Deskripsi

Class ini merepresentasikan produk di dalam DssPromosi.

Atribut

- - id : string
Merepresentasikan id dari produk.
- - nama : string
Merepresentasikan nama produk.
- - segmen_utama : string
Merepresentasikan segmen target utama dari produk.
- - segmen_kedua : string
Merepresentasikan segmen target kedua dari produk.
- - segmen_ketiga : string
Merepresentasikan segmen target ketiga dari produk.
- - tanggal_launching : string
Merepresentasikan tanggal launching dari produk.

Method

- + Produk ()

Merupakan konstruktor class, tanpa atribut terdefinisi.

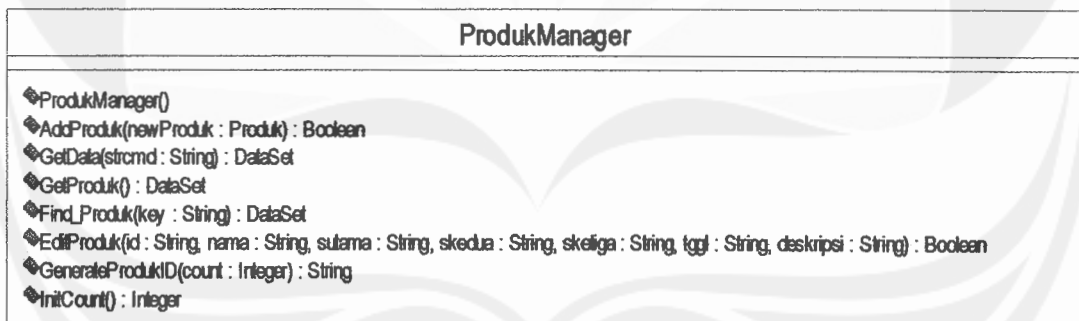
- + Produk(aid : String, nama : String, dskripsi : String, utama : String, kedua : String, ketiga : String, tgl : String)

Merupakan konstruktor class, dengan atribut terdefinisi.

Parameters

- aid : id produk
- nama : nama produk.
- dskripsi : deskripsi produk.
- utama : segmen target utama dari produk.
- kedua : segmen target kedua dari produk.
- ketiga : segmen target ketiga dari produk.
- tgl : tanggal launching produk.

2.2.4.5.2. Class Diagram Package DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.produk.ProdukManager



Gambar 2.56 Class DSSPROMOSI.produk.ProdukManager

Deskripsi

Class yang berperan sebagai *object control class* untuk aksi yang berhubungan dengan data produk.

Atribut

-

Method

- + ProdukManager()

Konstruktor class DSSPROMOSI. produk.ProdukManager. Buat instance baru DSSPROMOSI. produk.ProdukManager

• + addProduk (newproduk : DSSPROMOSI.produk.Produk) :
boolean

Digunakan untuk menambahkan data produk baru.

Parameters:

newproduk - DSSPROMOSI.produk.Produk yang akan ditambahkan.

Returns:

true jika produk berhasil ditambahkan.

• + EditProduk(id : String, nama : String, deskripsi :
String, sutama : String, skedua : String, sketiga : String,
tggl : String) : Boolean

Parameters:

- id : id produk.
- nama : nama produk.
- deskripsi: deskripsi produk.
- sutama : segmen target utama dari produk.
- skedua : segmen target kedua dari produk.
- sketiga : segmen target ketiga dari produk.
- tggl : eye catching score ketiga produk.

Returns:

True jika data produk berhasil diubah.

• + GetData(strcmd : string) : DataSet

Ambil data produk pada basis data sesuai field yang diinginkan.

Parameters:

Strcmd - query untuk mengambil data produk pada field tertentu.

Returns:

Set data yang berisi data produk yang tersimpan pada basis data.

• + GetUmbul () : DataSet

Ambil semua data produk pada basis data.

Parameters:

-

Returns:

Set data yang berisi semua data produk yang tersimpan pada basis data.

• + GenerateProdukId (count : integer) : String

Mendapatkan produk id yang telah ter-generate.

Parameters:

Count - id maximum dari produk yang telah di counter yang tersimpan di database.

Returns:

String id.

• + InItCount () : integer

Mendapatkan maximum produk id dan di tambah 1.

Parameters:

-

Returns:

id yang telah di counter.

• + Find_Produk (key : string) : DataSet

Mendapatkan class DSSPROMOSI.periklanan.mediacetak.mediacetakoutdoor.umbul.Umbul untuk data umbul-umbul tertentu.

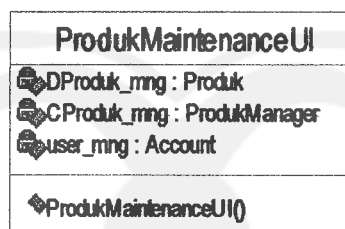
Parameters:

key - nama produk yang akan dicari.

Returns:

Set data produk untuk produk tertentu.

2.2.4.5.3. Class Diagram Package DSSPROMOSI. produk.Produk MaintenanceUI



Gambar 2.57 Class DSSPROMOSI.produk.ProdukMaintenanceUI

Deskripsi

Class ini merupakan GUI yang berperan sebagai *boundary class* yang digunakan pengelolaan data produk.

Atribut

• - DUmbul_mng : DSSPROMOSI.produk.Produk

Merepresentasikan *object entity class* untuk pengelolaan produk saat melakukan perubahan terhadap data produk.

- - CUmbul_mng : DSSPROMOSI.produk.ProdukManager

Merepresentasikan *object control class* untuk pengelolaan data produk saat melakukan perubahan terhadap data produk.

- - user_mng : DSSPROMOSI.account.Account

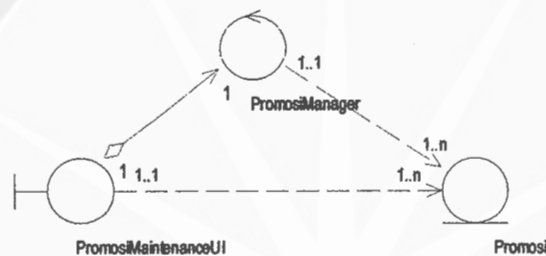
Merepresentasikan *object control class* untuk pengelolaan session user.

Method

- + ProdukMaintenanceUI()

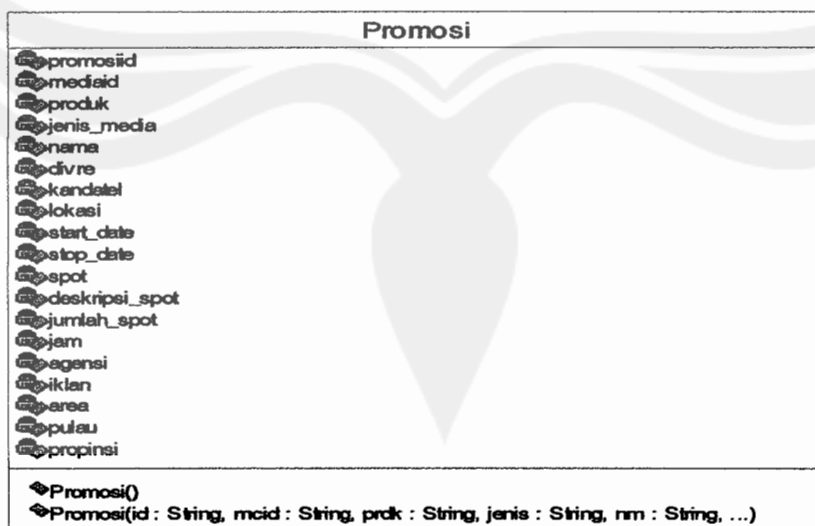
Merupakan konstruktor class, tanpa atribut terdefinisi.

2.2.4.6. Class Diagram Package DSSPROMOSI.Promosi



Gambar 2.58 Class DSSPROMOSI.promosi

2.2.4.6.1. Class Diagram Package DSSPROMOSI.promosi.Promosi



Gambar 2.59 Class DSSPROMOSI.promosi.Promosi

Deskripsi

Class ini merepresentasikan promosi di dalam DssPromosi.

Atribut

- - `promosiid` : string
Merepresentasikan id dari promosi
- - `mediaid` : string
Merepresentasikan media id sebagai tempat promosi.
- - `produk` : string
Merepresentasikan nama produk.
- - `jenis_media` : string
Merepresentasikan jenis media promosi, elektronik atau media cetak.
- - `nama` : string
Merepresentasikan nama media promosi.
- - `divre` : string
Merepresentasikan lokasi divre dari promosi yang dilakukan.
- - `kandatel`: string
Merepresentasikan lokasi kandatel dari promosi yang dilakukan.
- - `lokasi` : string
Merepresentasikan lokasi promosi dilakukan.
- - `Start_date` : date
Merepresentasikan tanggal dimulainya promosi.
- - `Stop_date` : date
Merepresentasikan tanggal berakhirnya promosi.
- - `spot` : string
Merepresentasikan spot iklan dari promosi.
- - `deskripsi_spot`: string
Merepresentasikan deskripsi spot dari iklan produk yang dipromosikan.
- - `jumlah_spot` : string
Merepresentasikan jumlah spot dari iklan promosi.
- - `jam` : string

Merepresentasikan jam siar dari promosi di media elektronik.

- - agensi : string

Merepresentasikan agensi atau penanggung jawab media promosi.

- - iklan : integer

Merepresentasikan tarif iklan promosi tersebut

- - area : string

Merepresentasikan jangkauan promosi, nasional atau internasional.

- - pulau : string

Merepresentasikan jangkauan pulau dari promosi.

- - propinsi : string

Merepresentasikan jangkauan propinsi dari propinsi.

Method

- + Promosi()

Merupakan konstruktor class, tanpa atribut terdefinisi.

- + Promosi(id : String, mcid : String, prdk : String, jenis : String, nm : String, div : String, knctl : String, lks : String, str : String, stp : String, sp : String, ds : String, js : String, jam : String, agen : String, Hiklan : Integer, covarea : String, pulau : String, prop : String)

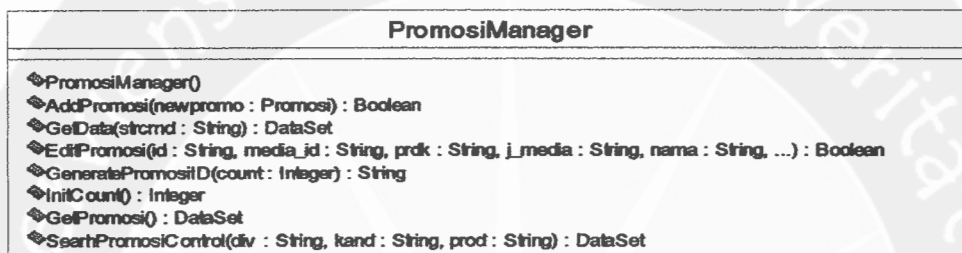
Merupakan konstruktor class, dengan atribut terdefinisi.

Parameters

- id : id promosi
- mcid : media id.
- prdk : produk
- jenis : jenis media
- nama : nama media.
- div : lokasi divre.
- knct : lokasi kandatel.
- lks : lokasi promosi.
- str : tanggal mulai promosi.
- stp : tanggal berakhirnya promosi.

- sp : spot iklan.
- ds : deskripsi spot iklan.
- js : jumlah spot iklan.
- jam : jam siar iklan.
- agen : agensi media iklan.
- Hiklan : tarif iklan.
- covarea : jangkauan promosi.
- pulau : jangkauan pulau promosi.
- propinsi: jangkauan dari promosi.

2.2.4.6.2. Class Diagram Package DSSPROMOSI.promosi. PromosiManager



Gambar 2.60 Class DSSPROMOSI.promosi.PromosiManager

Deskripsi

Class yang berperan sebagai *object control class* untuk aksi yang berhubungan dengan data promosi.

Atribut

-

Method

- + PromosiManager()

Konstruktor class DSSPROMOSI.promosi.PromosiManager. Buat instance baru DSSPROMOSI.promosi.PromosiManager.

- + addPromosi(newpromo : DSSPROMOSI.promosi.Promosi) : boolean

Digunakan untuk menambahkan data promosi baru.

Parameters:

newpromo - DSSPROMOSI.promosi.Promosi. yang akan ditambahkan.

Returns:

true jika promosi berhasil ditambahkan.

• + EditPromosi(id : String, media_id : String, prdk : String, j_media : String, nama : String, divre : String, kand : String, lks : String, strt : String, stp : String, spot : String, des_spot : String, jum_spot : String, jam : String, agen : String, tarif : String, area : String, pulau : String, prop : String) : Boolean

Parameters:

- id : id promosi
- media_id: media id.
- prdk : produk
- j_media : jenis media
- nama : nama media.
- divre : lokasi divre.
- kand : lokasi kandatel.
- lks : lokasi promosi.
- strt : tanggal mulai promosi.
- stp : tanggal berakhirnya promosi.
- spot : spot iklan.
- des_spot: deskripsi spot iklan.
- jum_spot: jumlah spot iklan.
- jam : jam siar iklan.
- agen : agensi media iklan.
- tarif : tarif iklan.
- area : jangkauan promosi.
- pulau : jangkauan pulau promosi.
- prop : jangkauan dari promosi.

Returns:

True jika data promosi berhasil diubah.

• + GetData(strcmd : string) : DataSet

Ambil data promosi pada basis data sesuai field yang diinginkan.

Parameters:

Strcmd - query untuk mengambil data promosi pada field tertentu.

Returns:

Set data yang berisi data promosi yang tersimpan pada basis data.

- + GetPromosi() : DataSet

Ambil semua data promosi pada basis data.

Parameters:

-

Returns:

Set data yang berisi semua data promosi yang tersimpan pada basis data.

- + GeneratePromosiId (count : integer) : String

Mendapatkan promosi id yang telah ter-generate.

Parameters:

Count - id maximum dari promosi yang telah di counter yang tersimpan di database.

Returns:

String id.

- + InItCount () : integer

Mendapatkan maximum promosi id dan di tambah 1.

Parameters:

-

Returns:

id yang telah di counter.

- + SearchPromosi (div : string, kand : string, prod : String) : DataSet

Mendapatkan class DSSPROMOSI.promosi.Promosi untuk data promosi tertentu.

Parameters:

div - divre promosi yang akan dicari.

kand - kandatel promosi yang akan dicari.

Prod - produk yang dipromosikan.

Returns:

Set data promosi untuk promosi tertentu.

**2.2.4.6.3. Class Diagram Package DSSPROMOSI.promotion.
PromosiMaintenanceUI**



Gambar 2.61 Class DSSPROMOSI.promosi.PromosiMaintenanceUI

Deskripsi

Class ini merupakan GUI yang berperan sebagai *boundary class* yang digunakan pengelolaan data promosi.

Atribut

- - DPromosi_mng : DSSPROMOSI.promosi.Promosi
Merepresentasikan *object entity class* untuk pengelolaan promosi saat melakukan perubahan terhadap data promosi.
- - CPromosi_mng : DSSPROMOSI.promosi.PromosiManager
Merepresentasikan *object control class* untuk pengelolaan data promosi saat melakukan perubahan terhadap data promosi.
- - user_mng : DSSPROMOSI.account.Account
Merepresentasikan *object control class* untuk pengelolaan session user.

Method

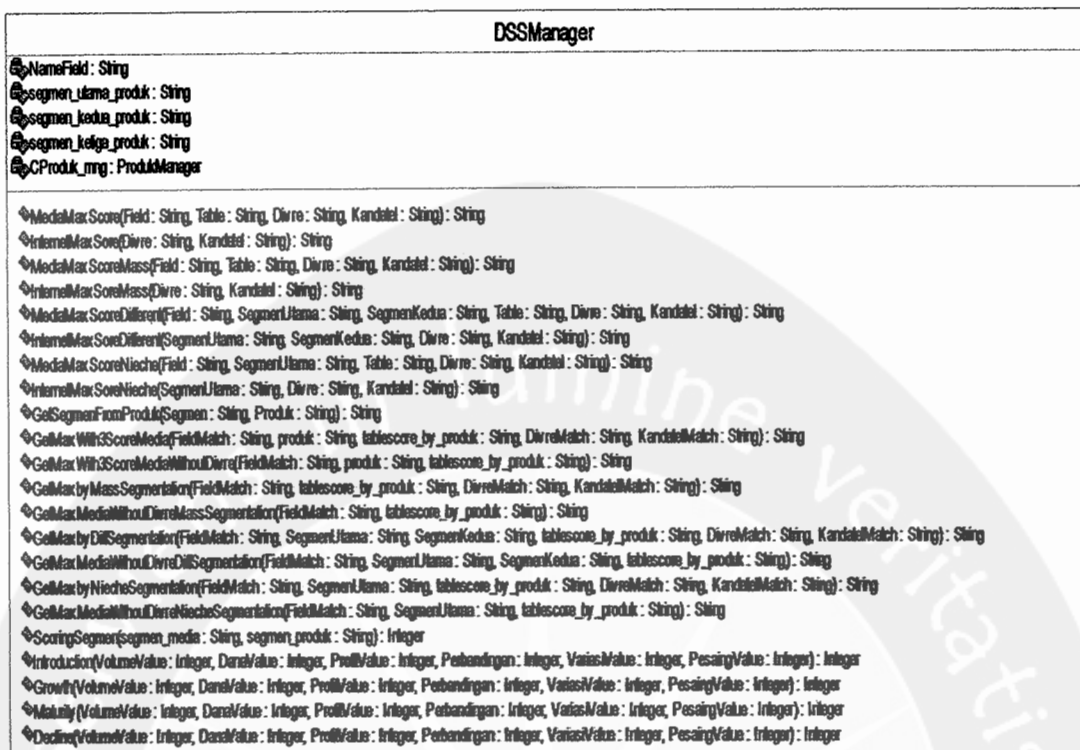
- + PromosiMaintenanceUI()
Merupakan konstruktor class, tanpa atribut terdefinisi.

2.2.4.7. Class Diagram Package DSSPROMOSI.DSS



Gambar 2.62 Class Package DSS

2.2.4.7.1. Class Diagram Package DSSPROMOSI.DSS.DSSManager



Gambar 2.63 Class DSSPROMOSI.DSS.DSSManager

Deskripsi

Class yang berperan sebagai *object control class* untuk aksi yang berhubungan dengan sistem pendukung keputusan promosi.

Atribut

- - NameField : string

Merepresentasikan nama media yang mempunyai score yang tinggi dan segmen yang sesuai dengan produk yang akan dipromosikan.

- - segmen_utama_produk : string

Merepresentasikan segmen utama dari produk.

- - segmen_kedua_produk : string

Merepresentasikan segmen kedua dari produk.

- - segmen_ketiga_produk : string

Merepresentasikan segmen ketiga dari produk.

- - CProduk_mng : ProdukManager

Merepresentasikan *object control class* untuk kelas produk.

Method

- + DSSManager()

Konstruktor class DSSPROMOSI.DSS.DSSPromosiManager. tanpa atribut terdefinisi.

- + GetSegmenFromProduk(segmen : String, produk : String) : String

Mendapatkan segmen produk, baik untuk segmen utama, segmen kedua atau segmen ketiga sesuai dengan produk yang telah terdefinisi.

Parameters:

- segmen : segmen produk
- produk : nama produk

Returns:

Segmen produk.

- + GetMaxWith3ScoreMedia(FieldMatch : String, produk : String, tablescore_by_produk : String, DivreMatch : String, KandatelMatch : String) : String

Mendapatkan media periklanan yang terbaik, sesuai dengan divre dan kandatel tempat dilaksanakannya promosi serta sesuai dengan segmen dari produk yang akan ditawarkan.

Parameters:

- FieldMatch - media periklanan
- Produk - produk yang akan dipromosikan
- Tablescore_by_produk - table media periklanan
- Divre_match - lokasi divre
- Kandatel_match - lokasi kandatel

Returns:

Media periklanan yang terbaik.

- + GetMaxWith3ScoreMediaWithoutDivre(FieldMatch : String, produk : String, tablescore_by_produk : String) : String

Mendapatkan media periklanan yang terbaik, sesuai dengan segmen dari produk yang akan ditawarkan.

Parameters:

- FieldMatch - media periklanan
- Produk - produk yang akan dipromosikan

Tablescore_by_produk - table media periklanan

Returns:

Media periklanan yang terbaik.

• + GetMaxbyMassSegmentation (FieldMatch : String, tablescore_by_produk : String, DivreMatch : String, KandatelMatch : String) : String

Mendapatkan media periklanan yang terbaik, sesuai dengan divre dan kandatel tempat dilaksanakannya promosi serta sesuai dengan seluruh segmen dari produk yang akan ditawarkan.

Parameters:

FieldMatch - media periklanan

Tablescore_by_produk - table media periklanan

Divre_match - lokasi divre

Kandatel_match - lokasi kandatel

Returns:

Media periklanan yang terbaik.

• + GetMaxMediaWithoutDivreMassSegmentation (FieldMatch : String, tablescore_by_produk : String) : String

Mendapatkan media periklanan yang terbaik, sesuai dengan seluruh segmen dari produk yang akan ditawarkan.

Parameters:

FieldMatch - media periklanan

Tablescore_by_produk - table media periklanan

Returns:

Media periklanan yang terbaik.

• GetMaxbyDiffSegmentation (FieldMatch : String, SegmenUtama : String, SegmenKedua : String, tablescore_by_produk : String, DivreMatch : String, KandatelMatch : String) : String

Mendapatkan media periklanan yang terbaik, sesuai dengan divre dan kandatel tempat dilaksanakannya promosi serta sesuai dengan segmen utama dan segmen kedua target promosi dari produk yang akan ditawarkan.

Parameters:

FieldMatch - media periklanan
SegmenUtama- segmen utama target promosi
SegmenKedua- segmen kedua target promosi
Tablescore_by_produk - table media periklanan
Divre_match - lokasi divre
Kandatel_match - lokasi kandatel

Returns:

Media periklanan yang terbaik.

• + GetMaxMediaWithoutDivreDiffSegmentation (FieldMatch : String, SegmenUtama : String, SegmenKedua : String, tablescore_by_produk : String) : String

Mendapatkan media periklanan yang terbaik, sesuai dengan segmen utama dan segmen kedua target promosi dari produk yang akan ditawarkan.

Parameters:

FieldMatch - media periklanan
SegmenUtama- segmen utama target promosi
SegmenKedua- segmen kedua target promosi
Tablescore_by_produk - table media periklanan

Returns:

Media periklanan yang terbaik.

• GetMaxbyNicheSegmentation (FieldMatch : String, SegmenUtama : String, tablescore_by_produk : String, DivreMatch : String, KandatelMatch : String) : String

Mendapatkan media periklanan yang terbaik, sesuai dengan divre dan kandatel tempat dilaksanakannya promosi serta sesuai dengan segmen utama target promosi dari produk yang akan ditawarkan.

Parameters:

FieldMatch - media periklanan
SegmenUtama- segmen utama target promosi
Tablescore_by_produk - table media periklanan
Divre_match - lokasi divre
Kandatel_match - lokasi kandatel

Returns:

Media periklanan yang terbaik.

• + `GetMaxMediaWithoutDivreNicheSegmentation (FieldMatch : String, SegmenUtama : String, tablescore_by_produk : String) : String`

Mendapatkan media periklanan yang terbaik, sesuai dengan segmen utama target promosi dari produk yang akan ditawarkan.

Parameters:

FieldMatch - media periklanan

SegmenUtama- segmen utama target promosi

Tablescore_by_produk - table media periklanan

Returns:

Media periklanan yang terbaik.

• + `ScoringSegmen(segmen_media : String, segmen_produk : String) : Integer`

Mendapatkan score untuk kesesuaian segmen media dengan segmen produk.

Parameters:

segmen_media- segmen media periklanan.

segmen_produk - segmen produk.

Returns:

score kesesuaian segmen media periklanan dengan segmen produk.

• + `Introduction (VolumeValue : Integer, DanaValue: Integer, ProfitValue: Integer, Perbandingan: Integer, VariasiValue: Integer, PesaingValue: Integer) : Integer`

Mendapatkan tahap *Introduction* siklus hidup produk dari faktor-faktor yang terdefinisi.

Parameters:

VolumeValue - volume penjualan

DanaValue - dana promosi/distribusi produk

ProfitValue - keuntungan saat ini dari penjualan produk

Perbandingan- perbandingan jumlah pengeluaran dengan jumlah penjualan

VariasiValue- Difersifikasi produk

PesaingValue- Intensitas Pesaing Bisnis

Returns:

Score stage perkenalan (Introduction).

• Growth (VolumeValue : Integer, DanaValue: Integer, ProfitValue: Integer, Perbandingan: Integer, VariasiValue: Integer, PesaingValue: Integer) : Integer

Mendapatkan tahap *Growth* siklus hidup produk dari faktor-faktor yang terdefinisi.

Parameters:

VolumeValue - volume penjualan

DanaValue - dana promosi/distribusi produk

ProfitValue - keuntungan saat ini dari penjualan produk

Perbandingan- perbandingan jumlah pengeluaran dengan jumlah penjualan

VariasiValue- Difersifikasi produk

PesaingValue- Intensitas Pesaing Bisnis

Returns:

Score stage pertumbuhan (Growth).

• Maturity (VolumeValue : Integer, DanaValue: Integer, ProfitValue: Integer, Perbandingan: Integer, VariasiValue: Integer, PesaingValue: Integer) : Integer

Mendapatkan tahap *Maturity* siklus hidup produk dari faktor-faktor yang terdefinisi.

Parameters:

VolumeValue - volume penjualan

DanaValue - dana promosi/distribusi produk

ProfitValue - keuntungan saat ini dari penjualan produk

Perbandingan- perbandingan jumlah pengeluaran dengan jumlah penjualan

VariasiValue- Difersifikasi produk

PesaingValue- Intensitas Pesaing Bisnis

Returns:

Score stage kematangan (Maturity).

- Decline (VolumeValue : Integer, DanaValue: Integer, ProfitValue: Integer, Perbandingan: Integer, VariasiValue: Integer, PesaingValue: Integer) : Integer

Mendapatkan tahap *Decline* siklus hidup produk dari faktor-faktor yang terdefinisi.

Parameters:

VolumeValue - volume penjualan

DanaValue - dana promosi/distribusi produk

ProfitValue - keuntungan saat ini dari penjualan produk

Perbandingan- perbandingan jumlah pengeluaran dengan jumlah penjualan

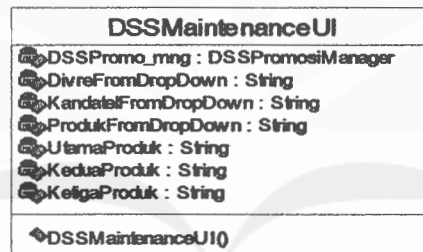
VariasiValue- Difersifikasi produk

PesaingValue- Intensitas Pesaing Bisnis

Returns:

Score stage kemunduran (*Decline*).

2.2.4.7.2. Class Diagram Package DSSPROMOSI.DSS.DSSMaintenanceUI



Gambar 2.64 Class DSSPROMOSI.DSS.DSSMaintenanceUI

Deskripsi

Class ini merupakan GUI yang berperan sebagai *boundary class* dari sistem pendukung keputusan.

Atribut

- - DSSPromo_mng : DSSPROMOSI.DSS.DSSManager
Merepresentasikan *object control class* untuk sistem pendukung keputusan.
- - DivreFromDropDown : String
Merepresentasikan lokasi divre.
- - KandatelFromDropDown : String
Merepresentasikan lokasi kandatel.

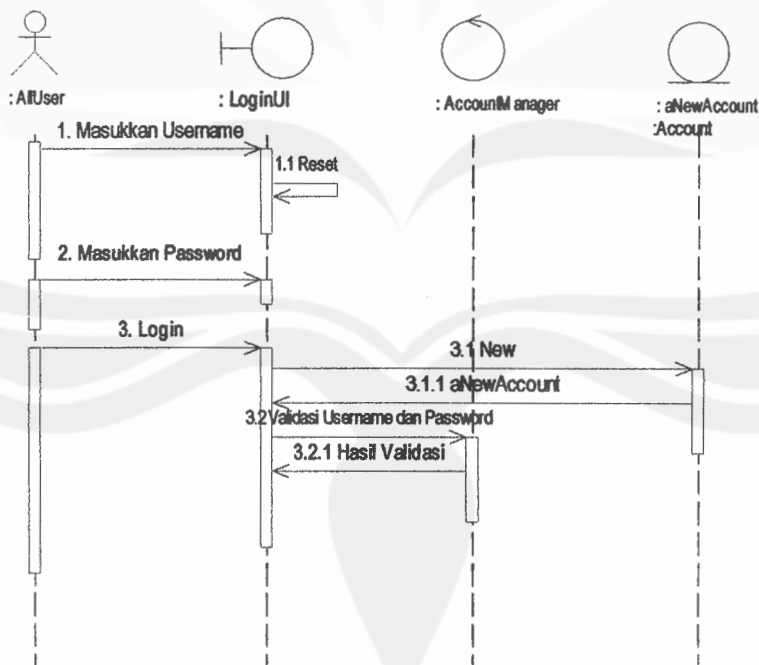
- - ProdukFromDropDown : String
Merepresentasikan produk yang akan dipromosikan.
- - UtamaProduk : String
Merepresentasikan segmen utama produk yang akan dipromosikan.
- - KeduaProduk : String
Merepresentasikan segmen kedua produk yang akan dipromosikan.
- - KetigaProduk : String
Merepresentasikan segmen ketiga produk yang akan dipromosikan.

Method

- + DSSMaintenanceUI ()
Merupakan konstruktor class, tanpa atribut terdefinisi.

2.3. Realisasi Use Case

2.3.1. Use Case : Login



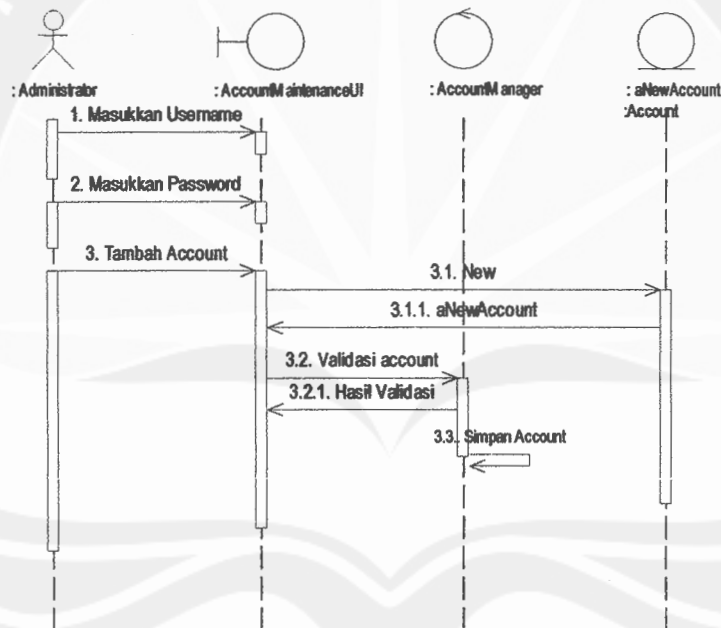
Gambar 2.65 Design Sequence Diagram : Use Case Login

Flow of events :

1. User menampilkan antarmuka untuk login yaitu *boundary class* LoginUI, lalu memasukkan user name.
2. User akan memasukkan password untuk login.
3. LoginUI akan melakukan *create instance user*.
4. Instance user yang baru akan dikembalikan ke LoginUI.
5. instance user akan dikembalikan ke *control class* AccountManager untuk dilakukan validasi username dan password user.
6. Control class UserManager mengembalikan hasil validasi ke *boundary class* LoginUI.

2.3.2. Use Case : Pengelolaan Data Operator

2.3.2.1. Penambahan Data Operator



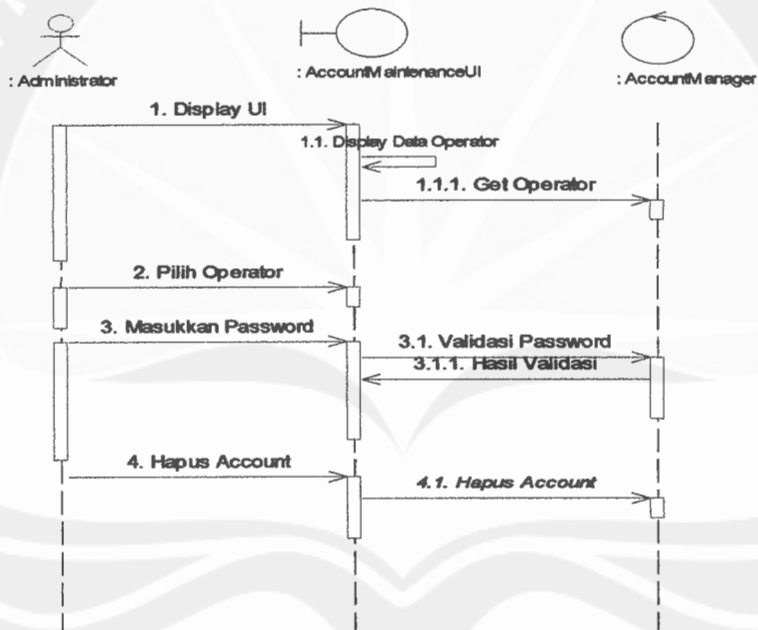
Gambar 2.66 Design Sequence Diagram : Use Case Pengelolaan Data Operator - Penambahan data Operator

Flow of events :

1. System akan menampilkan antarmuka untuk penambahan data operator, yaitu *boundary class* AccountMaintenanceUI dan mengeset ulang (reset) tampilannya dengan mengosongkan semua

- TextBox, administrator memasukkan username dari data operator yang akan diinputkan.
2. Administrator memasukkan password dari data operator yang akan diinputkan.
 3. AccountMaintenanceUI akan membentuk instance dari user.
 4. Instance aNewAccount yang terbentuk akan dikirimkan kembali ke PasienMaintenanceUI.
 5. Instance aNewAccount yang telah terbentuk akan dikembalikan ke AccountManager untuk dilakukan validasi account.
 6. Hasil validasi akan dikembalikan ke AccountMaintenanceUI.
 7. aNewAccount yang terbentuk akan disimpan di database.

2.3.2.2. Penghapusan Data Operator



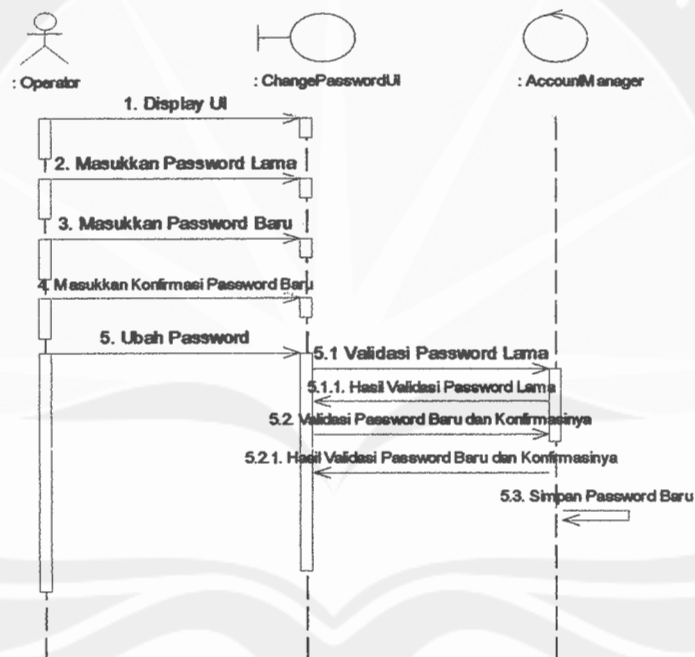
Gambar 2.67 Design Sequence Diagram : Use Case Pengelolaan Data Operator - Penghapusan Data Operator

Flow of events :

1. System akan menampilkan antarmuka untuk penghapusan data operator, yaitu boundary class AccountMaintenanceUI.
2. AccountMaintenanceUI akan menampilkan data operator yang tersimpan di basis data.

3. AccountMaintenanceUI akan memanggil fungsi GetOperator untuk mendapatkan semua data operator.
4. Administrator akan memilih operator yang akan dihapus dari basis data.
5. Administrator akan memasukkan password admin.
6. Password yang masukkan akan divalidasi pada AccountManager.
7. Hasil validasi password administrator akan dikembalikan ke AccountMaintenanceUI dan admin dapat melakukan penghapusan data operator.
8. Data operator dihapus dari basis data.

2.3.3. Use Case : Ubah Password



Gambar 2.68 Design Sequence Diagram : Use Case Ubah Password

Flow of events :

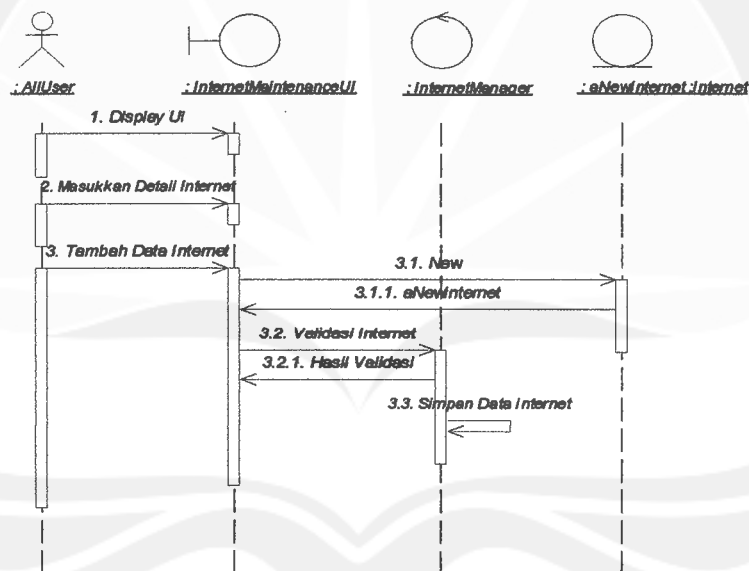
1. System akan menampilkan antarmuka untuk perubahan password, yaitu boundary class ChangePasswordeUI.
2. User memasukkan password lama yang akan diganti.
3. User memasukkan password baru.
4. User memasukkan konfirmasi password baru.

5. System akan melakukan validasi dari password lama user ke object control class AccountManager.
6. hasil validasi dari password lama akan dikembalikan ke ChangePasswordUI.
7. password baru dan konfirmasinya akan dikirimkan ke object control class AccountManager untuk dilakukan validasi, password baru dan konfirmasinya ini harus sesuai.
8. password baru dan konfirmasinya yang telah divalidasi akan dikembalikan ke ChangePasswordUI.
9. AccountManager akan menyimpan password baru user ke basis data.

2.3.4. Use Case : Pengelolaan Data Periklanan

2.3.4.1. Pengelolaan Data Internet

2.3.4.1.1. Penambahan Data Internet



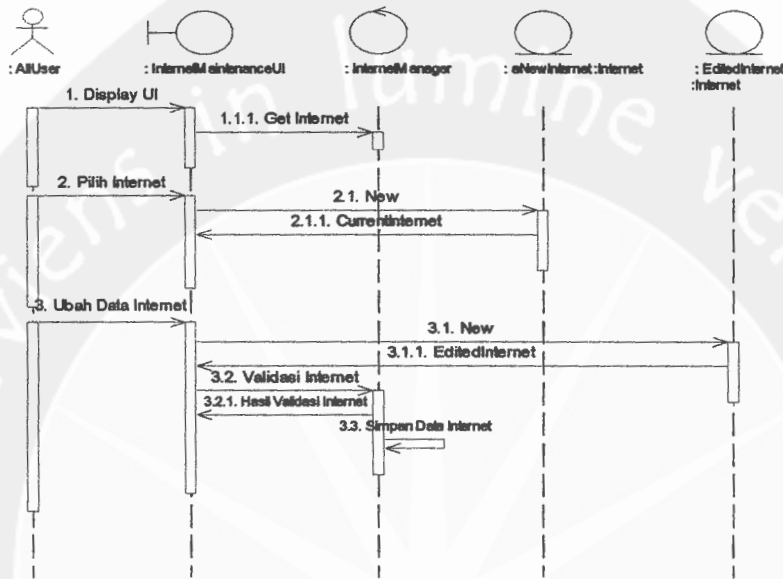
Gambar 2.69 Design Sequence Diagram : Use Case Penambahan Data Internet

Flow of events :

1. System akan menampilkan InternetMaintenanceUI sebagai object boundary class.
2. Aktor memasukkan detail data internet.
3. InternetMaintenanceUI akan membentuk instance dari data internet melalui object entity class Internet.
4. aNewInternet yang merupakan instance yang terbentuk akan dikembalikan ke InternetMaintenanceUI.

5. Instance yang terbentuk akan divalidasi melalui InternetManager.
6. Hasil validasi dari instance yang terbentuk akan dikembalikan ke InternetMaintenanceUI.
7. Data internet yang valid akan disimpan ke basis data melalui object control class InternetManager.

2.3.4.1.2. Perubahan Data Internet



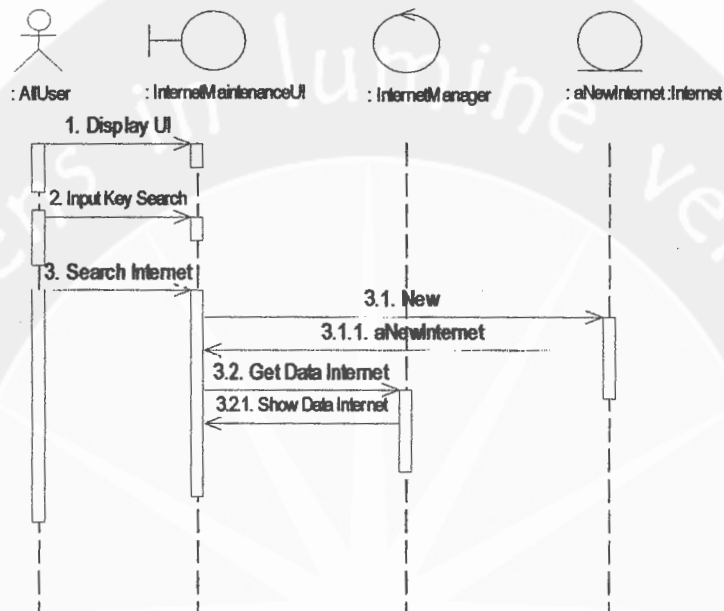
Gambar 2.70 Design Sequence Diagram : Use Case Perubahan Data Internet

Flow of events :

1. System akan menampilkan InternetMaintenanceUI sebagai object boundary class.
2. Sistem akan memanggil fungsi GetInternet() untuk mendapatkan semua data internet melalui object control class InternetManager.
3. Aktor memilih data internet yang akan diubah.
4. InternetMaintenanceUI akan memicu terbentuknya instance dari data internet yang akan diubah.
5. Instance dari data internet yang diambil dari basis data ini akan dikembalikan ke boundary class InternetMaintenanceUI untuk dilakukan perubahan.
6. Aktor akan mengubah data internet yang dipilih.
7. InternetMaintenanceUI akan membentuk instance EditedInternet terhadap data internet yang diubah.

8. Instance EditedInternet yang terbentuk ini akan dilakukan validasi ke control class InternetManager.
9. Hasil validasi instance EditedInternet yang baru ini akan dikembalikan ke boundary class InternetMaintenanceUI.
10. Perubahan terhadap data internet akan disimpan ke basis data.

2.3.4.1.3. Pencarian Data Internet

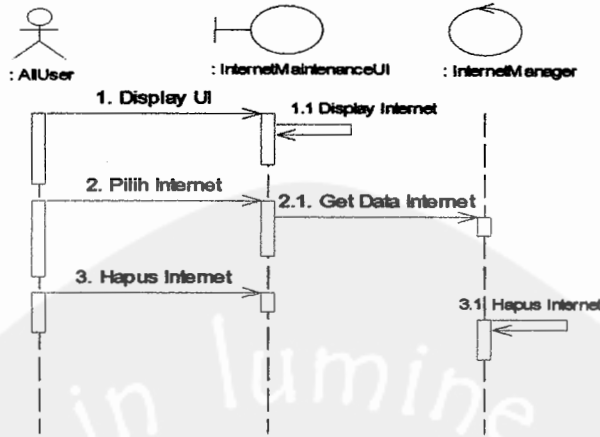


Gambar 2.71 Design Sequence Diagram : Use Case pencarian Data Internet

Flow of events :

1. System akan menampilkan boundary class InternetMaintenanceUI terhadap semua data internet.
2. Aktor akan memasukkan search key untuk situs internet yang akan dicari.
3. InternetMaintenanceUI akan melakukan validasi search key.
4. Boundary class InternetMaintenanceUI akan memanggil fungsi GerInternet untuk mendapatkan data internet yang dicari.
5. Boundary class InternetMaintenanceUI akan membentuk instance baru untuk data internet yang akan dicari.
6. instance yang terbentuk akan dikembalikan ke InternetMaintenanceUI.
7. Sistem akan menampilkan data internet yang dicari.

2.3.4.1.4. Penghapusan Data Internet

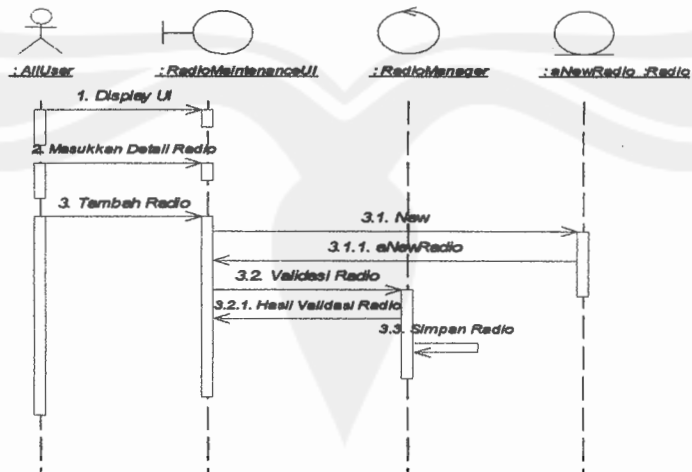


Gambar 2.72 Design Sequence Diagram : Use Case Penghapusan Data Internet
 Flow of events :

1. User memilih dan melakukan penghapusan terhadap data internet.
2. Boundary Class InternetMaintenancUI menampilkan data internet yang ingin dihapus.
3. Boundary Class InternetMaintenancUI meminta konfirmasi penghapusan dan akan dikirim ke control class InternetManager.
4. Control class InternetManager akan melakukan penghapusan dari database.

2.3.4.2. Pengelolaan Data Radio

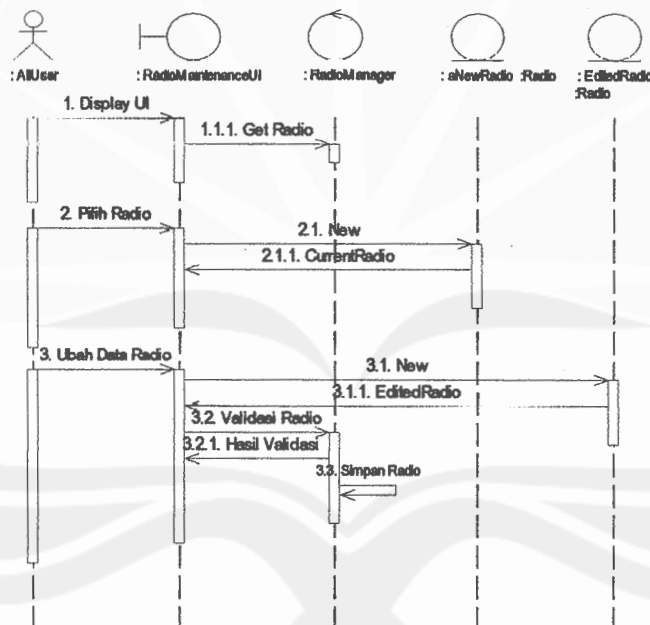
2.3.4.2.1. Penambahan Data Radio



Gambar 2.73 Design Sequence Diagram : Use Case Penambahan Data Radio
 Flow of events :

1. System akan menampilkan RadioMaintenanceUI sebagai object boundary class.
2. Aktor memasukkan detail data Radio.
3. RadioMaintenanceUI akan membentuk instance dari data radio melalui object entity class Radio.
4. aNewRadio yang merupakan instance yang terbentuk akan dikembalikan ke RadioMaintenanceUI.
5. Instance yang terbentuk akan divalidasi melalui RadioManager.
6. Hasil validasi dari instance yang terbentuk akan dikembalikan ke RadioMaintenanceUI.
7. Data radio yang valid akan disimpan ke basis data melalui object control class RadioManager.

2.3.4.2.2. Perubahan Data Radio



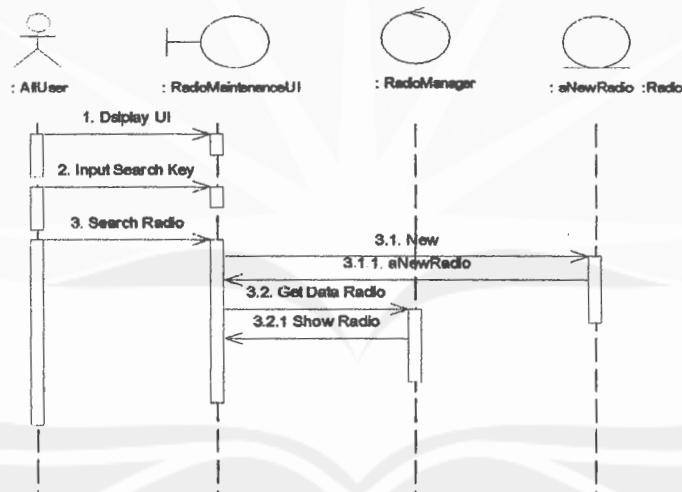
Gambar 2.74 Design Sequence Diagram : Use Case Pengubahan Data Radio

Flow of events :

1. System akan menampilkan RadioMaintenanceUI sebagai object boundary class.
2. Sistem akan memanggil fungsi GetRadio () untuk mendapatkan semua data radio melalui object control class RadioManager.
3. Aktor memilih data radio yang akan diubah.

4. RadioMaintenanceUI akan memicu terbentuknya instance dari data radio yang akan diubah.
5. Instance dari data radio yang diambil dari basis data ini akan dikembalikan ke boundary class RadioMaintenanceUI untuk dilakukan perubahan.
6. Aktor akan mengubah data radio yang dipilih.
7. RadioMaintenanceUI akan membentuk instance EditedRadio terhadap data radio yang diubah.
8. Instance Radio yang terbentuk ini akan dilakukan validasi ke control class RadioManager.
9. Hasil validasi instance EditedRadio yang baru ini akan dikembalikan ke boundary class RadioMaintenanceUI.
10. Perubahan terhadap data radio akan disimpan ke basis data.

2.3.4.2.3. Pencarian Data Radio



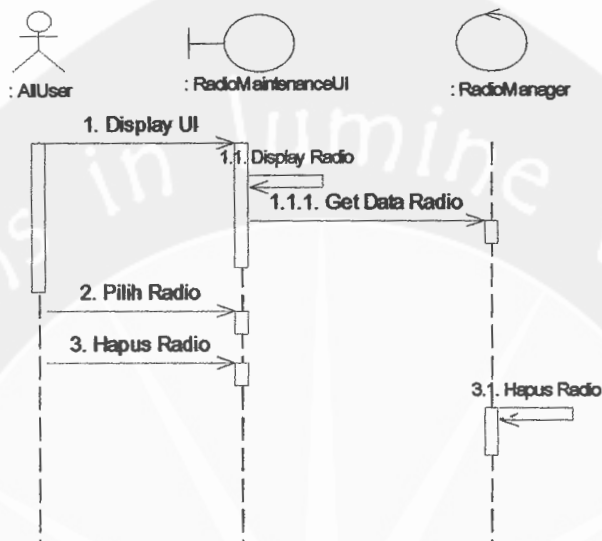
Gambar 2.75 Design Sequence Diagram : Use Case pencarian Data Radio

Flow of events :

1. System akan menampilkan boundary class RadioMaintenanceUI terhadap semua data radio.
2. Aktor akan memasukkan search key untuk radio yang akan dicari.
3. RadioMaintenanceUI akan melakukan validasi search key.
4. Boundary class RadioMaintenanceUI akan memanggil fungsi GerRadio untuk mendapatkan data radio yang dicari.

5. Boundary class RadioMaintenanceUI akan membentuk instance baru untuk data radio yang akan dicari.
6. instance yang terbentuk akan dikembalikan ke RadioMaintenanceUI.
7. Sistem akan menampilkan data radio yang dicari.

2.3.4.2.4. Penghapusan Data Radio



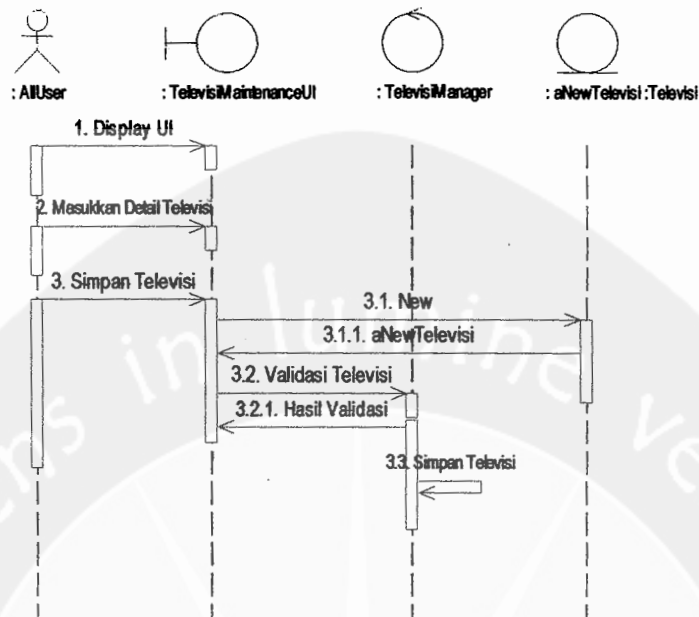
Gambar 2.76 Design Sequence Diagram : Use Case Penghapusan Data Radio

Flow of events :

1. User memilih dan melakukan penghapusan terhadap data radio.
2. Boundary Class RadioMaintenanceUI menampilkan data radio yang ingin dihapus.
3. Boundary Class RadioMaintenanceUI meminta konfirmasi penghapusan dan akan dikirim ke control class RadioManager.
4. Control class RadioManager akan melakukan penghapusan dari database.

2.3.4.3. Pengelolaan Data Televisi

2.3.4.3.1. Penambahan Data Televisi

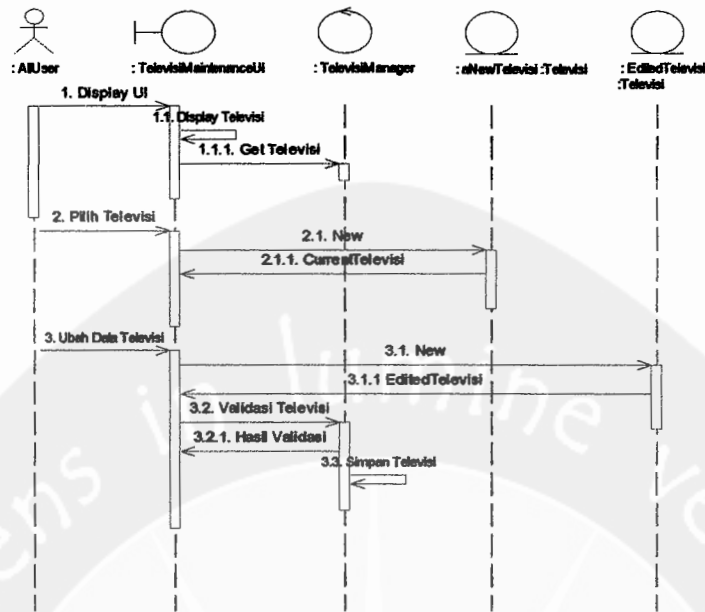


Gambar 2.77 Design Sequence Diagram : Use Case Penambahan Data Televisi

Flow of events :

1. System akan menampilkan TelevisiMaintenanceUI sebagai object boundary class.
2. Aktor memasukkan detail data televisi.
3. TelevisiMaintenanceUI akan membentuk instance dari data televisi melalui object entity class Televisi.
4. aNewTelevisi yang merupakan instance yang terbentuk akan dikembalikan ke TelevisiMaintenanceUI.
5. Instance yang terbentuk akan divalidasi melalui TelevisiManager.
6. Hasil validasi dari instance yang terbentuk akan dikembalikan ke TelevisiMaintenanceUI.
7. Data televisi yang valid akan disimpan ke basis data melalui object control class TelevisiManager.

2.3.4.3.2. Perubahan Data Televisi

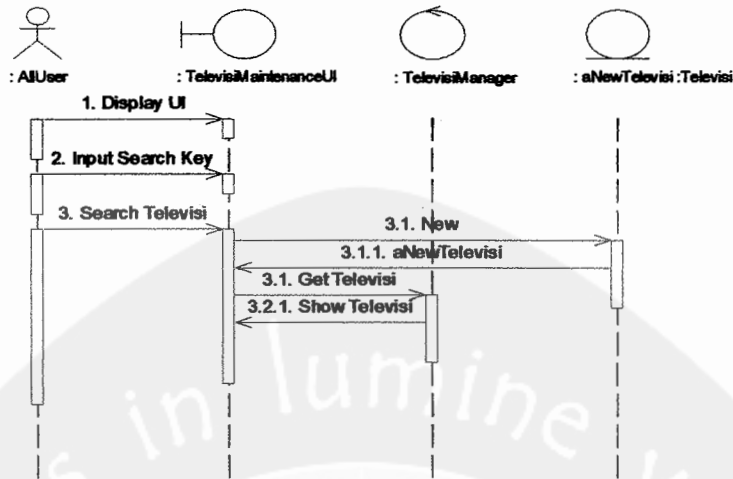


Gambar 2.78 Design Sequence Diagram : Use Case Perubahan Data Televisi

Flow of events :

1. System akan menampilkan TelevisiMaintenanceUI sebagai object boundary class.
2. Sistem akan memanggil fungsi GetTelevisi () untuk mendapatkan semua data televisi malalui object control class TelevisiManager.
3. Aktor memilih data televisi yang akan diubah.
4. TelevisiMaintenanceUI akan memicu terbentuknya instance dari data televisi yang akan diubah.
5. Instance dari data televisi yang diambil dari basis data ini akan dikembalikan ke boundary class TelevisiMaintenanceUI untuk dilakukan perubahan.
6. Aktor akan mengubah data televisi yang dipilih.
7. TelevisiMaintenanceUI akan membentuk instance EditedTelevisi terhadap data televisi yang diubah.
8. Instance yang terbentuk ini akan dilakukan validasi ke control class TelevisiManager.
9. Hasil validasi instance EditedTelevisi yang baru ini akan dikembalikan ke boundary class TelevisiMaintenanceUI.
10. Perubahan terhadap data televisi akan disimpan ke basis data.

2.3.4.3.3. Pencarian Data Televisi

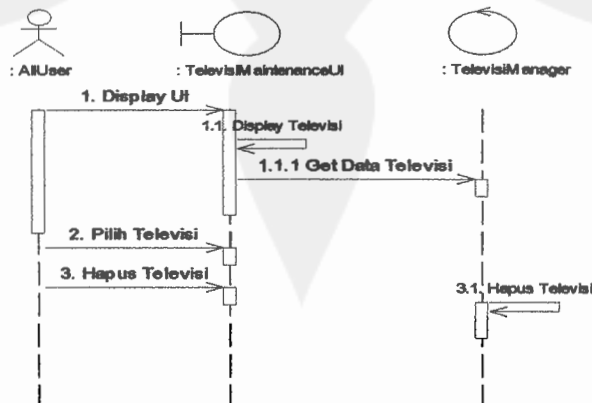


Gambar 2.79 Design Sequence Diagram : Use Case Pencarian Data Televisi

Flow of events :

1. System akan menampilkan boundary class TelevisiMaintenanceUI terhadap semua data televisi.
2. Aktor akan memasukkan search key untuk televisi yang akan dicari.
3. TelevisiMaintenanceUI akan melakukan validasi search key.
4. Boundary class TelevisiMaintenanceUI akan memanggil fungsi GerTelevisi untuk mendapatkan data televisi yang dicari.
5. Boundary class TelevisiMaintenanceUI akan membentuk instance baru untuk data televisi yang akan dicari.
6. instance yang terbentuk akan dikembalikan ke TelevisiMaintenanceUI.
7. Sistem akan menampilkan data televisi yang dicari.

2.3.4.3.4. Penghapusan Data Televisi



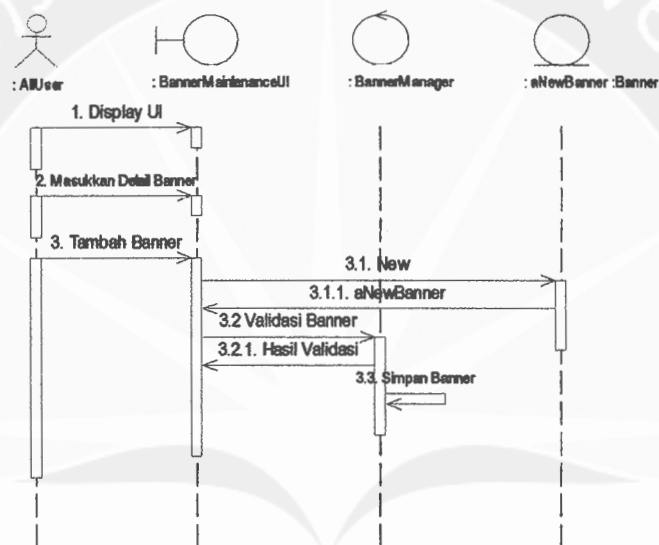
Gambar 2.80 Design Sequence Diagram : Use Case Penghapusan Data Televisi

Flow of events :

1. User memilih dan melakukan penghapusan terhadap data televisi.
2. *Boundary Class* TelevisiMaintenancUI menampilkan data televisi yang ingin dihapus.
3. *Boundary Class* TelevisiMaintenancUI meminta konfirmasi penghapusan dan akan dikirim ke *control class* TelevisiManager.
4. *Control class* TelevisiManager akan melakukan penghapusan dari database.

2.3.4.4. Pengelolaan Data Banner

2.3.4.4.1. Penambahan Data Banner



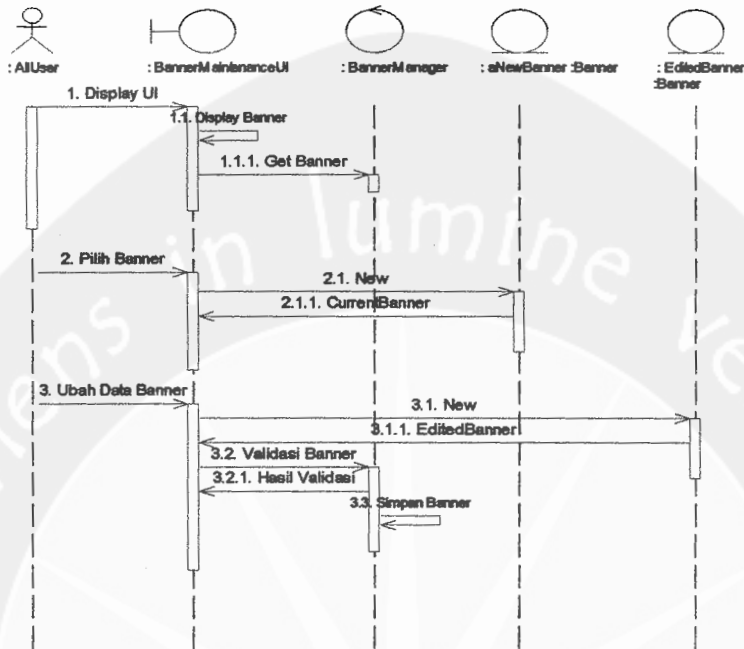
Gambar 2.81 Design Sequence Diagram : Use Case Penambahan Data Banner

Flow of events :

1. System akan menampilkan BannerMaintenanceUI sebagai object boundary class.
2. Aktor memasukkan detail data banner.
3. BannerMaintenanceUI akan membentuk instance dari data banner melalui object entity class Banner.
4. aNewBanner yang merupakan instance yang terbentuk akan dikembalikan ke BannerMaintenanceUI.
5. Instance yang terbentuk akan divalidasi melalui BannerManager.
6. Hasil validasi dari instance yang terbentuk akan dikembalikan ke BannerMaintenanceUI.

7. Data banner yang valid akan disimpan ke basis data melalui object control class BannerManager.

2.3.4.4.2. Perubahan Data Banner



Gambar 2.82 Design Sequence Diagram : Use Case Perubahan Data Banner

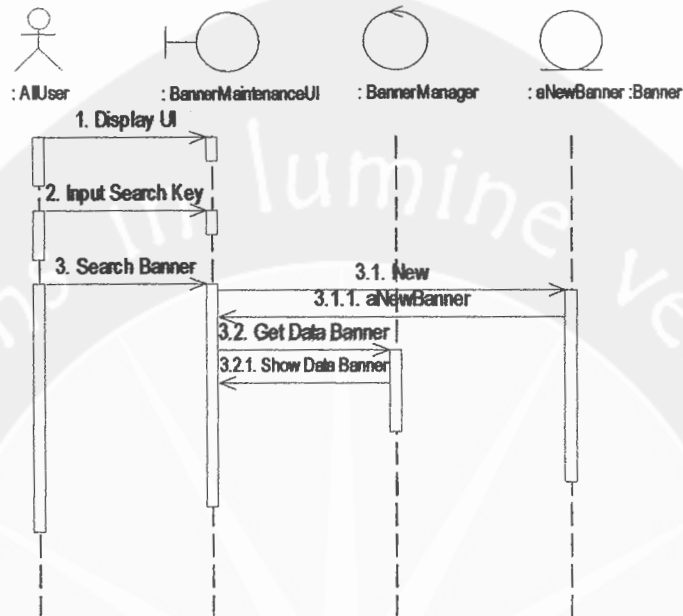
Flow of events :

1. System akan menampilkan BannerMaintenanceUI sebagai object boundary class.
2. Sistem akan memanggil fungsi GetBanner () untuk mendapatkan semua data banner melalui object control class BannerManager.
3. Aktor memilih data banner yang akan diubah.
4. BannerMaintenanceUI akan memicu terbentuknya instance dari data banner yang akan diubah.
5. Instance dari data banner yang diambil dari basis data ini akan dikembalikan ke boundary class BannerMaintenanceUI untuk dilakukan perubahan.
6. Aktor akan mengubah data banner yang dipilih.
7. BannerMaintenanceUI akan membentuk instance EditedBanner terhadap data banner yang diubah.
8. Instance yang terbentuk ini akan dilakukan validasi ke control class BannerManager.

9. Hasil validasi instance EditedBanner yang baru ini akan dikembalikan ke boundary class BannerMaintenanceUI.

10. Perubahan terhadap data banner akan disimpan ke basis data.

2.3.4.4.3. Pencarian Data Banner

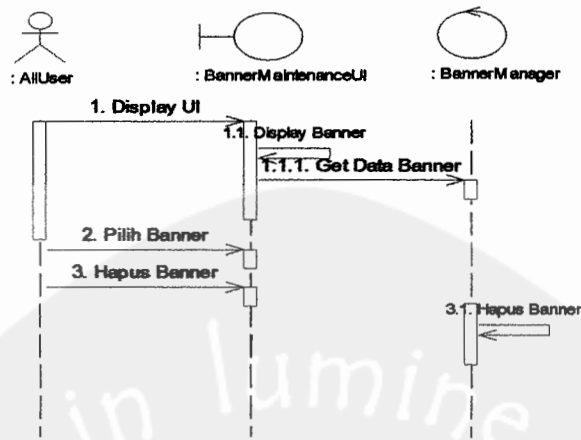


Gambar 2.83 Design Sequence Diagram : Use Case Pencarian Data Banner

Flow of events :

1. System akan menampilkan boundary class BannerMaintenanceUI terhadap semua data banner.
2. Aktor akan memasukkan search key untuk banner yang akan dicari.
3. BannerMaintenanceUI akan melakukan validasi search key.
4. Boundary class BannerMaintenanceUI akan memanggil fungsi GerBanner untuk mendapatkan data banner yang dicari.
5. Boundary class BannerMaintenanceUI akan membentuk instance baru untuk data banner yang akan dicari.
6. instance yang terbentuk akan dikembalikan ke BannerMaintenanceUI.
7. Sistem akan menampilkan data banner yang dicari.

2.3.4.4.4. Penghapusan Data Banner

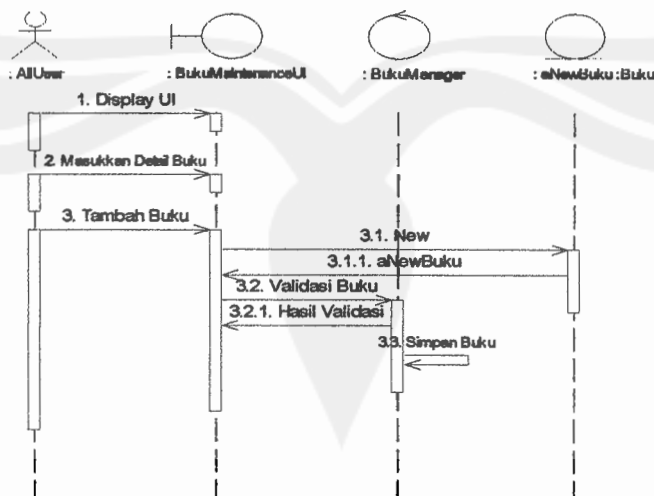


Gambar 2.84 Design Sequence Diagram : Use Case Penghapusan Data Televisi
Flow of events :

1. User memilih dan melakukan penghapusan terhadap data banner.
2. *Boundary Class* BannerMaintenancUI menampilkan data banner yang ingin dihapus.
3. *Boundary Class* BannerMaintenancUI meminta konfirmasi penghapusan dan akan dikirim ke *control class* BannerManager.
4. *Control class* BannerManager akan melakukan penghapusan dari *database*.

2.3.4.5. Pengelolaan Data Buku

2.3.4.5.1. Penambahan Data Buku

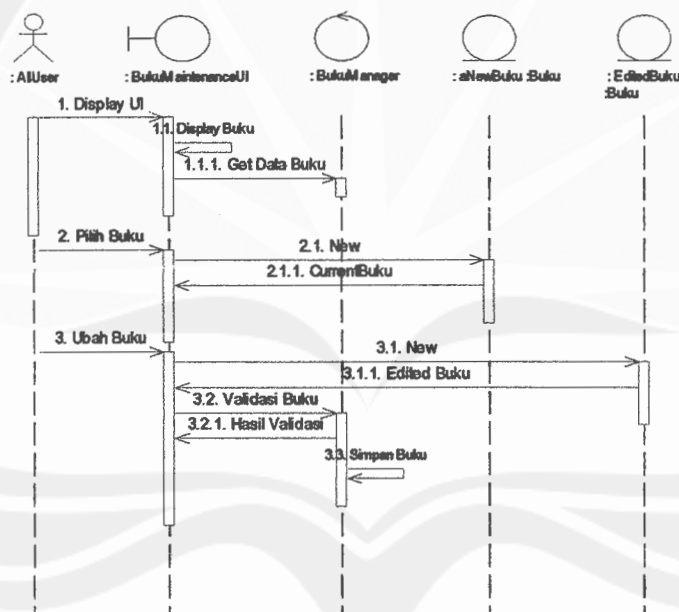


Gambar 2.85 Design Sequence Diagram : Use Case Penambahan Data Buku

Flow of events :

1. System akan menampilkan BukuMaintenanceUI sebagai object boundary class.
2. Aktor memasukkan detail data buku.
3. BukuMaintenanceUI akan membentuk instance dari data buku melalui object entity class Buku.
4. aNewBuku yang merupakan instance yang terbentuk akan dikembalikan ke BukuMaintenanceUI.
5. Instance yang terbentuk akan divalidasi melalui BukuManager.
6. Hasil validasi dari instance yang terbentuk akan dikembalikan ke BukuMaintenanceUI.
7. Data buku yang valid akan disimpan ke basis data melalui object control class BukuManager.

2.3.4.5.2. Perubahan Data Buku



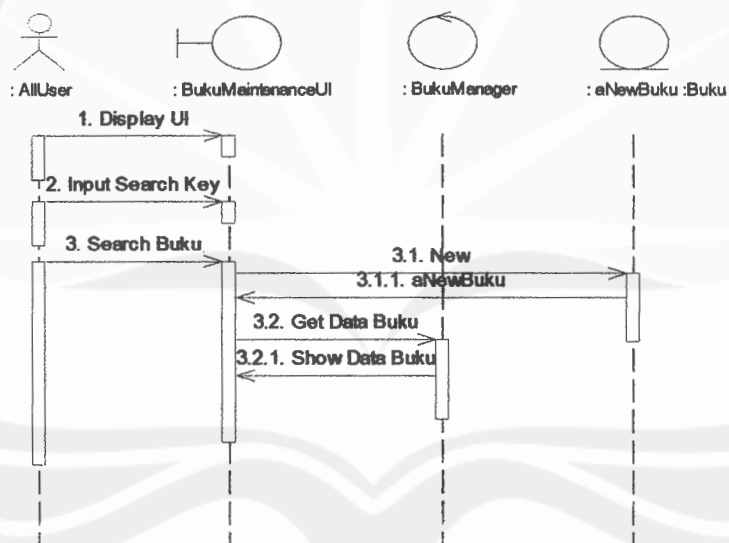
Gambar 2.86 Design Sequence Diagram : Use Case Perubahan Data Buku

Flow of events :

1. System akan menampilkan BukuMaintenanceUI sebagai object boundary class.
2. Sistem akan memanggil fungsi GetBuku () untuk mendapatkan semua data buku melalui object control class BukuManager.
3. Aktor memilih data buku yang akan diubah.

4. BukuMaintenanceUI akan memicu terbentuknya instance dari data buku yang akan diubah.
5. Instance dari data buku yang diambil dari basis data ini akan dikembalikan ke boundary class BukuMaintenanceUI untuk dilakukan perubahan.
6. Aktor akan mengubah data buku yang dipilih.
7. BukuMaintenanceUI akan membentuk instance EditedBuku terhadap data buku yang diubah.
8. Instance yang terbentuk ini akan dilakukan validasi ke control class BukuManager.
9. Hasil validasi instance EditedBuku yang baru ini akan dikembalikan ke boundary class BukuMaintenanceUI.
10. Perubahan terhadap data buku akan disimpan ke basis data.

2.3.4.5.3. Pencarian Data Buku



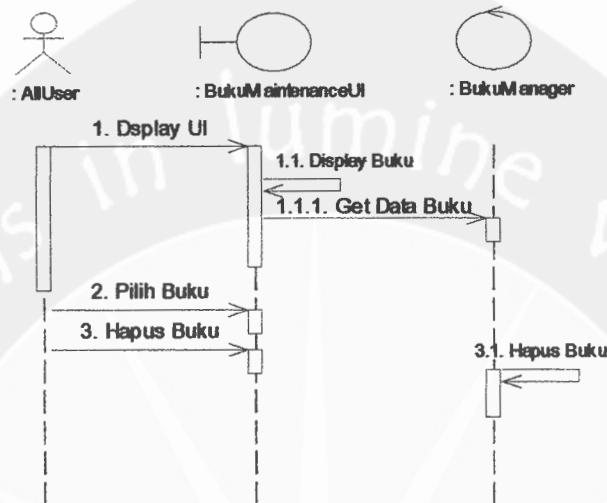
Gambar 2.87 Design Sequence Diagram : Use Case Pencarian Data Buku

Flow of events :

1. System akan menampilkan boundary class BukuMaintenanceUI terhadap semua data buku.
2. Aktor akan memasukkan search key untuk buku yang akan dicari.
3. BukuMaintenanceUI akan melakukan validasi search key.
4. Boundary class BukuMaintenanceUI akan memanggil fungsi GerBuku untuk mendapatkan data buku yang dicari.

5. Boundary class BukuMaintenanceUI akan membentuk instance baru untuk data buku yang akan dicari.
6. instance yang terbentuk akan dikembalikan ke BukuMaintenanceUI.
7. Sistem akan menampilkan data buku yang dicari.

2.3.4.5.4. Penghapusan Data Buku



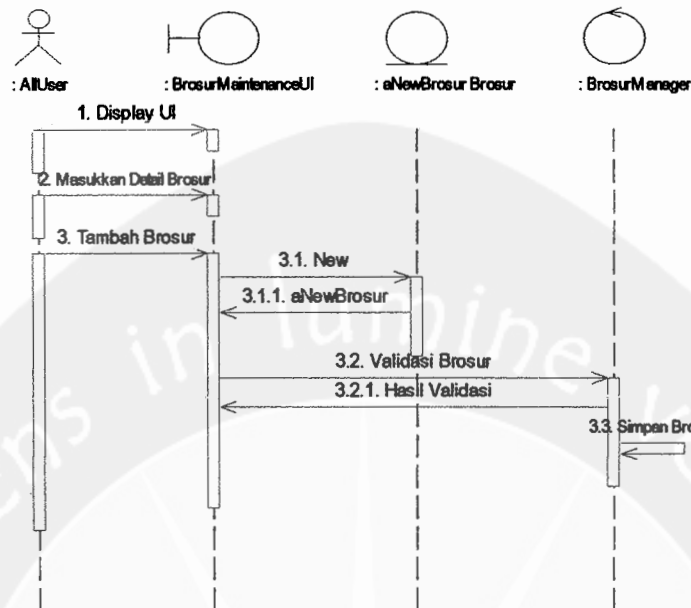
Gambar 2.88 Design Sequence Diagram : Use Case Penghapusan Data Buku

Flow of events :

1. User memilih dan melakukan penghapusan terhadap data buku.
2. Boundary Class BukuMaintenanceUI menampilkan data buku yang ingin dihapus.
3. Boundary Class BukuMaintenanceUI meminta konfirmasi penghapusan dan akan dikirim ke control class BukuManager.
4. Control class BukuManager akan melakukan penghapusan dari database.

2.3.4.6. Pengelolaan Data Brosur

2.3.4.6.1. Penambahan Data Brosur

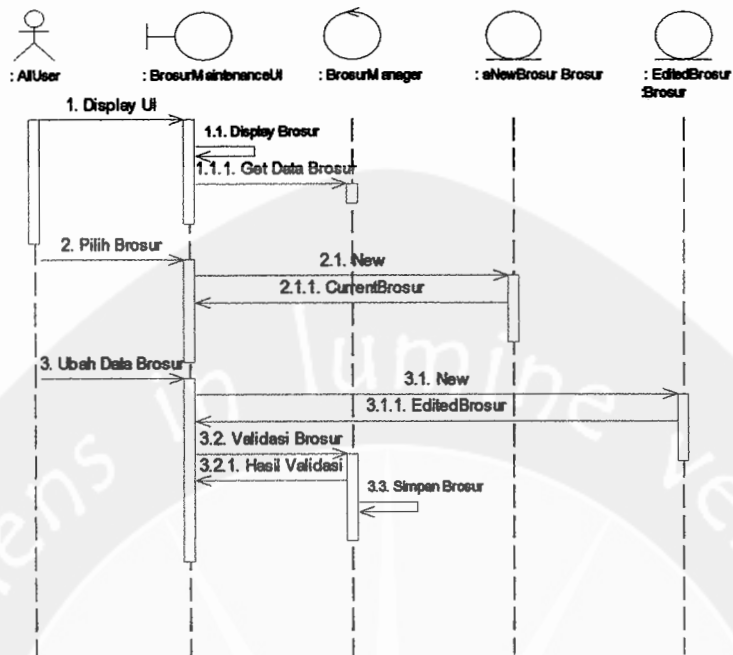


Gambar 2.89 Design Sequence Diagram : Use Case Penambahan Data Brosur

Flow of events :

1. System akan menampilkan BrosurMaintenanceUI sebagai object boundary class.
2. Aktor memasukkan detail data brosur.
3. BrosurMaintenanceUI akan membentuk instance dari data brosur melalui object entity class Brosur.
4. aNewBrosur yang merupakan instance yang terbentuk akan dikembalikan ke BrosurMaintenanceUI.
5. Instance yang terbentuk akan divalidasi melalui BrosurManager.
6. Hasil validasi dari instance yang terbentuk akan dikembalikan ke BrosurMaintenanceUI.
7. Data brosur yang valid akan disimpan ke basis data melalui object control class BrosurManager.

2.3.4.6.2. Perubahan Data Brosur

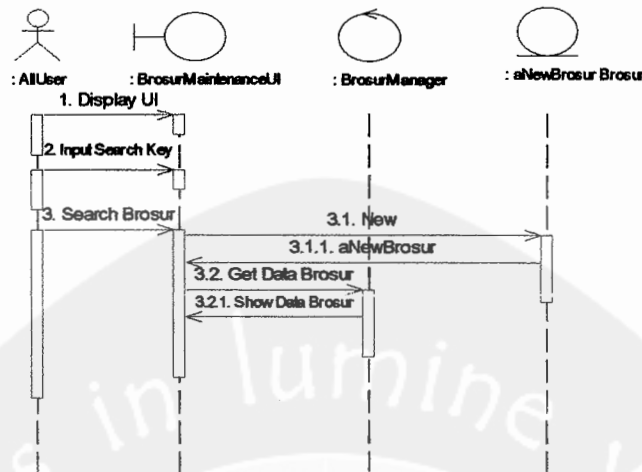


Gambar 2.90 Design Sequence Diagram : Use Case Perubahan Data Brosur

Flow of events :

1. System akan menampilkan BrosurMaintenanceUI sebagai object boundary class.
2. Sistem akan memanggil fungsi GetBrosur () untuk mendapatkan semua data brosur malalui object control class BrosurManager.
3. Aktor memilih data brosur yang akan diubah.
4. BrosurMaintenanceUI akan memicu terbentuknya instance dari data brosur yang akan diubah.
5. Instance dari data brosur yang diambil dari basis data ini akan dikembalikan ke boundary class BrosurMaintenanceUI untuk dilakukan perubahan.
6. Aktor akan mengubah data brosur yang dipilih.
7. BrosurMaintenanceUI akan membentuk instance EditedBrosur terhadap data brosur yang diubah.
8. Instance yang terbentuk ini akan dilakukan validasi ke control class BrosurManager.
9. Hasil validasi instance EditedBrosur yang baru ini akan dikembalikan ke boundary class BrosurMaintenanceUI.
10. Perubahan terhadap data brosur akan disimpan ke basis data.

2.3.4.6.3. Pencarian Data Brosur

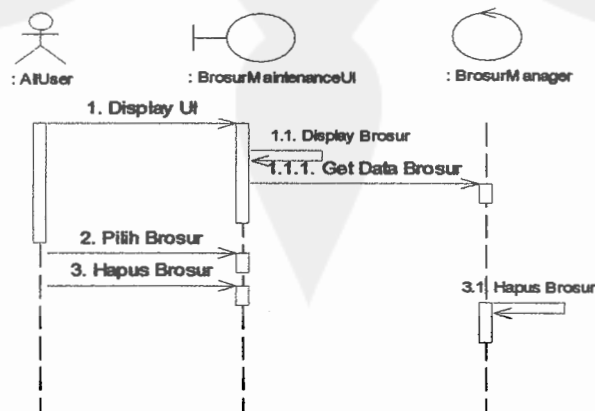


Gambar 2.91 Design Sequence Diagram : Use Case Pencarian Data Brosur

Flow of events :

1. System akan menampilkan boundary class BrosurMaintenanceUI terhadap semua data brosur.
2. Aktor akan memasukkan search key untuk brosur yang akan dicari.
3. BrosurMaintenanceUI akan melakukan validasi search key.
4. Boundary class BrosurMaintenanceUI akan memanggil fungsi GerBrosur untuk mendapatkan data brosur yang dicari.
5. Boundary class BrosurMaintenanceUI akan membentuk instance baru untuk data brosur yang akan dicari.
6. instance yang terbentuk akan dikembalikan ke BrosurMaintenanceUI.
7. Sistem akan menampilkan data brosur yang dicari.

2.3.4.6.4. Penghapusan Data Brosur



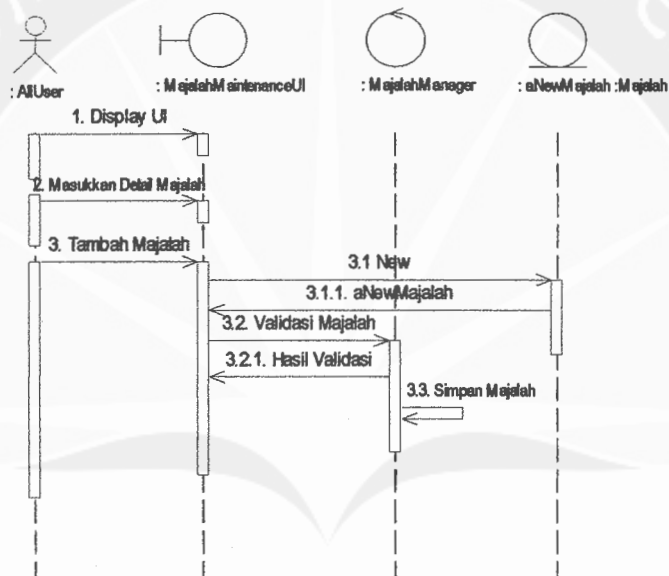
Gambar 2.92 Design Sequence Diagram : Use Case Penghapusan Data Brosur

Flow of events :

1. User memilih dan melakukan penghapusan terhadap data brosur.
2. *Boundary Class* BrosurMaintenancUI menampilkan data brosur yang ingin dihapus.
3. *Boundary Class* BrosurMaintenancUI meminta konfirmasi penghapusan dan akan dikirim ke *control class* BrosurManager.
4. *Control class* BrosurManager akan melakukan penghapusan dari *database*.

2.3.4.7. . Pengelolaan Data Majalah

2.3.4.7.1. Penambahan Data Majalah



Gambar 2.93 Design Sequence Diagram : Use Case Penambahan Data Majalah

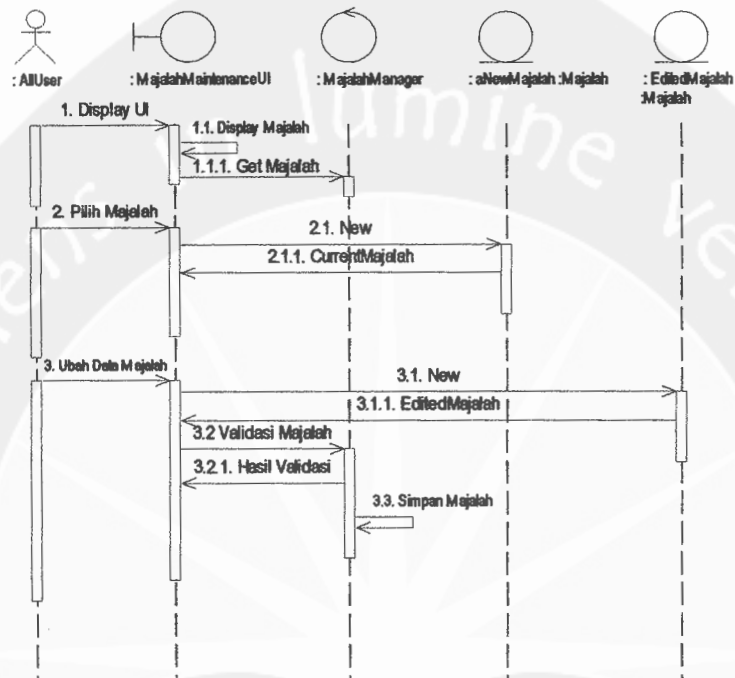
Flow of events :

1. System akan menampilkan *MajalahMaintenanceUI* sebagai object boundary class.
2. Aktor memasukkan detail data majalah.
3. *MajalahMaintenanceUI* akan membentuk instance dari data majalah melalui object entity class *Majalah*.
4. *aNewMajalah* yang merupakan instance yang terbentuk akan dikembalikan ke *MajalahMaintenanceUI*.
5. Instance yang terbentuk akan divalidasi melalui *MajalahManager*.

6. Hasil validasi dari instance yang terbentuk akan dikembalikan ke `MajalahMaintenanceUI`.

7. Data majalah yang valid akan disimpan ke basis data melalui object control class `MajalahManager`.

2.3.4.7.2. Perubahan Data Majalah

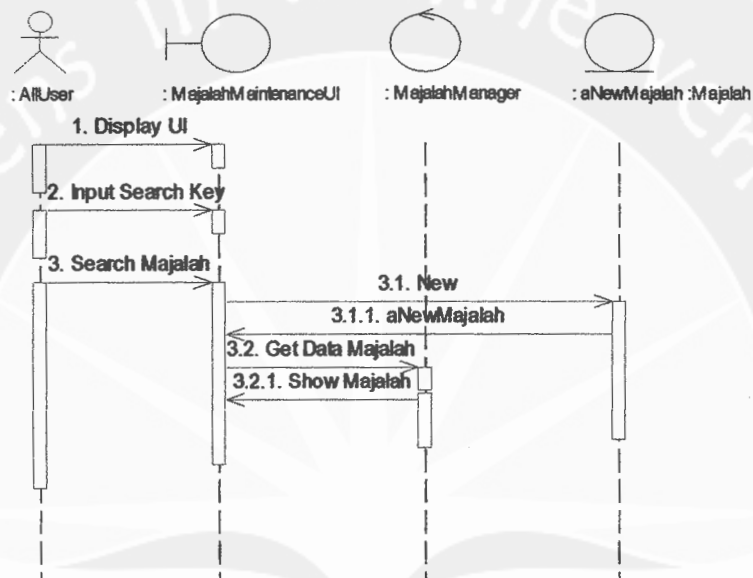


Gambar 2.94 Design Sequence Diagram : Use Case Perubahan Data Majalah
Flow of events :

1. System akan menampilkan `MajalahMaintenanceUI` sebagai object boundary class.
2. Sistem akan memanggil fungsi `GetMajalah ()` untuk mendapatkan semua data majalah melalui object control class `MajalahManager`.
3. Aktor memilih data majalah yang akan diubah.
4. `MajalahMaintenanceUI` akan memicu terbentuknya instance dari data majalah yang akan diubah.
5. Instance dari data majalah yang diambil dari basis data ini akan dikembalikan ke boundary class `MajalahMaintenanceUI` untuk dilakukan perubahan.
6. Aktor akan mengubah data majalah yang dipilih.

7. MajalahMaintenanceUI akan membentuk instance EditedMajalah terhadap data majalah yang diubah.
8. Instance yang terbentuk ini akan dilakukan validasi ke control class MajalahManager.
9. Hasil validasi instance EditedMajalah yang baru ini akan dikembalikan ke boundary class MajalahMaintenanceUI.
10. Perubahan terhadap data majalah akan disimpan ke basis data.

2.3.4.7.3. Pencarian Data Majalah

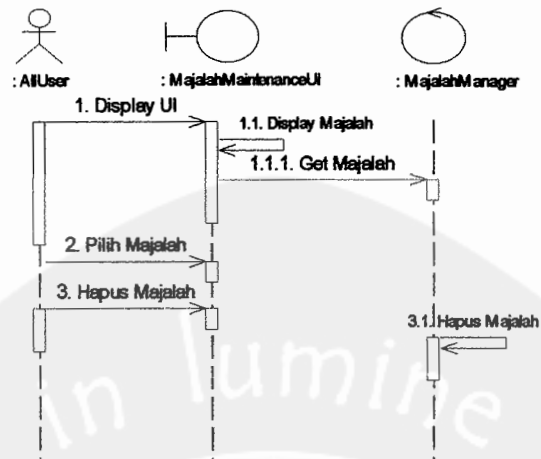


Gambar 2.95 Design Sequence Diagram : Use Case Pencarian Data Majalah

Flow of events :

1. System akan menampilkan boundary class MajalahMaintenanceUI terhadap semua data majalah.
2. Aktor akan memasukkan search key untuk majalah yang akan dicari.
3. MajalahMaintenanceUI akan melakukan validasi search key.
4. Boundary class MajalahMaintenanceUI akan memanggil fungsi GerMajalah untuk mendapatkan data majalah yang dicari.
5. Boundary class MajalahMaintenanceUI akan membentuk instance baru untuk data majalah yang akan dicari.
6. instance yang terbentuk akan dikembalikan ke MajalahMaintenanceUI.
7. Sistem akan menampilkan data majalah yang dicari.

2.3.4.7.4. Penghapusan Data Majalah



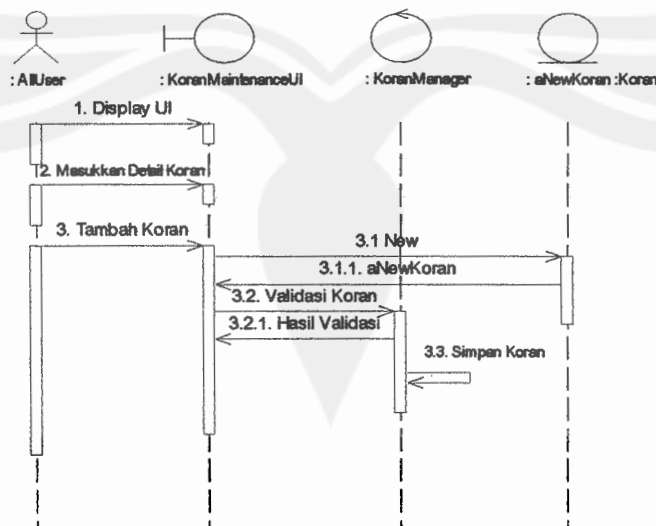
Gambar 2.96 Design Sequence Diagram : Use Case Penghapusan Data Majalah

Flow of events :

1. User memilih dan melakukan penghapusan terhadap data majalah.
2. Boundary Class MajalahMaintenanceUI menampilkan data majalah yang ingin dihapus.
3. Boundary Class MajalahMaintenanceUI meminta konfirmasi penghapusan dan akan dikirim ke control class MajalahManager.
4. Control class MajalahManager akan melakukan penghapusan dari database.

2.3.4.8. Pengelolaan Data Koran

2.3.4.8.1. Penambahan Data Koran

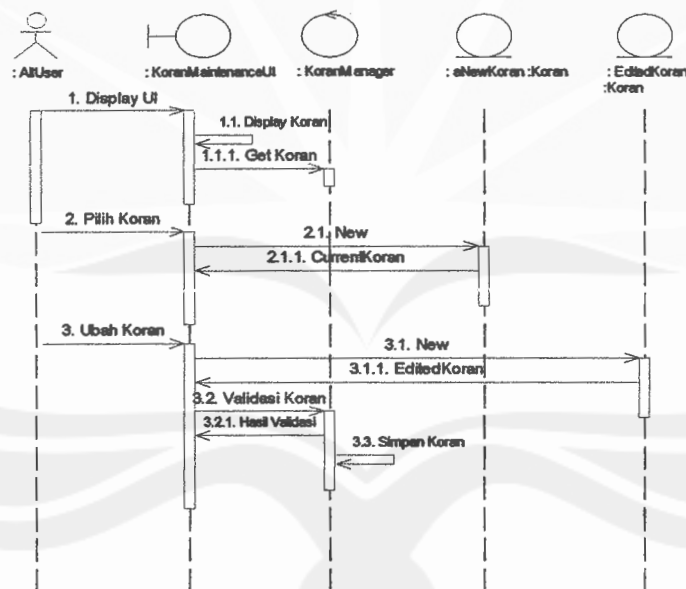


Gambar 2.97 Design Sequence Diagram : Use Case Penambahan Data Koran

Flow of events :

1. System akan menampilkan KoranMaintenanceUI sebagai object boundary class.
2. Aktor memasukkan detail data koran.
3. KoranMaintenanceUI akan membentuk instance dari data koran melalui object entity class Koran.
4. aNewKoran yang merupakan instance yang terbentuk akan dikembalikan ke KoranMaintenanceUI.
5. Instance yang terbentuk akan divalidasi melalui KoranManager.
6. Hasil validasi dari instance yang terbentuk akan dikembalikan ke KoranMaintenanceUI.
7. Data koran yang valid akan disimpan ke basis data melalui object control class KoranManager.

2.3.4.8.2. Perubahan Data Koran



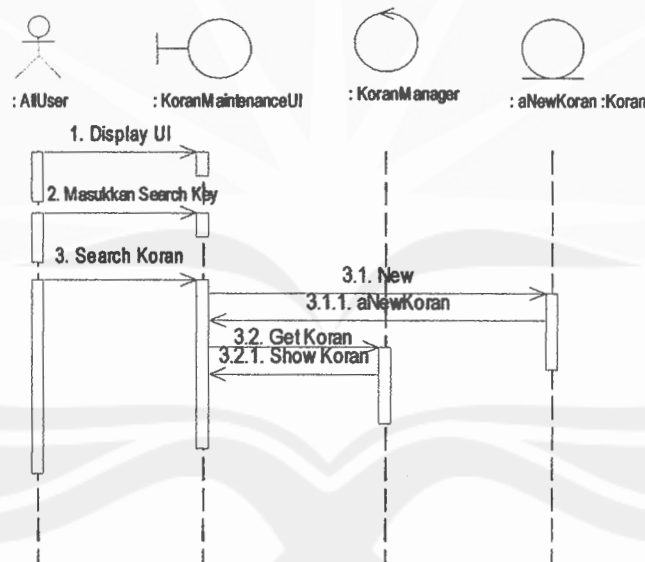
Gambar 2.98 Design Sequence Diagram : Use Case Perubahan Data Koran

Flow of events :

1. System akan menampilkan KoranMaintenanceUI sebagai object boundary class.
2. Sistem akan memanggil fungsi GetKoran () untuk mendapatkan semua data koran melalui object control class KoranManager.
3. Aktor memilih data koran yang akan diubah.

4. KoranMaintenanceUI akan memicu terbentuknya instance dari data koran yang akan diubah.
5. Instance dari data koran yang diambil dari basis data ini akan dikembalikan ke boundary class KoranMaintenanceUI untuk dilakukan perubahan.
6. Aktor akan mengubah data koran yang dipilih.
7. KoranMaintenanceUI akan membentuk instance EditedKoran terhadap data koran yang diubah.
8. Instance yang terbentuk ini akan dilakukan validasi ke control class KoranManager.
9. Hasil validasi instance EditedKoran yang baru ini akan dikembalikan ke boundary class KoranMaintenanceUI.
10. Perubahan terhadap data koran akan disimpan ke basis data.

2.3.4.8.3. Pencarian Data Koran



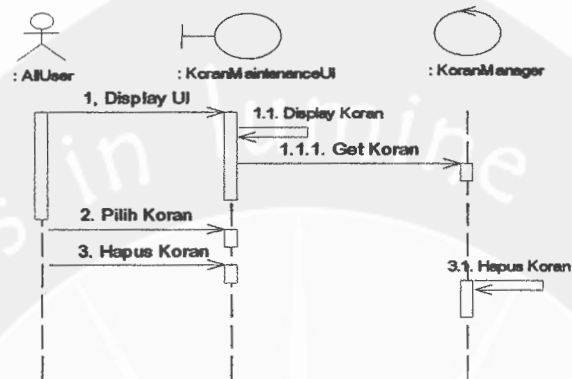
Gambar 2.99 Design Sequence Diagram : Use Case Pencarian Data Koran

Flow of events :

1. System akan menampilkan boundary class KoranMaintenanceUI terhadap semua data koran.
2. Aktor akan memasukkan search key untuk koran yang akan dicari.
3. KoranMaintenanceUI akan melakukan validasi search key.
4. Boundary class KoranMaintenanceUI akan memanggil fungsi GerKoran untuk mendapatkan data koran yang dicari.

5. Boundary class KoranMaintenanceUI akan membentuk instance baru untuk data koran yang akan dicari.
6. instance yang terbentuk akan dikembalikan ke KoranMaintenanceUI.
7. Sistem akan menampilkan data koran yang dicari.

2.3.4.8.4. Penghapusan Data Koran



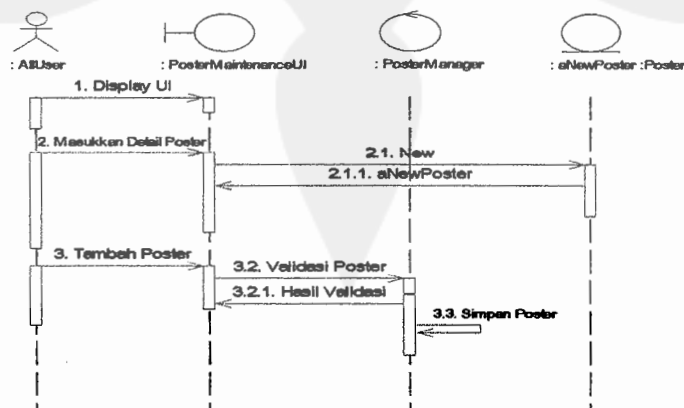
Gambar 2.100 Design Sequence Diagram : Use Case Penghapusan Data koran

Flow of events :

1. User memilih dan melakukan penghapusan terhadap data koran.
2. Boundary Class KoranMaintenanceUI menampilkan data koran yang ingin dihapus.
3. Boundary Class KoranMaintenanceUI meminta konfirmasi penghapusan dan akan dikirim ke control class KoranManager.
4. Control class KoranManager akan melakukan penghapusan dari database.

2.3.4.9. Pengelolaan Data Poster

2.3.4.9.1. Penambahan Data Poster

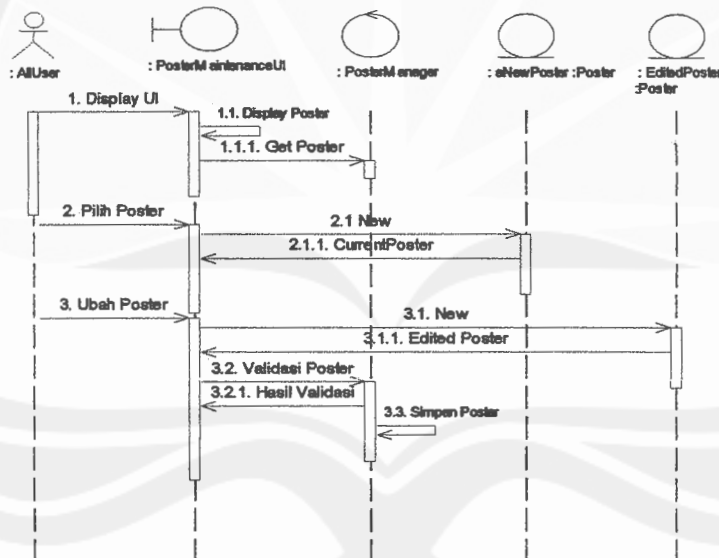


Gambar 2.101 Design Sequence Diagram : Use Case Penambahan Data Poster

Flow of events :

1. System akan menampilkan PosterMaintenanceUI sebagai object boundary class.
2. Aktor memasukkan detail data poster.
3. PosterMaintenanceUI akan membentuk instance dari data poster melalui object entity class Poster.
4. aNewPoster yang merupakan instance yang terbentuk akan dikembalikan ke PosterMaintenanceUI.
5. Instance yang terbentuk akan divalidasi melalui PosterManager.
6. Hasil validasi dari instance yang terbentuk akan dikembalikan ke PosterMaintenanceUI.
7. Data poster yang valid akan disimpan ke basis data melalui object control class PosterManager.

2.3.4.9.2. Pengubahan Data Poster



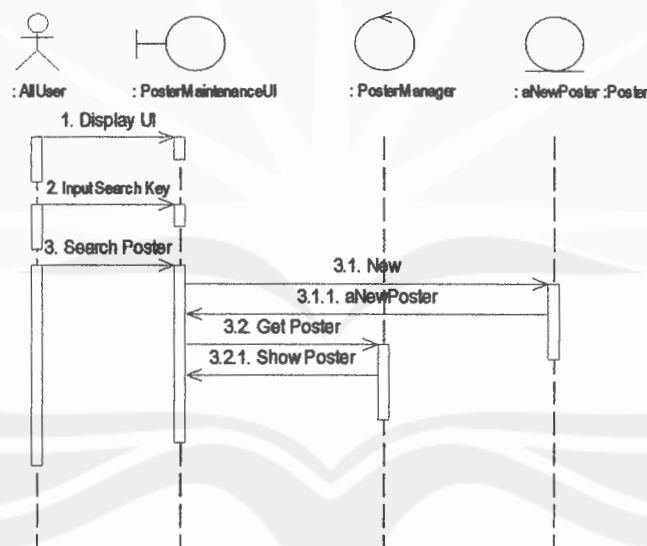
Gambar 2.102 Design Sequence Diagram : Use Case Pengubahan Data Poster

Flow of events :

1. System akan menampilkan PosterMaintenanceUI sebagai object boundary class.
2. Sistem akan memanggil fungsi GetPoster () untuk mendapatkan semua data poster melalui object control class PosterManager.
3. Aktor memilih data poster yang akan diubah.

4. PosterMaintenanceUI akan memicu terbentuknya instance dari data poster yang akan diubah.
5. Instance dari data poster yang diambil dari basis data ini akan dikembalikan ke boundary class PosterMaintenanceUI untuk dilakukan perubahan.
6. Aktor akan mengubah data poster yang dipilih.
7. PosterMaintenanceUI akan membentuk instance EditedPoster terhadap data poster yang diubah.
8. Instance yang terbentuk ini akan dilakukan validasi ke control class PosterManager.
9. Hasil validasi instance EditedPoster yang baru ini akan dikembalikan ke boundary class PosterMaintenanceUI.
10. Perubahan terhadap data poster akan disimpan ke basis data.

2.3.4.9.3. Pencarian Data Poster



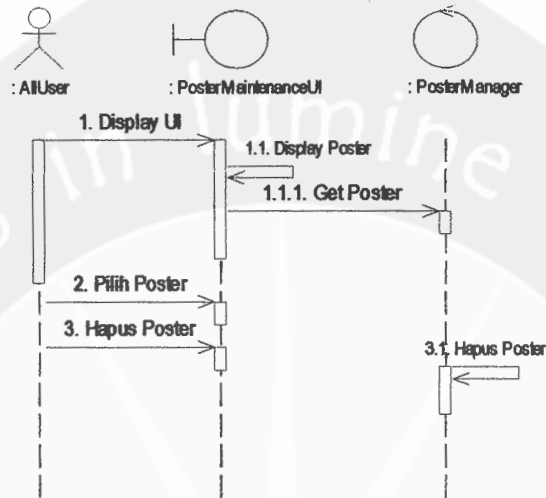
Gambar 2.103 Design Sequence Diagram : Use Case Pencarian Data Poster

Flow of events :

1. System akan menampilkan boundary class PosterMaintenanceUI terhadap semua data poster.
2. Aktor akan memasukkan search key untuk poster yang akan dicari.
3. PosterMaintenanceUI akan melakukan validasi search key.
4. Boundary class PosterMaintenanceUI akan memanggil fungsi GerPoster untuk mendapatkan data poster yang dicari.

5. Boundary class PosterMaintenanceUI akan membentuk instance baru untuk data poster yang akan dicari.
6. instance yang terbentuk akan dikembalikan ke PosterMaintenanceUI.
7. Sistem akan menampilkan data poster yang dicari.

2.3.4.9.4. Penghapusan Data Poster



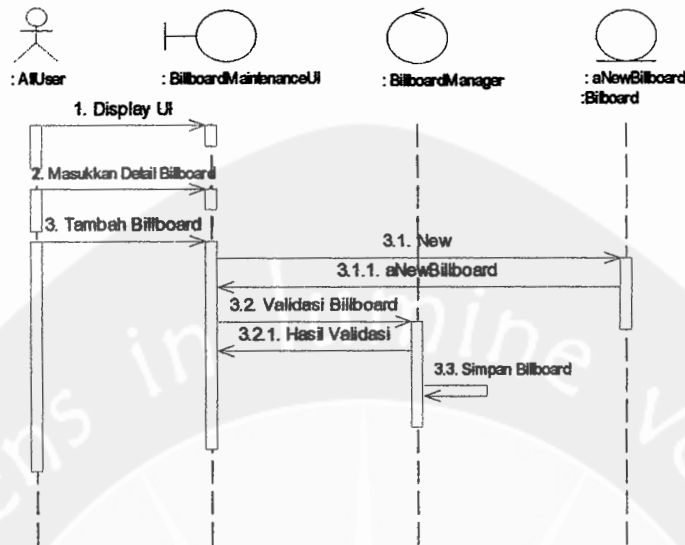
Gambar 2.104 Design Sequence Diagram : Use Case Penghapusan Data Poster

Flow of events :

1. User memilih dan melakukan penghapusan terhadap data poster.
2. Boundary Class PosterMaintenanceUI menampilkan data televisi yang ingin dihapus.
3. Boundary Class PosterMaintenanceUI meminta konfirmasi penghapusan dan akan dikirim ke control class PosterManager.
4. Control class PosterManager akan melakukan penghapusan dari database.

2.3.4.10. Pengelolaan Data Billboard

2.3.4.10.1. Penambahan Data Billboard

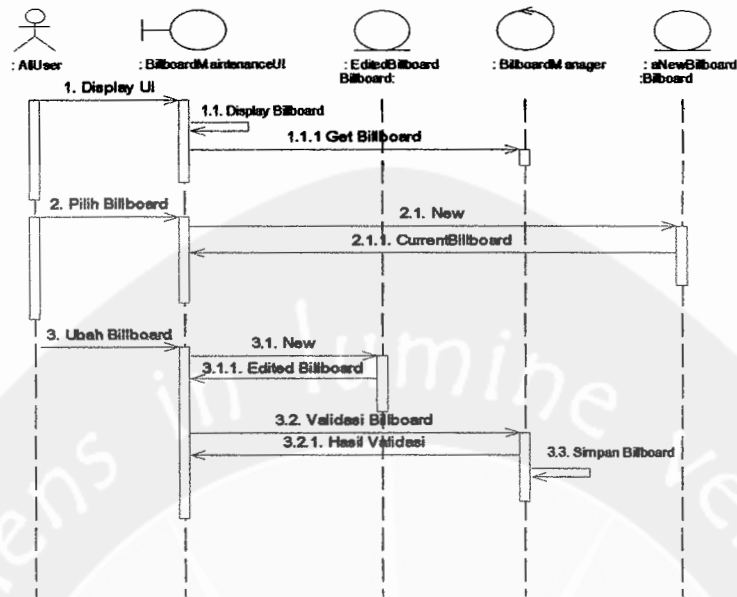


Gambar 2.105 Design Sequence Diagram : Use Case Penambahan Data Billboard

Flow of events :

1. System akan menampilkan BillboardMaintenanceUI sebagai object boundary class.
2. Aktor memasukkan detail data billboard.
3. BillboardMaintenanceUI akan membentuk instance dari data billboard melalui object entity class Billboard.
4. aNewBillboard yang merupakan instance yang terbentuk akan dikembalikan ke BillboardMaintenanceUI.
5. Instance yang terbentuk akan divalidasi melalui BillboardManager.
6. Hasil validasi dari instance yang terbentuk akan dikembalikan ke BillboardMaintenanceUI.
7. Data billboard yang valid akan disimpan ke basis data melalui object control class BillboardManager.

2.3.4.10.2. Perubahan Data Billboard

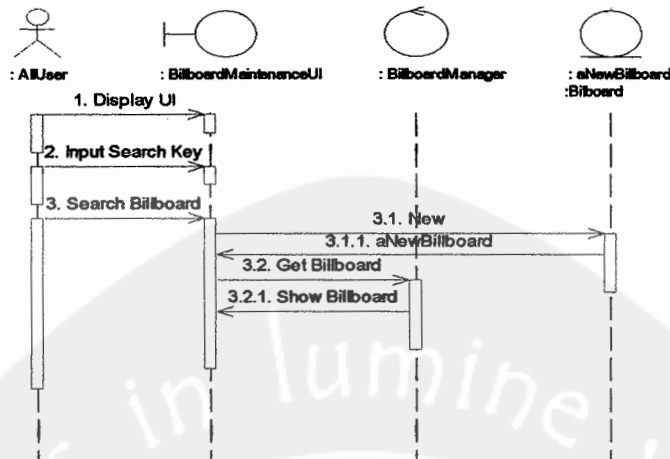


Gambar 2.106 Design Sequence Diagram : Use Case Perubahan Data Billboard

Flow of events :

1. System akan menampilkan BillboardMaintenanceUI sebagai object boundary class.
2. Sistem akan memanggil fungsi GetBillboard () untuk mendapatkan semua data billboard malalui object control class BillboardManager.
3. Aktor memilih data billboard yang akan diubah.
4. BillboardMaintenanceUI akan memicu terbentuknya instance dari data billboard yang akan diubah.
5. Instance dari data billboard yang diambil dari basis data ini akan dikembalikan ke boundary class BillboardMaintenanceUI untuk dilakukan perubahan.
6. Aktor akan mengubah data billboard yang dipilih.
7. BillboardMaintenanceUI akan membentuk instance EditedBillboard terhadap data billboard yang diubah.
8. Instance yang terbentuk ini akan dilakukan validasi ke control class BillboardManager.
9. Hasil validasi instance EditedBillboard yang baru ini akan dikembalikan ke boundary class BillboardMaintenanceUI.
10. Perubahan terhadap data billboard akan disimpan ke basis data.

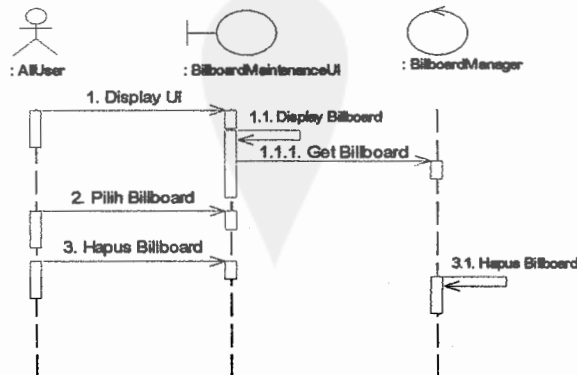
2.3.4.10.3. Pencarian Data Billboard



Gambar 2.107 Design Sequence Diagram : Use Case Pencarian Data Billboard
Flow of events :

1. System akan menampilkan boundary class BillboardMaintenanceUI terhadap semua data billboard.
2. Aktor akan memasukkan search key untuk billboard yang akan dicari.
3. BillboardMaintenanceUI akan melakukan validasi search key.
4. Boundary class BillboardMaintenanceUI akan memanggil fungsi GerBillboard untuk mendapatkan data billboard yang dicari.
5. Boundary class BillboardMaintenanceUI akan membentuk instance baru untuk data billboard yang akan dicari.
6. instance yang terbentuk akan dikembalikan ke BillboardMaintenanceUI.
7. Sistem akan menampilkan data billboard yang dicari.

2.3.4.10.4. Penghapusan Data Billboard



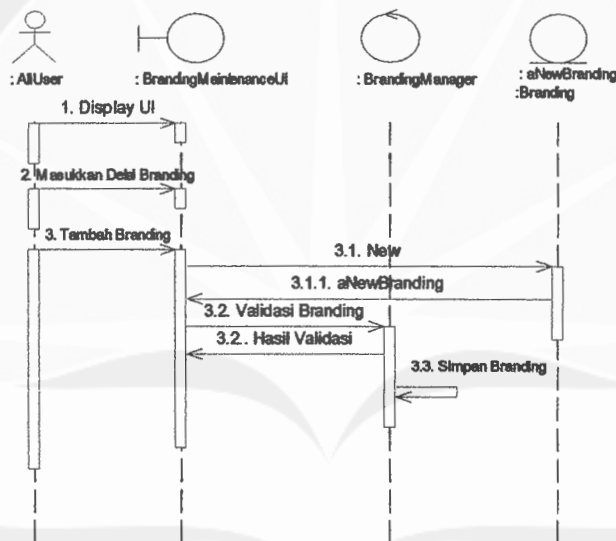
Gambar 2.108 Design Sequence Diagram : Use Case Penghapusan Data Billboard

Flow of events :

1. User memilih dan melakukan penghapusan terhadap data billboard.
2. Boundary Class BillboardMaintenancUI menampilkan data billboard yang ingin dihapus.
3. Boundary Class BillboardMaintenancUI meminta konfirmasi penghapusan dan akan dikirim ke control class BillboardManager.
4. Control class BillboardManager akan melakukan penghapusan dari database.

2.3.4.11. Pengelolaan Data Branding

2.3.4.11.1. Penambahan Data Branding



Gambar 2.109 Design Sequence Diagram : Use Case Penambahan Data Branding

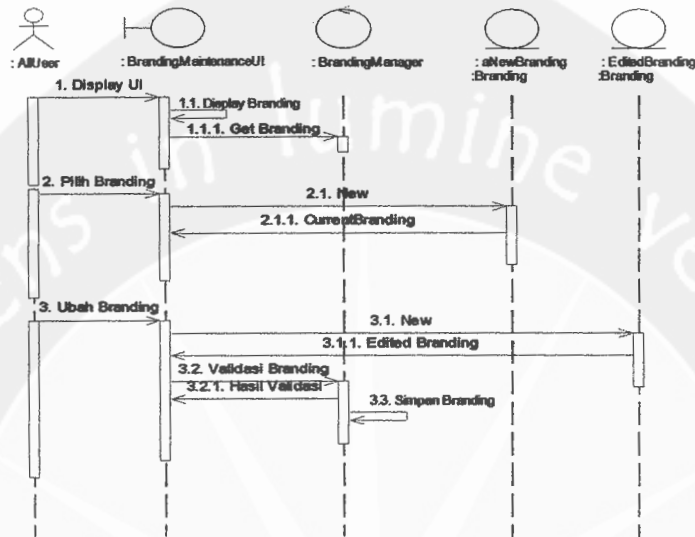
Flow of events :

1. System akan menampilkan BrandingMaintenanceUI sebagai object boundary class.
2. Aktor memasukkan detail data branding.
3. BrandingMaintenanceUI akan membentuk instance dari data branding melalui object entity class Branding.
4. aNewBranding yang merupakan instance yang terbentuk akan dikembalikan ke BrandingMaintenanceUI.
5. Instance yang terbentuk akan divalidasi melalui BrandingManager.

6. Hasil validasi dari instance yang terbentuk akan dikembalikan ke BrandingMaintenanceUI.

7. Data branding yang valid akan disimpan ke basis data melalui object control class BrandingManager.

2.3.4.11.2. Pengubahan Data Branding

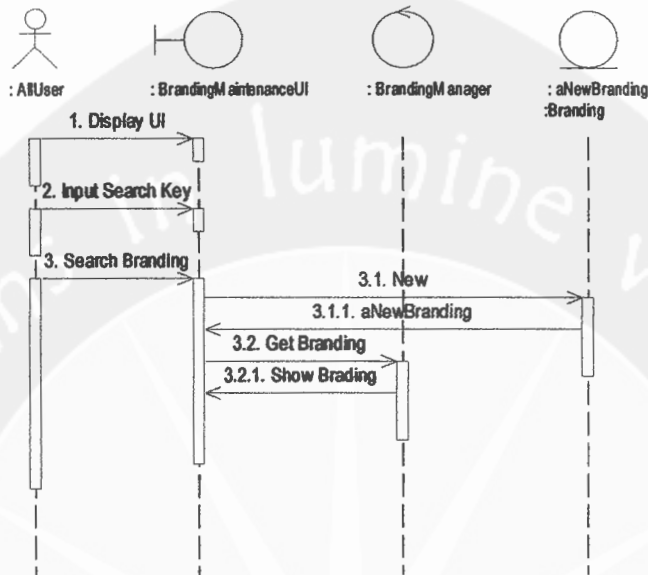


Gambar 2.110 Design Sequence Diagram : Use Case Pengubahan Data Branding
Flow of events :

1. System akan menampilkan BrandingMaintenanceUI sebagai object boundary class.
2. Sistem akan memanggil fungsi GetBranding () untuk mendapatkan semua data branding melalui object control class BrandingManager.
3. Aktor memilih data branding yang akan diubah.
4. BrandingMaintenanceUI akan memicu terbentuknya instance dari data branding yang akan diubah.
5. Instance dari data branding yang diambil dari basis data ini akan dikembalikan ke boundary class BrandingMaintenanceUI untuk dilakukan perubahan.
6. Aktor akan mengubah data branding yang dipilih.
7. BrandingMaintenanceUI akan membentuk instance EditedBranding terhadap data branding yang diubah.
8. Instance yang terbentuk ini akan dilakukan validasi ke control class BrandingManager.

9. Hasil validasi instance EditedBranding yang baru ini akan dikembalikan ke boundary class BrandingMaintenanceUI.
10. Perubahan terhadap data branding akan disimpan ke basis data.

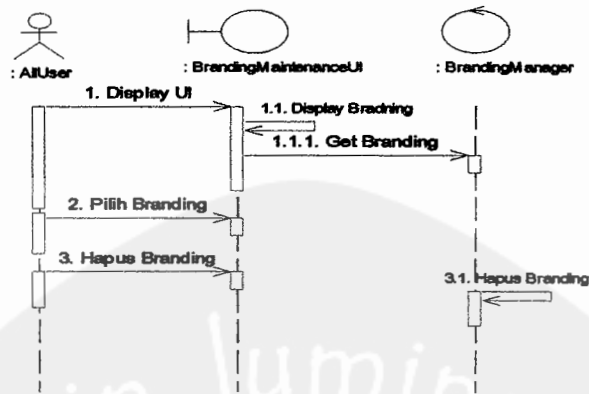
2.3.4.11.3. Pencarian Data Branding



Gambar 2.111 Design Sequence Diagram : Use Case Pencarian Data Branding
 Flow of events :

1. System akan menampilkan boundary class BradingMaintenanceUI terhadap semua data brading.
2. Aktor akan memasukkan search key untuk brading yang akan dicari.
3. BradingMaintenanceUI akan melakukan validasi search key.
4. Boundary class BradingMaintenanceUI akan memanggil fungsi GerBrading untuk mendapatkan data brading yang dicari.
5. Boundary class BradingMaintenanceUI akan membentuk instance baru untuk data brading yang akan dicari.
6. instance yang tebentuk akan dikembalikan ke BradingMaintenanUI.
7. Sistem akan menampilkan data brading yang dicari.

2.3.4.11.4. Penghapusan Data Branding

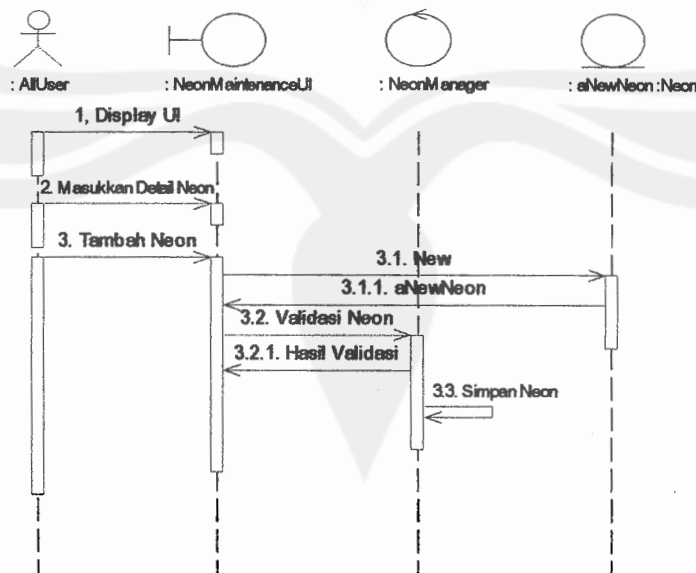


Gambar 2.112 Design Sequence Diagram : Use Case Penghapusan Data Branding
 Flow of events :

1. User memilih dan melakukan penghapusan terhadap data branding.
2. Boundary Class BrandingMaintenanceUI menampilkan data branding yang ingin dihapus.
3. Boundary Class BrandingMaintenanceUI meminta konfirmasi penghapusan dan akan dikirim ke control class BrandingManager.
4. Control class BrandingManager akan melakukan penghapusan dari database.

2.3.4.12. Pengelolaan Data Neon

2.3.4.12.1. Penambahan Data Neon

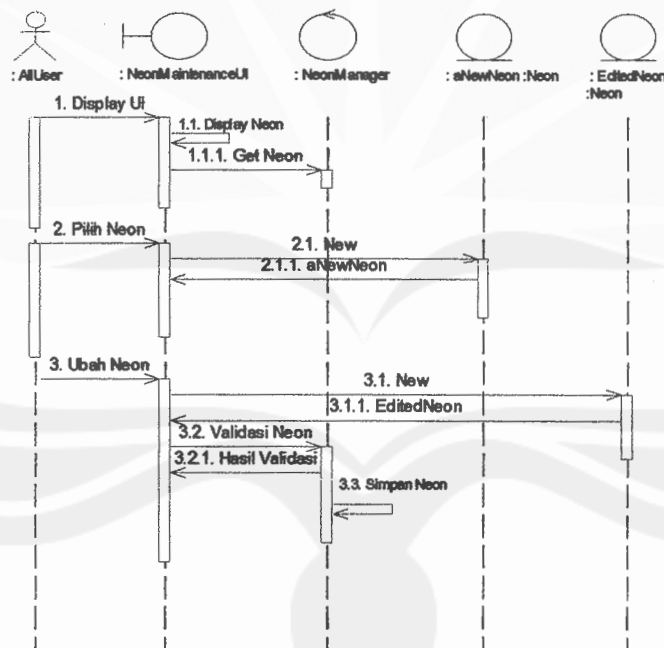


Gambar 2.113 Design Sequence Diagram : Use Case Penambahan Data Neon

Flow of events :

1. System akan menampilkan NeonMaintenanceUI sebagai object boundary class.
2. Aktor memasukkan detail data neon.
3. NeonMaintenanceUI akan membentuk instance dari data neon melalui object entity class Neon.
4. aNewNeon yang merupakan instance yang terbentuk akan dikembalikan ke NeonMaintenanceUI.
5. Instance yang terbentuk akan divalidasi melalui NeonManager.
6. Hasil validasi dari instance yang terbentuk akan dikembalikan ke NeonMaintenanceUI.
7. Data neon yang valid akan disimpan ke basis data melalui object control class NeonManager.

2.3.4.12.2. Pengubahan Data Neon



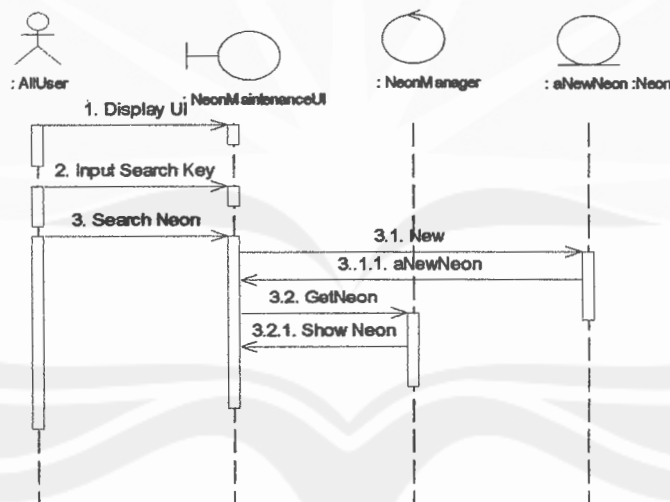
Gambar 2.114 Design Sequence Diagram : Use Case Pengubahan Data Neon

Flow of events :

1. System akan menampilkan NeonMaintenanceUI sebagai object boundary class.
2. Sistem akan memanggil fungsi GetNeon () untuk mendapatkan semua data neon melalui object control class NeonManager.

3. Aktor memilih data neon yang akan diubah.
4. NeonMaintenanceUI akan memicu terbentuknya instance dari data neon yang akan diubah.
5. Instance dari data neon yang diambil dari basis data ini akan dikembalikan ke boundary class NeonMaintenanceUI untuk dilakukan perubahan.
6. Aktor akan mengubah data neon yang dipilih.
7. NeonMaintenanceUI akan membentuk instance EditedNeon terhadap data neon yang diubah.
8. Instance yang terbentuk ini akan dilakukan validasi ke control class NeonManager.
9. Hasil validasi instance EditedNeon yang baru ini akan dikembalikan ke boundary class NeonMaintenanceUI.
10. Perubahan terhadap data neon akan disimpan ke basis data.

2.3.4.12.3. Pencarian Data Neon



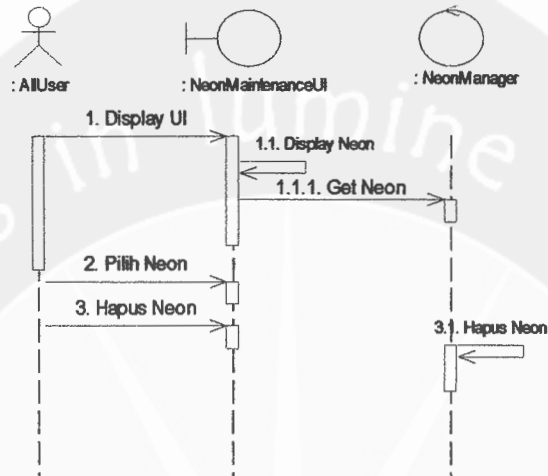
Gambar 2.115 Design Sequence Diagram : Use Case Pencarian Data Neon

Flow of events :

1. System akan menampilkan boundary class NeonMaintenanceUI terhadap semua data neon.
2. Aktor akan memasukkan search key untuk neon yang akan dicari.
3. NeonMaintenanceUI akan melakukan validasi search key.
4. Boundary class NeonMaintenanceUI akan memanggil fungsi GerNeon untuk mendapatkan data neon yang dicari.

5. Boundary class NeonMaintenanceUI akan membentuk instance baru untuk data neon yang akan dicari.
6. instance yang terbentuk akan dikembalikan ke NeonMaintenanceUI.
7. Sistem akan menampilkan data neon yang dicari.

2.3.4.12.4. Penghapusan Data Neon



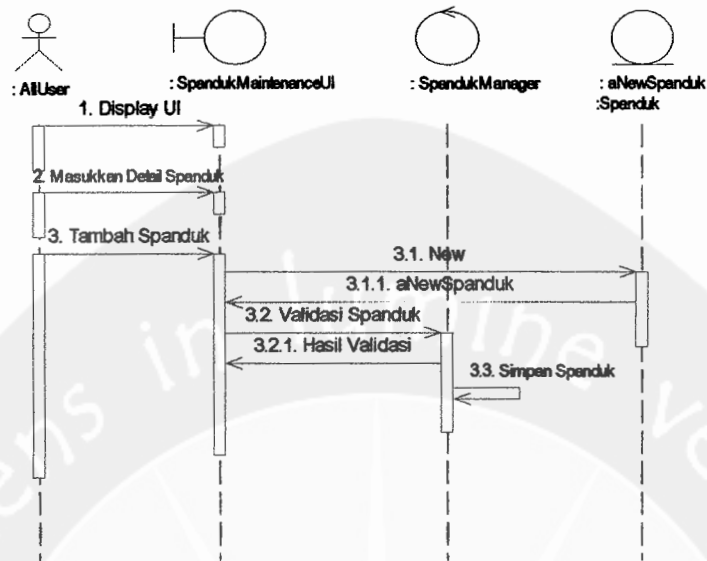
Gambar 2.116 Design Sequence Diagram : Use Case Penghapusan Data Neon

Flow of events :

1. User memilih dan melakukan penghapusan terhadap data neon.
2. Boundary Class NeonMaintenanceUI menampilkan data neon yang ingin dihapus.
3. Boundary Class NeonMaintenanceUI meminta konfirmasi penghapusan dan akan dikirim ke control class NeonManager.
4. Control class NeonManager akan melakukan penghapusan dari database.

2.3.4.13. Pengelolaan Data Spanduk

2.3.4.13.1. Penambahan Data Spanduk

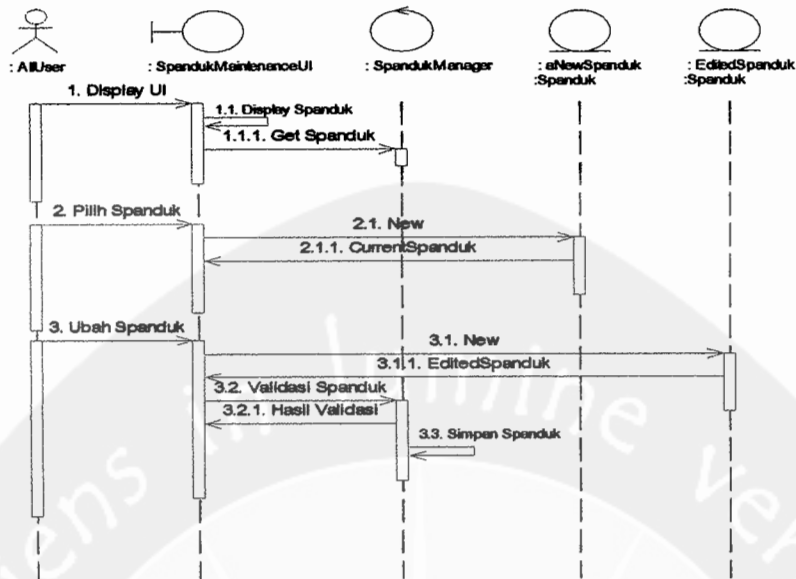


Gambar 2.117 Design Sequence Diagram : Use Case Penambahan Data Spanduk

Flow of events :

1. System akan menampilkan SpandukMaintenanceUI sebagai object boundary class.
2. Aktor memasukkan detail data spanduk.
3. SpandukMaintenanceUI akan membentuk instance dari data spanduk melalui object entity class Spanduk.
4. aNewSpanduk yang merupakan instance yang terbentuk akan dikembalikan ke SpandukMaintenanceUI.
5. Instance yang terbentuk akan divalidasi melalui SpandukManager.
6. Hasil validasi dari instance yang terbentuk akan dikembalikan ke SpandukMaintenanceUI.
7. Data spanduk yang valid akan disimpan ke basis data melalui object control class SpandukManager.

2.3.4.13.2. Perubahan Data Spanduk

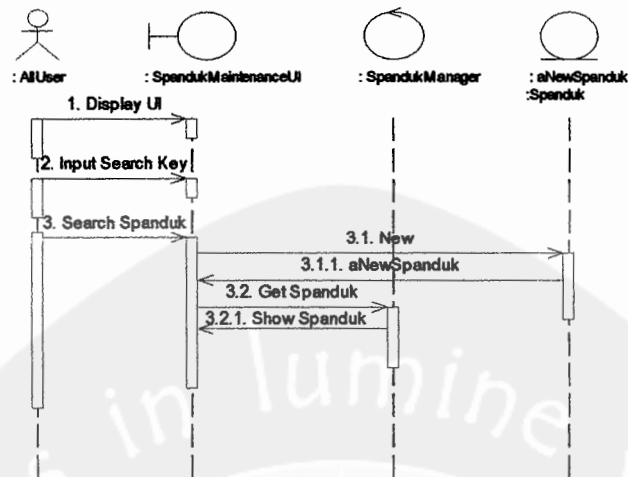


Gambar 2.118 Design Sequence Diagram : Use Case Perubahan Data Spanduk

Flow of events :

1. System akan menampilkan SpandukMaintenanceUI sebagai object boundary class.
2. Sistem akan memanggil fungsi GetSpanduk () untuk mendapatkan semua data spanduk malalui object control class SpandukManager.
3. Aktor memilih data spanduk yang akan diubah.
4. SpandukMaintenanceUI akan memicu terbentuknya instance dari data spanduk yang akan diubah.
5. Instance dari data spanduk yang diambil dari basis data ini akan dikembalikan ke boundary class SpandukMaintenanceUI untuk dilakukan perubahan.
6. Aktor akan mengubah data spanduk yang dipilih.
7. SpandukMaintenanceUI akan membentuk instance EditedSpanduk terhadap data spanduk yang diubah.
8. Instance yang terbentuk ini akan dilakukan validasi ke control class SpandukManager.
9. Hasil validasi instance EditedSpanduk yang baru ini akan dikembalikan ke boundary class SpandukMaintenanceUI.
10. Perubahan terhadap data spanduk akan disimpan ke basis data.

2.3.4.13.3. Pencarian Data Spanduk

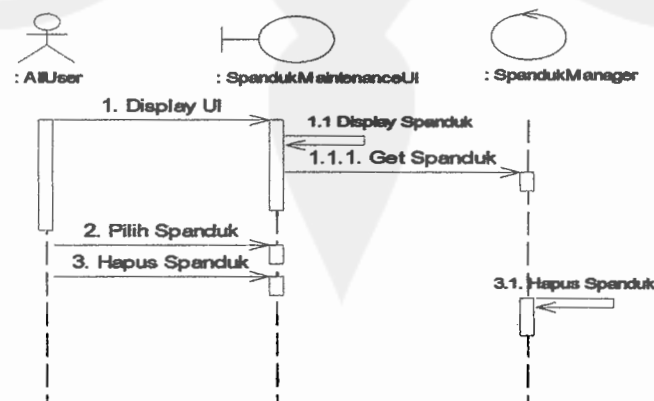


Gambar 2.119 Design Sequence Diagram : Use Case Pencarian Data Spanduk

Flow of events :

1. System akan menampilkan boundary class SpandukMaintenanceUI terhadap semua data spanduk.
2. Aktor akan memasukkan search key untuk spanduk yang akan dicari.
3. SpandukMaintenanceUI akan melakukan validasi search key.
4. Boundary class SpandukMaintenanceUI akan memanggil fungsi GerSpanduk untuk mendapatkan data spanduk yang dicari.
5. Boundary class SpandukMaintenanceUI akan membentuk instance baru untuk data spanduk yang akan dicari.
6. instance yang terbentuk akan dikembalikan ke SpandukMaintenanceUI.
7. Sistem akan menampilkan data spanduk yang dicari.

2.3.4.13.4. Penghapusan Data Spanduk



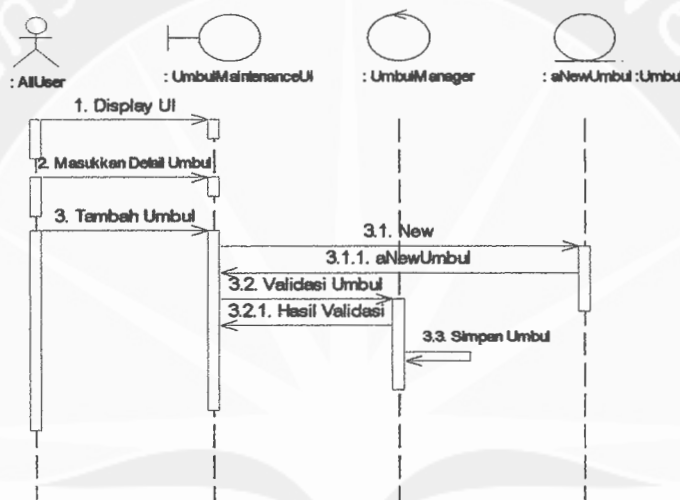
Gambar 2.120 Design Sequence Diagram : Use Case Penghapusan Data Spanduk

Flow of events :

1. User memilih dan melakukan penghapusan terhadap data spanduk.
2. *Boundary Class* SpandukMaintenancUI menampilkan data spanduk yang ingin dihapus.
3. *Boundary Class* SpandukMaintenancUI meminta konfirmasi penghapusan dan akan dikirim ke *control class* SpandukManager.
4. *Control class* SpandukManager akan melakukan penghapusan dari *database*.

2.3.4.14. Pengelolaan Data Umbul-Umbul

2.3.4.14.1. Penambahan Data Umbul

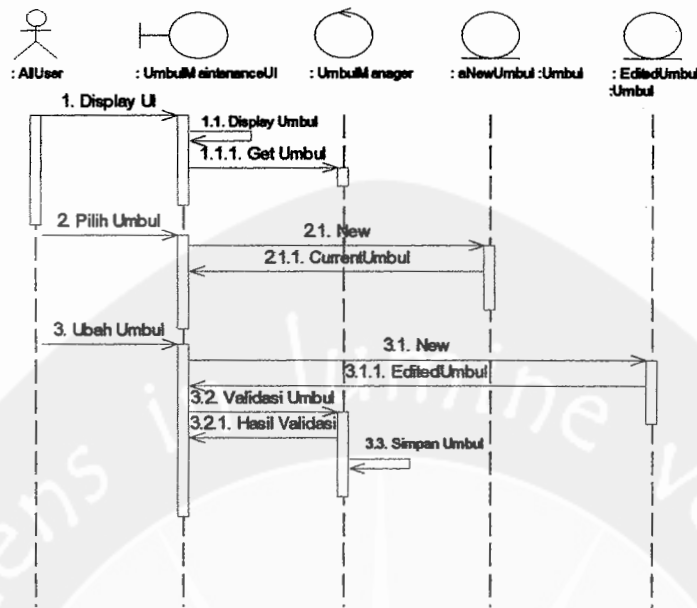


Gambar 2.121 Design Sequence Diagram : Use Case Penambahan Data Umbul

Flow of events :

1. System akan menampilkan UmbulMaintenanceUI sebagai object boundary class.
2. Aktor memasukkan detail data umbul.
3. UmbulMaintenanceUI akan membentuk instance dari data umbul melalui object entity class Umbul.
4. aNewUmbul yang merupakan instance yang terbentuk akan dikembalikan ke UmbulMaintenanceUI.
5. Instance yang terbentuk akan divalidasi melalui UmbulManager.
6. Hasil validasi dari instance yang terbentuk akan dikembalikan ke UmbulMaintenanceUI.
7. Data umbul yang valid akan disimpan ke basis data melalui object control class UmbulManager.

2.3.4.14.2. Perubahan Data Umbul

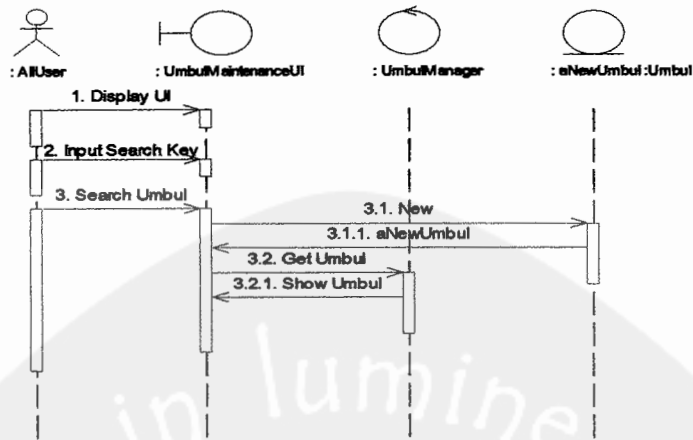


Gambar 2.122 Design Sequence Diagram : Use Case Perubahan Data Umbul

Flow of events :

1. System akan menampilkan UmbulMaintenanceUI sebagai object boundary class.
2. Sistem akan memanggil fungsi GetUmbul () untuk mendapatkan semua data umbul melalui object control class UmbulManager.
3. Aktor memilih data umbul yang akan diubah.
4. UmbulMaintenanceUI akan memicu terbentuknya instance dari data umbul yang akan diubah.
5. Instance dari data umbul yang diambil dari basis data ini akan dikembalikan ke boundary class UmbulMaintenanceUI untuk dilakukan perubahan.
6. Aktor akan mengubah data umbul yang dipilih.
7. UmbulMaintenanceUI akan membentuk instance EditedUmbul terhadap data umbul yang diubah.
8. Instance yang terbentuk ini akan dilakukan validasi ke control class UmbulManager.
9. Hasil validasi instance EditedUmbul yang baru ini akan dikembalikan ke boundary class UmbulMaintenanceUI.
10. Perubahan terhadap data umbul akan disimpan ke basis data.

2.3.4.14.3. Pencarian Data Umbul

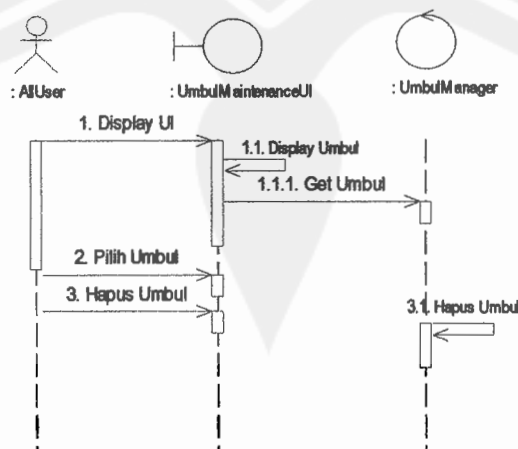


Gambar 2.123 Design Sequence Diagram : Use Case Pencarian Data Umbul

Flow of events :

1. System akan menampilkan boundary class UmbulMaintenanceUI terhadap semua data umbul.
2. Aktor akan memasukkan search key untuk umbul yang akan dicari.
3. UmbulMaintenanceUI akan melakukan validasi search key.
4. Boundary class UmbulMaintenanceUI akan memanggil fungsi GerUmbul untuk mendapatkan data umbul yang dicari.
5. Boundary class UmbulMaintenanceUI akan membentuk instance baru untuk data umbul yang akan dicari.
6. instance yang terbentuk akan dikembalikan ke UmbulMaintenanceUI.
7. Sistem akan menampilkan data umbul yang dicari.

2.3.4.14.4. Penghapusan Data Umbul



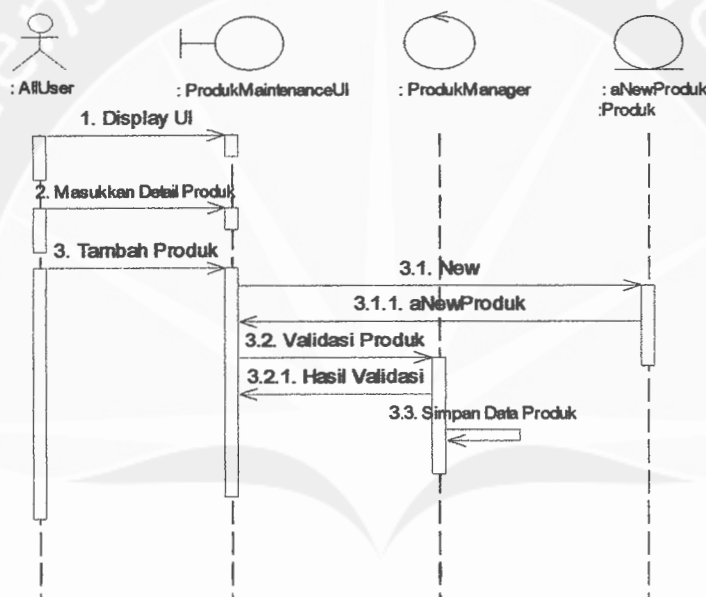
Gambar 2.124 Design Sequence Diagram : Use Case Penghapusan Data Umbul

Flow of events :

1. User memilih dan melakukan penghapusan terhadap data umbul.
2. *Boundary Class* UmbulMaintenanceUI menampilkan data umbul yang ingin dihapus.
3. *Boundary Class* UmbulMaintenanceUI meminta konfirmasi penghapusan dan akan dikirim ke *control class* UmbulManager.
4. *Control class* UmbulManager akan melakukan penghapusan dari *database*.

2.3.4.15. Pengelolaan Data Produk

2.3.4.15.1. Penambahan Data Produk



Gambar 2.125 Design Sequence Diagram : Use Case Penambahan Data Produk

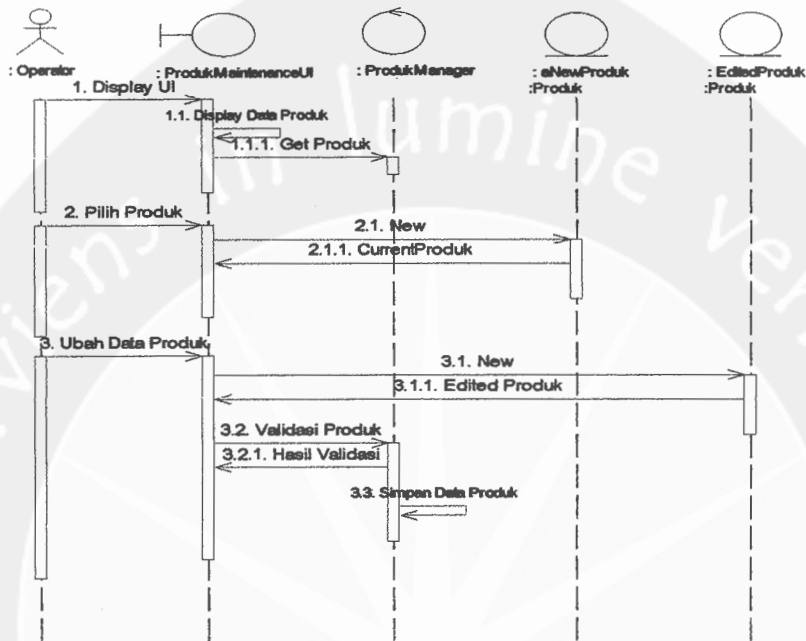
Flow of events :

1. System akan menampilkan ProdukMaintenanceUI sebagai object boundary class.
2. Aktor memasukkan detail data produk.
3. ProdukMaintenanceUI akan membentuk instance dari data produk melalui object entity class Produk.
4. aNewProduk yang merupakan instance yang terbentuk akan dikembalikan ke ProdukMaintenanceUI.
5. Instance yang terbentuk akan divalidasi melalui ProdukManager.

6. Hasil validasi dari instance yang terbentuk akan dikembalikan ke ProdukMaintenanceUI.

7. Data produk yang valid akan disimpan ke basis data melalui object control class ProdukManager.

2.3.4.15.2. Perubahan Data Produk



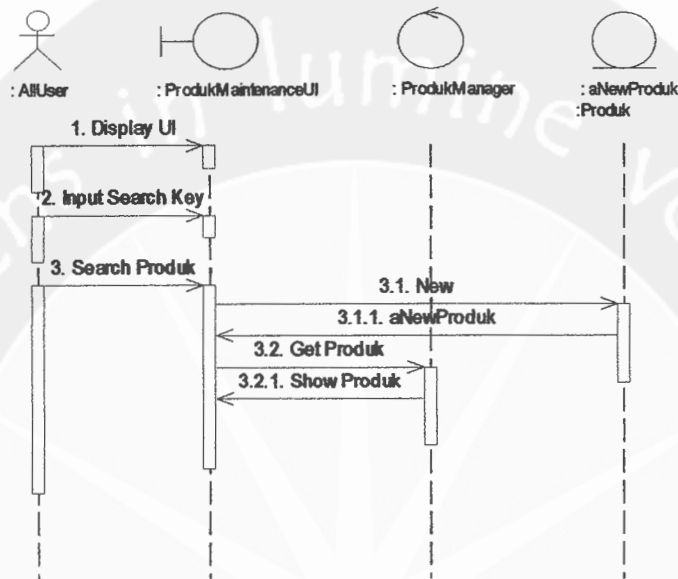
Gambar 2.126 Design Sequence Diagram : Use Case Perubahan Data Produk

Flow of events :

1. System akan menampilkan ProdukMaintenanceUI sebagai object boundary class.
2. Sistem akan memanggil fungsi GetProduk () untuk mendapatkan semua data produk melalui object control class ProdukManager.
3. Aktor memilih data produk yang akan diubah.
4. ProdukMaintenanceUI akan memicu terbentuknya instance dari data produk yang akan diubah.
5. Instance dari data produk yang diambil dari basis data ini akan dikembalikan ke boundary class ProdukMaintenanceUI untuk dilakukan perubahan.
6. Aktor akan mengubah data produk yang dipilih.
7. ProdukMaintenanceUI akan membentuk instance EditedProduk terhadap data produk yang diubah.

8. Instance yang terbentuk ini akan dilakukan validasi ke control class `ProdukManager`.
9. Hasil validasi instance `EditedProduk` yang baru ini akan dikembalikan ke boundary class `ProdukMaintenanceUI`.
10. Perubahan terhadap data produk akan disimpan ke basis data.

2.3.4.15.3. Pencarian Data Produk



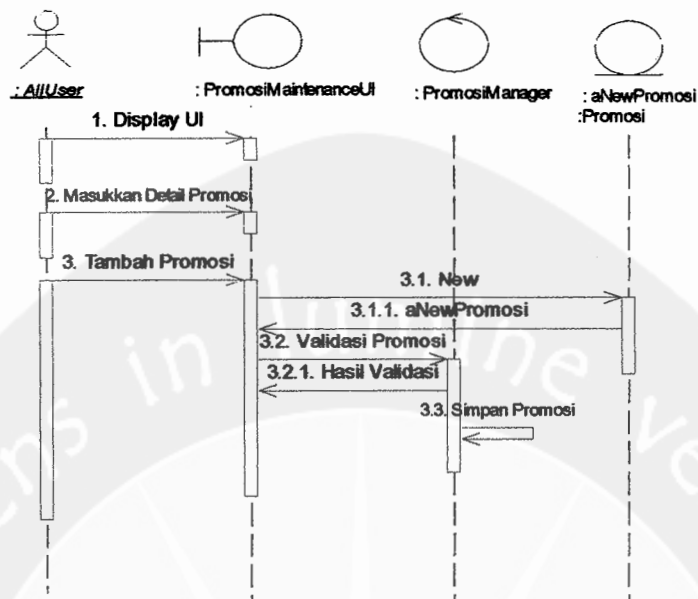
Gambar 2.127 Design Sequence Diagram : Use Case Pencarian Data Produk

Flow of events :

1. System akan menampilkan boundary class `ProdukMaintenanceUI` terhadap semua data produk.
2. Aktor akan memasukkan search key untuk produk yang akan dicari.
3. `ProdukMaintenanceUI` akan melakukan validasi search key.
4. Boundary class `ProdukMaintenanceUI` akan memanggil fungsi `GerProduk` untuk mendapatkan data produk yang dicari.
5. Boundary class `ProdukMaintenanceUI` akan membentuk instance baru untuk data produk yang akan dicari.
6. instance yang terbentuk akan dikembalikan ke `ProdukMaintenanceUI`.
7. Sistem akan menampilkan data produk yang dicari.

2.3.4.16. Pengelolaan Data Promosi

2.3.4.16.1. Penambahan Data Promosi

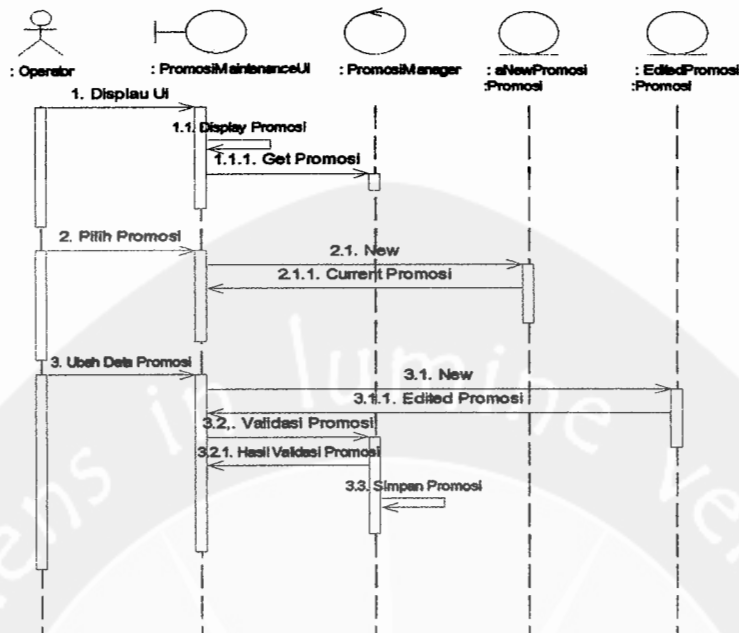


Gambar 2.128 Design Sequence Diagram : Use Case Penambahan Data Promosi

Flow of events :

1. System akan menampilkan PromosiMaintenanceUI sebagai object boundary class.
2. Aktor memasukkan detail data promosi.
3. PromosiMaintenanceUI akan membentuk instance dari data promosi melalui object entity class Promosi.
4. aNewPromosi yang merupakan instance yang terbentuk akan dikembalikan ke PromosiMaintenanceUI.
5. Instance yang terbentuk akan divalidasi melalui PromosiManager.
6. Hasil validasi dari instance yang terbentuk akan dikembalikan ke PromosiMaintenanceUI.
7. Data promosi yang valid akan disimpan ke basis data melalui object control class PromosiManager.

2.3.4.16.2. Perubahan Data Promosi

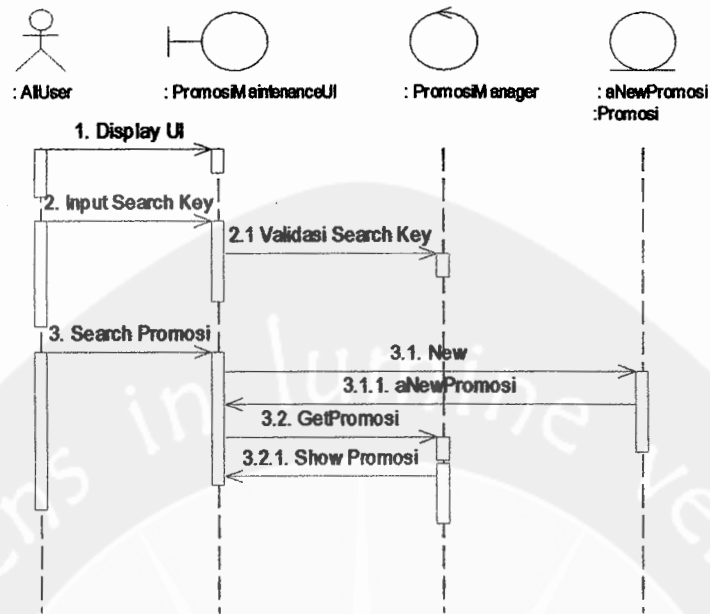


Gambar 2.129 Design Sequence Diagram : Use Case Perubahan Data Promosi

Flow of events :

1. System akan menampilkan PromosiMaintenanceUI sebagai object boundary class.
2. Sistem akan memanggil fungsi `GetPromosi ()` untuk mendapatkan semua data promosi malalui object control class PromosiManager.
3. Aktor memilih data promosi yang akan diubah.
4. PromosiMaintenanceUI akan memicu terbentuknya instance dari data promosi yang akan diubah.
5. Instance dari data promosi yang diambil dari basis data ini akan dikembalikan ke boundary class PromosiMaintenanceUI untuk dilakukan perubahan.
6. Aktor akan mengubah data promosi yang dipilih.
7. PromosiMaintenanceUI akan membentuk instance EditedPromosi terhadap data promosi yang diubah.
8. Instance yang terbentuk ini akan dilakukan validasi ke control class PromosiManager.
9. Hasil validasi instance EditedPromosi yang baru ini akan dikembalikan ke boundary class PromosiMaintenanceUI.
10. Perubahan terhadap data promosi akan disimpan ke basis data.

2.3.4.16.3. Pencarian Data Promosi

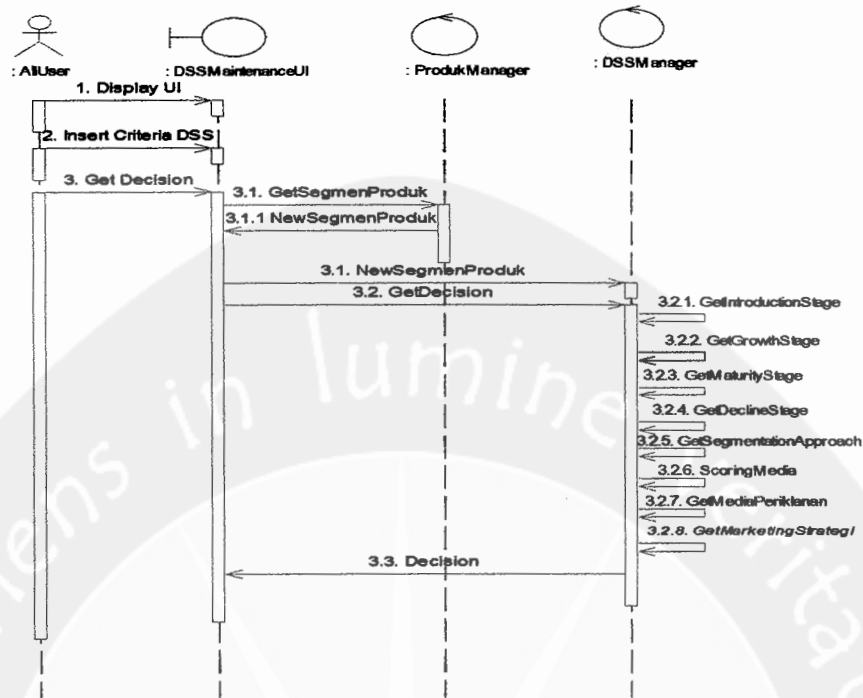


Gambar 2.130 Design Sequence Diagram : Use Case Pencarian Data Promosi

Flow of events :

1. System akan menampilkan boundary class PromosiMaintenanceUI terhadap semua data promosi.
2. Aktor akan memasukkan search key untuk data promosi yang akan dicari.
3. PromosiMaintenanceUI akan melakukan validasi search key.
4. Boundary class PromosiMaintenanceUI akan memanggil fungsi GerPromosi untuk mendapatkan data promosi yang dicari.
5. Boundary class PromosiMaintenanceUI akan membentuk instance baru untuk data promosi yang akan dicari.
6. instance yang terbentuk akan dikembalikan ke PromosiMaintenanceUI.
7. Sistem akan menampilkan data promosi yang dicari.

2.3.4.17. SPK Promosi Periklanan

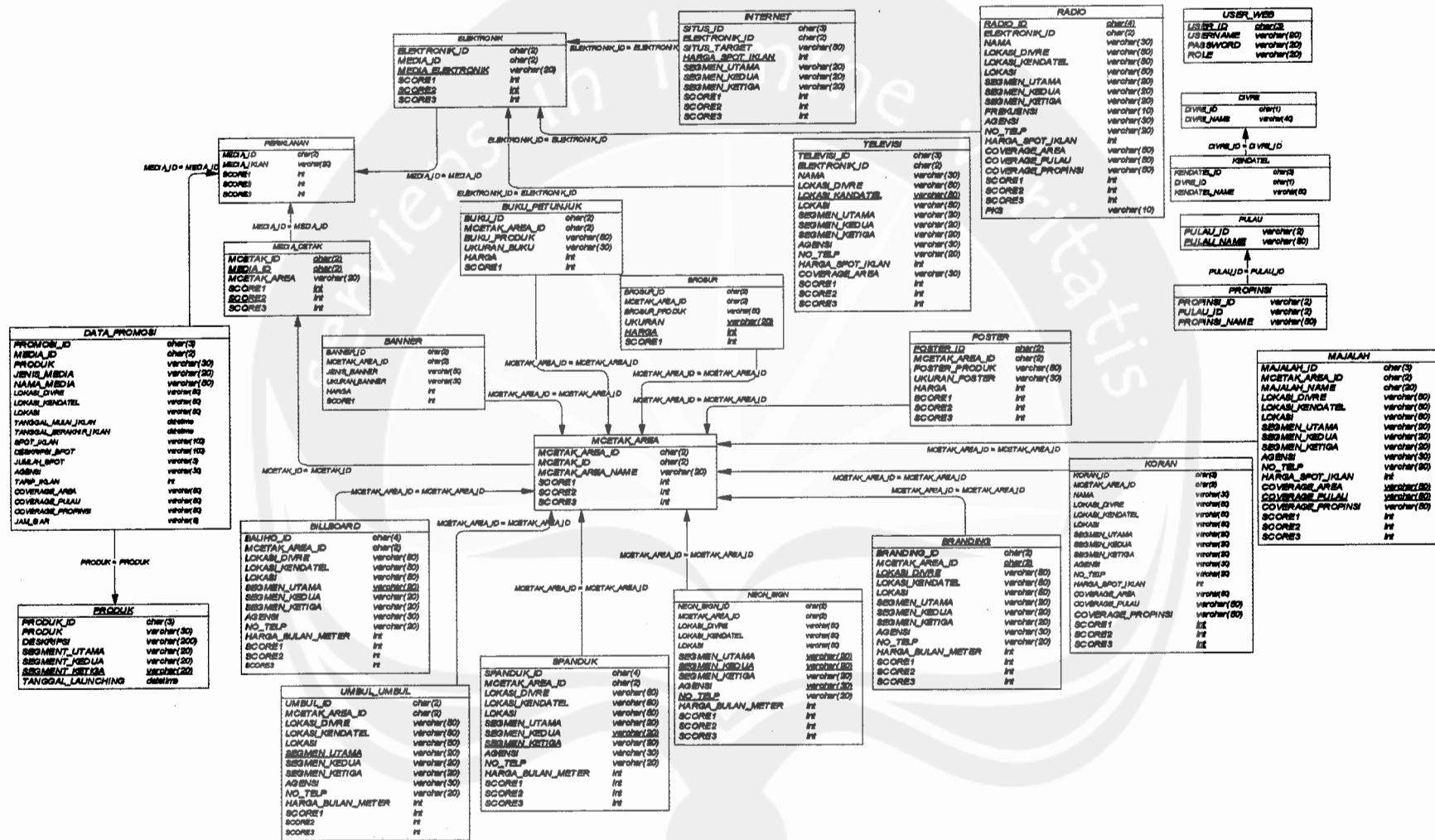


Gambar 2.131 Design Sequence Diagram : Use Case SPK Promosi Periklanan
Flow of events :

1. System akan menampilkan boundary class DSSMaintenanceUI untuk mengambil keputusan.
2. Aktor akan memasukkan kriteria untuk penentuan keputusan.
3. System akan mencari segmen pruduk (segmen utama, segmen kedua dan segmen ketiga)
4. Dari kriteria-kriteria yang telah diinputkan, akan didapatkan tahap siklus hidup produk, baik itu Introduction, Growth, Maturity ataupun Decline.
5. System akan menghitung score masing-masing media periklanan
6. System akan menghitung nilai/kesesuaian segmen produk dan segmen media periklanan sesuai dengan pendekatan segmentasi pemasaran yang digunakan.
7. System akan mendapatkan media periklanan yang paling tepat untuk melakukan promosi suatu produk
8. System akan mencari strategi pemasaran dari kriteria yang telah diinputkan
9. Keputusan akan dihasilkan.

3. Deskripsi Perancangan Persistenst Data

3.1. Basis Data



Gambar 3.128 Phisycal Data Model

3.1.1. Tabel User_web

Tabel ini merepresentasikan entitas User.

Tabel 3.1 Tabel User_web.

Field	Data Type	NULL	Description
User_id	Char(3)	No	User id(PK)
Username	Varchar(20)	NO	Username
Password	Varchar(20)	No	Password
Role	Varchar(20)	No	Role

3.1.2. Tabel Periklanan

Tabel ini merepresentasikan entitas Periklanan.

Tabel 3.2 Tabel Periklanan.

Field	Data Type	NULL	Description
Media_ID	char(2)	NO	Id dari media promosi (PK)
Media_Iklan	Varchar(20)	No	Media Iklan
Score1	Integer	No	Score utama
Score2	Integer	Yes	Score kedua
Score3	Integer	Yes	Score ketiga

3.1.3. Tabel Elektronik

Tabel ini merepresentasikan entitas Elektronik.

Tabel 3.3 Tabel Elektronik

Field	Data Type	NULL	Description
Elektronik_id	Char(2)	NO	Id media elektronik (PK)
Media_Id	Varchar(2)	No	Id media periklanan (FK)
Media_elektronik	Varchar(20)	No	Nama media elektronik.
Score1	Integer	No	Score utama
Score2	Integer	Yes	Score kedua
Score3	Integer	Yes	Score ketiga

3.1.4. Tabel Internet

Tabel ini merepresentasikan entitas internet.

Tabel 3.4 Tabel Internet

Field	Data Type	NULL	Description
Situs_id	Char(3)	NO	Id situs internet (PK)
Elektronik_id	Char(2)	No	Id media elektronik (FK)

Situs_target	Varchar(20)	No	Nama situs
Harga_Spot_Iklan	Integer	No	Tarif iklan.
Segmen_Utama	Varchar(20)	No	Segmen target utama
Segmen_Kedua	Varchar(20)	Yes	Segmen target kedua
Segmen_Ketiga	Varchar(20)	Yes	Segmen target ketiga
Score1	Integer	No	Score utama
Score2	Integer	Yes	Score kedua
Score3	Integer	Yes	Score ketiga

3.1.5. Tabel Televisi

Tabel ini merepresentasikan entitas Televisi.

Tabel 3.5 Tabel Televisi

Field	Data Type	NULL	Description
Televisi_id	Char(3)	NO	Id televisi (PK)
Elektronik_id	Char(2)	No	Id media elektronik (FK)
Nama	Varchar(30)	No	Nama stasiun televisi
Lokasi_divre	Varchar (50)	No	Lokasi divre
Lokasi_kandatel	Varchar (50)	No	Lokasi kandatel
Lokasi	Varchar (50)	Yes	Lokasi
Segmen_Utama	Varchar(20)	No	Segmen target utama
Segmen_Kedua	Integer	Yes	Segmen target kedua
Segmen_Ketiga	Integer	Yes	Segmen target ketiga
Agensi	Integer	Yes	Agensi media periklanan
No_Telp	Varchar(20)	Yes	Nomor telepon
Harga_Spot_Iklan	integer	No	Tarif Iklan
Coverage_area	Varchar(30)	No	Jangkauan siar
Score1	Integer	No	Score utama
Score2	Integer	Yes	Score kedua
Score3	Integer	Yes	Score ketiga

3.1.6. Tabel Radio

Tabel ini merepresentasikan entitas Radio.

Tabel 3.6 Tabel Radio

Field	Data Type	NULL	Description
-------	-----------	------	-------------

Radio_id	Char(3)	NO	Id televisi (PK)
Elektronik_id	Char(2)	No	Id media elektronik(FK)
Nama	Varchar(30)	No	Nama stasiun televisi
Lokasi_divre	Varchar (50)	No	Lokasi divre
Lokasi_kandatel	Varchar (50)	No	Lokasi kandatel
Lokasi	Varchar (50)	Yes	Lokasi
Segmen_Utama	Varchar(20)	No	Segmen target utama
Segmen_Kedua	Varchar(20)	Yes	Segmen target kedua
Segmen_Ketiga	Varchar(20)	Yes	Segmen target ketiga
Frekuensi	Varchar(10)	Yes	Frekuensi radio
Agensi	Varchar(30)	Yes	Agensi media periklanan
No_Telp	Varchar(20)	Yes	Nomor telepon
Harga_Spot_Iklan	integer	No	Tarif Iklan
Coverage_area	Varchar(30)	No	Jangkauan siar
Coverage_pulau	Varchar(20)	No	Jangkauan siar pulau
Coverage_propinsi	Varchar(20)	No	Jangkauan siar propinsi
Score1	Integer	No	Score utama
Score2	Integer	Yes	Score kedua
Score3	Integer	Yes	Score ketiga
PKS	Varchar(10)	No	Perjanjian Kerja Sama

3.1.7. Tabel Media_Cetak

Tabel ini merepresentasikan entitas Media_Cetak.

Tabel 3.7 Tabel Media_Cetak

Field	Data Type	NULL	Description
Mcetak_id	Char(2)	NO	Id media cetak (PK)
Media_Id	Char(2)	No	Id media periklanan (FK)
Media_Area	Varchar(20)	No	Media cetak area.
Score1	Integer	No	Score utama
Score2	Integer	Yes	Score kedua
Score3	Integer	Yes	Score ketiga

3.1.8. Tabel MCetak_Area

Tabel ini merepresentasikan entitas MCetak_Area.

Tabel 3.8 Tabel MCetak_Area

Field	Data Type	NULL	Description
Mcetak_area_id	Char(2)	NO	Id media cetak area(PK)
Mcetak_Id	Char(2)	No	Id media cetak (FK)
MCetak_Area_Name	Varchar(20)	No	Nama media cetak area.
Score1	Integer	No	Score utama
Score2	Integer	Yes	Score kedua
Score3	Integer	Yes	Score ketiga

3.1.9. Tabel Banner

Tabel ini merepresentasikan entitas Banner.

Tabel 3.9 Tabel Banner

Field	Data Type	NULL	Description
Banner_id	Char(2)	NO	Id banner(PK)
Mcetak_area_Id	Char(2)	No	Id media cetak area (FK)
Jenis_banner	Varchar(30)	No	Jenis benner.
Ukuran_banner	Varchar(30)	Yes	Ukuran banner
Harga	integer	Yes	Tarif iklan banner
Score	Integer	Yes	Score banner

3.1.10. Tabel Buku_Petunjuk

Tabel ini merepresentasikan entitas Buku_Petunjuk.

Tabel 3.10 Tabel Buku_Petunjuk

Field	Data Type	NULL	Description
Buku_id	Char(2)	NO	Id buku(PK)
Mcetak_area_Id	Char(2)	No	Id media cetak area (FK)
Buku_Produk	Varchar(30)	No	Buku petunjuk produk
Ukuran_buku	Varchar(30)	Yes	Ukuran buku
Harga	integer	Yes	Tarif iklan buku
Score1	Integer	No	Score utama buku
Score2	Integer	Yes	Score kedua buku
Score3	Integer	Yes	Score ketiga buku

3.1.11. Tabel Brosur

Tabel ini merepresentasikan entitas Brosur.

3.1.15. Tabel Billboard

Tabel ini merepresentasikan entitas Billboard.

Tabel 3.15 Tabel Billboard

Field	Data Type	NULL	Description
Baliho_id	Char(2)	NO	Id billboard (PK)
Mcetak_area_Id	Char(2)	No	Id media cetak area(FK)
Lokasi_divre	Varchar (50)	No	Lokasi divre
Lokasi_kandatel	Varchar (50)	No	Lokasi kandatel
Lokasi	Varchar (50)	Yes	Lokasi
Segmen_Utama	Varchar(20)	No	Segmen target utama
Segmen_Kedua	Varchar(20)	Yes	Segmen target kedua
Segmen_Ketiga	Varchar(20)	Yes	Segmen target ketiga
Agensi	Varchar(30)	Yes	Agensi media periklanan
No_Telp	Varchar(20)	Yes	Nomor telepon
Harga_Bulan_Meter	integer	No	Tarif Iklan
Score1	Integer	No	Score utama
Score2	Integer	Yes	Score kedua
Score3	Integer	Yes	Score ketiga

3.1.16. Tabel Branding

Tabel ini merepresentasikan entitas Branding.

Tabel 3.16 Tabel Branding

Field	Data Type	NULL	Description
Branding_id	Char(2)	NO	Id branding (PK)
Mcetak_area_Id	Char(2)	No	Id media cetak area(FK)
Lokasi_divre	Varchar (50)	No	Lokasi divre
Lokasi_kandatel	Varchar (50)	No	Lokasi kandatel
Lokasi	Varchar (50)	Yes	Lokasi
Segmen_Utama	Varchar(20)	No	Segmen target utama
Segmen_Kedua	Varchar(20)	Yes	Segmen target kedua
Segmen_Ketiga	Varchar(20)	Yes	Segmen target ketiga
Agensi	Varchar(30)	Yes	Agensi media periklanan
No_Telp	Varchar(20)	Yes	Nomor telepon
Harga_Bulan_Meter	integer	No	Tarif Iklan
Score1	Integer	No	Score utama

Score2	Integer	Yes	Score kedua
Score3	Integer	Yes	Score ketiga

3.1.17. Tabel Neon_Sign

Tabel ini merepresentasikan entitas Neon_Sign.

Tabel 3.17 Tabel Neon_Sign

Field	Data Type	NULL	Description
Neon_sign_id	Char(2)	NO	Id neon sign (PK)
Mcetak_area_Id	Char(2)	No	Id media cetak area(FK)
Lokasi_divre	Varchar (50)	No	Lokasi divre
Lokasi_kandatel	Varchar (50)	No	Lokasi kandatel
Lokasi	Varchar (50)	Yes	Lokasi
Segmen_Utama	Varchar(20)	No	Segmen target utama
Segmen_Kedua	Varchar(20)	Yes	Segmen target kedua
Segmen_Ketiga	Varchar(20)	Yes	Segmen target ketiga
Agensi	Varchar(30)	Yes	Agensi media periklanan
No_Telp	Varchar(20)	Yes	Nomor telepon
Harga_Bulan_Meter	integer	No	Tarif Iklan
Score1	Integer	No	Score utama
Score2	Integer	Yes	Score kedua
Score3	Integer	Yes	Score ketiga

3.1.18. Tabel Spanduk

Tabel ini merepresentasikan entitas Spanduk.

Tabel 3.18 Tabel Spanduk

Field	Data Type	NULL	Description
Spanduk_id	Char(2)	NO	Id spanduk (PK)
Mcetak_area_Id	Char(2)	No	Id media cetak area(FK)
Lokasi_divre	Varchar (50)	No	Lokasi divre
Lokasi_kandatel	Varchar (50)	No	Lokasi kandatel
Lokasi	Varchar (50)	Yes	Lokasi
Segmen_Utama	Varchar(20)	No	Segmen target utama
Segmen_Kedua	Varchar(20)	Yes	Segmen target kedua
Segmen_Ketiga	Varchar(20)	Yes	Segmen target ketiga

Agensi	Varchar(30)	Yes	Agensi media periklanan
No_Telp	Varchar(20)	Yes	Nomor telepon
Harga_Bulan_Meter	integer	No	Tarif Iklan
Score1	Integer	No	Score utama
Score2	Integer	Yes	Score kedua
Score3	Integer	Yes	Score ketiga

3.1.19. Tabel Umbul_Umbul

Tabel ini merepresentasikan entitas Umbul_Umbul.

Tabel 3.19 Tabel Umbul_Umbul

Field	Data Type	NULL	Description
Umbul_id	Char(2)	NO	Id umbul-umbul (PK)
Mcetak_area_Id	Char(2)	No	Id media cetak area (FK)
Lokasi_divre	Varchar (50)	No	Lokasi divre
Lokasi_kandatel	Varchar (50)	No	Lokasi kandatel
Lokasi	Varchar (50)	Yes	Lokasi
Segmen_Utama	Varchar(20)	No	Segmen target utama
Segmen_Kedua	Varchar(20)	Yes	Segmen target kedua
Segmen_Ketiga	Varchar(20)	Yes	Segmen target ketiga
Agensi	Varchar(30)	Yes	Agensi media periklanan
No_Telp	Varchar(20)	Yes	Nomor telepon
Harga_Bulan_Meter	integer	No	Tarif Iklan
Score1	Integer	No	Score utama
Score2	Integer	Yes	Score kedua
Score3	Integer	Yes	Score ketiga

3.1.20. Tabel Divre

Tabel ini merepresentasikan entitas Divre.

Tabel 3.20 Tabel Divre

Field	Data Type	NULL	Description
Divre_id	Char(1)	No	Id Divre (PK)
Divre_name	Varchar(40)	NO	Lokasi Divre

3.1.21. Tabel Kandatel

Tabel ini merepresentasikan entitas kandatel.

Tabel 3.21 Tabel Kandatel

Field	Data Type	NULL	Description
Kandatel_id	Char(3)	No	Id Kandatel (PK)
Divre_id	Char(1)	No	Id Divre (FK)
Kandatel_name	Varchar(50)	NO	Lokasi Kandatel

3.1.22. Tabel Pulau

Tabel ini merepresentasikan entitas Pulau.

Tabel 3.22 Tabel Pulau

Field	Data Type	NULL	Description
Pulau_id	Char(2)	No	Id Pulau (PK)
Pulau_name	Varchar(20)	NO	Nama Pulau

3.1.23. Tabel Propinsi

Tabel ini merepresentasikan entitas Propinsi.

Tabel 3.21 Tabel Propinsi

Field	Data Type	NULL	Description
Propinsi_id	Char(3)	No	Id propinsi (PK)
Pulau_id	Char(2)	No	Id pualu (FK)
Pulau_name	Varchar(30)	NO	Nama propinsi

3.1.24. Tabel Data_Promosi

Tabel ini merepresentasikan entitas Data Promosi.

Tabel 3.24 Tabel Data_Promosi

Field	Data Type	NULL	Description
Promosi_id	Char(2)	NO	Id promosi (PK)
Media_Id	Char(2)	No	Id media cetak area (FK)
Produk	Varchar (30)	No	Produk (FK)
Jenis_Media	Varchar (20)	No	Jenis media
Nama_Media	Varchar (30)	NO	Nama media
Lokasi_divre	Varchar (50)	No	Lokasi divre

Lokasi_kandatel	Varchar (50)	No	Lokasi kandatel
Lokasi	Varchar (50)	Yes	Lokasi
Tanggal_Mulai_Iklan	Date	No	Tanggal mulai promosi iklan
Tanggal_Berakhir_Iklan	Date	NO	Tanggal berakhir promosi iklan
Spot_Iklan	Varchar (20)	No	Spot Iklan
Deskripsi_Spot	Varchar (50)	Yes	Deskripsi spot
Jumlah_Spot	Varchar (3)	Yes	Jumlah spot
Jam_Siar	Varchar (5)	Yes	Jam siar (untuk radio)
Segmen_Utama	Varchar(20)	No	Segmen target utama
Segmen_Kedua	Varchar(20)	Yes	Segmen target kedua
Segmen_Ketiga	Varchar(20)	Yes	Segmen target ketiga
Agensi	Varchar(30)	Yes	Agensi media periklanan
Tarif_Iklan	integer	No	Tarif Iklan
Coverage_area	Varchar(30)	No	Jangkauan area iklan
Coverage_pulau	Varchar(20)	No	Jangkauan pulau
Coverage_propinsi	Varchar(20)	No	Jangkauan propinsi

3.1.25. Tabel Produk

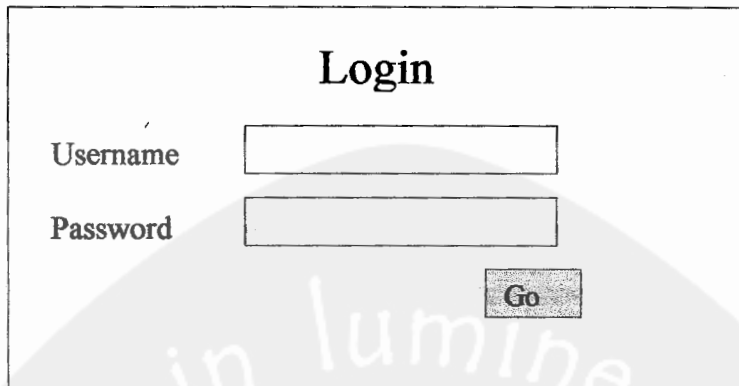
Tabel ini merepresentasikan entitas Produk.

Tabel 3.24 Tabel Produk

Field	Data Type	NULL	Description
Produk_Id	Char(3)	NO	Id
Nama	Varchar(30)	No	Komunikasi periklanan (PK)
Deskripsi	Varchar(200)	No	Deskripsi produk
Segmen_Utama	Varchar(20)	No	Segmen target utama
Segmen_Kedua	Varchar(20)	Yes	Segmen target kedua
Segmen_Ketiga	Varchar(20)	Yes	Segmen target ketiga
Tanggal_Launching	date	No	Tanggal launching produk

4. Deskripsi Perancangan Antarmuka

4.1. Use Case Login



The image shows a simple login interface. At the top center, the word "Login" is displayed in a large, bold font. Below it, there are two text input fields. The first is labeled "Username" and the second is labeled "Password". To the right of the "Password" field is a button labeled "Go". The entire interface is enclosed in a rectangular border.

Gambar 4.1 Rancangan antarmuka Login

Deskripsi

- Rancangan antarmuka ini diimplementasikan pada use case Login.
- TextBox pertama digunakan untuk menerima masukkan Username.
- TextBox kedua digunakan untuk menerima masukkan Password.
- Tombol Go digunakan untuk aksi login user.

Event

- Login

```
onClick_Login()  
{  
    SQL Statement :  
    SELECT * FROM User_TAB  
    WHERE Username = '[txtUserName]'  
    AND Password = '[txtPassword]'  
}
```

Urutan aksi yang terjadi :

1. User memasukkan username dan password dan role pada tempat yang telah tersedia.
2. User menekan tombol Go.
3. Sistem akan memeriksa apakah username ada pada database. Jika tidak ada maka ditampilkan pesan kesalahan. Jika ada, maka passwordnya akan dibandingkan dengan password user serta rolenya dalam database. Jika tidak sama maka ditampilkan pesan kesalahan.

4.2. Use Case Pengelolaan Data Operator

The screenshot shows a web interface for managing operators. On the left is a sidebar with a [Home](#) link. The top right corner has a [Logout](#) link. The main content area includes a search bar with a **Search** button. Below the search bar is a table with a [Delete](#) link in the first column. Underneath the table is a form with the following fields: Username (text input), Password (text input), Confirm Password (text input), and Role (dropdown menu). At the bottom of the form are **Save** and **Cancel** buttons.

Gambar 4.2 Rancangan antarmuka Data Operator

Deskripsi

- Halaman ini digunakan oleh user untuk melakukan pengelolaan data operator.
- Pada halaman ini terdapat 3 tombol, yaitu search, save dan cancel serta terdapat 3 buah link, yaitu link delete, logout dan home.
 - DataGrid digunakan untuk menampilkan semua data operator ataupun hasil pencarian.
 - Tombol Search digunakan untuk aksi Pencarian Data.
 - Tombol Save digunakan untuk aksi Penambahan operator.
 - Tombol Cancel digunakan untuk aksi reset textbox.
 - Link Edit digunakan untuk aksi Pengeditan operator.
 - Link Delete digunakan untuk aksi Penghapusan operator.
 - Link Home digunakan untuk kembali ke halaman utama.
 - Link Logout digunakan untuk kembali ke halaman login.

Event

- Search Data
onClick_Search()


```

{
    SQL Statement :
    select user_id,username,role from user_web where
    username=' [txtusername] '
}

```

Urutan aksi yang terjadi :

1. Aktor memasukkan nama user yang akan dicari pada textbox yang telah disediakan.
2. Aktor Klik tombol Search.
3. Sistem akan melakukan pengecekan terhadap keyword pencarian, jika data yang sesuai dengan keyword pencarian ada di database, maka data akan ditampilkan pada DataGrid.

- Save Data

```

onClick_Save()
{
    SQL Statement :
    insert into user_web values
    (' [txtUserID] ', ' [txtUsername] ', ' [txtPassword] ', ' [DDRoleName] )
}

```

Urutan aksi yang terjadi :

1. User memasukkan detail data user pada TextBox atau pada DropDownList yang tersedia.
2. User menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang dimasukkan, jika valid maka data tersebut akan ditambahkan ke database. Hasilnya ditampilkan pada DataGrid.

- Cancel

Urutan aksi yang terjadi :

1. User klik tombol cancel.
2. Sistem akan melakukan clear data pada textbox dan dropdownlist yang ada.

- Delete Data

```

onClick_Delete()
{
    SQL Statement :
    delete from user_web where user_id = ' [txtid] '
}

```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan tombol Delete pada item yang akan dihapus.
2. Sistem akan menghapus data tersebut dari database.

- Link Logout

Urutan aksi yang terjadi :

1. User klik Link Logout.
2. Sistem akan menampilkan halaman login.

- Link Home

Urutan aksi yang terjadi :

1. User klik Link Home.
2. Sistem akan menampilkan halaman utama user.

4.3. Use Case Ubah Password

The image shows a web form for changing a password. It consists of three input fields stacked vertically, labeled 'Password Lama', 'Password Baru', and 'Confirm Password'. Below these fields are two buttons: 'Update' and 'Cancel'. The form is enclosed in a rectangular border.

Gambar 4.3 Rancangan antarmuka Ubah Password

Deskripsi

- Halaman ini digunakan oleh user untuk melakukan pengelolaan ubah password.
- Pada halaman ini terdapat 2 tombol, yaitu update dan cancel
 - Tombol update digunakan untuk aksi pengubahan data password user.
 - Tombol Cancel digunakan untuk aksi kembali ke halaman utama user.

Event

- Update Data

```
onClick_Update ()
{
```

```

SQL Statement :
update user_web set password = '[txtnewpass]' where
username='[sessionname]'
}

```

Urutan aksi yang terjadi :

1. User memasukkan password lama, password baru dan konfirmasi password baru pada TextBox yang tersedia.
2. User menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang dimasukkan, jika valid maka data tersebut akan di-update ke database dan akan kembali ke halaman pengelolaan data operator.

- Cancel

Urutan aksi yang terjadi :

1. User klik tombol cancel.
2. Sistem akan melakukan clear data pada textbox dan yang ada dan akan kembali ke halaman pengelolaan data operator.

4.4. Use Case Pengelolaan Data Internet

The screenshot shows a web application interface for internet data management. It features a navigation menu with 'Home' and 'Logout' links. A search bar is present at the top. Below the search bar is a table with 'Edit Delete' buttons. The main form contains the following fields and controls:

- Situs target:
- Harga Spot Iklan:
- Segmen Utama:
- Segmen Utama:
- Segmen Utama:
- 1st Score:
- 2nd Score:
- 3rd Score:
- Save:
- Cancel:

Gambar 4.4 Rancangan antarmuka Pengelolaan Data Internet

Deskripsi

- Halaman ini digunakan oleh user untuk melakukan pengelolaan data Internet.
- Pada halaman ini terdapat 3 tombol, yaitu search, save dan cancel serta terdapat 4 buah link, yaitu link edit, delete, logout dan home.
 - DataGrid digunakan untuk menampilkan semua data internet ataupun hasil pencarian.
 - Tombol Search digunakan untuk aksi Pencarian Data.
 - Tombol Save digunakan untuk aksi Penambahan internet.
 - Tombol Cancel digunakan untuk aksi reset textbox.
 - Link Edit digunakan untuk aksi Pengeditan internet.
 - Link Delete digunakan untuk aksi Penghapusan internet.
 - Link Home digunakan untuk kembali ke halaman utama.
 - Link Logout digunakan untuk kembali ke halaman login.

Event

- Search Data

```
onClick_Search()  
{  
    SQL Statement :  
    select situs_id, situs_target, harga_spot_iklan, segmen_utama,  
    segmen_kedua, segmen_ketiga, score1, score2, score3 from internet  
    where situs_target like '[%txtkey%]'  
}
```

Urutan aksi yang terjadi :

1. Aktor memasukkan situs internet yang akan dicari pada textbox yang telah disediakan.
 2. Aktor Klik tombol Search.
 3. Sistem akan melakukan pengecekan terhadap keyword pencarian, jika data yang sesuai dengan keyword pencarian ada di database, maka data akan ditampilkan pada DataGrid.
- Save Data

```
onClick_Save()  
{  
    SQL Statement :  
    insert into internet values ('[txtSitusID]', '[ElektronikId]',  
    '[txtSitusTarget]', '[txtTarifIklan]', '[DDSTarget1]', '[DDSTarge  
t2]', '[DDSTarget3]', '[txtEyeCatching1]', '[txtEyeCatching2]', '[  
txtEyeCatching3]')  
}
```

Urutan aksi yang terjadi :

1. User memasukkan detail data internet pada TextBox atau pada DropDownList yang tersedia.
2. User menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang dimasukkan, jika valid maka data tersebut akan ditambahkan ke database. Hasilnya ditampilkan pada DataGrid.

- Cancel

Urutan aksi yang terjadi :

1. User klik tombol cancel.
2. Sistem akan melakukan clear data pada textbox dan dropdownlist yang ada.

- Edit Data

```
onClick_Edit()  
{  
    SQL Statement :  
    update internet set situs_target='[txtsitus]',  
    harga_spot_iklan='[txtharga]',segmen_utama='[Ddutama]',segmen_  
    kedua='[Ddkedua]',segmen_ketiga='[Ddketiga]',score1='[txtsc1]'  
    ,score2='[txtsc2]',score3='[txtsc3]' where situs_id='[txtid]'  
}
```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan link Edit pada item yang akan diubah, maka data tersebut akan tertampil pada textbox dan droopdown list yang tersedia.
2. User mengubah data. Kemudian user menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang diubah, jika valid maka sistem akan mengubah data dalam database. Hasilnya ditampilkan pada DataGrid.

- Delete Data

```
onClick_Delete()  
{  
    SQL Statement :  
    delete from internet where situs_id = '[txtid]'  
}
```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan tombol Delete pada item yang akan dihapus.
2. Sistem akan menghapus data tersebut dari database.

- Link Logout

Urutan aksi yang terjadi :

1. User klik Link Logout.
2. Sistem akan menampilkan halaman login.

- Link Home

Urutan aksi yang terjadi :

1. User klik Link Home.
2. Sistem akan menampilkan halaman utama user.

4.5. Use Case Pengelolaan Data Radio

Home Logout

Divre
Kandatel

Edit Delete			
-----------------------------	--	--	--

Nama Radio Coverage Area
Frekuensi Pulau
Devisi Region Propinsi
Kandatel 1st Score
Lokasi 2nd Score
Agensi 3rd Score
No Telp Apakah sudah pernah dilakukan perjanjian kerja sama
Harga Spot
Segmen Utama
Segmen Kedua
Segmen Ketiga

Gambar 4.5 Rancangan antarmuka Pengelolaan Data Radio

Deskripsi

- Halaman ini digunakan oleh user untuk melakukan pengelolaan data radio.

- Pada halaman ini terdapat 3 tombol, yaitu search, save dan cancel serta terdapat 4 buah link, yaitu link edit,delete, logout dan home.
 - DataGrid digunakan untuk menampilkan semua data radio ataupun hasil pencarian.
 - Tombol Search digunakan untuk aksi Pencarian Data.
 - Tombol Save digunakan untuk aksi Penambahan radio.
 - Tombol Cancel digunakan untuk aksi reset textbox.
 - Link Edit digunakan untuk aksi Pengeditan radio.
 - Link Delete digunakan untuk aksi Penghapusan radio.
 - Link Home digunakan untuk kembali ke halaman utama.
 - Link Logout digunakan untuk kembali ke halaman login.

Event

- Search Data

```

onClick_Search()
{
    SQL Statement :
    select radio_id,nama,lokasi_divre,lokasi_kandatel,lokasi,
    segmen_utama,segmen_kedua,segmen_ketiga,frekuensi,agensi,no_telp,
    Harga_spot_iklan,coverage_area,coverage_pulau,coverage_propinsi,
    score1,score2,score3,pks from radio where
    lokasi_divre='[Dddiv]' and lokasi_kandatel='[Ddkand]'
}

```

Urutan aksi yang terjadi :

1. Aktor memilih divre dan kandatel dari data radio yang akan dicari pada dropdownlist yang telah disediakan.
 2. Aktor Klik tombol Search.
 3. Sistem akan melakukan pengecekan terhadap keyword pencarian, sistem akan menampilkan data radio pada DataGrid.
- Save Data

```

onClick_Save()
{
    SQL Statement :
    insert into radio values
    ([txtRadioID]','[txtElektronikID]','[txtNamaRadio]','[DDDivreArea]','[DDKandatelArea]','[txtLokasiKoran]','[DDSTarget1]','[DDSTarget2]','[DDSTarget3]','[txtFrekuensi]','[txtAgensi]','[txtNoTelp]','[txtHargaIklan]','[DDCoveregeArea]','[DDCoveregePulau]','[DDCoveregePropinsi]','[txtEyeCatching1]','[txtEyeCatching2]','[txtEyeCatching3]','[txtPerjanjian]');}

```

Urutan aksi yang terjadi :

1. User memasukkan detail data radio pada TextBox atau pada DropDownList yang tersedia.
2. User menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang dimasukkan, jika valid maka data tersebut akan ditambahkan ke database. Hasilnya ditampilkan pada DataGrid.

- Cancel

Urutan aksi yang terjadi :

1. User klik tombol cancel.
2. Sistem akan melakukan clear data pada textbox dan dropdownlist yang ada.

- Edit Data

```
onClick_Edit()  
{  
    SQL Statement :  
    update radio set nama = '[txtnama]',lokasi_divre=  
    '[DDdiv]',lokasi_kandatel= '[DDkand]',lokasi= '[txtlokasi]',  
    agensi= '[txtagensi]',no_telp= '[txtnotelp]',harga_spot_iklan=  
    '[txtharga]',segmen_utama= '[DDutama]',segmen_kedua=  
    '[DDkedua]',segmen_ketiga= '[Ddketiga]',coverage_area=  
    '[DDarea]',coverage_pulau= '[Ddpulau]',coverage_propinsi=  
    '[Ddprop]',score1= '[txtscore1]',score2= '[txtscore2]',score3=  
    '[txtscore3]',pks= '[txtpks]' where radio_id= '[txtid]'  
}
```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan link Edit pada item yang akan diubah, maka data tersebut akan tertampil pada textbox dan droopdown list yang tersedia.
2. User mengubah data. Kemudian user menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang diubah, jika valid maka sistem akan mengubah data dalam database. Hasilnya ditampilkan pada DataGrid.

- Delete Data

```
onClick_Delete()  
{  
    SQL Statement :  
    delete from radio where radio_id = '[txtid]'  
}
```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan tombol Delete pada item yang akan dihapus.
 2. Sistem akan menghapus data tersebut dari database.
- Link Logout
- Urutan aksi yang terjadi :
1. User klik Link Logout.
 2. Sistem akan menampilkan halaman login.
- Link Home
- Urutan aksi yang terjadi :
1. User klik Link Home.
 2. Sistem akan menampilkan halaman utama user.

4.6. Use Case Pengelolaan Data Televisi

The screenshot shows a web interface for managing TV data. On the left is a sidebar with a 'Home' link. The main content area has a search section with dropdown menus for 'Divre' and 'Kandatel', and a 'Search' button. Below this is a table with columns for 'Edit' and 'Delete' actions. The form below the table includes various input fields: 'Stasiun Televisi', 'Devisi Region' (with a dropdown), 'Kandatel' (with a dropdown), 'Lokasi' (with a dropdown), 'Segmen Utama' (with a dropdown), 'Segmen Kedua' (with a dropdown), 'Segmen Ketiga' (with a dropdown), 'Agensi', 'No Telp', 'Harga Spot Iklan', 'Coverage Area' (with a dropdown), '1st Score', '2nd Score', and '3rd Score'. At the bottom right are 'Save' and 'Cancel' buttons.

Gambar 4.6 Rancangan antarmuka Pengelolaan Data Televisi

Deskripsi

- Halaman ini digunakan oleh user untuk melakukan pengelolaan data televisi.

- Pada halaman ini terdapat 3 tombol, yaitu search, save dan cancel serta terdapat 4 buah link, yaitu link edit,delete, logout dan home.
 - DataGrid digunakan untuk menampilkan semua data televisi ataupun hasil pencarian.
 - Tombol Search digunakan untuk aksi Pencarian Data.
 - Tombol Save digunakan untuk aksi Penambahan televisi.
 - Tombol Cancel digunakan untuk aksi reset textbox.
 - Link Edit digunakan untuk aksi Pengeditan televisi.
 - Link Delete digunakan untuk aksi Penghapusan televisi.
 - Link Home digunakan untuk kembali ke halaman utama.
 - Link Logout digunakan untuk kembali ke halaman login.

Event

- Search Data

```
onClick_Search()
{
    SQL Statement :
    select televisi_id,nama,lokasi_divre,lokasi_kandatel,lokasi,
    segmen_utama,segmen_kedua,segmen_ketiga,frekuensi,agensi,no_telp,Harga_spot_iklan,coverage_area,coverage_pulau,coverage_propinsi,score1,score2,score3 from televisi where lokasi_divre='[DDdiv]' and lokasi_kandatel='[DDkand]'
```

Urutan aksi yang terjadi :

1. Aktor memilih divre dan kandatel dari data televisi yang akan dicari pada dropdownlist yang telah disediakan.
2. Aktor Klik tombol Search.
3. Sistem akan melakukan pengecekan terhadap keyword pencarian, sistem akan menampilkan data televisi pada DataGrid.

- Save Data

```
onClick_Save()
{
    SQL Statement :
    insert into televisi values ([ID]',
    '[txtElektronikID]', '[txtNamaTelevisi]', '[DDDivreArea]', '[DDKandatelArea]', '[txtLokasiTelevisi]', '[DDSTarget1]', '[DDSTarget2]', '[DDSTarget3]', '[txtAgensi]', '[txtNoTelp]', '[txtHargaIklan]', '[DDCoveregeArea]', '[DDCoveregePulau]', '[DDCoveregePropinsi]', '[txtEyeCatching1]', '[txtEyeCatching2]', '[txtEyeCatching3]',
    ;
}
```

Urutan aksi yang terjadi :

1. User memasukkan detail data televisi pada TextBox atau pada DropDownList yang tersedia.
 2. User menekan Tombol Save.
 3. Sistem akan melakukan pengecekan terhadap data yang dimasukkan, jika valid maka data tersebut akan ditambahkan ke database. Hasilnya ditampilkan pada DataGrid.
- Cancel

Urutan aksi yang terjadi :

1. User klik tombol cancel.
2. Sistem akan melakukan clear data pada textbox dan dropdownlist yang ada.

- Edit Data

```
onClick_Edit()
{
    SQL Statement :
    update televisi set nama = '[txtnama]',lokasi_divre=
    '[DDdiv]',lokasi_kandatel= '[DDkand]',lokasi= '[txtlokasi]',
    agensi= '[txtagensi]',no_telp= '[txtnotelp]',harga_spot_iklan=
    '[txtharga]',segmen_utama= '[DDutama]',segmen_kedua=
    '[DDkedua]',segmen_ketiga= '[Ddketiga]',coverage_area=
    '[DDarea]',coverage_pulau= '[Ddpulau]',coverage_propinsi=
    '[Ddprop]',score1= '[txtscore1]',score2= '[txtscore2]',score3=
    '[txtscore3]' where televisi_id= '[txtid]'
}
```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan link Edit pada item yang akan diubah, maka data tersebut akan tertampil pada textbox dan droopdown list yang tersedia.
2. User mengubah data. Kemudian user menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang diubah, jika valid maka sistem akan mengubah data dalam database. Hasilnya ditampilkan pada DataGrid.

- Delete Data

```
onClick_Delete()
{
    SQL Statement :
    delete from televisi where televisi_id = '[txtid]'
}
```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan tombol Delete pada item yang akan dihapus.

2. Sistem akan menghapus data tersebut dari database.

- Link Logout

Urutan aksi yang terjadi :

1. User klik Link Logout.

2. Sistem akan menampilkan halaman login.

- Link Home

Urutan aksi yang terjadi :

1. User klik Link Home.

2. Sistem akan menampilkan halaman utama user.

4.7. Use Case Pengelolaan Data Banner

The screenshot shows a web interface for banner management. It features a navigation bar with 'Home' and 'Logout' links. Below the navigation bar is a search bar with a 'Search' button. A data grid displays banner information, with 'Edit Delete' links for each row. Below the grid are form fields for 'Jenis Banner', 'Ukuran Banner', 'Harga', and 'Eye Catching Score', each with a corresponding input box. At the bottom of the form are 'Save' and 'Cancel' buttons.

Gambar 4.7 Rancangan antarmuka Pengelolaan Data Banner

Deskripsi

- Halaman ini digunakan oleh user untuk melakukan pengelolaan data banner.
- Pada halaman ini terdapat 3 tombol, yaitu search, save dan cancel serta terdapat 4 buah link, yaitu link edit,delete, logout dan home.
 - DataGrid digunakan untuk menampilkan semua data banner ataupun hasil pencarian.

- Tombol Search digunakan untuk aksi Pencarian Data.
- Tombol Save digunakan untuk aksi Penambahan banner.
- Tombol Cancel digunakan untuk aksi reset textbox.
- Link Edit digunakan untuk aksi Pengeditan banner.
- Link Delete digunakan untuk aksi Penghapusan banner.
- Link Home digunakan untuk kembali ke halaman utama.
- Link Logout digunakan untuk kembali ke halaman login.

Event

- Search Data

```
onClick_Search()
{
    SQL Statement :
    select banner_id,jenis_banner , ukuran_banner ,harga ,score1
    from banner where jenis_banner like ' [%key%]'
```

Urutan aksi yang terjadi :

1. Aktor key search dari jenis banner dari data banner yang akan dicari pada textbox yang telah disediakan.
2. Aktor Klik tombol Search.
3. Sistem akan melakukan pengecekan terhadap keyword pencarian, sistem akan menampilkan data banner pada DataGridView.

- Save Data

```
onClick_Save()
{
    SQL Statement :
    insert into banner values ([ID]','[AreaI]','[txtJenis]','
    '[txtUkuran]','[txtHarga]','[txtEyeCatching1]'
```

Urutan aksi yang terjadi :

1. User memasukkan detail data banner pada TextBox yang tersedia.
2. User menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang dimasukkan, jika valid maka data tersebut akan ditambahkan ke database. Hasilnya ditampilkan pada DataGridView.

- Cancel

Urutan aksi yang terjadi :

1. User klik tombol cancel.

2. Sistem akan melakukan clear data pada textbox yang ada.

- Edit Data

```
onClick_Edit()  
{  
    SQL Statement :  
    update banner set jenis_banner='[txtjns]',  
    ukuran_banner='[txtukrn]', harga='[txthrg]', score1='[txtscore]'  
    where banner_id='[txtid]'  
}
```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan link Edit pada item yang akan diubah, maka data tersebut akan tertampil pada textbox yang tersedia.
2. User mengubah data. Kemudian user menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang diubah, jika valid maka sistem akan mengubah data dalam database. Hasilnya ditampilkan pada DataGrid.

- Delete Data

```
onClick_Delete()  
{  
    SQL Statement :  
    delete from banner where banner_id = '[txtid]'  
}
```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan tombol Delete pada item yang akan dihapus.
2. Sistem akan menghapus data tersebut dari database.

- Link Logout

Urutan aksi yang terjadi :

1. User klik Link Logout.
2. Sistem akan menampilkan halaman login.

- Link Home

Urutan aksi yang terjadi :

1. User klik Link Home.
2. Sistem akan menampilkan halaman utama user.

4.8. Use Case Pengelolaan Data Brosur

The screenshot shows a web application interface for managing brochures. It features a navigation bar with a 'Home' link on the left and a 'Logout' link on the right. Below the navigation bar is a search bar with a 'Search' button. A data grid displays a list of brochures, with 'Edit' and 'Delete' links for each row. Below the data grid is a form for adding or editing a brochure, with fields for 'Jenis Brosur', 'Ukuran Brosur', 'Harga', and 'Eye Catching Score'. The form includes 'Save' and 'Cancel' buttons.

Gambar 4.8 Rancangan antarmuka Pengelolaan Data Brosur

Deskripsi

- Halaman ini digunakan oleh user untuk melakukan pengelolaan data brosur.
- Pada halaman ini terdapat 3 tombol, yaitu search, save dan cancel serta terdapat 4 buah link, yaitu link edit,delete, logout dan home.
 - DataGridView digunakan untuk menampilkan semua data brosur ataupun hasil pencarian.
 - Tombol Search digunakan untuk aksi Pencarian Data.
 - Tombol Save digunakan untuk aksi Penambahan brosur.
 - Tombol Cancel digunakan untuk aksi reset textbox.
 - Link Edit digunakan untuk aksi Pengeditan brosur.
 - Link Delete digunakan untuk aksi Penghapusan brosur.
 - Link Home digunakan untuk kembali ke halaman utama.
 - Link Logout digunakan untuk kembali ke halaman login.

Event

- Search Data
`onClick_Search()`
{

```

SQL Statement :
select brosur_id,jenis_brosur , ukuran_brosur ,harga ,score1
from brosur where jenis_brosur like ' [%key%]'
}

```

Urutan aksi yang terjadi :

1. Aktor key search dari jenis brosur dari data brosur yang akan dicari pada textbox yang telah disediakan.
2. Aktor Klik tombol Search.
3. Sistem akan melakukan pengecekan terhadap keyword pencarian, sistem akan menampilkan data brosur pada DataGridView.

- Save Data

```

onClick_Save()
{
    SQL Statement :
    insert into brosur values ([ID]','[AreaI]','[txtJenis]','
    '[txtUkuran]','[txtHarga]','[txtEyeCatching1]'
}

```

Urutan aksi yang terjadi :

1. User memasukkan detail data banner pada TextBox yang tersedia.
2. User menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang dimasukkan, jika valid maka data tersebut akan ditambahkan ke database. Hasilnya ditampilkan pada DataGridView.

- Cancel

Urutan aksi yang terjadi :

1. User klik tombol cancel.
2. Sistem akan melakukan clear data pada textbox yang ada.

- Edit Data

```

onClick_Edit()
{
    SQL Statement :
    update      brosur      set      jenis_brosur='[txtjns]',
    ukuran_brosur='[txtukrn]',harga='[txthrg]',score1='[txtscore]'
    where brosur_id='[txtid]'
}

```

Urutan aksi yang terjadi :

1. User memilih data pada DataGridView lalu menekan link Edit pada item yang akan diubah, maka data tersebut akan tertampil pada textbox yang tersedia.

2. User mengubah data. Kemudian user menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang diubah, jika valid maka sistem akan mengubah data dalam database. Hasilnya ditampilkan pada DataGrid.

- Delete Data

```
onClick_Delete()
{
    SQL Statement :
    delete from brosur where brosur_id = '[txtid]'
}
```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan tombol Delete pada item yang akan dihapus.
2. Sistem akan menghapus data tersebut dari database.

- Link Logout

Urutan aksi yang terjadi :

1. User klik Link Logout.
2. Sistem akan menampilkan halaman login.

- Link Home

Urutan aksi yang terjadi :

1. User klik Link Home.
2. Sistem akan menampilkan halaman utama user.

4.9. Use Case Pengelolaan Data Buku

		Logout
Home	<input type="text"/>	<input type="button" value="Search"/>
	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
Jenis Buku	<input type="text"/>	
Ukuran Brosur	<input type="text"/>	
Harga	<input type="text"/>	
Eye Catching Score	<input type="text"/>	
	<input type="button" value="Save"/>	<input type="button" value="Cancel"/>

Gambar 4.9 Rancangan antarmuka Pengelolaan Data Buku

Deskripsi

- Halaman ini digunakan oleh user untuk melakukan pengelolaan data Buku Petunjuk.
- Pada halaman ini terdapat 3 tombol, yaitu search, save dan cancel serta terdapat 4 buah link, yaitu link edit,delete, logout dan home.
 - DataGrid digunakan untuk menampilkan semua data buku petunjuk ataupun hasil pencarian.
 - Tombol Search digunakan untuk aksi Pencarian Data.
 - Tombol Save digunakan untuk aksi Penambahan buku petunjuk.
 - Tombol Cancel digunakan untuk aksi reset textbox.
 - Link Edit digunakan untuk aksi Pengeditan buku petunjuk.
 - Link Delete digunakan untuk aksi Penghapusan buku petunjuk.
 - Link Home digunakan untuk kembali ke halaman utama.
 - Link Logout digunakan untuk kembali ke halaman login.

Event

- Search Data

```
onClick_Search()  
{  
    SQL Statement :  
    select buku_id,jenis_buku, ukuran_buku,harga ,score1,  
score2,score3 from buku_petunjuk where jenis_buku like  
    '[%key%]'  
}
```

Urutan aksi yang terjadi :

1. Aktor key search dari jenis brosur dari data buku petunjuk yang akan dicari pada textbox yang telah disediakan.
2. Aktor Klik tombol Search.
3. Sistem akan melakukan pengecekan terhadap keyword pencarian, sistem akan menampilkan data buku petunjuk pada DataGrid.

- Save Data

```
onClick_Save()  
{  
    SQL Statement :  
    insert into buku_petunjuk values ([ID]','[AreaI]','  
    '[txtJenis]', '[txtUkuran]','[txtHarga]','[txtEyeCatching1]'  
}
```

Urutan aksi yang terjadi :

1. User memasukkan detail data buku petunjuk pada TextBox yang tersedia.
2. User menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang dimasukkan, jika valid maka data tersebut akan ditambahkan ke database. Hasilnya ditampilkan pada DataGridView.

- Cancel

Urutan aksi yang terjadi :

1. User klik tombol cancel.
2. Sistem akan melakukan clear data pada textbox yang ada.

- Edit Data

```
onClick_Edit()
{
    SQL Statement :
    update      buku_petunjuk      set      jenis_buku='[txtjns]',
    ukuran_buku='[txtukrn]',harga='[txthrg]',score1='[txtscore]'
    where buku_id='[txtid]'
```

Urutan aksi yang terjadi :

1. User memilih data pada DataGridView lalu menekan link Edit pada item yang akan diubah, maka data tersebut akan tertampil pada textbox yang tersedia.
2. User mengubah data. Kemudian user menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang diubah, jika valid maka sistem akan mengubah data dalam database. Hasilnya ditampilkan pada DataGridView.

- Delete Data

```
onClick_Delete()
{
    SQL Statement :
    delete from buku_petunjuk where buku_id = '[txtid]'
```

Urutan aksi yang terjadi :

1. User memilih data pada DataGridView lalu menekan tombol Delete pada item yang akan dihapus.
2. Sistem akan menghapus data tersebut dari database.

- Link Logout

Urutan aksi yang terjadi :

1. User klik Link Logout.

2. Sistem akan menampilkan halaman login.

- Link Home

Urutan aksi yang terjadi :

1. User klik Link Home.

2. Sistem akan menampilkan halaman utama user.

4.10. Use Case Pengelolaan Data Koran

The screenshot shows a web application interface for managing newspaper data. On the left, there is a sidebar with a 'Home' link. The main content area has a 'Logout' link in the top right corner. Below the sidebar, there are two dropdown menus for 'Divre' and 'Kandatel', followed by a 'Search' button. A data grid is displayed below the search, with 'Edit' and 'Delete' links in the first column. The main form contains several input fields: 'Nama Koran', 'Devisi Region', 'Kandatel', 'Lokasi', 'Agensi', 'No Telp', 'Harga Spot Iklan', 'Segmen Utama', 'Segmen Kedua', 'Segmen Ketiga', 'Coverage Area', 'Pulau', 'Propinsi', '1st Score', '2nd Score', and '3rd Score'. 'Save' and 'Cancel' buttons are located at the bottom right of the form.

Gambar 4.10 Rancangan antarmuka Pengelolaan Data Koran

Deskripsi

- Halaman ini digunakan oleh user untuk melakukan pengelolaan data koran.
- Pada halaman ini terdapat 3 tombol, yaitu search, save dan cancel serta terdapat 4 buah link, yaitu link edit, delete, logout dan home.
 - DataGrid digunakan untuk menampilkan semua data koran ataupun hasil pencarian.

- Tombol Search digunakan untuk aksi Pencarian Data.
- Tombol Save digunakan untuk aksi Penambahan koran.
- Tombol Cancel digunakan untuk aksi reset textbox.
- Link Edit digunakan untuk aksi Pengeditan koran.
- Link Delete digunakan untuk aksi Penghapusan koran.
- Link Home digunakan untuk kembali ke halaman utama.
- Link Logout digunakan untuk kembali ke halaman login.

Event

- Search Data

```
onClick_Search()
{
    SQL Statement :
    select koran_id,nama,lokasi_divre,lokasi_kandatel,lokasi,
    segmen_utama,segmen_kedua,segmen_ketiga,agensi,no_telp,Harga_s
    pot_iklan,coverage_area,coverage_pulau,coverage_propinsi,score
    1,score2,score3 from koran where lokasi_divre='[Dddiv]' and
    lokasi_kandatel='[Ddkand]'
}
```

Urutan aksi yang terjadi :

1. Aktor memilih divre dan kandatel dari data koran yang akan dicari pada dropdownlist yang telah disediakan.
2. Aktor Klik tombol Search.
3. Sistem akan melakukan pengecekan terhadap keyword pencarian, sistem akan menampilkan data koran pada DataGrid.

- Save Data

```
onClick_Save()
{
    SQL Statement :
    insert into koran values ([txtkoranID]',
    '[MCetakID]', '[txtNamakoran]', '[DDDivreArea]', '[DDKandatelArea
    ]', '[txtLokasi]', '[DDSTarget1]', '[DDSTarget2]', '[DDSTarget3]',
    '[txtAgensi]', '[txtNoTelp]', '[txtHargaIklan]', '[DDCoveregeArea
    ]', '[DDCoveregePulau]', '[DDCoveregePropinsi]', '[txtEyeCatching
    1]', '[txtEyeCatching2]', '[txtEyeCatching3]';
}
```

Urutan aksi yang terjadi :

1. User memasukkan detail data koran pada TextBox atau pada DropDownList yang tersedia.
2. User menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang dimasukkan, jika valid maka data tersebut akan ditambahkan ke database. Hasilnya ditampilkan pada DataGrid.

- Cancel

Urutan aksi yang terjadi :

1. User klik tombol cancel.
2. Sistem akan melakukan clear data pada textbox dan dropdownlist yang ada.

- Edit Data

```
onClick_Edit()  
{  
    SQL Statement :  
    update koran set nama = '[txtnama]',lokasi_divre=  
    '[DDdiv]',lokasi_kandatel= '[DDkand]',lokasi= '[txtlokasi]',  
    agensi= '[txtagensi]',no_telp= '[txtnotelp]',harga_spot_iklan=  
    '[txtharga]',segmen_utama= '[DDutama]',segmen_kedua=  
    '[DDkedua]',segmen_ketiga= '[Ddketiga]',coverage_area=  
    '[DDarea]',coverage_pulau= '[Ddpulau]',coverage_propinsi=  
    '[Ddprop]',score1= '[txtscore1]',score2= '[txtscore2]',score3=  
    '[txtscore3]' where koran_id= '[txtid]'  
}
```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan link Edit pada item yang akan diubah, maka data tersebut akan tertampil pada textbox dan droopdown list yang tersedia.
2. User mengubah data. Kemudian user menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang diubah, jika valid maka sistem akan mengubah data dalam database. Hasilnya ditampilkan pada DataGrid.

- Delete Data

```
onClick_Delete()  
{  
    SQL Statement :  
    delete from koran where koran_id = '[txtid]'  
}
```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan tombol Delete pada item yang akan dihapus.
2. Sistem akan menghapus data tersebut dari database.

- Link Logout

Urutan aksi yang terjadi :

1. User klik Link Logout.
2. Sistem akan menampilkan halaman login.

- Link Home

Urutan aksi yang terjadi :

1. User klik Link Home.
2. Sistem akan menampilkan halaman utama user.

4.11. Use Case Pengelolaan Data Majalah

The screenshot shows a web application interface for managing magazine data. It features a navigation bar with 'Home' and 'Logout' links. The main content area includes search filters for 'Divre' and 'Kandatel' with a 'Search' button. Below the filters is a table with 'Edit' and 'Delete' links. The form contains various input fields and dropdown menus for details like 'Nama Majalah', 'Devisi Region', 'Kandatel', 'Lokasi', 'Agensi', 'No Telp', 'Harga Spot Iklan', 'Segmen Utama', 'Segmen Kedua', 'Segmen Ketiga', 'Coverage Area', 'Pulau', 'Propinsi', '1st Score', '2nd Score', and '3rd Score'. 'Save' and 'Cancel' buttons are located at the bottom right of the form.

Gambar 4.11 Rancangan antarmuka Pengelolaan Data Majalah

Deskripsi

- Halaman ini digunakan oleh user untuk melakukan pengelolaan data majalah.
- Pada halaman ini terdapat 3 tombol, yaitu search, save dan cancel serta terdapat 4 buah link, yaitu link edit, delete, logout dan home.
 - DataGrid digunakan untuk menampilkan semua data majalah ataupun hasil pencarian.
 - Tombol Search digunakan untuk aksi Pencarian Data.
 - Tombol Save digunakan untuk aksi Penambahan majalah.
 - Tombol Cancel digunakan untuk aksi reset textbox.
 - Link Edit digunakan untuk aksi Pengeditan majalah.

- Link Delete digunakan untuk aksi Penghapusan majalah.
- Link Home digunakan untuk kembali ke halaman utama.
- Link Logout digunakan untuk kembali ke halaman login.

Event

- Search Data

```
onClick_Search()
{
    SQL Statement :
    select
    majalah_id,majalah_name,lokasi_divre,lokasi_kandatel,lokasi,
    segmen_utama,segmen_kedua,segmen_ketiga,agensi,no_telp,Harga_s
    pot_iklan,coverage_area,coverage_pulau,coverage_propinsi,score
    1,score2,score3 from majalah where lokasi_divre='[Dddiv]' and
    lokasi_kandatel='[Ddkand]'
```

Urutan aksi yang terjadi :

1. Aktor memilih divre dan kandatel dari data majalah yang akan dicari pada dropdownlist yang telah disediakan.
2. Aktor Klik tombol Search.
3. Sistem akan melakukan pengecekan terhadap keyword pencarian, sistem akan menampilkan data majalah pada DataGrid.

- Save Data

```
onClick_Save()
{
    SQL Statement :
    insert into majalah values ([ID]',
    '[MCetakID]', '[txtNama]', '[DDDivreArea]', '[DDKandatelArea]', '[
    txtLokasi]', '[DDSTarget1]', '[DDSTarget2]', '[DDSTarget3]', '[txt
    Agensi]', '[txtNoTelp]', '[txtHargaIklan]', '[DDCoveregeArea]', '[
    DDcoveregePulau]', '[DDCoveregePropinsi]', '[txtEyeCatching1]', '[
    txtEyeCatching2]', '[txtEyeCatching3]');
```

Urutan aksi yang terjadi :

1. User memasukkan detail data koran pada TextBox atau pada DropDownList yang tersedia.
2. User menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang dimasukkan, jika valid maka data tersebut akan ditambahkan ke database. Hasilnya ditampilkan pada DataGrid.

- Cancel

Urutan aksi yang terjadi :

1. User klik tombol cancel.
2. Sistem akan melakukan clear data pada textbox dan dropdownlist yang ada.

- Edit Data

```

onClick_Edit()
{
    SQL Statement :
    update majalah set nama = '[txtnama]',lokasi_divre=
    '[DDdiv]',lokasi_kandatel= '[DDkand]',lokasi= '[txtlokasi]',
    agensi= '[txtagensi]',no_telp= '[txtnotelp]',harga_spot_iklan=
    '[txtharga]',segmen_utama= '[DDutama]',segmen_kedua=
    '[DDkedua]',segmen_ketiga= '[Ddketiga]',coverage_area=
    '[DDarea]',coverage_pulau= '[Ddpulau]',coverage_propinsi=
    '[Ddprop]',score1= '[txtscore1]',score2= '[txtscore2]',score3=
    '[txtscore3]' where majalah_id= '[txtid]'
}

```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan link Edit pada item yang akan diubah, maka data tersebut akan tertampil pada textbox dan droopdown list yang tersedia.
2. User mengubah data. Kemudian user menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang diubah, jika valid maka sistem akan mengubah data dalam database. Hasilnya ditampikan pada DataGrid.

- Delete Data

```

onClick_Delete()
{
    SQL Statement :
    delete from majalah where majalah_id = '[txtid]'
}

```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan tombol Delete pada item yang akan dihapus.
2. Sistem akan menghapus data tersebut dari database.

- Link Logout

Urutan aksi yang terjadi :

1. User klik Link Logout.
2. Sistem akan menampilkan halaman login.

- Link Home

Urutan aksi yang terjadi :

1. User klik Link Home.
2. Sistem akan menampilkan halaman utama user.

4.12. Use Case Pengelolaan Data Poster

The screenshot shows a web application interface for poster management. It features a sidebar with a 'Home' link, a top navigation bar with a 'Logout' link, and a main content area. The main content area includes a search bar with a 'Search' button, a table with 'Edit' and 'Delete' links, and a form with fields for 'Jenis Poster', 'Ukuran Brosur', 'Harga', '1st Score', '2nd Score', and '3rd Score', along with 'Save' and 'Cancel' buttons.

Gambar 4.12 Rancangan antarmuka Pengelolaan Data Poster

Deskripsi

- Halaman ini digunakan oleh user untuk melakukan pengelolaan data Poster.
- Pada halaman ini terdapat 3 tombol, yaitu search, save dan cancel serta terdapat 4 buah link, yaitu link edit,delete, logout dan home.
 - DataGrid digunakan untuk menampilkan semua data poster petunjuk ataupun hasil pencarian.
 - Tombol Search digunakan untuk aksi Pencarian Data.
 - Tombol Save digunakan untuk aksi Penambahan poster.
 - Tombol Cancel digunakan untuk aksi reset textbox.
 - Link Edit digunakan untuk aksi Pengeditan poster.
 - Link Delete digunakan untuk aksi Penghapusan poster.
 - Link Home digunakan untuk kembali ke halaman utama.
 - Link Logout digunakan untuk kembali ke halaman login.

Event

- Search Data

```

onClick_Search()
{
    SQL Statement :
    select poster_id, jenis_poster, ukuran_poster, harga , score1,
    score2, score3 from poster where jenis_poster like ' [%key%]'
}

```

Urutan aksi yang terjadi :

1. Aktor key search dari jenis brosur dari data poster yang akan dicari pada textbox yang telah disediakan.
2. Aktor Klik tombol Search.
3. Sistem akan melakukan pengecekan terhadap keyword pencarian, sistem akan menampilkan data poster pada DataGridView.

- Save Data

```

onClick_Save()
{
    SQL Statement :
    insert into poster values ([ID], '[AreaID]', '[txtJenis]',
    '[txtUkuran]', '[txtHarga]', '[txtEyeCatching1]'
}

```

Urutan aksi yang terjadi :

1. User memasukkan detail data poster pada TextBox yang tersedia.
2. User menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang dimasukkan, jika valid maka data tersebut akan ditambahkan ke database. Hasilnya ditampilkan pada DataGridView.

- Cancel

Urutan aksi yang terjadi :

1. User klik tombol cancel.
2. Sistem akan melakukan clear data pada textbox yang ada.

- Edit Data

```

onClick_Edit()
{
    SQL Statement :
    update poster set jenis_poster='[txtjns]', ukuran_poster=
    '[txtukrn]', harga='[txthrg]', score1='[txtscore]' where
    poster_id='[txtid]'
}

```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan link Edit pada item yang akan diubah, maka data tersebut akan tertampil pada textbox yang tersedia.
2. User mengubah data. Kemudian user menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang diubah, jika valid maka sistem akan mengubah data dalam database. Hasilnya ditampilkan pada DataGrid.

- Delete Data

```
onClick_Delete()  
{  
    SQL Statement :  
    delete from poster where poster_id = '[txtid]'  
}
```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan tombol Delete pada item yang akan dihapus.
2. Sistem akan menghapus data tersebut dari database.

- Link Logout

Urutan aksi yang terjadi :

1. User klik Link Logout.
2. Sistem akan menampilkan halaman login.

- Link Home

Urutan aksi yang terjadi :

1. User klik Link Home.
2. Sistem akan menampilkan halaman utama user.

4.13. Use Case Pengelolaan Data Billboard

The screenshot shows a web application interface for managing billboard data. On the left is a sidebar with a 'Home' link. The top right corner has a 'Logout' link. The main content area includes two dropdown menus for 'Divre' and 'Kandatel', followed by a 'Search' button. Below this is a table with two columns: 'Edit Delete' and an empty column. Further down are several input fields: 'Devisi Region' and 'Kandatel' (dropdowns), 'Lokasi', 'Agensi', 'No Telp', and 'Harga Spot Iklan' (text boxes), and 'Segmen Utama', 'Segmen Kedua', 'Segmen Ketiga', '1st Score', '2nd Score', and '3rd Score' (text boxes). At the bottom right are 'Save' and 'Cancel' buttons.

Gambar 4.13 Rancangan antarmuka Pengelolaan Data Billboard

Deskripsi

- Halaman ini digunakan oleh user untuk melakukan pengelolaan data billboard.
- Pada halaman ini terdapat 3 tombol, yaitu search, save dan cancel serta terdapat 4 buah link, yaitu link edit,delete, logout dan home.
 - DataGrid digunakan untuk menampilkan semua data billboard ataupun hasil pencarian.
 - Tombol Search digunakan untuk aksi Pencarian Data.
 - Tombol Save digunakan untuk aksi Penambahan billboard.
 - Tombol Cancel digunakan untuk aksi reset textbox.
 - Link Edit digunakan untuk aksi Pengeditan billboard.
 - Link Delete digunakan untuk aksi Penghapusan billboard.
 - Link Home digunakan untuk kembali ke halaman utama.
 - Link Logout digunakan untuk kembali ke halaman login.

Event

- Search Data

```
onClick_Search()  
{  
    SQL Statement :  
    select billboard_id, lokasi_divre, lokasi_kandatel, lokasi,  
    segmen_utama, segmen_kedua, segmen_ketiga, agensi, no_telp, Harga_b  
    ulan_meter, score1, score2, score3 from billboard where  
    lokasi_divre='[Dddiv]' and lokasi_kandatel='[Ddkand]'  
}
```

Urutan aksi yang terjadi :

1. Aktor memilih divre dan kandatel dari data billboard yang akan dicari pada dropdownlist yang telah disediakan.
2. Aktor Klik tombol Search.
3. Sistem akan melakukan pengecekan terhadap keyword pencarian, sistem akan menampilkan data billboard pada DataGrid.

- Save Data

```
onClick_Save()  
{  
    SQL Statement :  
    insert into billboard values ([ID]',  
    '[MCetakID]', '[DDDivreArea]', '[DDKandatelArea]', '[txtLokasi]',  
    '[DDSTarget1]', '[DDSTarget2]', '[DDSTarget3]', '[txtAgensi]', '[t  
    xtNoTelp]', '[txtHargaIklan]', '[txtEyeCatching1]', '[txtEyeCatch  
    ing2]', '[txtEyeCatching3]');  
}
```

Urutan aksi yang terjadi :

1. User memasukkan detail data billboard pada TextBox atau pada DropDownList yang tersedia.
2. User menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang dimasukkan, jika valid maka data tersebut akan ditambahkan ke database. Hasilnya ditampilkan pada DataGrid.

- Cancel

Urutan aksi yang terjadi :

1. User klik tombol cancel.
2. Sistem akan melakukan clear data pada textbox dan dropdownlist yang ada.

- Edit Data

```
onClick_Edit()  
{  
    SQL Statement :
```

```

update billboard set lokasi_divre= '[DDdiv]',lokasi_kandatel=
'[DDkand]',lokasi= '[txtlokasi]', agensi=
'[txtagensi]',no_telp= '[txtnotelp]',harga_bulan_meter=
'[txtharga]',segmen_utama= '[DDutama]',segmen_kedua=
'[DDkedua]',segmen_ketiga= '[Ddketiga]',score1=
'[txtscore1]',score2= '[txtscore2]',score3= '[txtscore3]'
where billboard_id= '[txtid]'
}

```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan link Edit pada item yang akan diubah, maka data tersebut akan tertampil pada textbox dan droopdown list yang tersedia.
2. User mengubah data. Kemudian user menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang diubah, jika valid maka sistem akan mengubah data dalam database. Hasilnya ditampilkan pada DataGrid.

- Delete Data

```

onClick_Delete()
{
    SQL Statement :
    delete from billboard where billboard_id = '[txtid]'
}

```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan tombol Delete pada item yang akan dihapus.
2. Sistem akan menghapus data tersebut dari database.

- Link Logout

Urutan aksi yang terjadi :

1. User klik Link Logout.
2. Sistem akan menampilkan halaman login.

- Link Home

Urutan aksi yang terjadi :

1. User klik Link Home.
2. Sistem akan menampilkan halaman utama user.

4.14. Use Case Pengelolaan Data Branding

The screenshot shows a web application interface for data management. It includes a search section with dropdown menus for 'Divre' and 'Kandatel', and a 'Search' button. Below the search is a table with 'Edit' and 'Delete' links. A form section contains various input fields: 'Devisi Region', 'Kandatel', 'Lokasi', 'Agensi', 'No Telp', 'Harga Spot Iklan', 'Segmen Utama', 'Segmen Kedua', 'Segmen Ketiga', '1st Score', '2nd Score', and '3rd Score'. 'Save' and 'Cancel' buttons are at the bottom right. A 'Home' link is on the left and a 'Logout' link is at the top right.

Gambar 4.14 Rancangan antarmuka Pengelolaan Data Branding

Deskripsi

- Halaman ini digunakan oleh user untuk melakukan pengelolaan data branding.
- Pada halaman ini terdapat 3 tombol, yaitu search, save dan cancel serta terdapat 4 buah link, yaitu link edit,delete, logout dan home.
 - DataGrid digunakan untuk menampilkan semua data branding ataupun hasil pencarian.
 - Tombol Search digunakan untuk aksi Pencarian Data.
 - Tombol Save digunakan untuk aksi Penambahan branding.
 - Tombol Cancel digunakan untuk aksi reset textbox.
 - Link Edit digunakan untuk aksi Pengeditan branding.
 - Link Delete digunakan untuk aksi Penghapusan branding.
 - Link Home digunakan untuk kembali ke halaman utama.
 - Link Logout digunakan untuk kembali ke halaman login.

Event

- Search Data

```
onClick_Search()  
{  
    SQL Statement :  
    select branding_id, lokasi_divre, lokasi_kandatel, lokasi,  
    segmen_utama, segmen_kedua, segmen_ketiga, agensi, no_telp, Harga_b  
    ulan_meter, score1, score2, score3 from branding where  
    lokasi_divre='[Dddiv]' and lokasi_kandatel='[Ddkand]'  
}
```

Urutan aksi yang terjadi :

1. Aktor memilih divre dan kandatel dari data branding yang akan dicari pada dropdownlist yang telah disediakan.
2. Aktor Klik tombol Search.
3. Sistem akan melakukan pengecekan terhadap keyword pencarian, sistem akan menampilkan data branding pada DataGrid.

- Save Data

```
onClick_Save()  
{  
    SQL Statement :  
    insert into branding values ([ID] ,  
    '[MCetakID]', '[DDDivreArea]', '[DDKandatelArea]', '[txtLokasi]',  
    '[DDSTarget1]', '[DDSTarget2]', '[DDSTarget3]', '[txtAgensi]', '[t  
    xtNoTelp]', '[txtHargaIklan]', '[txtEyeCatching1]', '[txtEyeCatch  
    ing2]', '[txtEyeCatching3]');  
}
```

Urutan aksi yang terjadi :

1. User memasukkan detail data branding pada TextBox atau pada DropDownList yang tersedia.
2. User menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang dimasukkan, jika valid maka data tersebut akan ditambahkan ke database. Hasilnya ditampilkan pada DataGrid.

- Cancel

Urutan aksi yang terjadi :

1. User klik tombol cancel.
2. Sistem akan melakukan clear data pada textbox dan dropdownlist yang ada.

- Edit Data

```
onClick_Edit()  
{  
    SQL Statement :
```

```

update branding set lokasi_divre= '[DDdiv]',lokasi_kandatel=
'[DDkand]',lokasi= '[txtlokasi]', agensi=
'[txtagensi]',no_telp= '[txtnotelp]',harga_bulan_meter=
'[txtharga]',segmen_utama= '[DDutama]',segmen_kedua=
'[DDkedua]',segmen_ketiga= '[Ddketiga]',score1=
'[txtscore1]',score2= '[txtscore2]',score3= '[txtscore3]'
where branding_id= '[txtid]'
}

```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan link Edit pada item yang akan diubah, maka data tersebut akan tertampil pada textbox dan droopdown list yang tersedia.
2. User mengubah data. Kemudian user menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang diubah, jika valid maka sistem akan mengubah data dalam database. Hasilnya ditampilkan pada DataGrid.

- Delete Data

```

onClick_Delete()
{
    SQL Statement :
    delete from branding where branding_id = '[txtid]'
}

```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan tombol Delete pada item yang akan dihapus.
2. Sistem akan menghapus data tersebut dari database.

- Link Logout

Urutan aksi yang terjadi :

1. User klik Link Logout.
2. Sistem akan menampilkan halaman login.

- Link Home

Urutan aksi yang terjadi :

1. User klik Link Home.
2. Sistem akan menampilkan halaman utama user.

4.15. Use Case Pengelolaan Data Neon Sign

The screenshot shows a web application interface for managing Neon Sign data. It features a sidebar with a 'Home' link, a top navigation bar with a 'Logout' link, and a main content area. The main area includes a search section with dropdown menus for 'Divre' and 'Kandatel', and a 'Search' button. Below the search is a table with columns for 'Edit' and 'Delete'. The main form contains fields for 'Devisi Region', 'Kandatel', 'Lokasi', 'Agensi', 'No Telp', and 'Harga Spot Iklan' on the left, and 'Segmen Utama', 'Segmen Kedua', 'Segmen Ketiga', '1st Score', '2nd Score', and '3rd Score' on the right. 'Save' and 'Cancel' buttons are at the bottom right.

Gambar 4.15 Rancangan antarmuka Pengelolaan Data Neon Sign

Deskripsi

- Halaman ini digunakan oleh user untuk melakukan pengelolaan data neon sign.
- Pada halaman ini terdapat 3 tombol, yaitu search, save dan cancel serta terdapat 4 buah link, yaitu link edit, delete, logout dan home.
 - DataGrid digunakan untuk menampilkan semua data neon sign ataupun hasil pencarian.
 - Tombol Search digunakan untuk aksi Pencarian Data.
 - Tombol Save digunakan untuk aksi Penambahan neon sign.
 - Tombol Cancel digunakan untuk aksi reset textbox.
 - Link Edit digunakan untuk aksi Pengeditan neon sign.
 - Link Delete digunakan untuk aksi Penghapusan neon sign.
 - Link Home digunakan untuk kembali ke halaman utama.
 - Link Logout digunakan untuk kembali ke halaman login.

Event

- Search Data

```
onClick_Search()
{
    SQL Statement :
    select neon_id, lokasi_divre, lokasi_kandatel, lokasi,
    segmen_utama, segmen_kedua, segmen_ketiga, agensi, no_telp, Harga_b
    ulan_meter, score1, score2, score3 from neon_sign where
    lokasi_divre='[Dddiv]' and lokasi_kandatel='[Ddkand]'
```

Urutan aksi yang terjadi :

1. Aktor memilih divre dan kandatel dari data neon sign yang akan dicari pada dropdownlist yang telah disediakan.
2. Aktor Klik tombol Search.
3. Sistem akan melakukan pengecekan terhadap keyword pencarian, sistem akan menampilkan data neon sign pada DataGrid.

- Save Data

```
onClick_Save()
{
    SQL Statement :
    insert into neon_sign values ([ID]',
    '[MCetakID]', '[DDDivreArea]', '[DDKandatelArea]', '[txtLokasi]',
    '[DDSTarget1]', '[DDSTarget2]', '[DDSTarget3]', '[txtAgensi]', '[t
    xtNoTelp]', '[txtHargaIklan]', '[txtEyeCatching1]', '[txtEyeCatch
    ing2]', '[txtEyeCatching3]');
```

Urutan aksi yang terjadi :

1. User memasukkan detail data neon sign pada TextBox atau pada DropDownList yang tersedia.
2. User menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang dimasukkan, jika valid maka data tersebut akan ditambahkan ke database. Hasilnya ditampilkan pada DataGrid.

- Cancel

Urutan aksi yang terjadi :

1. User klik tombol cancel.
2. Sistem akan melakukan clear data pada textbox dan dropdownlist yang ada.

- Edit Data

```
onClick_Edit()
{
    SQL Statement :
```

```

update neon_sign set lokasi_divre= '[DDdiv]',lokasi_kandatel=
'[DDkand]',lokasi= '[txtlokasi]', agensi=
'[txtagensi]',no_telp= '[txtnotelp]',harga_bulan_meter=
'[txtharga]',segmen_utama= '[DDutama]',segmen_kedua=
'[DDkedua]',segmen_ketiga= '[Ddketiga]',score1=
'[txtscore1]',score2= '[txtscore2]',score3= '[txtscore3]'
where neon_id= '[txtid]'
}

```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan link Edit pada item yang akan diubah, maka data tersebut akan tertampil pada textbox dan droopdown list yang tersedia.
2. User mengubah data. Kemudian user menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang diubah, jika valid maka sistem akan mengubah data dalam database. Hasilnya ditampilkan pada DataGrid.

- Delete Data

```

onClick_Delete()
{
    SQL Statement :
    delete from neon_sign where neon_id = '[txtid]'
}

```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan tombol Delete pada item yang akan dihapus.
2. Sistem akan menghapus data tersebut dari database.

- Link Logout

Urutan aksi yang terjadi :

1. User klik Link Logout.
2. Sistem akan menampilkan halaman login.

- Link Home

Urutan aksi yang terjadi :

1. User klik Link Home.
2. Sistem akan menampilkan halaman utama user.

4.16. Use Case Pengelolaan Data Spanduk

Gambar 4.16 Rancangan antarmuka Pengelolaan Data Spanduk

Deskripsi

- Halaman ini digunakan oleh user untuk melakukan pengelolaan data spanduk.
- Pada halaman ini terdapat 3 tombol, yaitu search, save dan cancel serta terdapat 4 buah link, yaitu link edit,delete, logout dan home.
 - DataGridView digunakan untuk menampilkan semua data spanduk ataupun hasil pencarian.
 - Tombol Search digunakan untuk aksi Pencarian Data.
 - Tombol Save digunakan untuk aksi Penambahan spanduk.
 - Tombol Cancel digunakan untuk aksi reset textbox.
 - Link Edit digunakan untuk aksi Pengeditan spanduk.
 - Link Delete digunakan untuk aksi Penghapusan spanduk.
 - Link Home digunakan untuk kembali ke halaman utama.
 - Link Logout digunakan untuk kembali ke halaman login.

Event

- Search Data

```
onClick_Search()
{
```

SQL Statement :

```
select spanduk_id, lokasi_divre, lokasi_kandatel, lokasi,
segmen_utama, segmen_kedua, segmen_ketiga, agensi, no_telp, Harga_b
ulan_meter, score1, score2, score3 from spanduk where
lokasi_divre='[Dddiv]' and lokasi_kandatel='[Ddkand]'
```

}

Urutan aksi yang terjadi :

1. Aktor memilih divre dan kandatel dari data spanduk yang akan dicari pada dropdownlist yang telah disediakan.
2. Aktor Klik tombol Search.
3. Sistem akan melakukan pengecekan terhadap keyword pencarian, sistem akan menampilkan data spanduk pada DataGrid.

- Save Data

onClick_Save()

{

SQL Statement :

```
insert into spanduk values ([ID]',
'MCetakID]', '[DDDivreArea]', '[DDKandatelArea]', '[txtLokasi]',
'[DDSTarget1]', '[DDSTarget2]', '[DDSTarget3]', '[txtAgensi]', '[t
xtNoTelp]', '[txtHargaIklan]', '[txtEyeCatching1]', '[txtEyeCatch
ing2]', '[txtEyeCatching3]');
```

}

Urutan aksi yang terjadi :

1. User memasukkan detail data spanduk pada TextBox atau pada DropDownList yang tersedia.
2. User menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang dimasukkan, jika valid maka data tersebut akan ditambahkan ke database. Hasilnya ditampilkan pada DataGrid.

- Cancel

Urutan aksi yang terjadi :

1. User klik tombol cancel.
2. Sistem akan melakukan clear data pada textbox dan dropdownlist yang ada.

- Edit Data

onClick_Edit()

{

SQL Statement :

```
update spanduk set lokasi_divre= '[DDdiv]', lokasi_kandatel=
'[DDkand]', lokasi= '[txtlokasi]', agensi=
'[txtagensi]', no_telp= '[txtnotelp]', harga_bulan_meter=
'[txtharga]', segmen_utama= '[DDutama]', segmen_kedua=
```

```

'[DDkedua]', segmen_ketiga= '[Ddketiga]', score1=
'[txtscore1]', score2= '[txtscore2]', score3= '[txtscore3]'
where spanduk_id= '[txtid]'
}

```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan link Edit pada item yang akan diubah, maka data tersebut akan tertampil pada textbox dan droopdown list yang tersedia.
2. User mengubah data. Kemudian user menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang diubah, jika valid maka sistem akan mengubah data dalam database. Hasilnya ditampilkan pada DataGrid.

- Delete Data

```

onClick_Delete()
{
    SQL Statement :
    delete from spanduk where spanduk_id = '[txtid]'
}

```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan tombol Delete pada item yang akan dihapus.
2. Sistem akan menghapus data tersebut dari database.

- Link Logout

Urutan aksi yang terjadi :

1. User klik Link Logout.
2. Sistem akan menampilkan halaman login.

- Link Home

Urutan aksi yang terjadi :

1. User klik Link Home.
2. Sistem akan menampilkan halaman utama user.

4.17. Use Case Pengelola Data Umbul-Umbul

The screenshot shows a web application interface for managing Umbul-Umbul data. It features a navigation bar with 'Home' and 'Logout' links. A search section includes dropdown menus for 'Divre' and 'Kandatel', and a 'Search' button. Below the search is a table with an 'Edit Delete' link. The main form contains fields for 'Devisi Region', 'Kandatel', 'Lokasi', 'Agensi', 'No Telp', and 'Harga Spot Iklan' on the left, and 'Segmen Utama', 'Segmen Kedua', 'Segmen Ketiga', '1st Score', '2nd Score', and '3rd Score' on the right. 'Save' and 'Cancel' buttons are at the bottom right.

Gambar 4.17 Rancangan antarmuka Pengelolaan Data Umbul-umbul

Deskripsi

- Halaman ini digunakan oleh user untuk melakukan pengelolaan data umbul-umbul.
- Pada halaman ini terdapat 3 tombol, yaitu search, save dan cancel serta terdapat 4 buah link, yaitu link edit,delete, logout dan home.
 - DataGrid digunakan untuk menampilkan semua data umbul-umbul ataupun hasil pencarian.
 - Tombol Search digunakan untuk aksi Pencarian Data.
 - Tombol Save digunakan untuk aksi Penambahan umbul-umbul.
 - Tombol Cancel digunakan untuk aksi reset textbox.
 - Link Edit digunakan untuk aksi Pengeditan umbul-umbul.
 - Link Delete digunakan untuk aksi Penghapusan umbul-umbul.
 - Link Home digunakan untuk kembali ke halaman utama.
 - Link Logout digunakan untuk kembali ke halaman login.

Event

- Search Data

onClick_Search()

```
{  
    SQL Statement :  
    select umbul_id, lokasi_divre, lokasi_kandatel, lokasi,  
    segmen_utama, segmen_kedua, segmen_ketiga, agensi, no_telp, Harga_b  
    ulan_meter, score1, score2, score3 from umbul_umbul where  
    lokasi_divre=' [Ddiv]' and lokasi_kandatel=' [Ddkand]'  
}
```

Urutan aksi yang terjadi :

1. Aktor memilih divre dan kandatel dari data umbul-umbul yang akan dicari pada dropdownlist yang telah disediakan.
2. Aktor Klik tombol Search.
3. Sistem akan melakukan pengecekan terhadap keyword pencarian, sistem akan menampilkan data umbul-umbul pada DataGridView.

- Save Data

```
onClick_Save()  
{  
    SQL Statement :  
    insert into umbul_umbul values ([ID]',  
    '[MCetakID]', '[DDDivreArea]', '[DDKandatelArea]', '[txtLokasi]',  
    '[DDSTarget1]', '[DDSTarget2]', '[DDSTarget3]', '[txtAgensi]', '[t  
    xtNoTelp]', '[txtHargaIklan]', '[txtEyeCatching1]', '[txtEyeCatching2]', '[txtEyeCatching3]');  
}
```

Urutan aksi yang terjadi :

1. User memasukkan detail data umbul-umbul pada TextBox atau pada DropDownList yang tersedia.
2. User menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang dimasukkan, jika valid maka data tersebut akan ditambahkan ke database. Hasilnya ditampilkan pada DataGridView.

- Cancel

Urutan aksi yang terjadi :

1. User klik tombol cancel.
2. Sistem akan melakukan clear data pada textbox dan dropdownlist yang ada.

- Edit Data

```
onClick_Edit()  
{  
    SQL Statement :  
    update umbul_umbul set lokasi_divre=  
    '[Ddiv]', lokasi_kandatel= '[Ddkand]', lokasi= '[txtlokasi]',
```

```

agensi= '[txtagensi]',no_telp=
'[txtnotelp]',harga_bulan_meter= '[txtharga]',segmen_utama=
'[DDutama]',segmen_kedua= '[DDkedua]',segmen_ketiga=
'[Ddketiga]',score1= '[txtscore1]',score2=
'[txtscore2]',score3= '[txtscore3]' where umbul_id= '[txtid]'
}

```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan link Edit pada item yang akan diubah, maka data tersebut akan tertampil pada textbox dan droopdown list yang tersedia.
2. User mengubah data. Kemudian user menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang diubah, jika valid maka sistem akan mengubah data dalam database. Hasilnya ditampikan pada DataGrid.

- Delete Data

```

onClick_Delete()
{
    SQL Statement :
    delete from umbul_umbul where umbul_id = '[txtid]'
}

```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan tombol Delete pada item yang akan dihapus.
2. Sistem akan menghapus data tersebut dari database.

- Link Logout

Urutan aksi yang terjadi :

1. User klik Link Logout.
2. Sistem akan menampilkan halaman login.

- Link Home

Urutan aksi yang terjadi :

1. User klik Link Home.
2. Sistem akan menampilkan halaman utama user.

4.18. Use Case Pengelola Data Produk

The screenshot shows a web application interface for product data management. It features a navigation bar with 'Home' and 'Logout' links. A search bar is located at the top, with a 'Search' button. Below the search bar is a table with columns for product details and actions. The table has a header row with 'Edit Delete' links. Below the table is a form for adding or editing product data. The form includes a 'Produk' field, a 'No Telp' field, a 'Tanggal Launching' field with three date pickers, and three dropdown menus for 'Segmen Utama', 'Segmen Kedua', and 'Segmen Ketiga'. At the bottom of the form are 'Save' and 'Cancel' buttons.

Gambar 4.18 Rancangan antarmuka Pengelolaan Data Produk

Deskripsi

- Halaman ini digunakan oleh user untuk melakukan pengelolaan data produk.
- Pada halaman ini terdapat 3 tombol, yaitu search, save dan cancel serta terdapat 4 buah link, yaitu link edit, delete, logout dan home.
 - DataGrid digunakan untuk menampilkan semua data produk ataupun hasil pencarian.
 - Tombol Search digunakan untuk aksi Pencarian Data.
 - Tombol Save digunakan untuk aksi Penambahan produk.
 - Tombol Cancel digunakan untuk aksi reset textbox.
 - Link Edit digunakan untuk aksi Pengeditan produk.
 - Link Delete digunakan untuk aksi Penghapusan produk.
 - Link Home digunakan untuk kembali ke halaman utama.

- Link Logout digunakan untuk kembali ke halaman login.

Event

- Search Data

```
onClick_Search()  
{  
    SQL Statement :  
    select produk_id,nama, deskripsi,segment_utama,segment_kedua,  
    segment_ketiga,tanggal_launching from produk where nama like  
    '%[key]%'  
}
```

Urutan aksi yang terjadi :

1. Aktor memasukkan keyword dari nama produk yang akan dicari pada textbox yang telah disediakan.
2. Aktor Klik tombol Search.
3. Sistem akan melakukan pengecekan terhadap keyword pencarian, sistem akan menampilkan data produk pada DataGrid.

- Save Data

```
onClick_Save()  
{  
    SQL Statement :  
    insert          into          produk          values  
    ('[D]', '[txtNamaProduk]', '[txtDeskripsiProduk]', '[DdsegmenUtama]', '[DdsegmenKedua]', '[DdsegmenKetiga]', STR_TO_DATE(''+  
    Launching+', GET_FORMAT(DATE, 'USA'))");  
}
```

Urutan aksi yang terjadi :

1. User memasukkan detail data produk pada TextBox atau pada DropDownList yang tersedia.
2. User menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang dimasukkan, jika valid maka data tersebut akan ditambahkan ke database. Hasilnya ditampilkan pada DataGrid.

- Cancel

Urutan aksi yang terjadi :

1. User klik tombol cancel.
2. Sistem akan melakukan clear data pada textbox dan dropdownlist yang ada.

- Edit Data

```
onClick_Edit()  
{
```

```

SQL Statement :
update produk set
nama='[txtnama]',deskripsi='[txtdesc]',segment_utama='[Ddutama
]',segment_kedua='[Ddkedua]',segment_ketiga='[Ddketiga]',tangg
al_launching=STR_TO_DATE('" + tgl + "',GET_FORMAT(DATE,'USA'))
where produk_id = '[id]';

```

}

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan link Edit pada item yang akan diubah, maka data tersebut akan tertampil pada textbox dan droopdown list yang tersedia.
2. User mengubah data. Kemudian user menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang diubah, jika valid maka sistem akan mengubah data dalam database. Hasilnya ditampikan pada DataGrid.

- Delete Data

```

onClick_Delete()
{
    SQL Statement :
    delete from produk where produk_id = '[txtid]'
}

```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan tombol Delete pada item yang akan dihapus.
2. Sistem akan menghapus data tersebut dari database.

- Link Logout

Urutan aksi yang terjadi :

1. User klik Link Logout.
2. Sistem akan menampilkan halaman login.

- Link Home

Urutan aksi yang terjadi :

1. User klik Link Home.
2. Sistem akan menampilkan halaman utama user.

4.19. Use Case Pengelolaan Data Promosi

Home [Logout](#)

Divre
Kandatel
Produk

Edit			
----------------------	--	--	--

Media Deskripsi Spot
Produk Jumlah Spot
Jenis Media Agensi
Nama Media Tarif Iklan
Devisi Region Coverage Area
Kandatel Pulau
Lokasi Propinsi
Start Date
Stop Date
Spot Iklan

Gambar 4.19 Rancangan antarmuka Pengelolaan Data Promosi

Deskripsi

- Halaman ini digunakan oleh user untuk melakukan pengelolaan data promosi.
- Pada halaman ini terdapat 3 tombol, yaitu search, save dan cancel serta terdapat 3 buah link, yaitu link edit, logout dan home.
 - DataGrid digunakan untuk menampilkan semua data promosi ataupun hasil pencarian.
 - Tombol Search digunakan untuk aksi Pencarian Data.
 - Tombol Save digunakan untuk aksi Penambahan promosi.
 - Tombol Cancel digunakan untuk aksi reset textbox.
 - Link Edit digunakan untuk aksi Pengeditan promosi.
 - Link Home digunakan untuk kembali ke halaman utama.
 - Link Logout digunakan untuk kembali ke halaman login.

Event

- Search Data

```
onClick_Search()  
{  
    SQL Statement :  
    select p.promosi_id,m.media_iklan,p.produk,p.jenis_media,  
    p.nama_media,p.lokasi_divre,p.lokasi_kandatel,p.lokasi,p.Tanggal  
    l_mulai_iklan,p.tanggal_berakhir_iklan,p.spot_iklan,p.deskripsi  
    _spot,p.jumlah_spot,p.agensi,p.tarif_iklan,p.coverage_area,p.co  
    verage_pulau,p.coverage_propinsi,p.jam_siar from data_promosi  
    p, periklanan m where p.media_id=m.media_id and  
    lokasi_divre='{Dddiv}' and lokasi_kandatel='{Ddkand}' and  
    produk='{DDprdk}'  
}
```

Urutan aksi yang terjadi :

1. Aktor memilih divre, kandatel dan produk dari data promosi yang akan dicari pada dropdownlist yang telah disediakan.
2. Aktor Klik tombol Search.
3. Sistem akan melakukan pengecekan terhadap keyword pencarian, sistem akan menampilkan data promosi pada DataGridView.

- Save Data

```
onClick_Save()  
{  
    SQL Statement :  
    insert          into          data_promosi          values  
    ('[ID]','[MID]','[DDProduk]','[DDJenisMedia]','[DDNamaMedia]','  
    '[DDDivreArea]','[DDKandatelArea]','[txtLokasi]',STR_TO_DATE(''  
    "+newpromo.Start+"',GET_FORMAT(DATE,'USA')),STR_TO_DATE(''+new  
    promo.Finish+"',GET_FORMAT(DATE,'USA')),'[txtSpot]','[txtDeskr  
    ipsiSpot]','[txtJumlahSpot]','[txtAgensi]','[txtHargaIklan]','  
    "[DDCoveregeArea]','[DDCoveregePulau]','[DDCoveregePropinsi]','  
    '[txtJamSiar]')";  
}
```

Urutan aksi yang terjadi :

1. User memasukkan detail data promosi pada TextBox atau pada DropDownList yang tersedia.
2. User menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang dimasukkan, jika valid maka data tersebut akan ditambahkan ke database. Hasilnya ditampilkan pada DataGridView.

- Cancel

Urutan aksi yang terjadi :

1. User klik tombol cancel.
2. Sistem akan melakukan clear data pada textbox dan dropdownlist yang ada.

- Edit Data

```
onClick_Edit()  
{  
    SQL Statement :  
    update data_promosi set media_id = '[area]',produk=  
    '[Ddproduk]',jenis_media= '[DdjenisM]',nama_media=  
    '[DdnamaM]',lokasi_divre= '[Dddivre]',lokasi_kandatel=  
    '[Ddkandatel]',lokasi= '[txtlokasi]',tanggal_mulai_iklan=  
    STR_TO_DATE('"+start+"',GET_FORMAT(DATE,'USA')),tanggal_berakh  
    ir_iklan= STR_TO_DATE('"+stop+"',GET_FORMAT(DATE,  
    'USA')),spot_iklan= '[txtspot]',deskripsi_spot=  
    '[txtdeskripsi]',jumlah_spot= '[txtjumlah]',agensi=  
    '[txtagensi]',tarif_iklan= '[txtiklan]',coverage_area=  
    '[Ddcovarea]',coverage_pulau= '[Ddpulau]',coverage_propinsi=  
    '[Ddprop]',jam_siar='[txtjam]' where promosi_id= '[id]';  
}
```

Urutan aksi yang terjadi :

1. User memilih data pada DataGrid lalu menekan link Edit pada item yang akan diubah, maka data tersebut akan tertampil pada textbox dan droopdown list yang tersedia.
2. User mengubah data. Kemudian user menekan Tombol Save.
3. Sistem akan melakukan pengecekan terhadap data yang diubah, jika valid maka sistem akan mengubah data dalam database. Hasilnya ditampilkan pada DataGrid.

- Link Logout

Urutan aksi yang terjadi :

1. User klik Link Logout.
2. Sistem akan menampilkan halaman login.

- Link Home

Urutan aksi yang terjadi :

1. User klik Link Home.
2. Sistem akan menampilkan halaman utama user.

4.20. Use Case SPK Promosi Periklanan

The screenshot shows a web interface for the SPK Promosi Periklanan system. At the top right, there is a **Logout** link. The main form contains the following elements:

- Divre**: A dropdown menu.
- Kandatel**: A dropdown menu.
- Produk**: A dropdown menu.
- Dana Promosi**: A text input field.
- Question**: Five radio buttons, each followed by a dropdown menu.
- Question**: A radio button labeled "Ya" and a radio button labeled "Tidak".
- Get Decision**: A button.
- Reset**: A button.
- DataGrid**: A table with 4 columns and 1 row.

Gambar 4.20 Rancangan antarmuka SPK Promosi Periklanan

Deskripsi

- Halaman ini digunakan oleh user untuk mendapatkan keputusan mengenai promosi periklanan
- Pada halaman ini terdapat 2 tombol, yaitu Get Decision dan Reset, serta link button Logout
 - DataGrid digunakan untuk menampilkan data media periklanan
 - Tombol Get Decision digunakan untuk aksi mendapatkan keputusan periklanan.
 - Tombol reset digunakan untuk aksi reset textbox/dropdownlist.
 - Logout digunakan untuk keluar dari sistem

Event

- Get Decision

Urutan aksi yang terjadi :

1. Aktor memilih divre, kandatel dan produk serta dana promosi pada dropdownlist dan textbox yang telah disediakan.

2. Aktor memilih pertanyaan yang sebagai dasar dalam pengambilan keputusan.

3. Sistem mendapatkan keputusan dari kriteria yang telah diinputkan oleh user.

- Reset

Urutan aksi yang terjadi :

1. Aktor klik tombol reset.

2. semua dropdownlist dan textbox akan kosong, serta datagrid tidak akan ditampilkan.

- Link Logout

Urutan aksi yang terjadi :

1. User klik Link Logout.

2. Sistem akan menampilkan halaman login.