

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan dari pembahasan-pembahasan pada bab-bab sebelumnya, maka dapat ditarik kesimpulan dari Tugas Akhir ini :

1. Sistem Informasi CBIA-DM (CBIADMIS) yang dirancang untuk kelompok CBIA-DM telah berhasil dibuat dan berjalan dengan baik untuk dapat melakukan fungsi yang diperlukan kelompok CBIA-DM.
2. Sistem Informasi CBIA-DM (CBIADMIS) dapat memberikan kontrol yang lebih komunikatif antara pengelola kelompok dengan anggotanya sendiri agar proses CBIA dalam kelompok dapat terlaksana dengan tepat guna.

6.2 Saran

Beberapa saran dan masukan yang dapat disampaikan penulis terhadap pembuatan CBIADMIS ini dimasa yang akan datang adalah :

1. Sistem diharapkan bisa dikembangkan lebih lanjut dengan memaksimalkan proses bisnis pengelolaan yang ada, contohnya pengintegrasian system dengan system rumah sakit untuk keperluan transfer data-data pasien yang diperlukan seperti data diet, terapi, dan hasil check up agar lebih valid adanya.
2. Sistem diharapkan bisa dikembangkan lebih lanjut dengan diberikan fungsi untuk dapat berinteraksi dengan multimedia sesuai dengan

perkembangan teknologi jaman seperti berupa pengelolaan upload foto untuk dokumentasi serta data pribadi.



DAFTAR PUSTAKA

- Anonim, 2008, *Kamus Besar Bahasa Indonesia*, Balai Pustaka, Jakarta.
- Anonim. 2010. *What Are Information Systems?*, <http://www.internationalgraduate.net/information-systems.htm>, diakses tanggal 19 Desember 2010.
- Ariyanto. 2005. *Mengembangkan Aplikasi Sistem Informasi Akademik Berbasis SMS dengan Java*. Jakarta: Penerbit Salemba Infotek.
- Arumdhani, Melati Tri. 2010. Perbedaan dari Daftar Pustaka, Kutipan dan Catatan Kaki. Posted to: <http://arumdhanimelati.blogspot.com/2010/11/daftar-pustaka-daftar-pustaka-adalah.html>
- Avison, D. and Elliot, S., *Scoping the Discipline of Information Systems*, media.wiley.com/product_data/excerpt/80/EHEP0008/EHEP000880.pdf, diakses tanggal 19 Desember 2010.
- Avison, D. E., and Fitzgerald, G., 2003, *Information Systems Development: Methodologies, Techniques and Tools*, 3rd ed, McGraw-Hill, London.
- Bin Ladjamuddin, Al-Bahra. 2004. *Konsep Sistem Basis Data dan Implementasinya*. Yogyakarta: Graha Ilmu.
- Burch, J., and Grudnitski, G., 1986, *Information Systems Theory and Practice*, John Wiley and Sons, New York.
- Deitel, H.M., P.J. Deitel, J. Listfield, T.R. Nieto, C. Yaeger, M. Zlatkina. 2002. *C# How To Program*. New Jersey: Prentice-Hall.
- Gondodiyoto, Sanyoto. 2003. *Audit Sistem Informasi: Pendekatan Konsep*. Jakarta: Media Global Edukasi.
- Hartayu, Th. B. Titien Siwi. 2010. *Impact of CBIA-DM Strategy on Diabetic Patients' Knowledge, Attitude and Practice in Diabetes Self-management, Diabetic*

Pateints' Adherence to Tratment Guideline, and Diabetic Patients' Quality of Life. Tidak diterbitkan. Penang: University Sains Malaysia.

Hartomo, Djoko. 2005. *Sistem Basis Data*. Yogyakarta: Graha Ilmu.

Hirt, Alan. 2007. *Pro SQL Server 2005 High Availability*. New York: Springer-Verlag New York.

Irfiyanda, S. 2009. Analisis dan Implementasi Informasi Pembayaran Rekening Air Berbasis Mobile (Studi Kasus Perusahaan Daerah Air Minum Tirta Kerta Raharja Kab. Tangerang). Tidak diterbitkan. Jakarta: UIN Syarif Hidayatullah.

Nugroho, W. Setyo. 2008. Daftar Pustaka dari Sumber Digital. Posted to <http://wsetyonugroho.wordpress.com/2008/09/04/daftar-pustaka-dari-sumber-digital/>

Pidwirny, M.. 2006. *Fundamentals of Physical Geography, 2nd Edition*. <http://www.physicalgeography.net/fundamentals/4b.html>.

Sinaga, Benyamin L..2004. *Pemrograman Berorientasi Objek dengan Java*. Yogyakarta: Gava Media.

Soegondo, Sidartawan, Ahmad Rudianto, Asman Manaf, dkk. 2006. *Konsensus Pengelolaan dan Pencegahan Diabetes Melitus Tipe 2 di Indonesia 2006*. Jakarta: PB Perkeni.

Wahidin. 2010. *Aplikasi SMS dengan PHP Untuk Orang awam*. Palembang: Maxikom.

Waseso, D.C. Edi. 2010. *Program Aplikasi Pelayanan Resep Obat Menggunakan SMS dan Website*. Tidak diterbitkan. Yogyakarta: Universitas Sanata Dharma Yogyakarta.

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

CBIADMIS

(CBIA-DM Information System)

Untuk:


Kelompok CBIA-DM

Dipersiapkan oleh:

Reinardus Kristyanto / 070705198

**Program Studi Teknik Informatika - Fakultas Teknologi
Industri**

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		DPPL-CBIADMIS		1/88
		Revisi		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh	RK							
Diperik sa oleh	IW, SP							
Disetuj ui oleh	IW, SP							

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1	Pendahuluan.....	5
1.1	Tujuan.....	5
1.2	Ruang Lingkup.....	5
1.3	Definisi dan Akronim.....	6
1.4	Referensi.....	8
2	Perancangan Sistem.....	10
2.1	Perancangan Arsitektur.....	10
2.2	Perancangan Rinci.....	11
2.2.1	Sequence Diagram.....	11
2.2.2	Diagram Kelas.....	50
2.2.3	Deskripsi Kelas.....	51
3	Perancangan Data.....	73
3.1	Dekomposisi Data.....	73
3.1.1	Deskripsi Entitas Data Account.....	73
3.1.2	Deskripsi Entitas Merk.....	73
3.1.3	Deskripsi Entitas Mobil.....	73
3.1.4	Deskripsi RefPerawatan.....	74
3.1.5	Deskripsi Entitas Perawatan.....	74
3.1.6	Deskripsi Entitas SK.....	75
3.2	PDM.....	77
4	Perancangan Antarmuka.....	80
4.1	Form Login (User).....	80
4.2	Form Ubah Password (User).....	81
4.3	Form Utama (User).....	82
4.4	Form Pengelolaan Account (Administrator).....	83
4.5	Form Pengelolaan Merk (Operasional).....	84
4.6	Form Pengelolaan Mobil (Operasional).....	85
4.7	Form Pengelolaan Update Perijinan (Operasional).....	86
4.8	Form Pengelolaan Referensi Perawatan (Operasional).....	87
4.9	Form Pengelolaan Perawatan (Operasional).....	88
4.10	Form Pengelolaan SK (Operasional).....	Error! Bookmark not defined.
4.11	Form Laporan Mobil (Direktur).....	Error! Bookmark not defined.
4.12	Form Laporan STNK (Direktur).....	Error! Bookmark not defined.
4.13	Form Laporan KIR (Direktur).....	Error! Bookmark not defined.
4.14	Form Laporan KP (Direktur).....	Error! Bookmark not defined.
4.15	Form Laporan SK (Direktur).....	Error! Bookmark not defined.
4.16	Form Laporan Perawatan (Direktur).....	Error! Bookmark not defined.

1 Pendahuluan

1.1 Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) bertujuan untuk mendefinisikan perancangan perangkat lunak yang akan dikembangkan. Dokumen DPPL tersebut digunakan oleh pengembang perangkat lunak sebagai acuan untuk implementasi pada tahap selanjutnya.

1.2 Ruang Lingkup

Perangkat Lunak CBIADMIS dikembangkan dengan tujuan untuk:

1. Menangani login.
2. Menangani logout.
3. Menangani pengelolaan data account.
4. Menangani pengelolaan ubah password.
5. Menangani pengelolaan data anggota.
6. Menangani pengelolaan check up.
7. Menangani pengelolaan daily check.
8. Menangani pengelolaan terapi.
9. Menangani pengelolaan diet.
10. Menangani pengelolaan SMS.
11. Menangani pengelolaan pelaporan list yang belum melakukan check-up.
12. Menangani pengelolaan scheduled and alert SMS.
13. Menangani autorespond SMS.
14. Menangani registrasi anggota.
15. Menangani unregistrasi anggota.

Perangkat Lunak ini berbasis desktop. Perangkat Lunak berjalan pada lingkungan dengan platform Windows XP.

Program Studi Teknik Informatika	DPPL – GMIS	5/ 88
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1.3 Definisi dan Akronim

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
SKPL	Merupakan spesifikasi kebutuhan dari perangkat lunak yang akan dikembangkan.
SKPL-CBIADMIS-XXX	Kode yang merepresentasikan kebutuhan pada CBIADMIS (CBIA-DM Information System) dimana XXX merupakan nomor fungsi produk.
CBIADMIS	Perangkat lunak yang menangani pengelolaan kelompok penderita Diabetes Mellitus dengan basis CBIA (CBIA-DM) melalui sms yang berbasis desktop.
CBIA	Metode pembelajaran yang menuntut keaktifan dari si pembelajar.
DM	Singkatan dari Diabetes Mellitus. Nama penyakit peningkatan kadar gula dalam darah yang disebabkan kurangnya insulin.
CBIA-DM	Metode pembelajaran CBIA yang diterapkan untuk para penderita DM.
Gula Darah Puasa	Gula Darah Puasa (Fasting Plasma Glucose) adalah kadar gula darah pada saat puasa. Dengan satuan mg/dL.
GDPP	Gula Darah Post Prandial (Post Prandial Glucose) adalah kadar gula darah pada saat 2 jam sesudah makan. Dengan satuan mg/dL.
GDS	Gula Darah Sewaktu (Random Blood Glucose) adalah kadar gula darah random. Dengan satuan mg/dL.

HbA1C	Kadar gula darah dalam darah. Ditunjukkan dalam prosentase (%).
LDL	Nilai atas kadar lipoprotein dalam darah. Satuan yang digunakan adalah mg/dL.
HDL	Nilai bawah kadar lipoprotein dalam darah. Satuan yang digunakan adalah mg/dL.
Cholesterol Total	Total kolesterol yang terdapat dalam darah. Dengan satuan mg/dL.
Trigliserid	Kadar lemak yang ada dalam darah. Satuan yang digunakan adalah mg/dL.
Ureum	Hasil metabolisme protein, terdapat dalam darah. Satuan yang digunakan adalah mg/dL.
Kreatinin	Merupakan zat racun dalam darah, produk sisa dari perombakan keratin fosfat yang terjadi dalam otot. Satuan yang digunakan mg/dL.

Role	Peran yang disediakan oleh sistem untuk masing-masing user.
Previlege	Wewenang yang dimiliki oleh user yang diberikan oleh sistem.
User	Semua pengguna komputer yang berhubungan langsung dengan sistem.
Administrator	User yang memiliki role sebagai pengelola akun dalam sistem.
Anggota	User yang memiliki role sebagai anggota penderita DM dalam sistem.
Control	User yang memiliki role sebagai pengawas yang melakukan pengawasan terhadap keteraturan check-up anggota serta dapat melakukan pengiriman sms manual peringatan lanjutan.
Desktop	Istilah yang digunakan saat user menggunakan komputer yang tidak memiliki koneksi ke internet.
Account	Istilah bahasa Inggris yang digunakan untuk menyebut kata akun.

1.4 Referensi

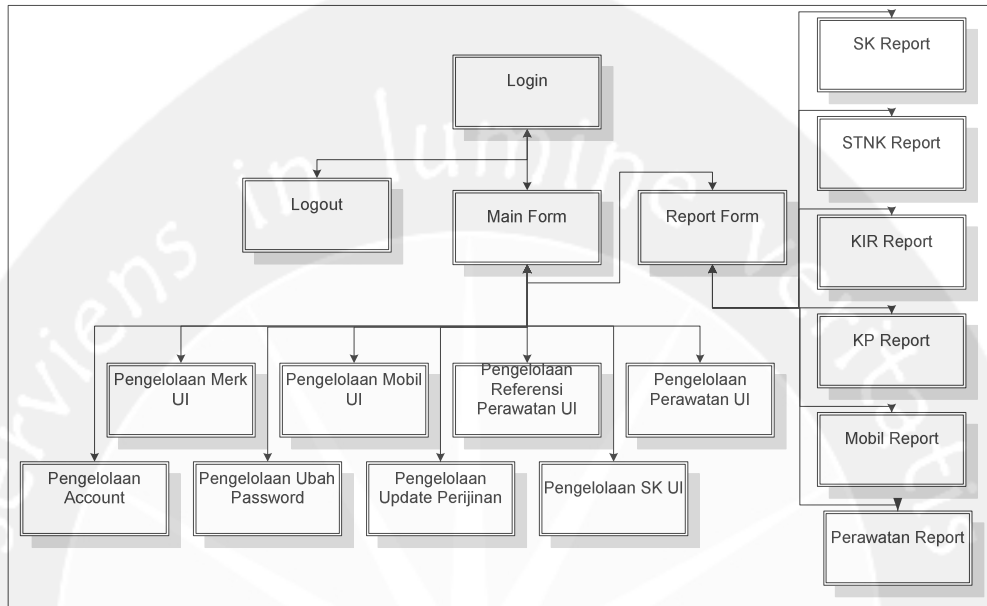
Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Kristyanto Reinardus, *Spesifikasi Kebutuhan Perangkat Lunak CBIADMIS*, Universitas Atma Jaya Yogyakarta, 2011.

2. Kristyanto Reinardus, *Spesifikasi Kebutuhan Perangkat Lunak GMIS*, Universitas Atma Jaya Yogyakarta, 2010.
3. Kristyanto Reinardus, *Deskripsi Perancangan Perangkat Lunak Soft*, Universitas Atma Jaya Yogyakarta, 2010.
4. Kristyanto Reinardus, *Deskripsi Perancangan Perangkat Lunak DC-CRIS*, Universitas Atma Jaya Yogyakarta, 2010.
5. Bennet Simon, McRobb Steve, Farmer Ray, *Object-Oriented System Analysis and Design Using UML*, McGraw-Hill Companies, 2002.
6. Boggs Wendy, Boggs Michael, *Mastering UML with Rational Rose 2002*, SYBEX Inc, 2002.
7. Deitel, *C# How to Program*, Prentice-Hall Inc, 2002.
8. MSDN Library-October 2005, Microsoft, 2005.

2 Perancangan Sistem

2.1 Perancangan Arsitektur

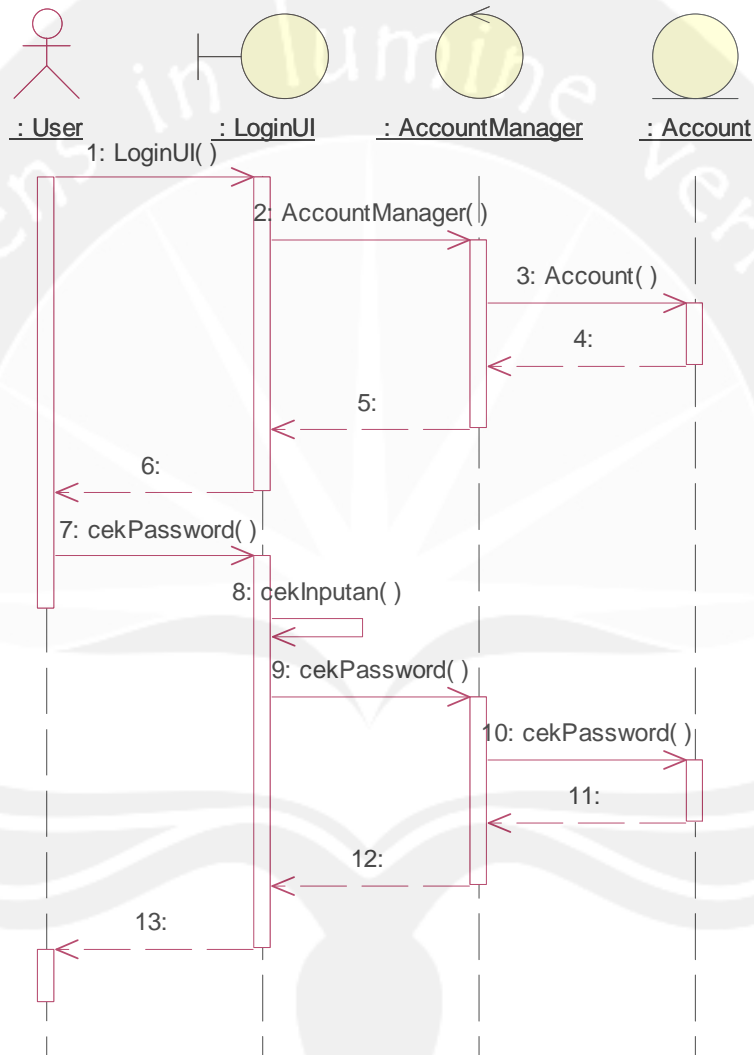


Gambar 1 Rancangan Arsitektur GMIS

2.2 Perancangan Rinci

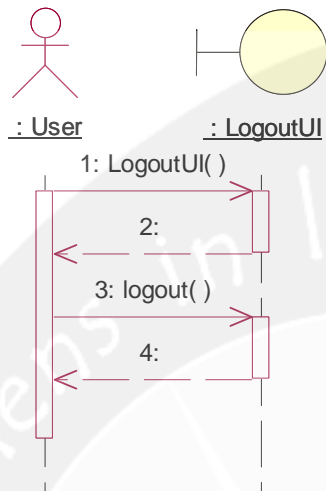
2.2.1 Sequence Diagram

2.2.1.1 Login



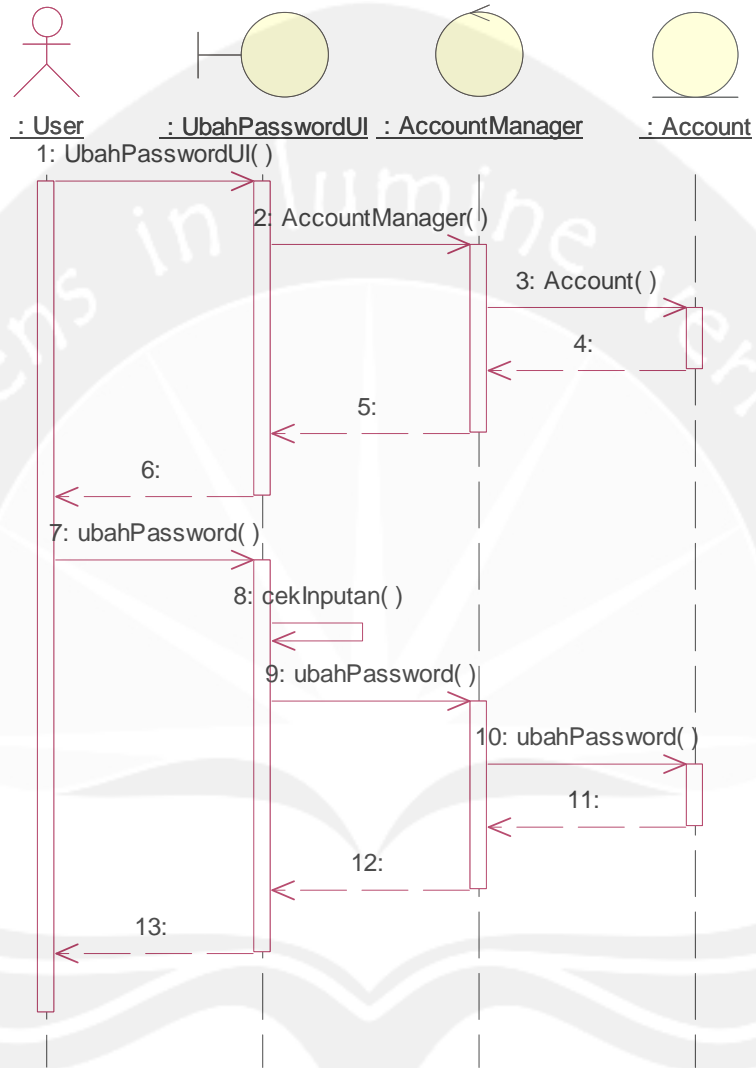
Gambar 2.2.1.1 Sequence Diagram - Login

2.2.1.2 Logout



Gambar 2.2.1.2 Sequence Diagram - Logout

2.2.1.3 Ubah Password

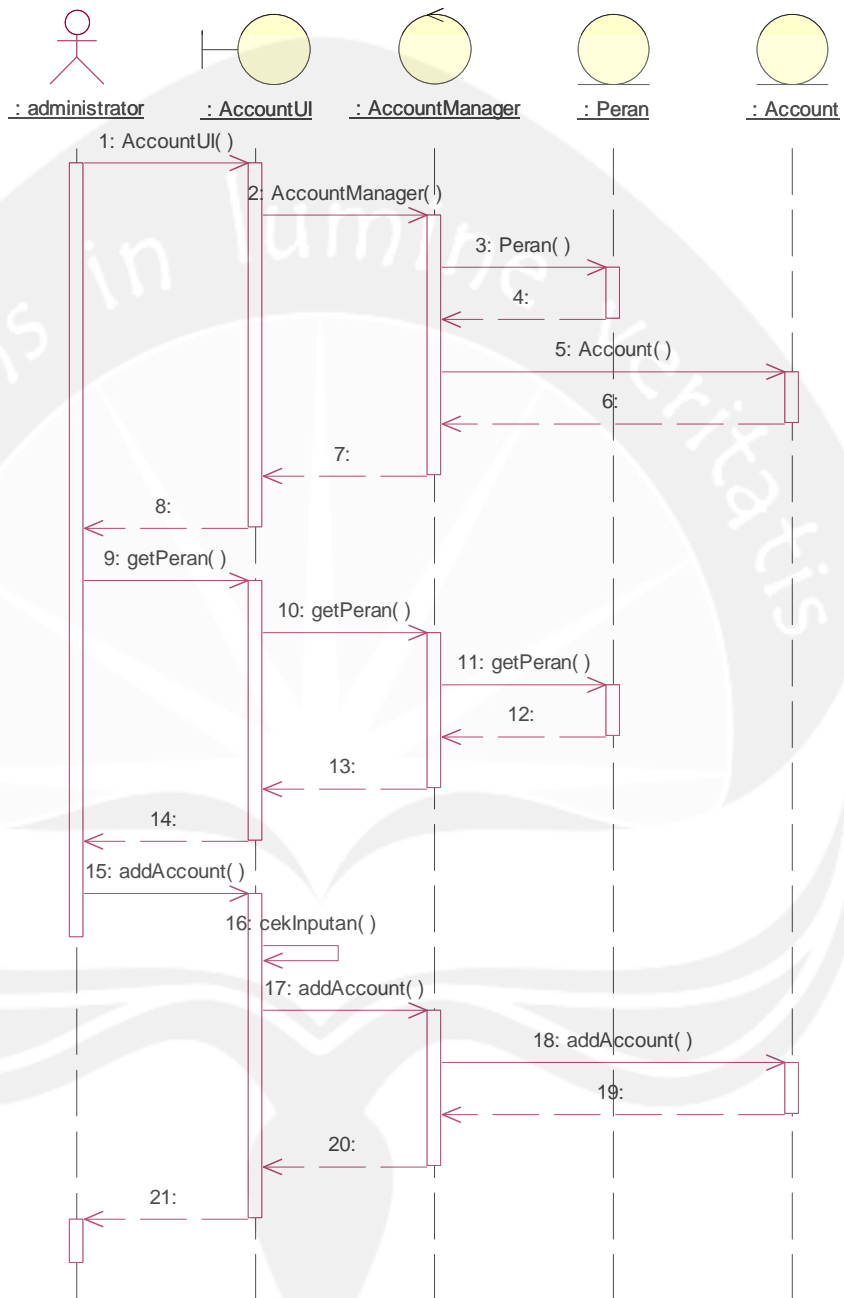


Gambar 2.2.1.3 Sequence Diagram - Ubah Password

2.2.1.4 Pengelolaan Account

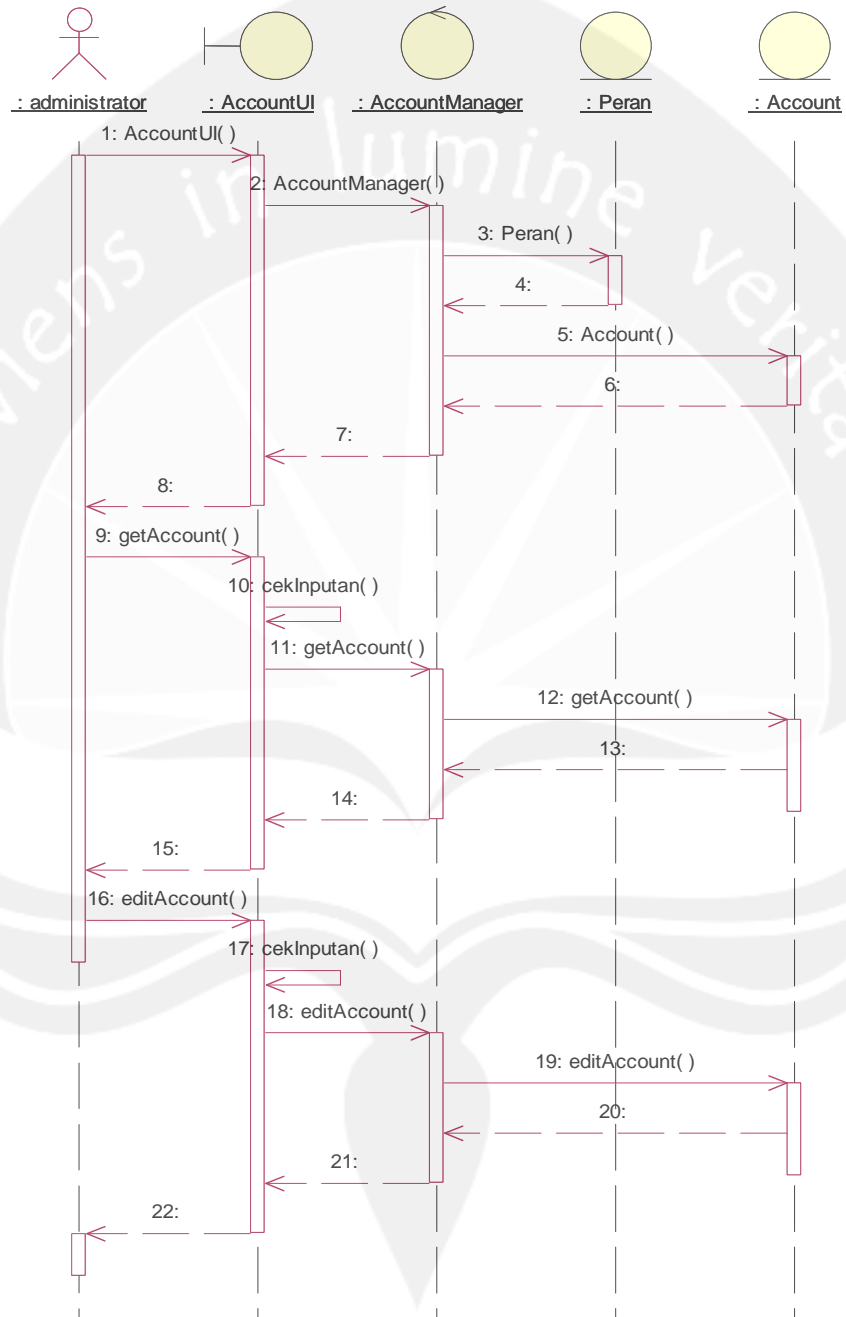
2.2.1.4.1 Add Data





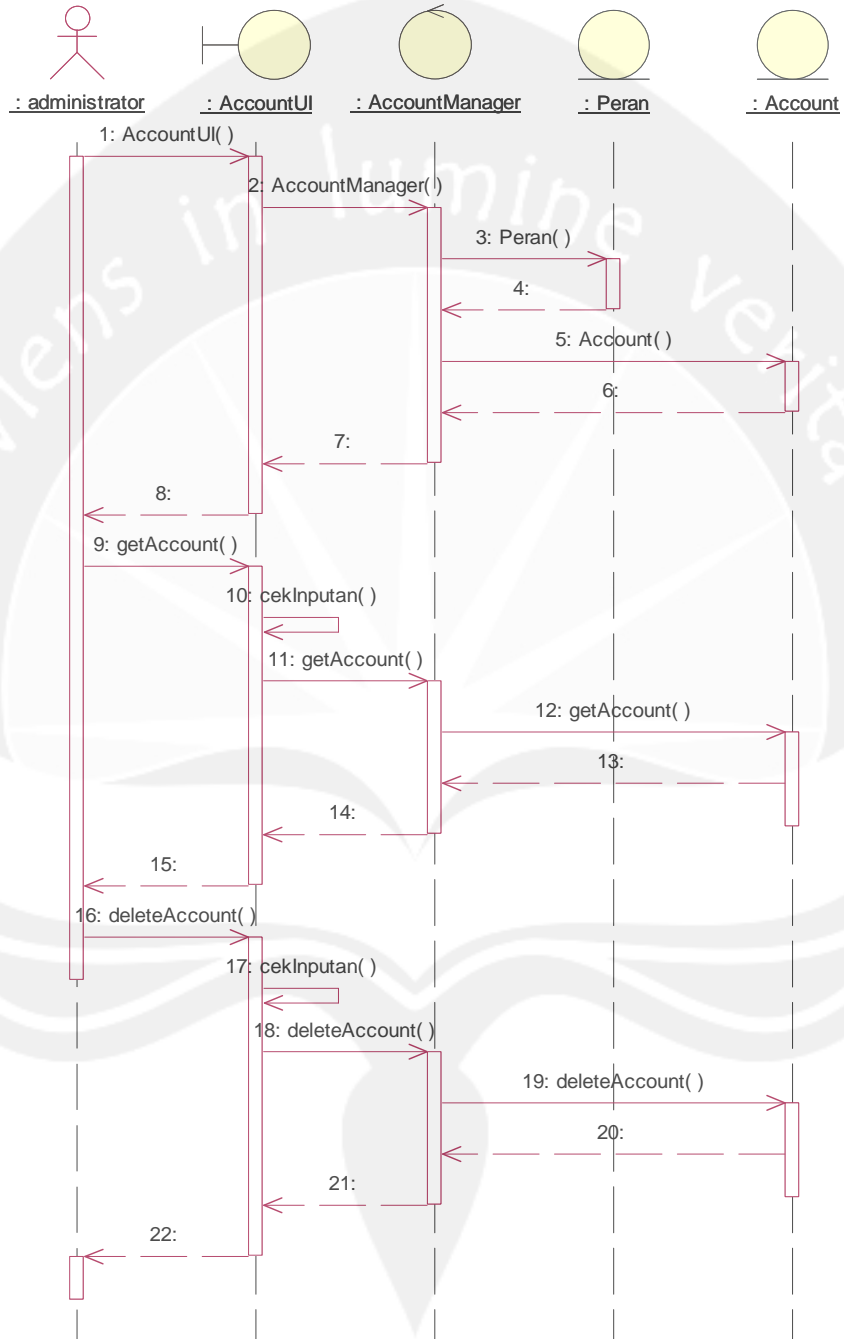
Gambar 2.2.1.4.1 Sequence Diagram - Add Data Account

2.2.1.4.2 Edit Data



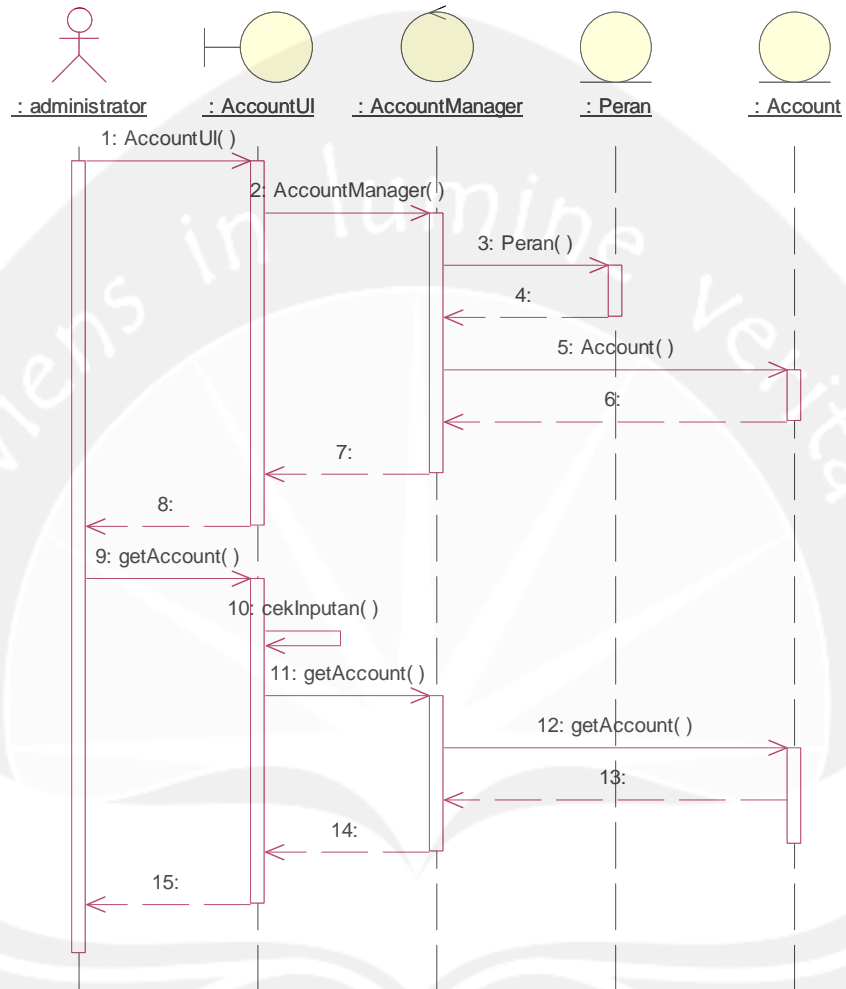
Gambar 2.2.1.4.2 Sequence Diagram - Edit Data Account

2.2.1.4.3 Delete Data



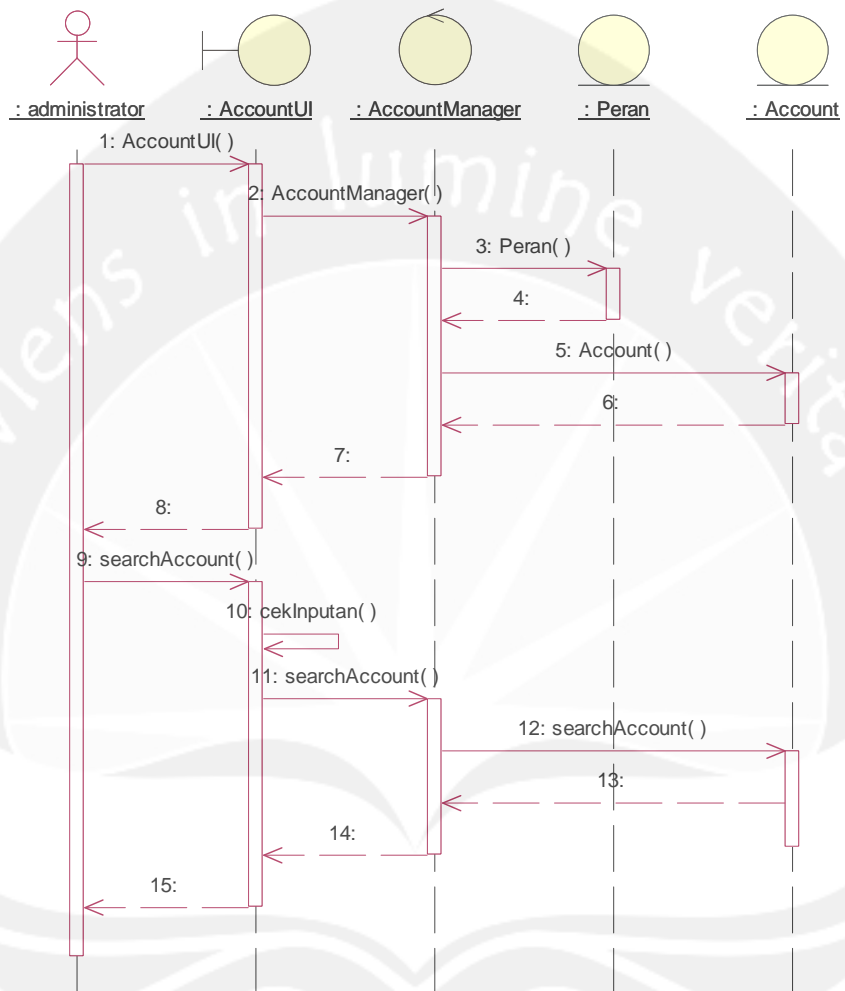
Gambar 2.2.1.4.3 Sequence Diagram - Delete Data Account

2.2.1.4.4 Display Data



Gambar 2.2.1.4.4 Sequence Diagram - Display Data Account

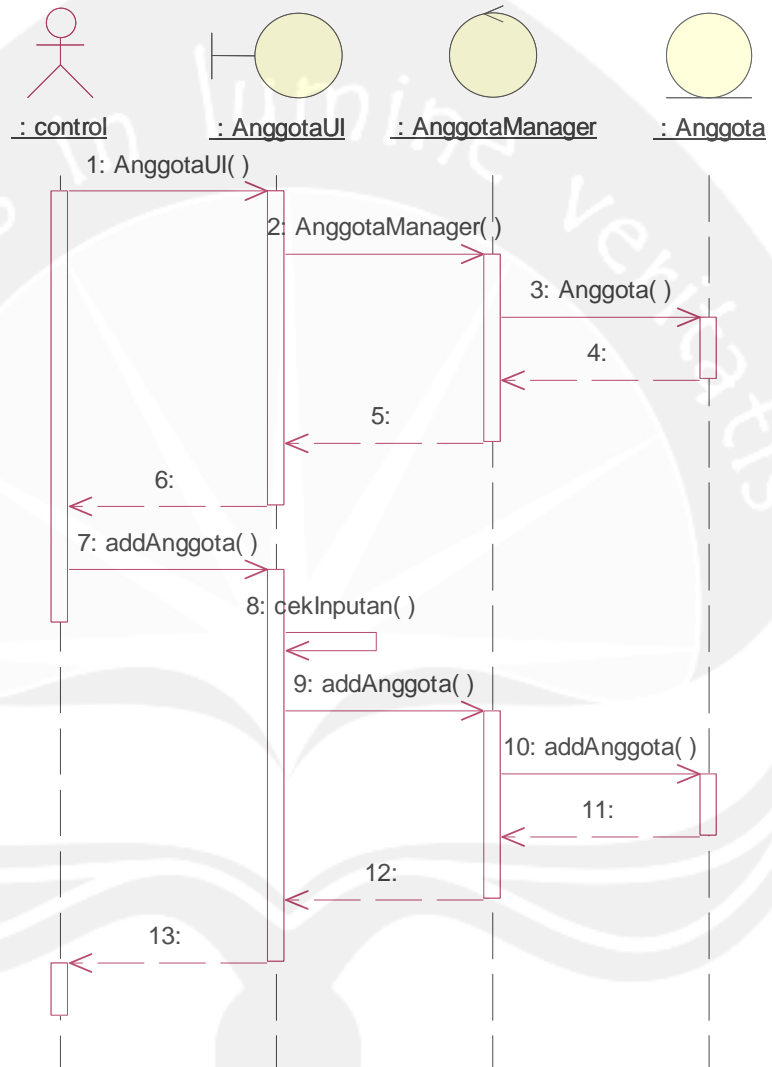
2.2.1.4.5 Search Data



Gambar 2.2.1.4.5 Sequence Diagram - Search Data Account

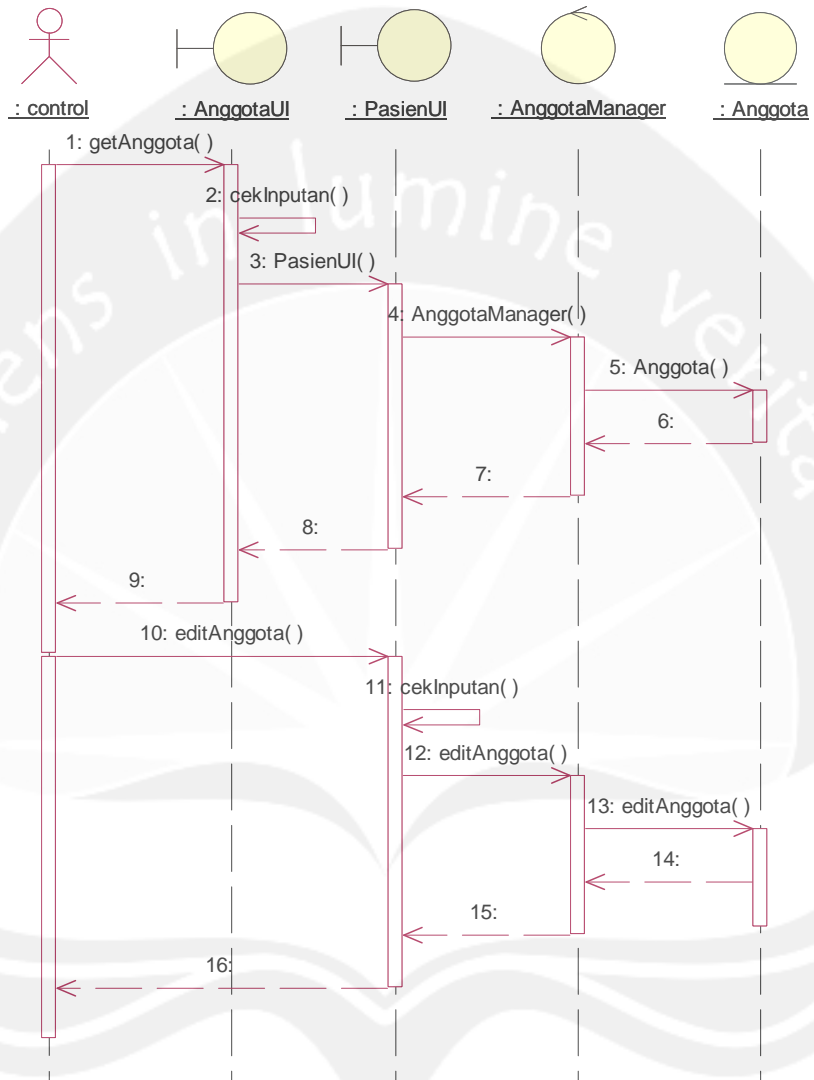
2.2.1.5 Pengelolaan Anggota

2.2.1.5.1 Add Data



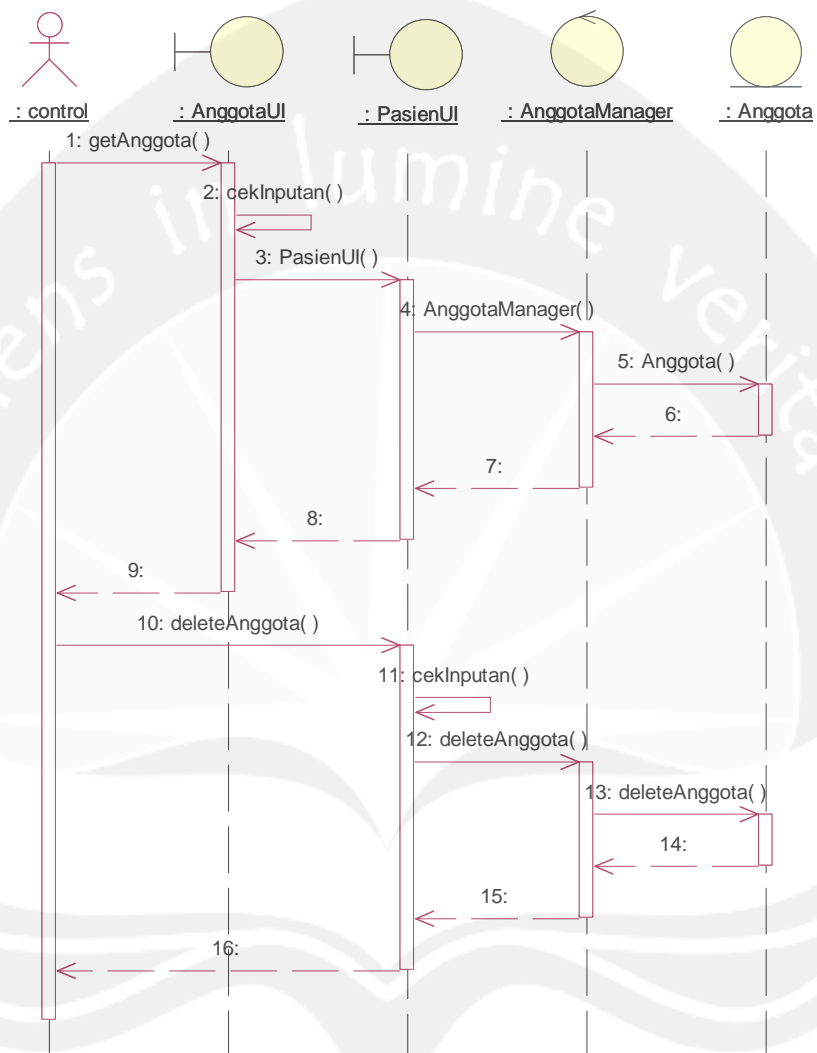
Gambar 2.2.1.5.1 Sequence Diagram - Add Data Anggota

2.2.1.5.2 Edit Data



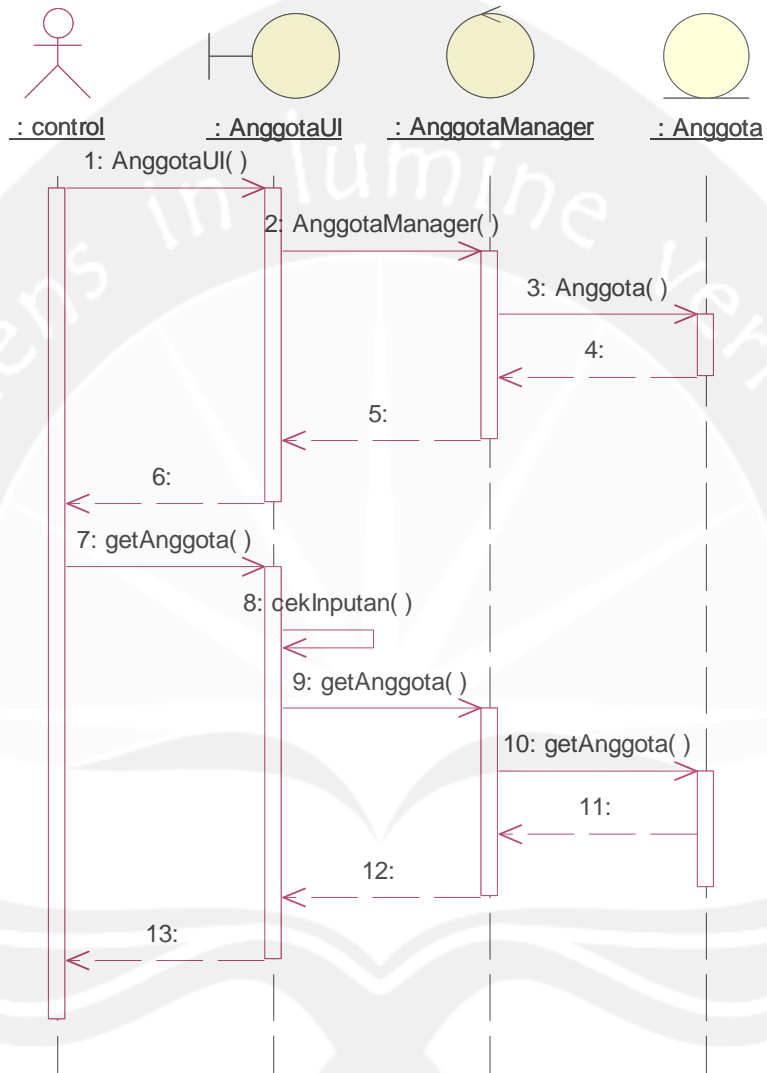
Gambar 2.2.1.5.2 Sequence Diagram - Edit Data Anggota

2.2.1.5.3 Delete Data



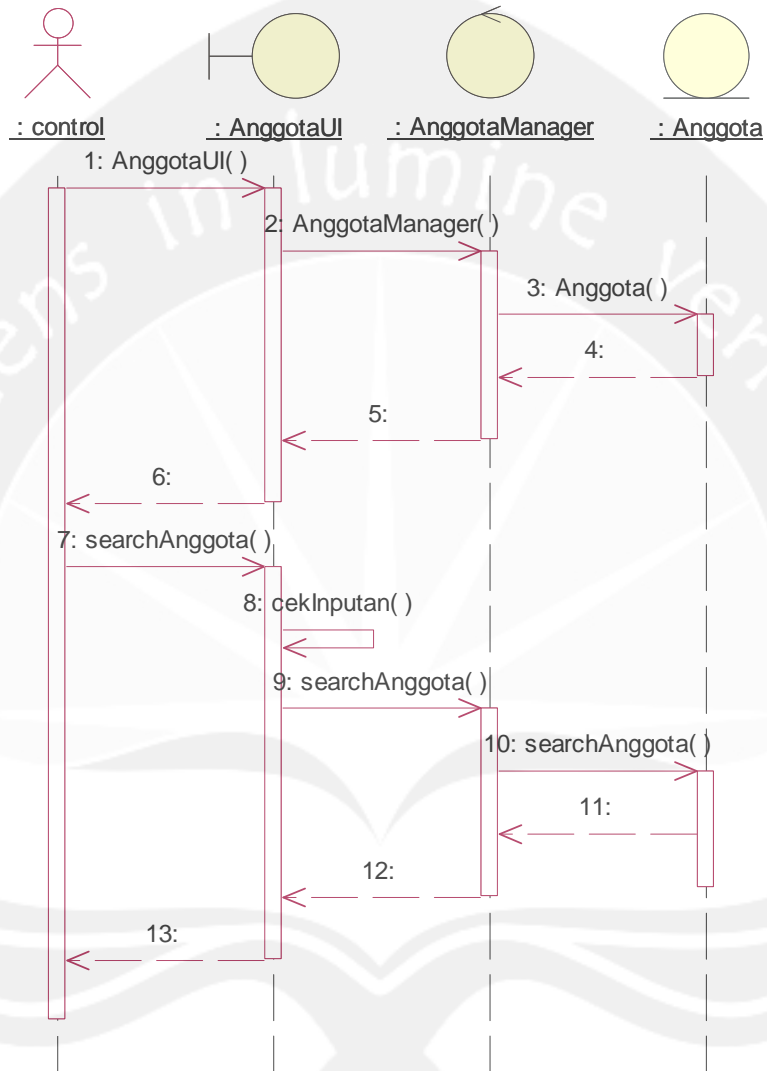
Gambar 2.2.1.5.3 Sequence Diagram - Delete Data Anggota

2.2.1.5.4 Display Data



Gambar 2.2.1.5.4 Sequence Diagram - Display Data Anggota

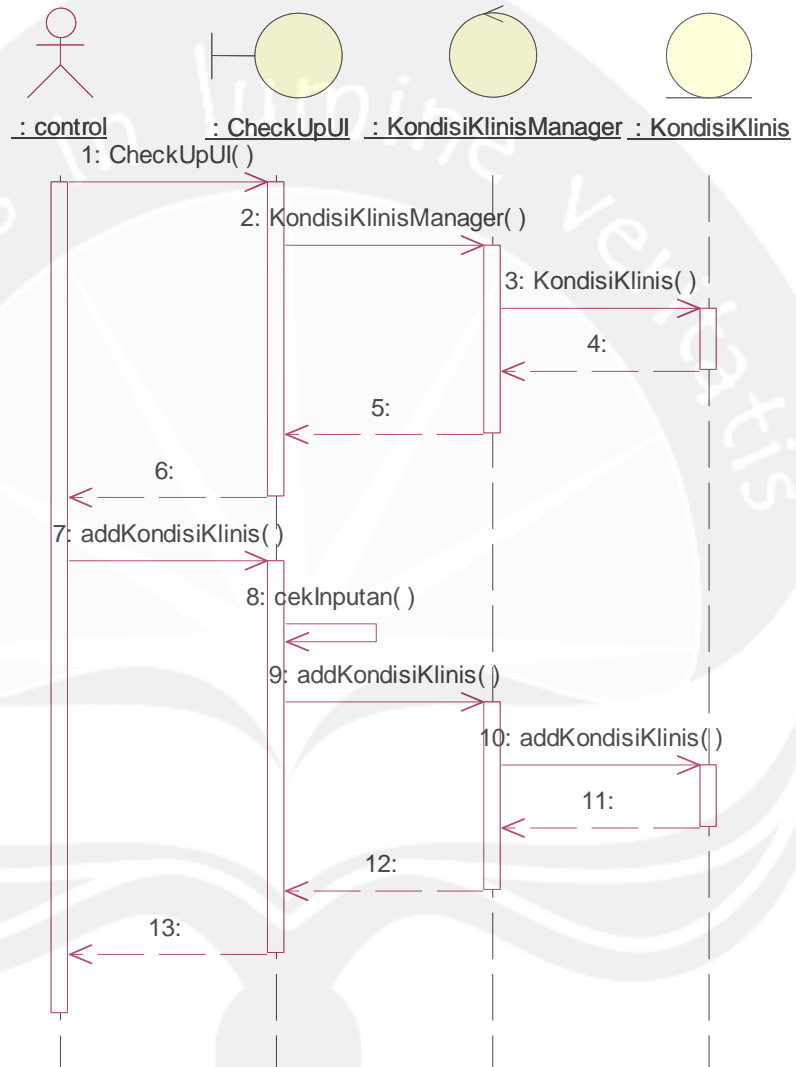
2.2.1.5.5 Search Data



Gambar 2.2.1.5.5 Sequence Diagram - Search Data Anggota

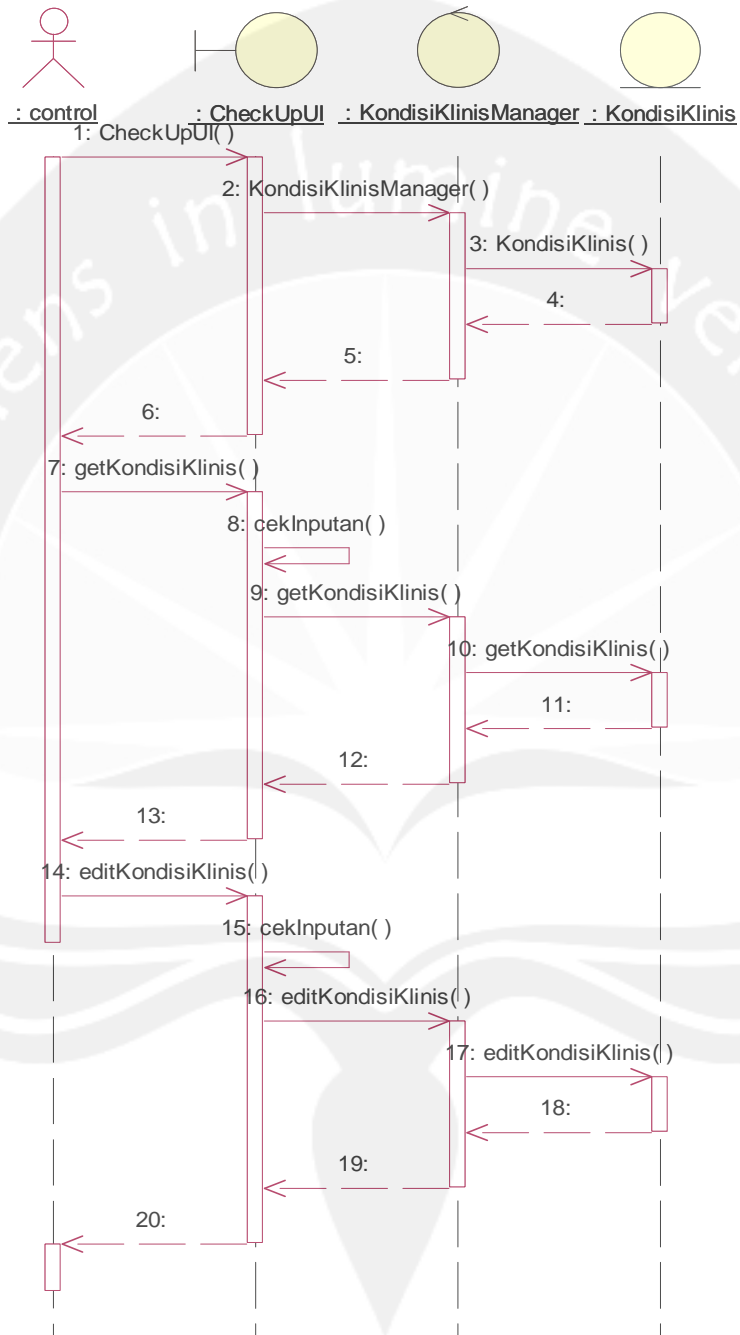
2.2.1.6 Pengelolaan Check Up

2.2.1.6.1 Add Data



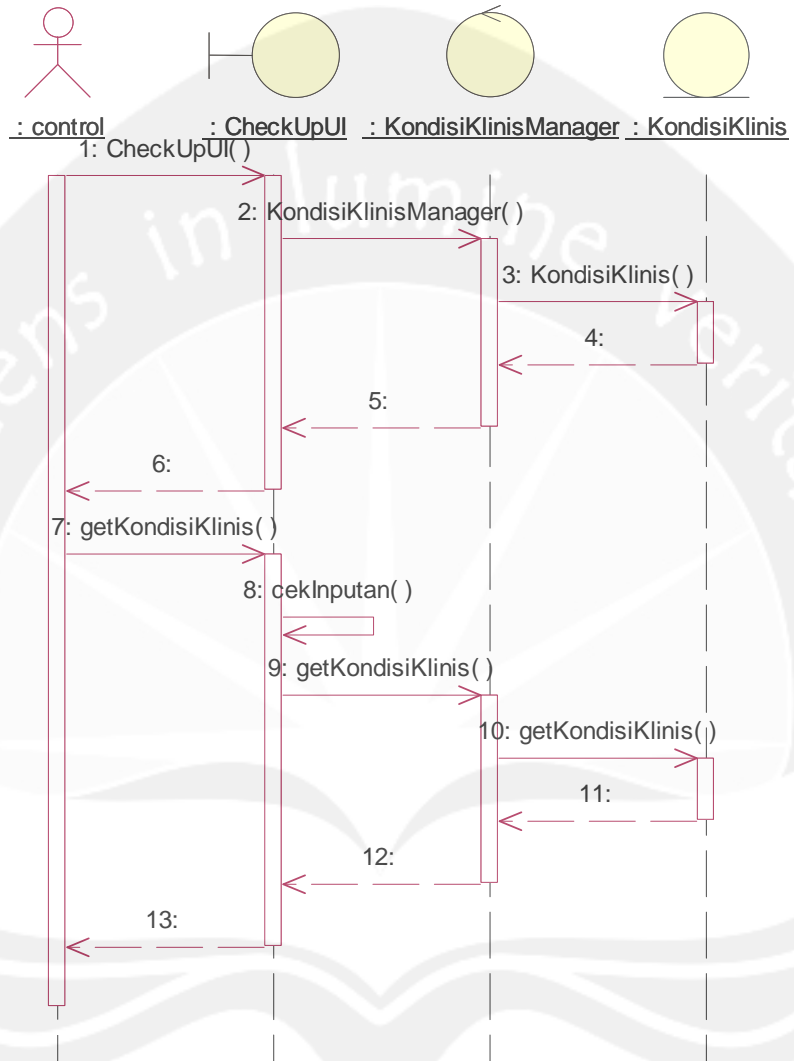
Gambar 2.2.1.6.1 Sequence Diagram - Add Data Check Up

2.2.1.6.2 Edit Data



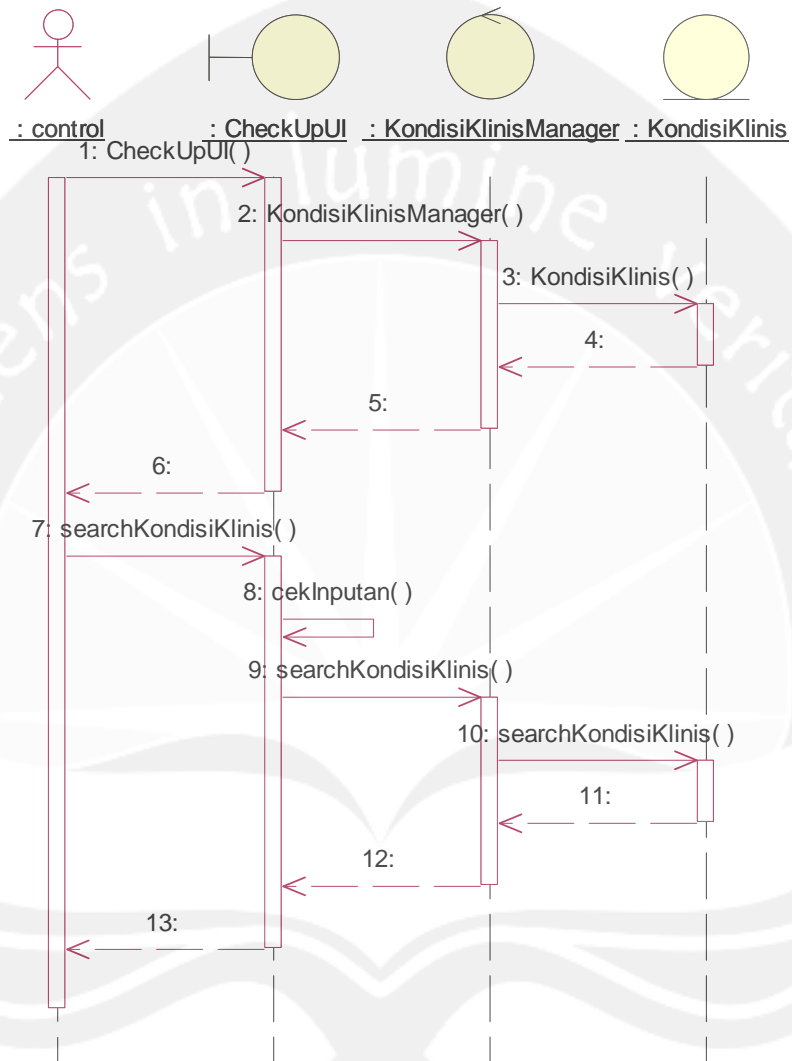
Gambar 2.2.1.6.2 Sequence Diagram - Edit Data Check Up

2.2.1.6.3 Display Data



Gambar 2.2.1.6.3 Sequence Diagram - Display Data Check Up

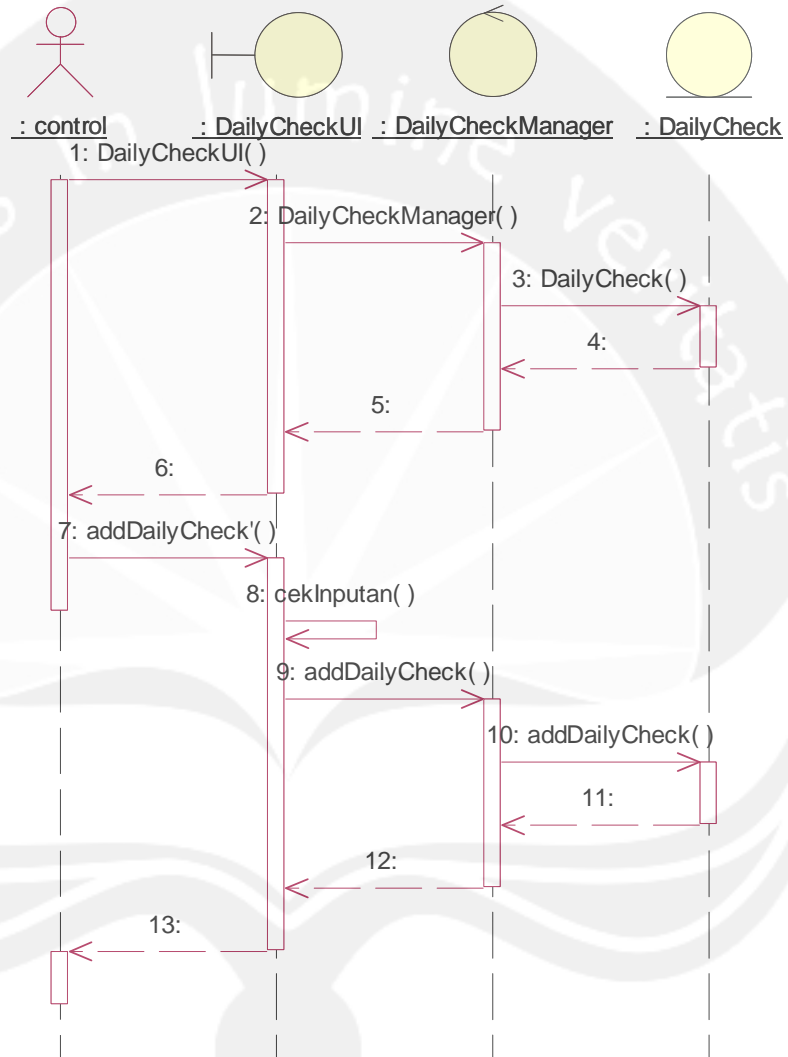
2.2.1.6.4 Search Data



Gambar 2.2.1.6.4 Sequence Diagram - Search Data Check Up

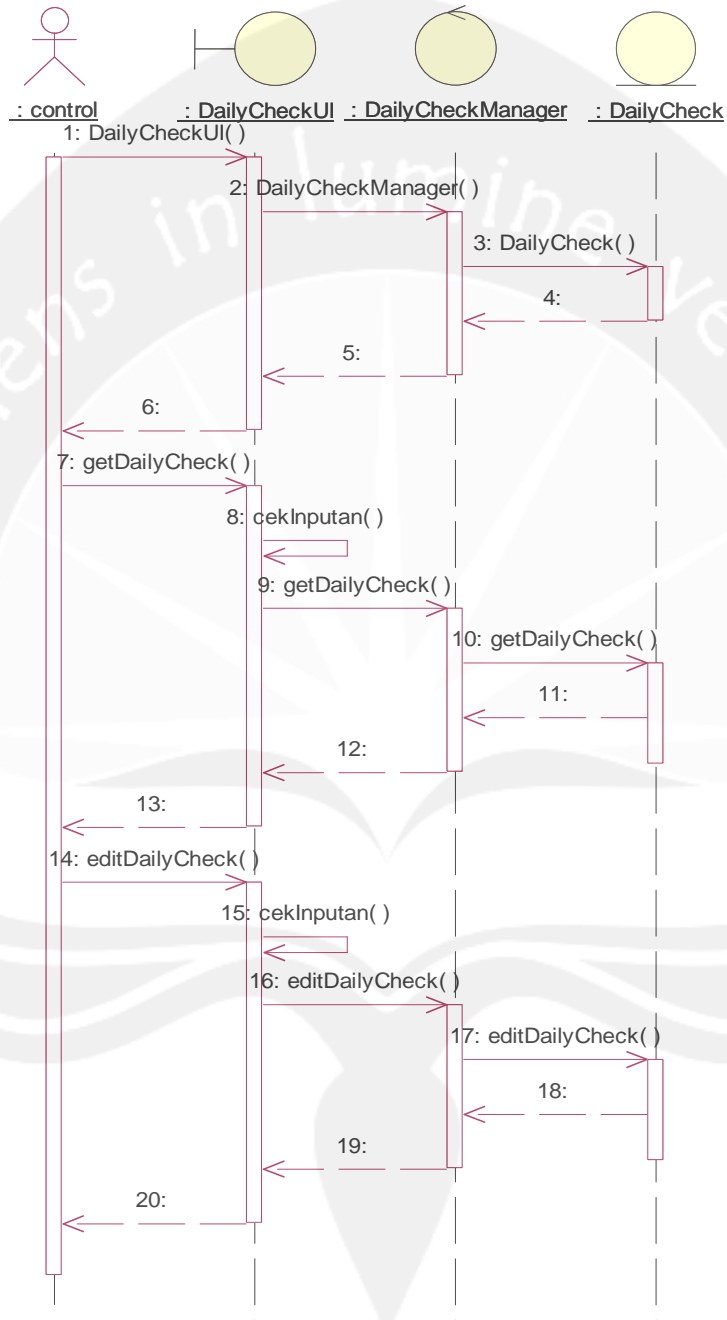
2.2.1.7 Pengelolaan Daily Check

2.2.1.7.1 Add Data



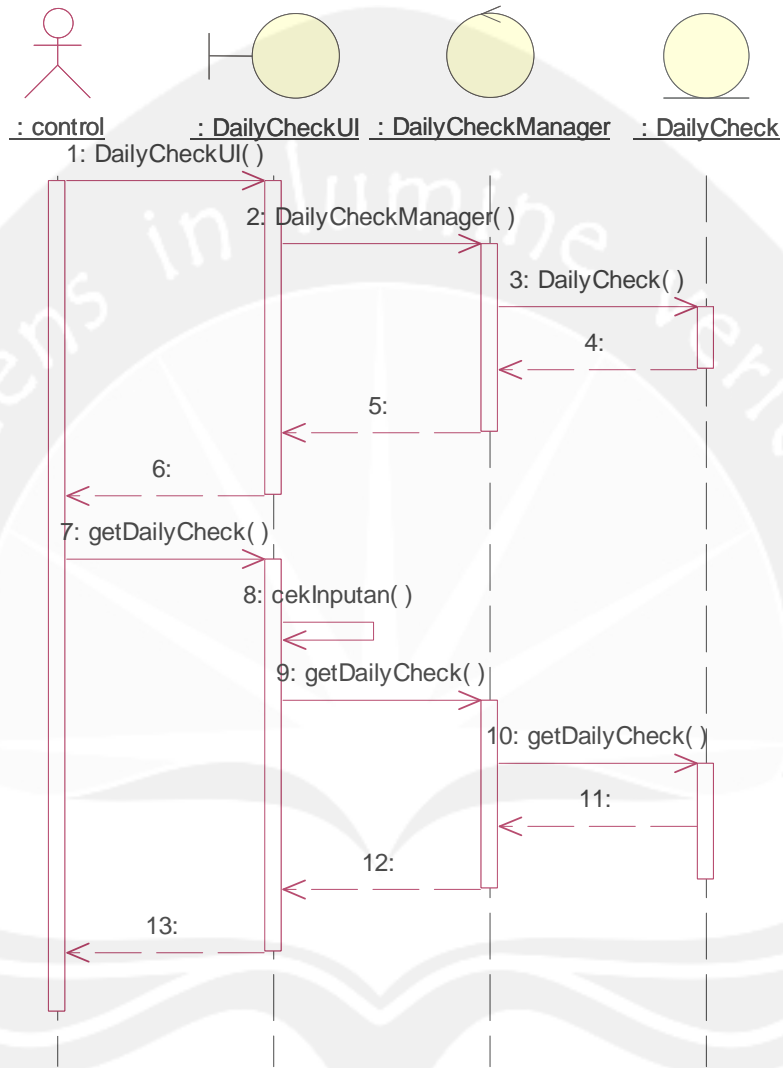
Gambar 2.2.1.7.1 Sequence Diagram - Add Data Daily Check

2.2.1.7.2 Edit Data



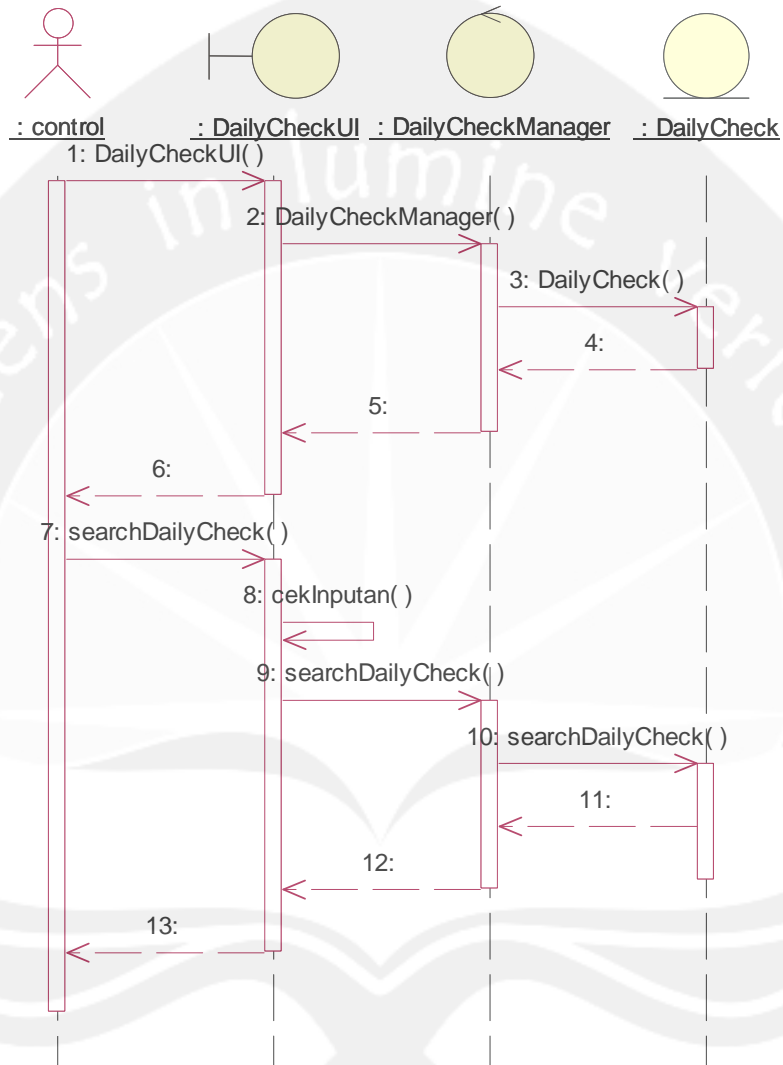
Gambar 2.2.1.7.2 Sequence Diagram - Edit Data Daily Check

2.2.1.7.3 Display Data



Gambar 2.2.1.7.3 Sequence Diagram - Display Data Daily Check

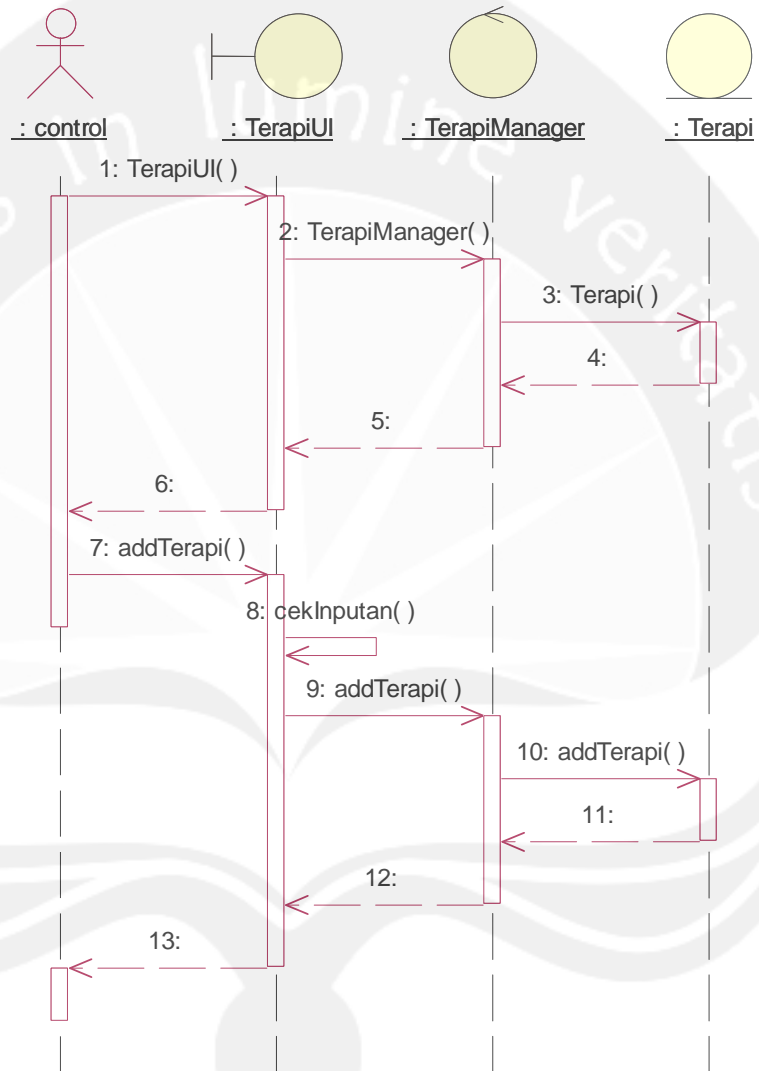
2.2.1.7.4 Search Data



Gambar 2.2.1.7.4 Sequence Diagram - Search Data Daily Check

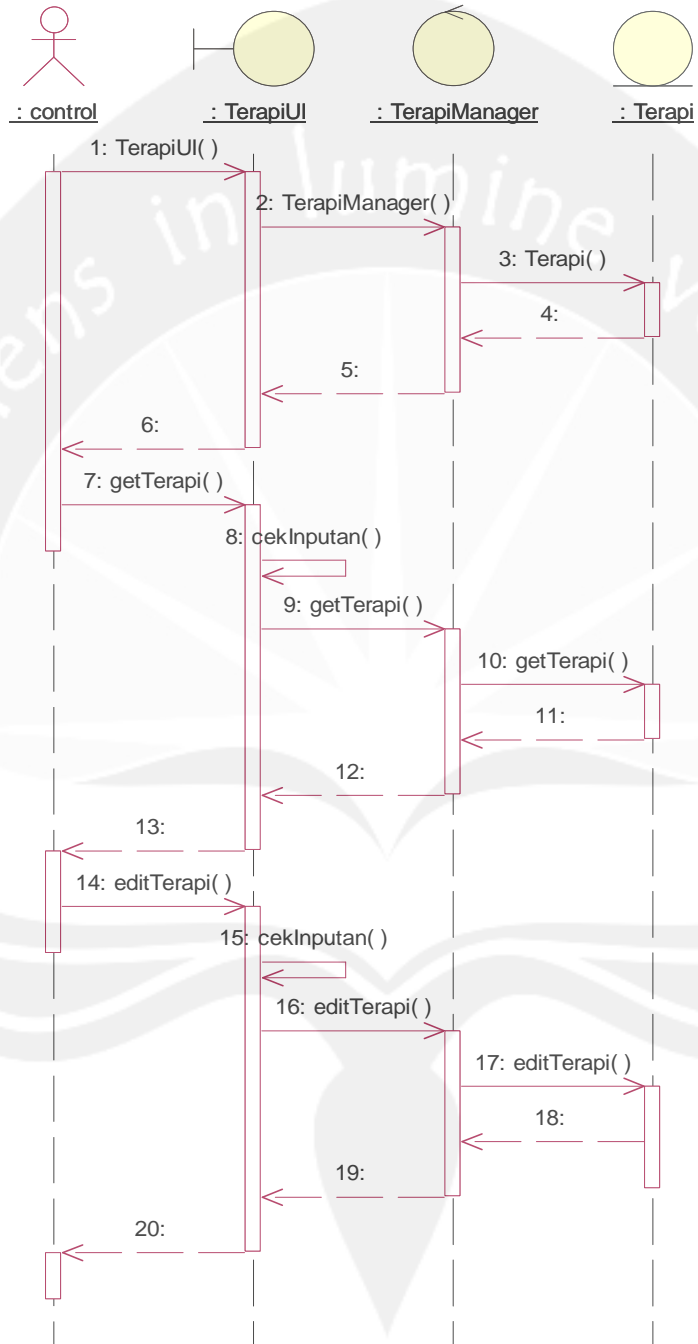
2.2.1.8 Pengelolaan Terapi

2.2.1.8.1 Add Data



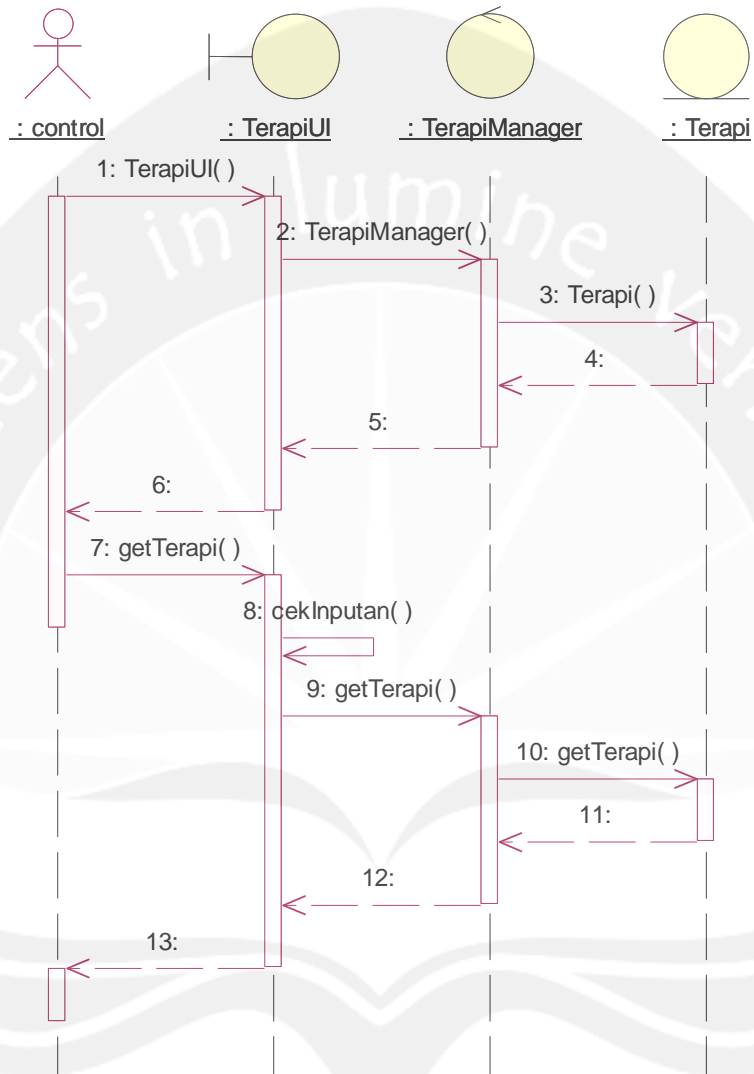
Gambar 2.2.1.8.1 Sequence Diagram - Add Data Terapi

2.2.1.8.2 Edit Data



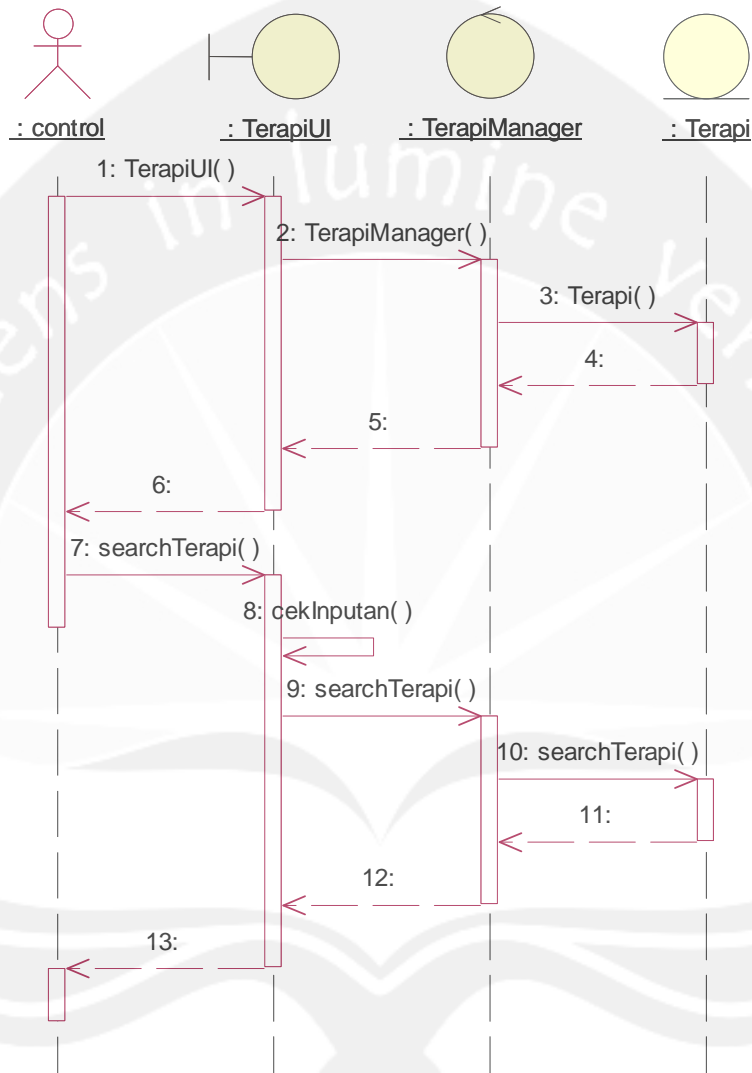
Gambar 2.2.1.8.2 Sequence Diagram - Edit Data Terapi

2.2.1.8.3 Display Data



Gambar 2.2.1.8.3 Sequence Diagram - Display Data Terapi

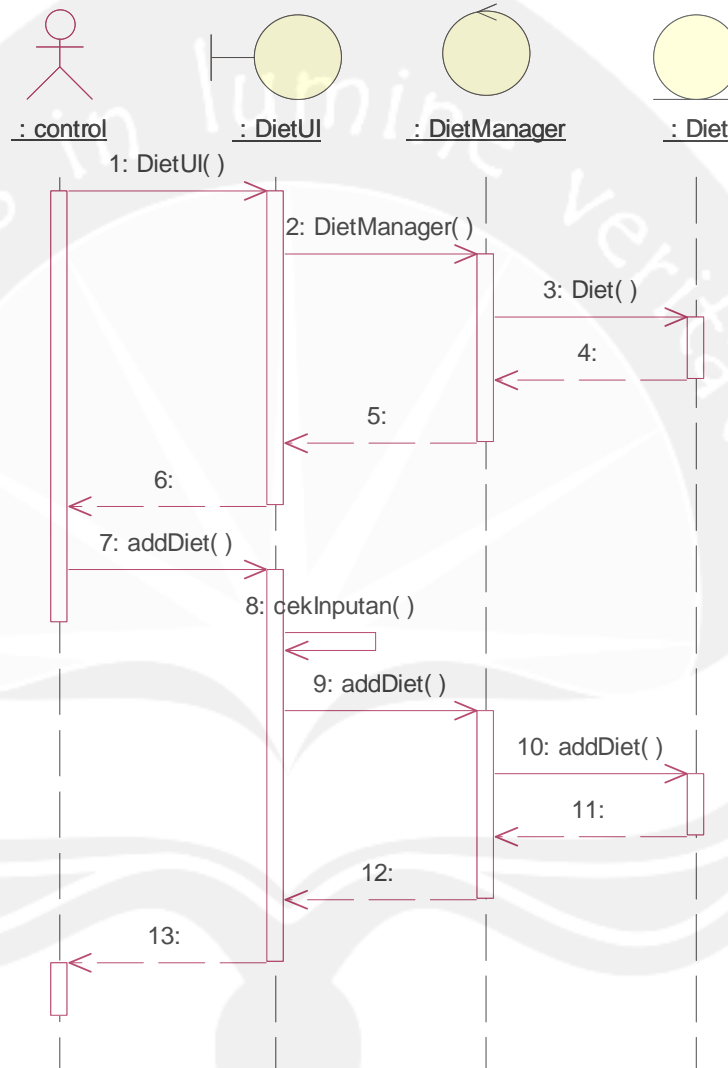
2.2.1.8.4 Search Data



Gambar 2.2.1.8.4 Sequence Diagram - Search Data Terapi

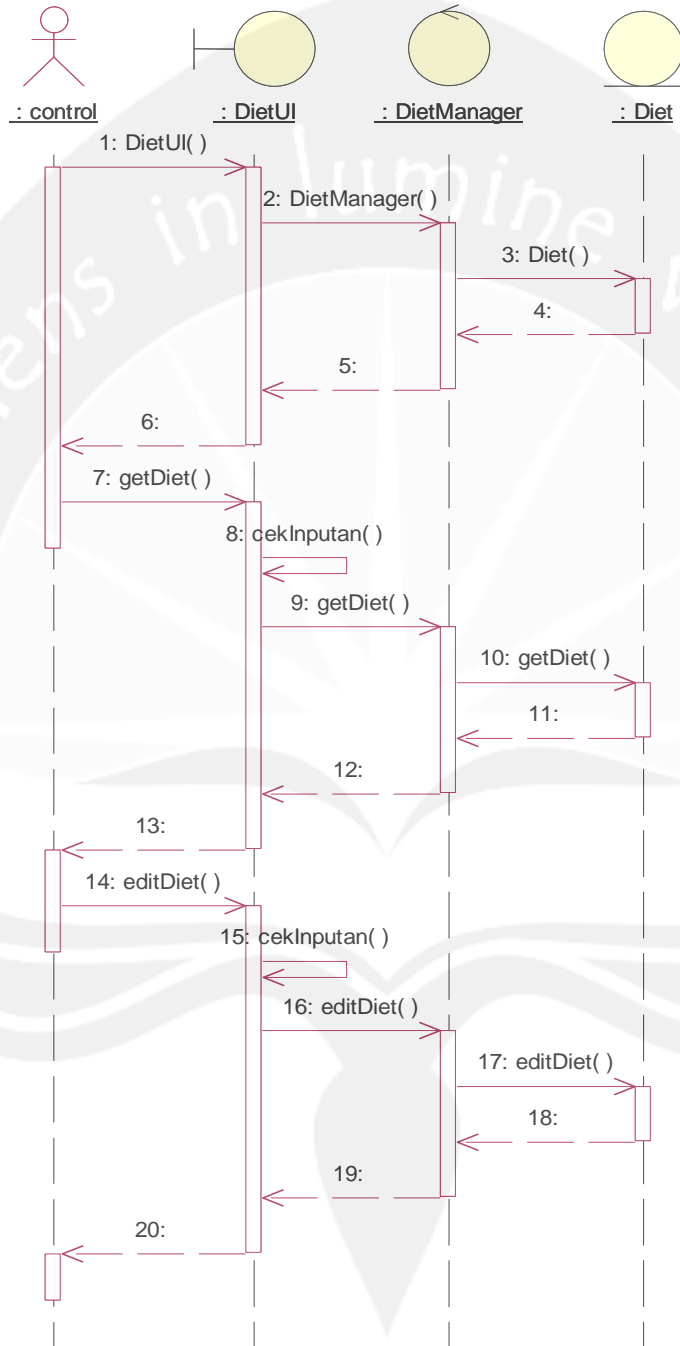
2.2.1.9 Pengelolaan Diet

2.2.1.9.1 Add Data



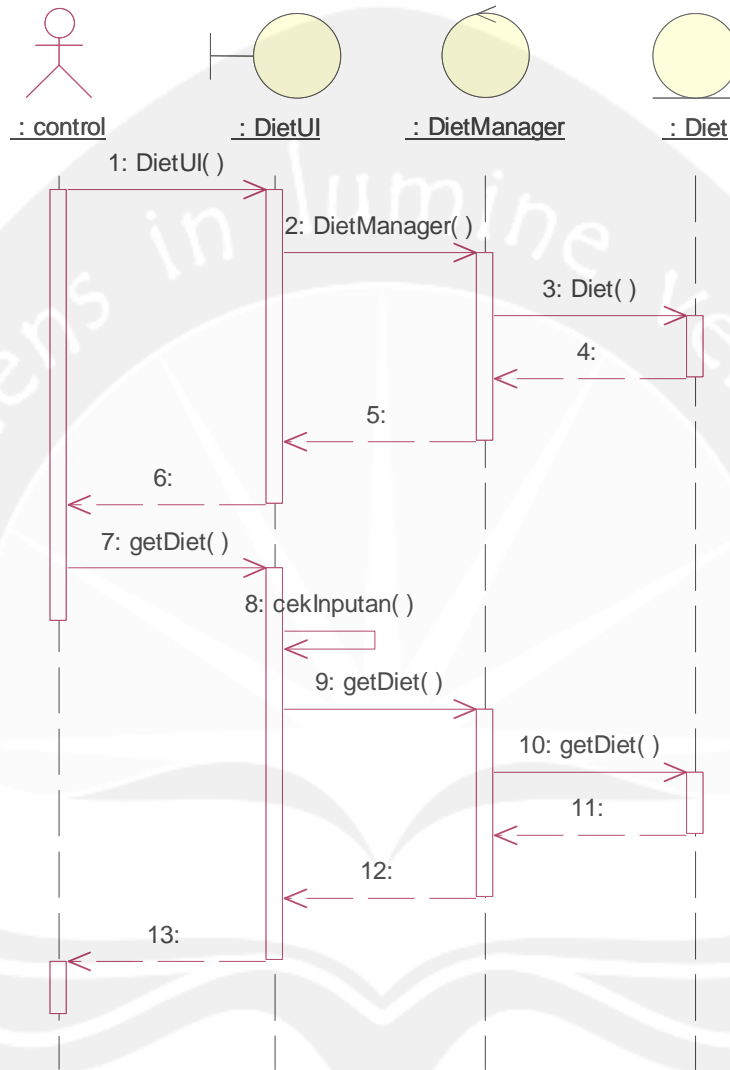
Gambar 2.2.1.9.1 Sequence Diagram - Add Data Diet

2.2.1.9.2 Edit Data



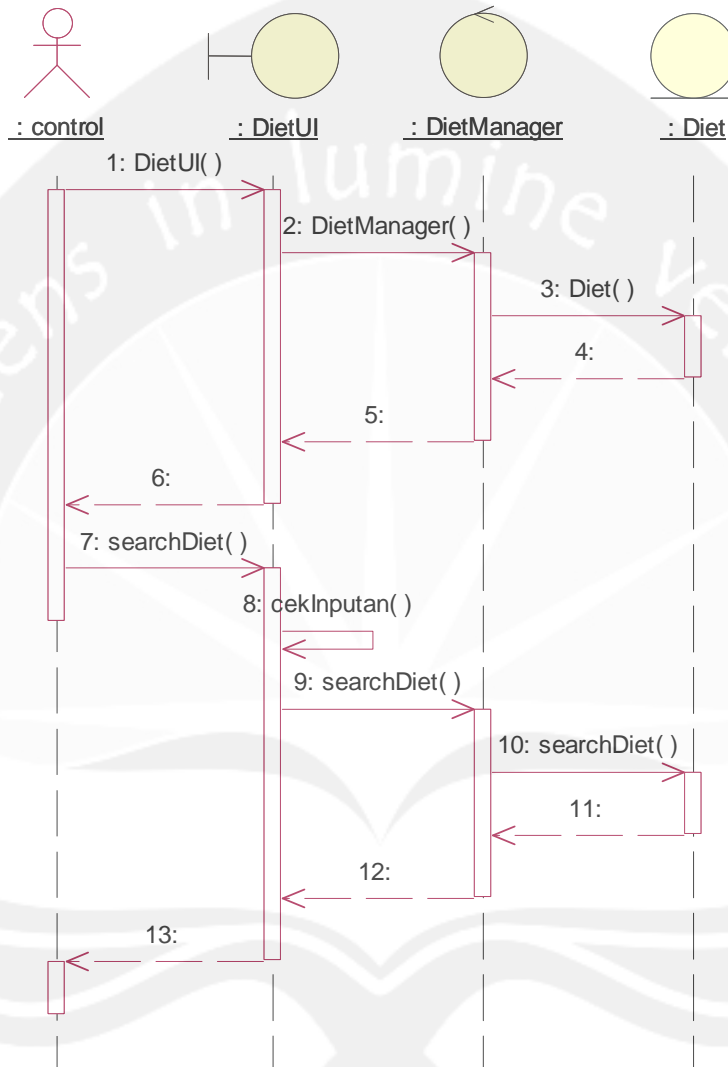
Gambar 2.2.1.9.2 Sequence Diagram - Edit Data Diet

2.2.1.9.3 Display Data



Gambar 2.2.1.9.3 Sequence Diagram - Display Data Diet

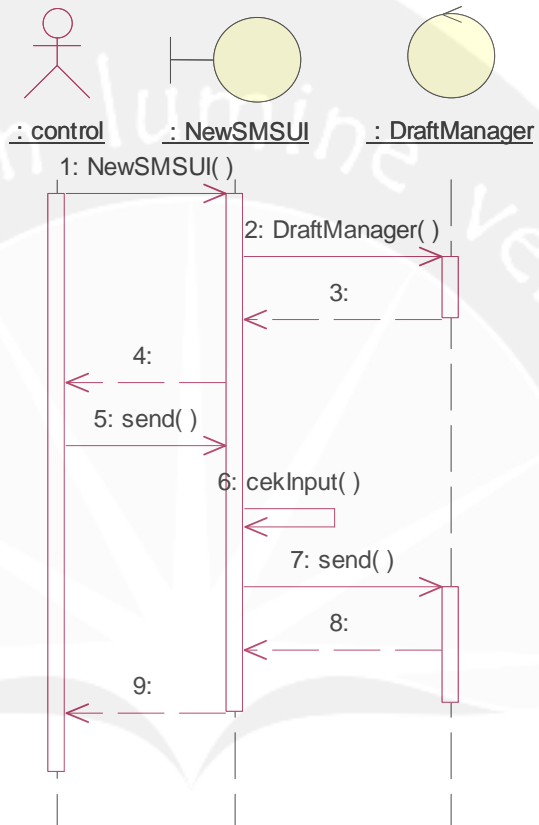
2.2.1.9.4 Search Data



Gambar 2.2.1.9.4 Sequence Diagram - Search Data Diet

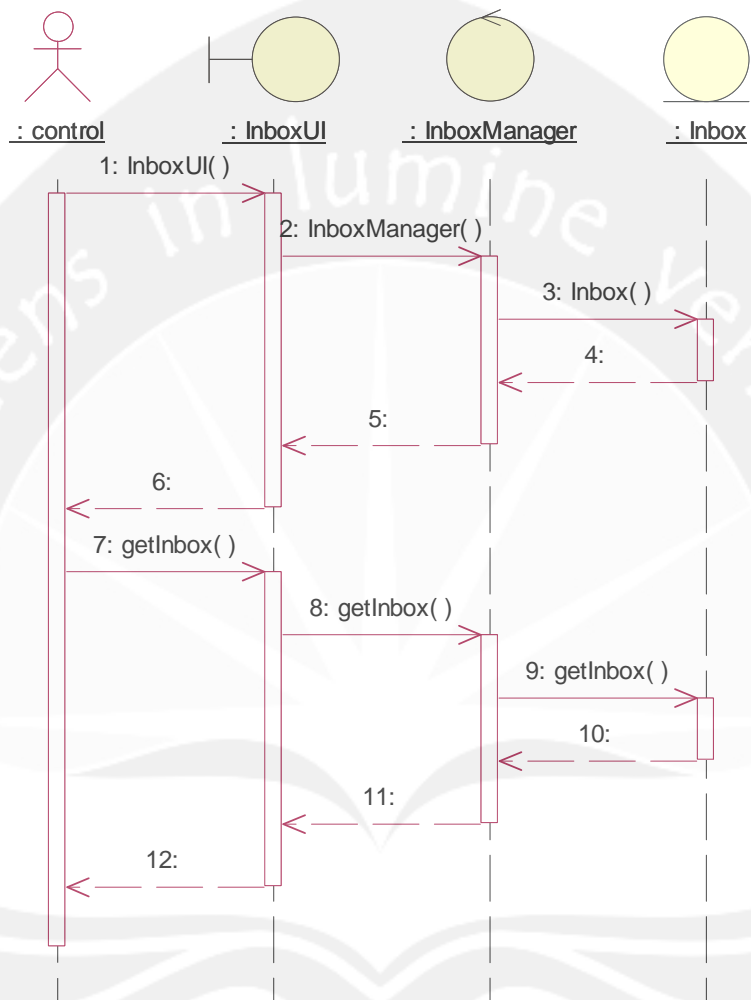
2.2.1.10 Pengelolaan SMS

2.2.1.10.1 New SMS



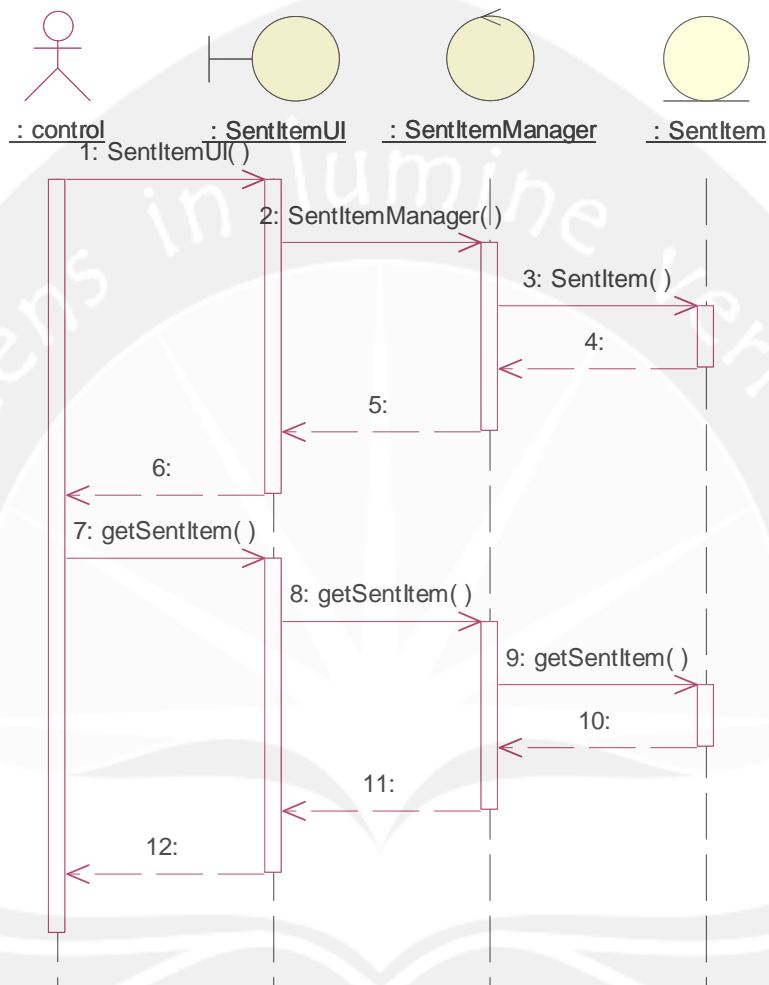
Gambar 2.2.1.10.1 Sequence Diagram - New SMS

2.2.1.10.2 Inbox



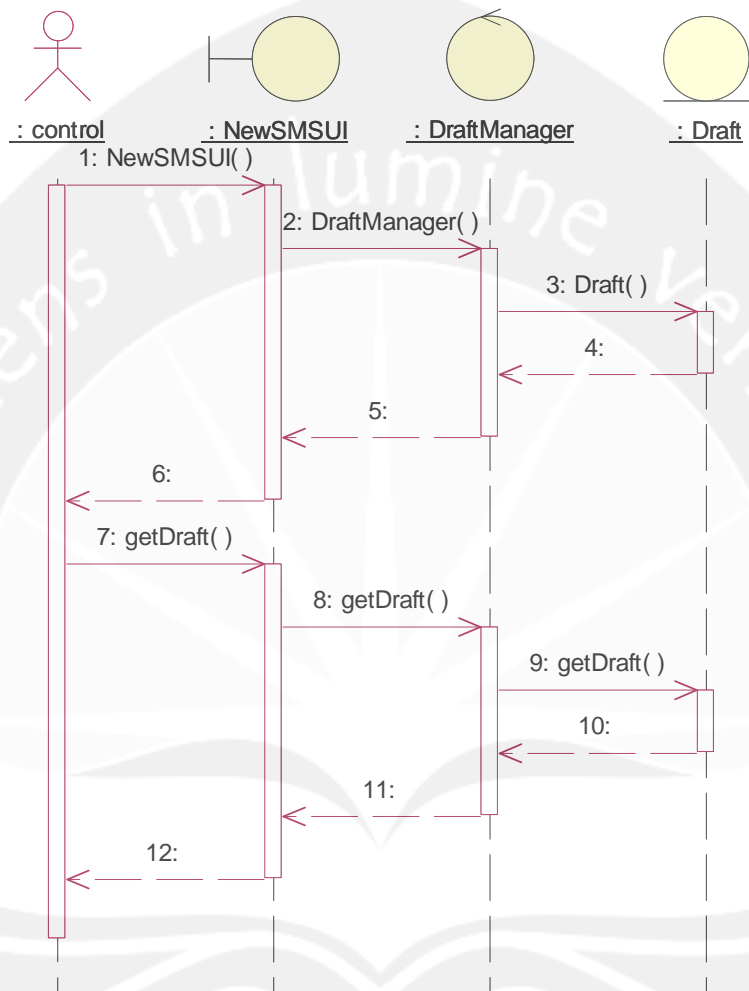
Gambar 2.2.1.10.2 Sequence Diagram - Inbox

2.2.1.10.3 Sent Item



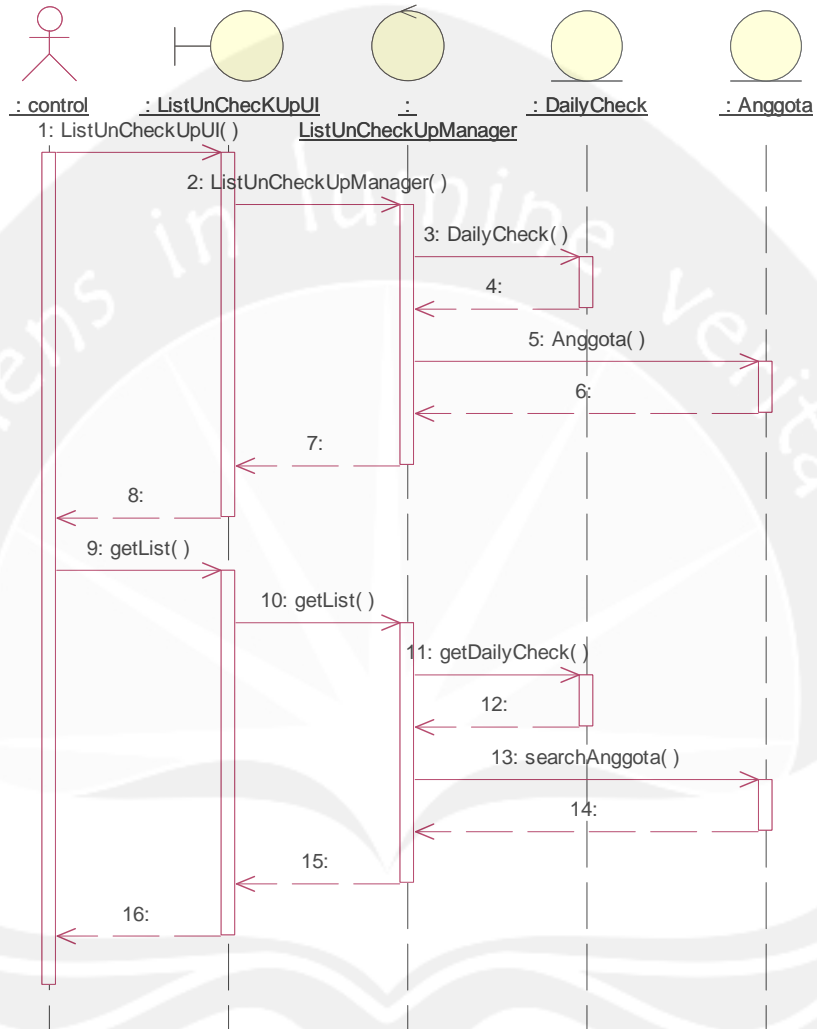
Gambar 2.2.1.10.3 Sequence Diagram - Sent Item

2.2.1.10.4 Draft



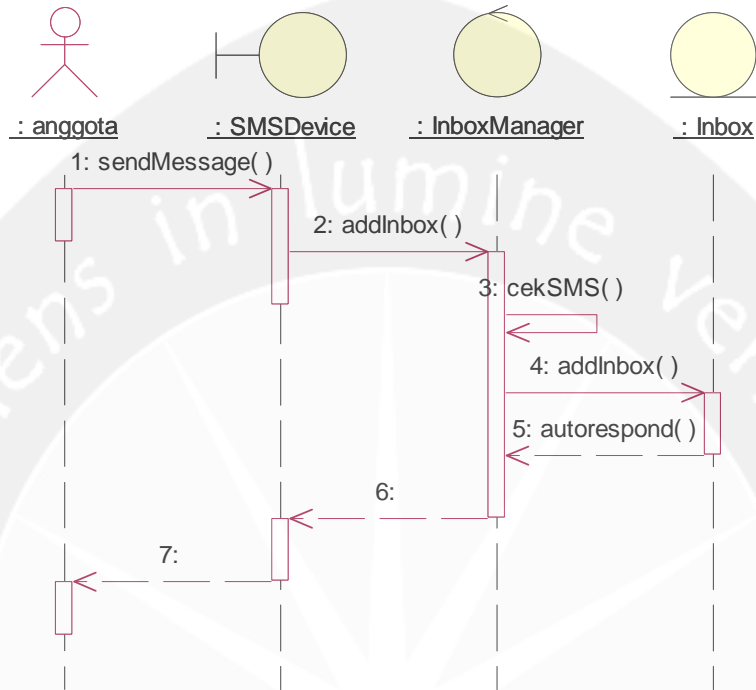
Gambar 2.2.1.10.4 Sequence Diagram - Draft

2.2.1.11 List UncekUp



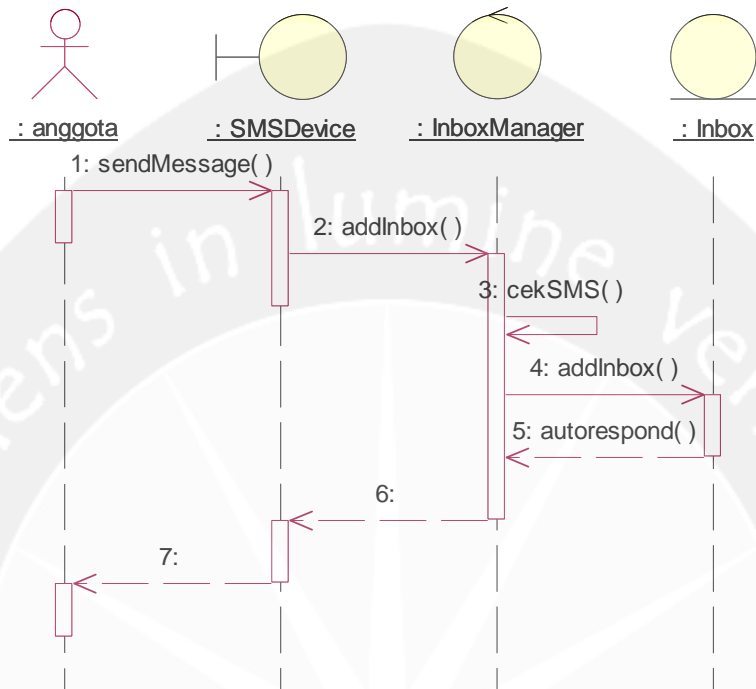
Gambar 2.2.1.11 Sequence Diagram - List Uncheck Up

2.2.1.12 Scheduled and Alert SMS



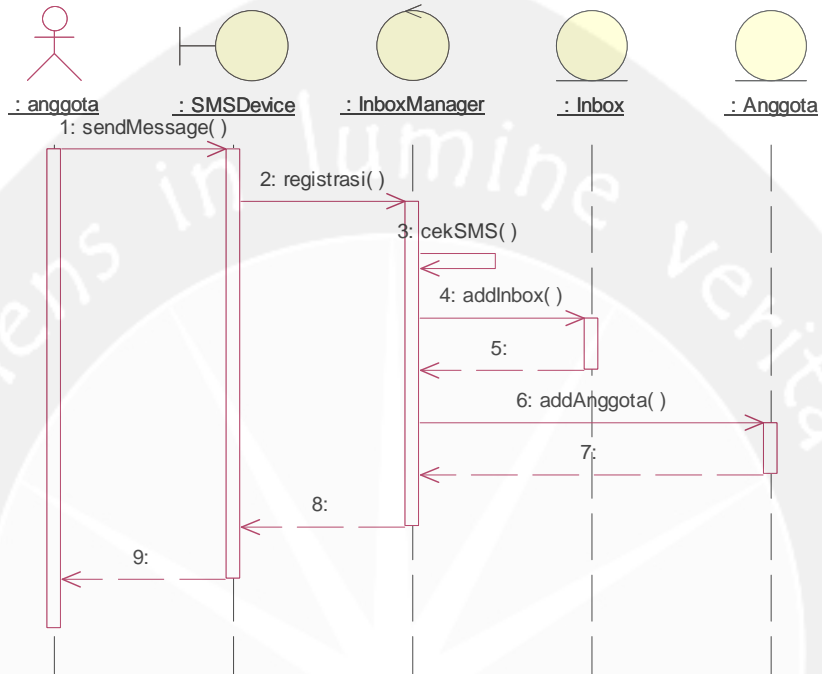
Gambar 2.2.12.12 Sequence Diagram - Scheduled and Alert SMS

2.2.1.13 Autorespond SMS



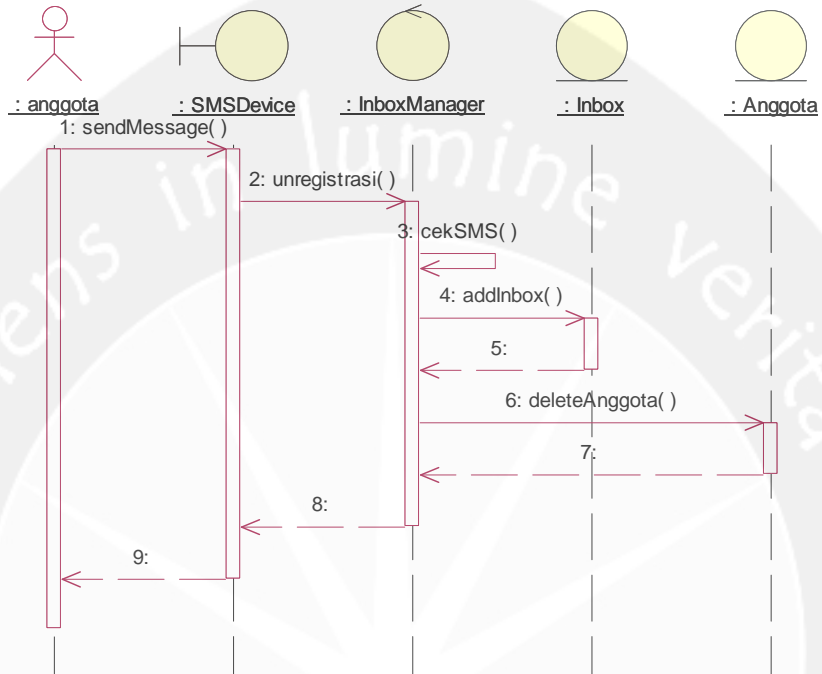
Gambar 2.2.1.13 Sequence Diagram - Autorespond SMS

2.2.1.14 Registrasi Anggota



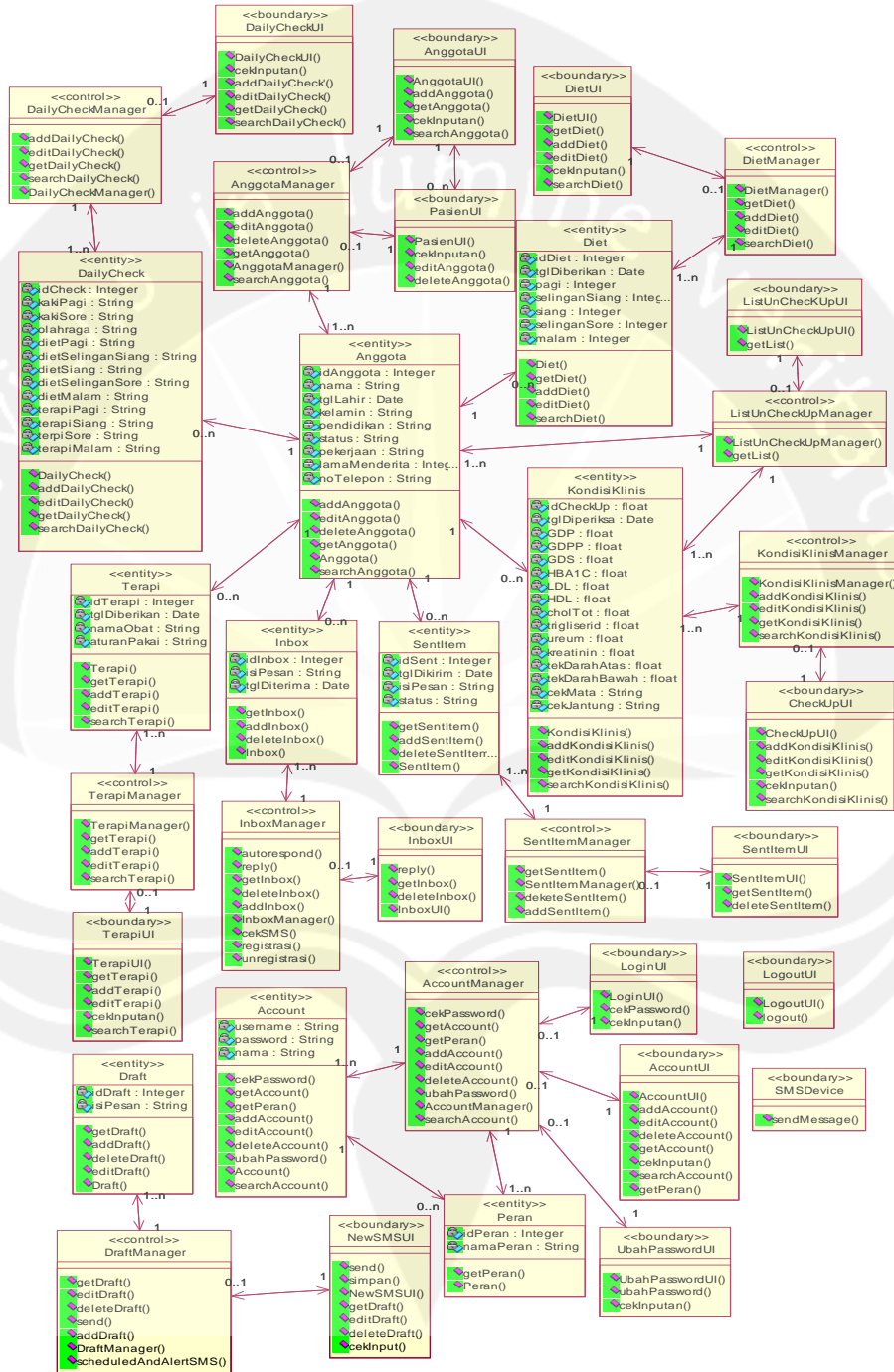
Gambar 2.2.1.14 Sequence Diagram - Registrasi Anggota

2.2.1.15 Unregistrasi Anggota



Gambar 2.2.1.15 Sequence Diagram - Unregistrasi Anggota

2.2.2 Diagram Kelas



Gambar 2.2.2 Diagram Kelas

2.2.3 Deskripsi Kelas

2.2.3.1 Specific Design Class LoginUI

LoginUI	<<boundary>>
+LoginUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+cekPassword(): boolean Operasi ini digunakan untuk mencari dan mengambil data login dari database dengan inputan keyword dari user.	
+cekInputan(): boolean Operasi ini digunakan untuk mengecek kelengkapan data yang diinputkan oleh user.	

2.2.3.2 Specific Design Class LogoutUI

LogoutUI	<<boundary>>
+LogoutUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+logout(): void Operasi ini digunakan untuk mengeluarkan user dari sistem.	

2.2.3.3 Specific Design Class AccountUI

AccountUI	<<boundary>>
+AccountUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+getAccount(): Data Table Operasi ini digunakan untuk mengambil data dari table account yang ingin ditampilkan user.	

```

+addAccount(): void
Operasi ini digunakan untuk menambah data account pada
database.
+editAccount(): void
Operasi ini digunakan untuk mengedit data account pada
database.
+cekInputan(): boolean
Operasi ini digunakan untuk mengecek kelengkapan data yang
diinputkan oleh user. Mereturnkan true jika data-data valid.
+deleteAccount(): void
Operasi ini digunakan untuk menghapus data account yang
diinputkan oleh user.
+searchAccount() : Data Table
Operasi ini digunakan untuk mencari data account tertentu
dengan keyword input dari user.

```

2.2.3.4 Specific Design Class UbahPasswordUI

UbahPasswordUI	<<boundary>>
<pre> +UbahPasswordUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +ubahPassword(): void Operasi ini digunakan untuk mengubah password yang dimiliki oleh user yang bersangkutan. +cekInputan(): boolean Operasi ini digunakan untuk mengecek kelengkapan data yang diinputkan oleh user. </pre>	

2.2.3.5 Specific Design Class CheckUpUI

CheckUpUI	<<boundary>>
<pre> +CheckUpUI() Default konstruktor, digunakan untuk inisialisasi semua </pre>	

attribute dari kelas ini.

+getKondisiKlinis(): Data Table
 Operasi ini digunakan untuk mengambil data dari database yang ingin ditampilkan user.

+cekInputan(): boolean
 Operasi ini digunakan untuk mengecek kelengkapan data yang diinputkan oleh user.

+addKondisiKlinis(): void
 Operasi ini digunakan untuk menambah data kondisi klinis pada database.

+editCheckUp(): void
 Operasi ini digunakan untuk mengedit data check up pada database.

+searchKondisiKlinis() : Data Table
 Operasi ini digunakan untuk mencari data kondisi klinis tertentu dengan keyword input dari user.

2.2.3.6 Specific Design Class DietUI

DietUI	<<boundary>>
<p>+DietUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+getDiet(): Data Table Operasi ini digunakan untuk mengambil data dari database yang ingin ditampilkan user.</p> <p>+cekInputan(): boolean Operasi ini digunakan untuk mengecek kelengkapan data yang diinputkan oleh user.</p> <p>+addDiet(): void Operasi ini digunakan untuk menambah data diet pada database.</p> <p>+editDiet(): void Operasi ini digunakan untuk mengedit data diet pada</p>	

database.

+searchDiet() : Data Table
 Operasi ini digunakan untuk mencari data diet tertentu dengan keyword input dari user.

2.2.3.7 Specific Design Class AnggotaUI

AnggotaUI	<<boundary>>
<pre> +AnggotaUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getAnggota(): Data Table Operasi ini digunakan untuk mengambil data dari database yang ingin ditampilkan user. +cekInputan(): boolean Operasi ini digunakan untuk mengecek kelengkapan data yang diinputkan oleh user. +addAnggota(): void Operasi ini digunakan untuk menambah data anggota pada database. +editAnggota(): void Operasi ini digunakan untuk mengedit data anggota pada database. +deleteAnggota(): void Operasi ini digunakan untuk menghapus data anggota pada database. +searchAnggota() : Data Table Operasi ini digunakan untuk mencari data anggota tertentu dengan keyword input dari user. </pre>	

2.2.3.8 Specific Design Class PasienUI

PasienUI	<<boundary>>
<pre> +PasienUI() </pre>	

Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

+cekInputan(): boolean
Operasi ini digunakan untuk mengecek kelengkapan data yang diinputkan oleh user.

+editAnggota(): void
Operasi ini digunakan untuk mengedit data anggota pada database.

+deleteAnggota(): void
Operasi ini digunakan untuk menghapus data anggota pada database.

2.2.3.9 Specific Design Class TerapiUI

TerapiUI	<<boundary>>
<p>+TerapiUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+getTerapi(): Data Table Operasi ini digunakan untuk mengambil data dari database yang ingin ditampilkan user.</p> <p>+cekInputan(): boolean Operasi ini digunakan untuk mengecek kelengkapan data yang diinputkan oleh user.</p> <p>+addTerapi(): void Operasi ini digunakan untuk menambah data terapi pada database.</p> <p>+editTerapi(): void Operasi ini digunakan untuk mengedit data terapi pada database.</p> <p>+searchTerapi() : Data Table Operasi ini digunakan untuk mencari data terapi tertentu dengan keyword input dari user.</p>	

2.2.3.10 Specific Design Class DailyCheckUI

DailyCheckUI	<<boundary>>
<pre>+DailyCheckUI() Default konstruktor, digunakan untuk inialisasi semua attribute dari kelas ini. +getDailyCheck(): Data Table Operasi ini digunakan untuk mengambil data dari database yang ingin ditampilkan user. +cekInputan(): boolean Operasi ini digunakan untuk mengecek kelengkapan data yang diinputkan oleh user. +addDailyCheck(): void Operasi ini digunakan untuk menambah data daily check pada database. +editDailyCheck(): void Operasi ini digunakan untuk mengedit data daily check pada database. +searchDailyCheck() : Data Table Operasi ini digunakan untuk mencari data daily check tertentu dengan keyword input dari user.</pre>	

2.2.3.11 Specific Design Class ListUnCheckUpUI

ListUnCheckUpUI	<<boundary>>
<pre>+ListUnCheckUpUI() Default konstruktor, digunakan untuk inialisasi semua attribute dari kelas ini. +getList(): Data Table Operasi ini digunakan untuk mengambil data dari database yang ingin ditampilkan user.</pre>	

2.2.3.12 Specific Design Class NewSMSUI

NewSMSUI	<<boundary>>
-----------------	---------------------------------

<pre> +NewSMSUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +send(): void Operasi ini digunakan untuk mengirimkan sms. +simpan(): void Operasi ini digunakan untuk menyimpan sms sebagai draft. +getDraft(): Data Table Operasi ini digunakan untuk mengambil data dari database yang ingin ditampilkan user. +editDraft(): void Operasi ini digunakan untuk melakukan perubahan pada data draft di database. +deleteDraft(): void Operasi ini digunakan untuk melakukan penghapusan pada data draft di database. +cekInputan(): boolean Operasi ini digunakan untuk mengecek kelengkapan data yang diinputkan oleh user. </pre>
--

2.2.3.13 Specific Design Class InboxUI

InboxUI	<<boundary>>
<pre> +InboxUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getInbox(): Data Table Operasi ini digunakan untuk mengambil data dari database yang ingin ditampilkan user. +reply(): void Operasi ini digunakan untuk membalas sms yang ada pada inbox. +deleteInbox(): void </pre>	

Operasi ini digunakan untuk menghapus sms yang ada pada inbox.

2.2.3.14 Specific Design Class SentItemUI

SentItemUI	<<boundary>>
<pre>+SentItemUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +deleteSentItem(): void Operasi ini digunakan untuk menghapus sms yang ada pada sent item. +getSentItem(): Data Table Operasi ini digunakan untuk mengambil data dari database yang ingin ditampilkan user.</pre>	

2.2.3.15 Specific Design Class SMSDeviceUI

MobilUI	<<boundary>>
<pre>+MobilUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +searchMobil(): Data Table Operasi ini digunakan untuk mencari data tertentu dengan keyword input dari user. +getMobil(): Data Table Operasi ini digunakan untuk mengambil data dari database yang ingin ditampilkan user. +editMobil(): void Operasi ini digunakan untuk mengedit data pada database. +cekInputan(): boolean Operasi ini digunakan untuk mengecek kelengkapan data yang diinputkan oleh user. +deleteMobil(): void</pre>	

Operasi ini digunakan untuk menghapus data dari database yang diidentifikasi oleh user.

+addMobil(): void

Operasi ini digunakan untuk menambahkan data yang diinputkan oleh user.

2.2.3.16 Specific Design Class AccountManager

AccountManager	<<control>>
<p>+AccountManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+getAccount(): Data Table Operasi ini digunakan untuk mengambil data dari table account yang ingin ditampilkan user.</p> <p>+getPeran(): Data Table Operasi ini digunakan untuk mengambil data dari table peran yang ingin ditampilkan user.</p> <p>+addAccount(): boolean Operasi ini digunakan untuk menambah data account pada database. Mereturnkan true jika operasi berhasil.</p> <p>+editAccount(): boolean Operasi ini digunakan untuk mengedit data account pada database. Mereturnkan true jika operasi berhasil.</p> <p>+deleteAccount(): boolean Operasi ini digunakan untuk menghapus data account yang diinputkan oleh user. Mereturnkan true jika operasi berhasil.</p> <p>+searchAccount(): Data Table Operasi ini digunakan untuk mencari data account tertentu dengan keyword input dari user.</p> <p>+cekPassword(): boolean Operasi ini digunakan untuk melakukan pengecekan kebenaran password yang ada di database dengan yang diinputkan oleh</p>	

user dimana untuk keperluan login.

+ubahPassword(): boolean

Operasi ini digunakan untuk melakukan ubah password pada database.

2.2.3.17 Specific Design Class KondisiKlinisManager

KondisiKlinisManager	<<control>>
<pre>+KondisiKlinisManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getKondisiKlinis(): Data Table Operasi ini digunakan untuk mengambil data dari table kondisi klinis yang ingin ditampilkan user. +addKondisiKlinis(): boolean Operasi ini digunakan untuk menambah data kondisi klinis pada database. Mereturnkan true jika operasi berhasil. +editKondisiKlinis(): boolean Operasi ini digunakan untuk mengedit data kondisi klinis pada database. Mereturnkan true jika operasi berhasil. +searchKondisiKlinis(): Data Table Operasi ini digunakan untuk mencari data kondisi klinis tertentu dengan keyword input dari user.</pre>	

2.2.3.18 Specific Design Class DietManager

DietManager	<<control>>
<pre>+DietManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDiet(): Data Table Operasi ini digunakan untuk mengambil data dari table diet yang ingin ditampilkan user. +addDiet(): boolean</pre>	

Operasi ini digunakan untuk menambah data diet pada database. Mereturnkan true jika operasi berhasil.

+editDiet(): boolean

Operasi ini digunakan untuk mengedit data diet pada database. Mereturnkan true jika operasi berhasil.

+searchDiet(): Data Table

Operasi ini digunakan untuk mencari data diet tertentu dengan keyword input dari user.

2.2.3.19 Specific Design Class AnggotaManager

AnggotaManager	<<control>>
<p>+AnggotaManager()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+getAnggota(): Data Table</p> <p>Operasi ini digunakan untuk mengambil data dari table karakteristikanggota yang ingin ditampilkan user.</p> <p>+addAnggota(): boolean</p> <p>Operasi ini digunakan untuk menambah data anggota pada database. Mereturnkan true jika operasi berhasil.</p> <p>+editAnggota(): boolean</p> <p>Operasi ini digunakan untuk mengedit data anggota pada database. Mereturnkan true jika operasi berhasil.</p> <p>+deleteAnggota(): boolean</p> <p>Operasi ini digunakan untuk menghapus data anggota yang diinputkan oleh user. Mereturnkan true jika operasi berhasil.</p> <p>+searchAnggota(): Data Table</p> <p>Operasi ini digunakan untuk mencari data anggota tertentu dengan keyword input dari user.</p>	

2.2.3.20 Specific Design Class TerapiManager

TerapiManager	<<control>>
----------------------	--------------------------------

<pre> +TerapiManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getTerapi(): Data Table Operasi ini digunakan untuk mengambil data dari table terapi yang ingin ditampilkan user. +addTerapi(): boolean Operasi ini digunakan untuk menambah data terapi pada database. Mereturnkan true jika operasi berhasil. +editTerapi(): boolean Operasi ini digunakan untuk mengedit data terapi pada database. Mereturnkan true jika operasi berhasil. +searchTerapi(): Data Table Operasi ini digunakan untuk mencari data terapi tertentu dengan keyword input dari user. </pre>
--

2.2.3.21 Specific Design Class DailyCheckManager

DailyCheckManager	<<control>>
<pre> +DailyCheckManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDailyCheck(): Data Table Operasi ini digunakan untuk mengambil data dari table terapi yang ingin ditampilkan user. +addDailyCheck(): boolean Operasi ini digunakan untuk menambah data terapi pada database. Mereturnkan true jika operasi berhasil. +editDailyCheck(): boolean Operasi ini digunakan untuk mengedit data terapi pada database. Mereturnkan true jika operasi berhasil. +searchDailyCheck(): Data Table Operasi ini digunakan untuk mencari data terapi tertentu </pre>	

Program Studi Teknik Informatika	DPPL – GMIS	62/ 88
----------------------------------	-------------	--------

dengan keyword input dari user.

2.2.3.22 Specific Design Class ListUnCheckUpManager

ListUnCheckUpManager	<<control>>
<pre>+ListUnCheckUpManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getList(): Data Table Operasi ini digunakan untuk mengambil data dari database yang ingin ditampilkan user.</pre>	

2.2.3.23 Specific Design Class DraftManager

DraftManager	<<control>>
<pre>+DraftManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDraft(): Data Table Operasi ini digunakan untuk mengambil data dari table draft yang ingin ditampilkan user. +addDraft(): boolean Operasi ini digunakan untuk menambah data draft pada database. Mereturnkan true jika operasi berhasil. +editDraft(): boolean Operasi ini digunakan untuk mengedit data draft pada database. Mereturnkan true jika operasi berhasil. +send(): boolean Operasi ini digunakan untuk mengirimkan sms. Mereturnkan true jika berhasil mengirim. +deleteDraft(): boolean Operasi ini digunakan untuk menghapus sms draft pada database. Mereturnkan true jika operasi berhasil. +scheduledAndAlertSMS(): void</pre>	

Operasi ini digunakan untuk mengirimkan sms pada waktu tertentu.

2.2.3.24 Specific Design Class InboxManager

InboxManager	<<control>>
<pre> +InboxManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getInbox(): Data Table Operasi ini digunakan untuk mengambil data dari table inbox yang ingin ditampilkan user. +addInbox(): void Operasi ini digunakan untuk menambah data inbox pada database. Mereturnkan true jika operasi berhasil. +reply(): boolean Operasi ini digunakan untuk mengirimkan sms balasan. Mereturnkan true jika operasi berhasil. +autorespond(): void Operasi ini digunakan untuk membalas sms yang masuk. Mereturnkan true jika operasi berhasil. +deleteInbox(): boolean Operasi ini digunakan untuk menghapus sms yang ada pada table inbox. Mereturnkan true jika operasi berhasil. +registrasi(): void Operasi ini digunakan untuk melakukan registrasi anggota ke dalam system melalui sms. +unregistrasi(): void Operasi ini digunakan untuk melakukan unregistrasi/penghapusan anggota dari table pada database. +cekSMS(): boolean Operasi ini digunakan untuk melakukan pengecekan terhadap sms yang masuk. Mereturnkan false jika tidak sesuai format. </pre>	

2.2.3.25 Specific Design Class SentItemManager

SentItemManager	<<control>>
<pre>+SentItemManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getSentItem(): Data Table Operasi ini digunakan untuk mengambil data dari table sent item yang ingin ditampilkan user. +addSentItem(): boolean Operasi ini digunakan untuk menambah data sent item pada database. Mereturnkan true jika operasi berhasil. +deleteSentItem(): boolean Operasi ini digunakan untuk menghapus data sent item pada database. Mereturnkan true jika operasi berhasil.</pre>	

2.2.3.26 Specific Design Class Account

Account	<<entity>>
<pre>-username: string -password: string -idPeran: integer -namaPeran: string -nama: string +Account() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getAccount(): Data Table Operasi ini digunakan untuk mengambil data dari table account yang ingin ditampilkan user. +addAccount(): boolean Operasi ini digunakan untuk menambah data account pada database. Mereturnkan true jika operasi berhasil. +editAccount(): boolean Operasi ini digunakan untuk mengedit data account pada</pre>	

database. Mereturnkan true jika operasi berhasil.

+ubahPassword(): boolean

Operasi ini digunakan untuk mengedit password pada table account pada database. Mereturnkan true jika operasi berhasil.

+deleteAccount(): boolean

Operasi ini digunakan untuk menghapus data account yang diinputkan oleh user. Mereturnkan true jika operasi berhasil.

+searchAccount(): Data Table

Operasi ini digunakan untuk mencari data account tertentu dengan keyword input dari user.

2.2.3.27 Specific Design Class Peran

Peran	<<entity>>
-idPeran: int -namaPeran: string	
+Peran() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getPeran(): Data Table Operasi ini digunakan untuk mengambil data dari table peran yang ingin ditampilkan user.	

2.2.3.28 Specific Design Class Anggota

Anggota	<<entity>>
-idAnggota: int -nama: string -tglLahir: date -kelamin: string -pendidikan: string -status: string -pekerjaan: string -lamaMenderita: int	

-noTelepon: string
<p>+Anggota() Default konstruktor, digunakan untuk inialisasi semua attribute dari kelas ini.</p> <p>+getAnggota(): Data Table Operasi ini digunakan untuk mengambil data dari table anggota yang ingin ditampilkan user.</p> <p>+addAnggota(): boolean Operasi ini digunakan untuk menambah data anggota pada database. Mereturnkan true jika operasi berhasil.</p> <p>+editAnggota(): boolean Operasi ini digunakan untuk mengedit data anggota pada database. Mereturnkan true jika operasi berhasil.</p> <p>+deleteAnggota(): boolean Operasi ini digunakan untuk menghapus data anggota yang diinputkan oleh user. Mereturnkan true jika operasi berhasil.</p> <p>+searchAnggota(): Data Table Operasi ini digunakan untuk mencari data anggota tertentu dengan keyword input dari user.</p>

2.2.3.29 Specific Design Class DailyCheck

DailyCheck	<<entity>>
<p>-idCheck: int</p> <p>-kakiPagi: string</p> <p>-kakiSore: string</p> <p>-olahraga: string</p> <p>-dietPagi: string</p> <p>-dietSelinganSiang: string</p> <p>-dietSiang: string</p> <p>-dietSelinganSore: string</p> <p>-dietMalam: string</p> <p>-terapiPagi: string</p> <p>-terapiSiang: string</p>	

<pre>-terapiSore: string -terapiMalam: string</pre>
<pre>+DailyCheck() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDailyCheck(): Data Table Operasi ini digunakan untuk mengambil data dari table daily check yang ingin ditampilkan user. +addDailyCheck(): boolean Operasi ini digunakan untuk menambah data daily check pada database. Mereturnkan true jika operasi berhasil. +editDailyCheck(): boolean Operasi ini digunakan untuk mengedit data daily check pada database. Mereturnkan true jika operasi berhasil. +searchDailyCheck(): Data Table Operasi ini digunakan untuk mencari data daily check tertentu dengan keyword input dari user.</pre>

2.2.3.30 Specific Design Class KondisiKlinis

KondisiKlinis	<<entity>>
<pre>-idCheckUp: int -tglDiperiksa: date -GDP: float -GDPP: float -GDS: float -HBA1C: float -LDL: float -HDL: float -cholesterol: float -trigliserid: float -ureum: float -kreatinin: float -tekDarahAtas: float -tekDarahBawah: float</pre>	

-cekMata: string
-cekJantung: string
+KondisiKlinis() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.
+getKondisiKlinis(): Data Table Operasi ini digunakan untuk mengambil data dari table kondisi klinis yang ingin ditampilkan user.
+addKondisiKlinis(): boolean Operasi ini digunakan untuk menambah data kondisi klinis pada database. Mereturnkan true jika operasi berhasil.
+editKondisiKlinis(): boolean Operasi ini digunakan untuk mengedit data kondisi klinis pada database. Mereturnkan true jika operasi berhasil.
+searchKondisiKlinis(): Data Table Operasi ini digunakan untuk mencari data kondisi klinis tertentu dengan keyword input dari user.

2.2.3.31 Specific Design Class Diet

Diet	<<entity>>
-idDiet: int -tglDiberikan: date -pagi: int -selinganSiang: int -siang: int -selinganSore: int -malam: int	
+Diet() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDiet(): Data Table Operasi ini digunakan untuk mengambil data dari table diet yang ingin ditampilkan user. +addDiet(): boolean	

Operasi ini digunakan untuk menambah data diet pada database. Mereturnkan true jika operasi berhasil.
 +editDiet(): boolean

Operasi ini digunakan untuk mengedit data diet pada database. Mereturnkan true jika operasi berhasil.
 +searchDiet(): Data Table

Operasi ini digunakan untuk mencari data diet tertentu dengan keyword input dari user.

2.2.3.32 Specific Design Class Terapi

Terapi	<<entity>>
-idTerapi: int -tglDiberikan: date -namaObat: string -aturanPakai: string	
+Terapi() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +searchTerapi(): Data Table Operasi ini digunakan untuk mencari data terapi tertentu dengan keyword input dari user. +getTerapi(): Data Table Operasi ini digunakan untuk mengambil data dari tarapi pada database yang ingin ditampilkan user. +editTerapi(): boolean Operasi ini digunakan untuk mengedit data terapi pada database. Mereturnkan true jika operasi berhasil. +addTerapi(): boolean Operasi ini digunakan untuk menambahkan data terapi yang diinputkan oleh user. Mereturnkan true jika operasi berhasil.	

2.2.3.33 Specific Design Class SentItem

SentItem	<<entity>>
-----------------	-------------------------------

<pre>-idSentItem: int -tglDikirim: date -isiPesan: string -status: string</pre>
<pre>+SentItem() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getSentItem(): Data Table Operasi ini digunakan untuk mengambil data dari sent item pada database yang ingin ditampilkan user. +deleteSentItem(): boolean Operasi ini digunakan untuk menghapus data sent item pada database. Mereturnkan true jika operasi berhasil. +addSentItem(): boolean Operasi ini digunakan untuk menambahkan data sent item yang diinputkan oleh user. Mereturnkan true jika operasi berhasil.</pre>

2.2.3.34 Specific Design Class Draft

Draft	<<entity>>
<pre>-idDraft: int -isiPesan: string</pre>	
<pre>+Draft() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDraft(): Data Table Operasi ini digunakan untuk mengambil data dari draft pada database yang ingin ditampilkan user. +editDraft(): boolean Operasi ini digunakan untuk mengedit data draft pada database. Mereturnkan true jika operasi berhasil. +addDraft(): boolean Operasi ini digunakan untuk menambahkan data draft yang diinputkan oleh user. Mereturnkan true jika operasi</pre>	

berhasil.
`+deleteDraft(): boolean`
 Operasi ini digunakan untuk menghapus data draft pada database.

2.2.3.35 Specific Design Class Inbox

Inbox	<<entity>>
<code>-idInbox: int</code> <code>-isiPesan: string</code> <code>-tglDikirim: date</code>	
<code>+Inbox()</code> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. <code>+getInbox(): Data Table</code> Operasi ini digunakan untuk mengambil data dari inbox pada database yang ingin ditampilkan user. <code>+deleteInbox(): boolean</code> Operasi ini digunakan untuk mengedit data inbox pada database. Mereturnkan true jika operasi berhasil. <code>+addInbox(): boolean</code> Operasi ini digunakan untuk menambahkan data inbox yang diinputkan oleh user. Mereturnkan true jika operasi berhasil.	

3 Perancangan Data

3.1 Dekomposisi Data

3.1.1 Deskripsi Entitas Data Account

Nama	Tipe	Panjang	Keterangan
USERNAME	Variable Character	20	Username user, Primary Key
PASSWORD	Variable Character	20	Password user
NAMA	Variable Character	30	Nama user
IDPERAN	Integer	-	ID Peran, Foreign Key

3.1.2 Deskripsi Entitas Peran

Nama	Tipe	Panjang	Keterangan
IDPERAN	Integer	-	ID peran, Primary Key
NAMAPERAN	Variable Character	30	Nama peran
ISDELETED	Boolean	-	Status peran, terhapus atau tidak

3.1.3 Deskripsi Entitas KarakteristikAnggota

Nama	Tipe	Panjang	Keterangan
IDANGGOTA	Integer	-	ID anggota, Primary Key
NAMA	Variable Character	30	Nama anggota
TANGGALLAHIR	Date	-	Tanggal lahir anggota
KELAMIN	Variable Character	2	Jenis kelamin anggota
PENDIDIKAN	Variable Character	30	Pendidikan terakhir anggota

STATUS	Variable Character	20	Status anggota, menikah atau belum
PEKERJAAN	Variable Character	30	Pekerjaan yang dimiliki anggota
LAMAMENDERITA	Integer	-	Tahun pertama divonis sakit DM
NOTELEPON	Variable Character	30	Nomor telepon anggota
ISDELETED	Boolean	-	Status anggota, terhapus atau tidak

3.1.4 Deskripsi Diet

Nama	Tipe	Panjang	Keterangan
IDDIET	Integer	-	ID diet, Primay Key
IDANGGOTA	Variable Character	30	ID anggota, Foreign Key
TGLDIBERIKAN	Boolean	-	Tanggal diet dikeluarkan/diberikan
PAGI	Integer	-	Diet pagi hari (dalam kalori)
SELINGANSIANG	Integer	-	Diet selingan siang hari (dalam kalori)
SIANG	Integer	-	Diet siang hari (dalam kalori)
SELINGANSORE	Integer	-	Diet selingan sore hari (dalam kalori)
MALAM	Integer	-	Diet malam hari (dalam kalori)

3.1.5 Deskripsi Terapi

Nama	Tipe	Panjang	Keterangan
IDTERAPI	Integer	-	ID terapi, Primary Key
IDANGGOTA	Integer	-	ID anggota, Foreign Key

TGLDIBERIKAN	Date	-	Tanggal terapi dikeluarkan/diberikan
NAMAOBAT	Variable Character	50	Nama-nama obat yang harus dikonsumsi
ATURANPAKAI	Variable Character	200	Aturan pakai dari obat yang diberikan

3.1.6 Deskripsi Entitas DailyCheck

Nama	Tipe	Panjang	Keterangan
IDCHECK	Integer	-	ID check, Primary Key
IDANGGOTA	Integer	-	ID anggota, Foreign Key
TGL	Date	-	Tanggal pelaporan kegiatan
KAKIPAGI	Variable Character	200	Keterangan pada perawatan kaki pagi
KAKISORE	Variable Character	200	Keterangan pada perawatan kaki sore
OLAHRAGA	Variable Character	200	Keterangan pada olahraga yang dilakukan
DIETPAGI	Variable Character	200	Keterangan pada diet pagi
DIETSELINGANSIANG	Variable Character	200	Keterangan pada diet selingan siang
DIETSIANG	Variable Character	200	Keterangan pada diet siang
DIETSELINGANSORE	Variable Character	200	Keterangan pada diet selingan sore
DIETMALAM	Variable Character	200	Keterangan pada diet malam

TERAPIPAGI	Variable Character	200	Keterangan pada terapi pagi
TERAPISIANG	Variable Character	200	Keterangan pada terapi selingan siang
TERAPISORE	Variable Character	200	Keterangan pada terapi sore
TERAPIMALAM	Variable Character	200	Keterangan pada terapi malam

3.1.7 Deskripsi Entitas CheckUp

Nama	Tipe	Panjang	Keterangan
IDCHECKUP	Integer	-	ID check up, Primary Key
IDANGGOTA	Integer	-	ID anggota, Foreign Key
TGLPERIKSA	Date	-	Tanggal pasien/anggota diperiksa
GDP	Float	-	Nilai Gula Darah Puasa dalam mg/dL
GDPP	Float	-	Nilai Gula Darah Post Prandial dalam mg/dL
GDS	Float	-	Nilai Gula Darah Sewaktu dalam mg/dL
HBA1C	Float	-	Nilai Hemoglobin dalam darah dalam %
LDL	Float	-	Nilai bawah lipoprotein dalam mg/dL
HDL	Float	-	Nilai atas lipoprotein dalam mg/dL
CHOLTOT	Float	-	Nilai kolesterol

			total dalam mg/dL
TRIGLISERID	Float	-	Nilai trigliserid dalam mg/dL
UREUM	Float	-	Nilai ureum dalam mg/dL
KREATININ	Float	-	Nilai kreatinin dalam mg/dL
TEKDARAHATAS	Float	-	Nilai atas tekanan darah dalam mmHg
TEKDARAHBAWAH	Float	-	Nilai bawah tekanan darah dalam mmHg
CEKMATA	Float	-	Keterangan hasil cek mata
CEKJANTUNG	Float	-	Keterangan hasil cek jantung

3.1.8 Deskripsi Entitas Inbox

Nama	Tipe	Panjang	Keterangan
IDINBOX	Integer	-	ID check, Primary Key
IDANGGOTA	Integer	-	ID anggota, Foreign Key
TGLDITERIMA	Date	-	Tanggal SMS diterima
ISIPESAN	Variable Character	200	Isi pesan

3.1.9 Deskripsi Entitas SentItem

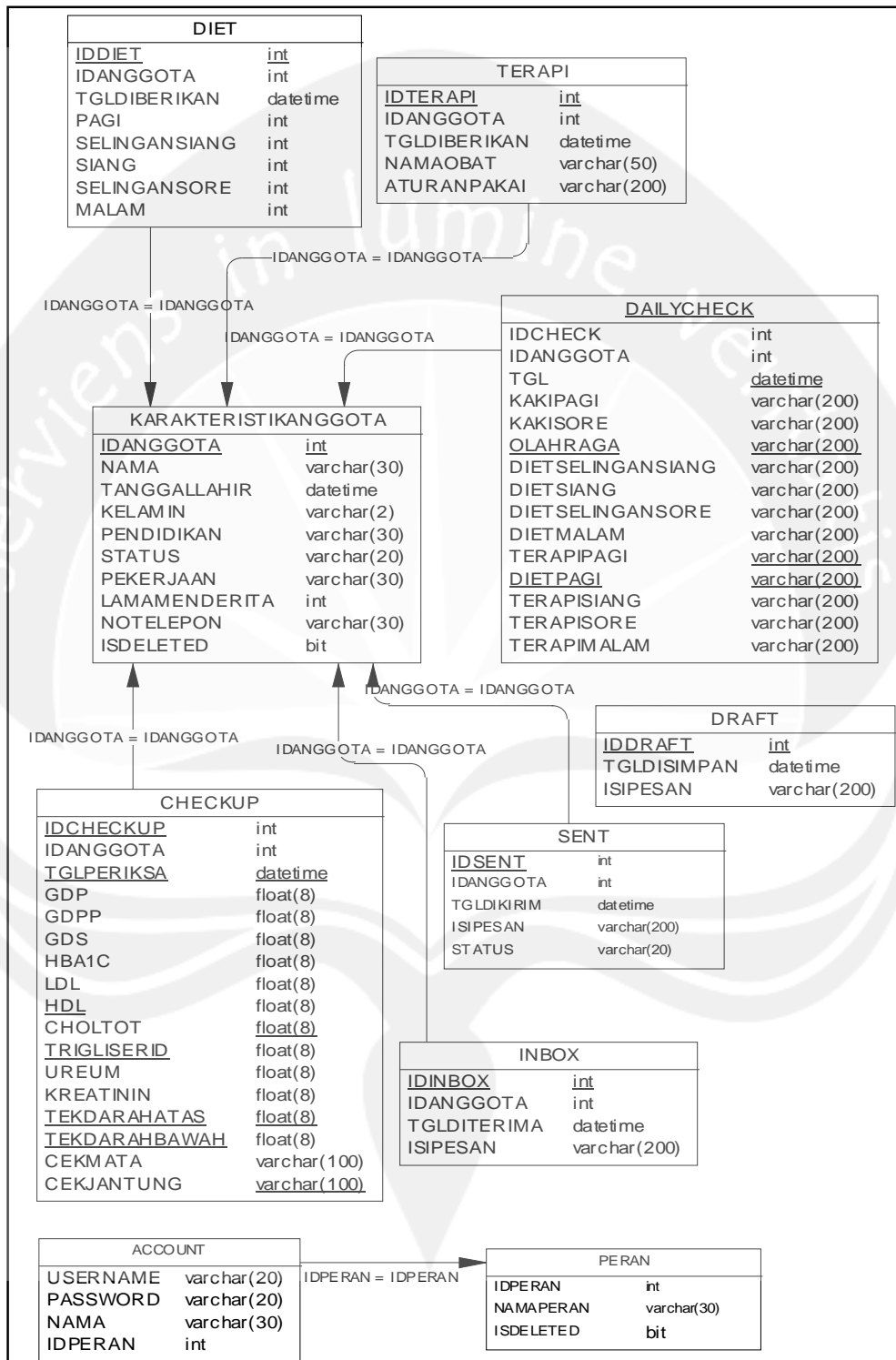
Nama	Tipe	Panjang	Keterangan
IDSENT	Integer	-	ID sent item, Primary Key
IDANGGOTA	Integer	-	ID anggota, Foreign Key
TGLDIKIRIM	Date	-	Tanggal SMS dikirim
ISIPESAN	Variable	200	Isi pesan

	Character		
STATUS	Variable Character	20	Status SMS, terkirim atau tidak

3.1.10 Deskripsi Entitas Draft

Nama	Tipe	Panjang	Keterangan
IDDRAFT	Integer	-	ID draft, Primary Key
TGLDISIMPAN	Date	-	Tanggal SMS disimpan
ISIPESAN	Variable Character	200	Isi pesan

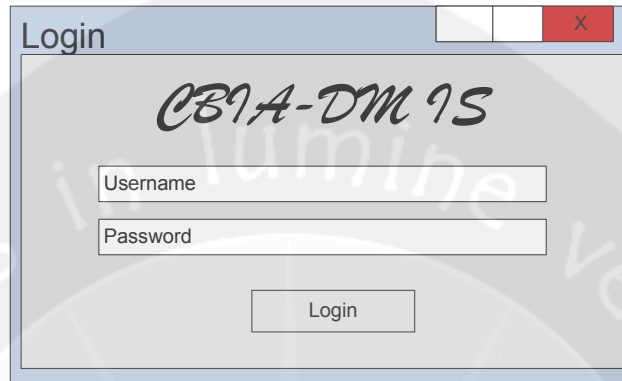
3.2 PDM



Gambar 3 Physical Data Model pengelolaan mobil GMIS Modul Mobil

4 Perancangan Antarmuka

4.1 Form Login (User)

The image shows a screenshot of a login window. The window has a title bar with the text 'Login' and a red close button with an 'X' icon. The main content area has a light gray background. At the top center, the text 'CBIADMIS' is written in a large, black, handwritten-style font. Below this, there are two text input fields. The first field is labeled 'Username' and the second is labeled 'Password'. Both fields are empty. Below the input fields, there is a single button labeled 'Login'.

Gambar 4.1 Rancangan Antarmuka Login (User)

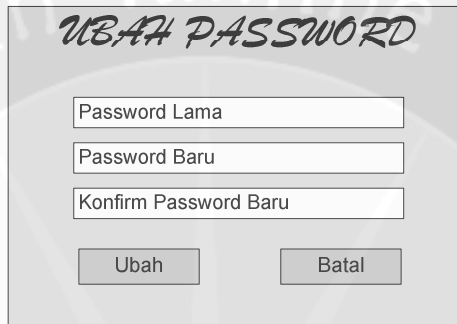
Form ini yang menggunakan adalah semua User yang memiliki akses ke dalam pengelolaan desktop. Form ini yang akan menjadi antarmuka bagi bagian User untuk dapat melakukan Login ke dalam sistem dan menggunakan aplikasi CBIADMIS. Button "Login" untuk verifikasi username dan password untuk masuk ke dalam sistem.

Untuk melakukan verifikasi password sistem mengambil inputan username dan password yang diberikan oleh user, mencarinya dalam database berdasarkan usernamanya dengan *query statement*.

Dengan username inputan yang menjadi parameter dan diambil dari tabel ACCOUNT. Jika menemukan username yang dimaksud dimana statusnya tidak terhapus, sistem akan mengambil data dan menampungnya dalam suatu variabel yang bertipe Account dan membawanya ke layer manager untuk kemudian dibandingkan antara password yang diinputkan dengan password yang di dapat dari dalam database. Maka dari itu username haruslah unik.

Setelah berhasil login, sistem akan menyimpan data-data login yang digunakan user ke dalam variabel bertipe Account.

4.2 Form Ubah Password (User)



The image shows a web form titled "UBAH PASSWORD". It features three text input fields stacked vertically: "Password Lama", "Password Baru", and "Konfirm Password Baru". At the bottom of the form, there are two buttons: "Ubah" and "Batal". The form is set against a light gray background with a faint watermark of a university logo and the motto "serviens in lumina veritatis".

Gambar 4.2 Rancangan Antarmuka Ubah Password (User)

Pada form ini user dapat melakukan perubahan password untuk proteksi account yang dimilikinya agar tidak sembarang orang dapat melakukan login menggunakan accountnya dan masuk ke dalam sistem GMIS.

Untuk melakukan pengubahan password ini ada 2 langkah, yang pertama agar tidak sembarang orang dapat mengganti user yang bersangkutan maka sistem akan meminta password lama (password yang masih berlaku). Sistem kemudian membandingkan password inputan tersebut dengan password yang memang digunakan user untuk login di awal saat user menjalankan sistem. Setelah itu, langkah yang kedua sistem akan merubah password user yang bersangkutan dengan password baru yang telah diinputkan dengan *query statement* sebagai berikut:

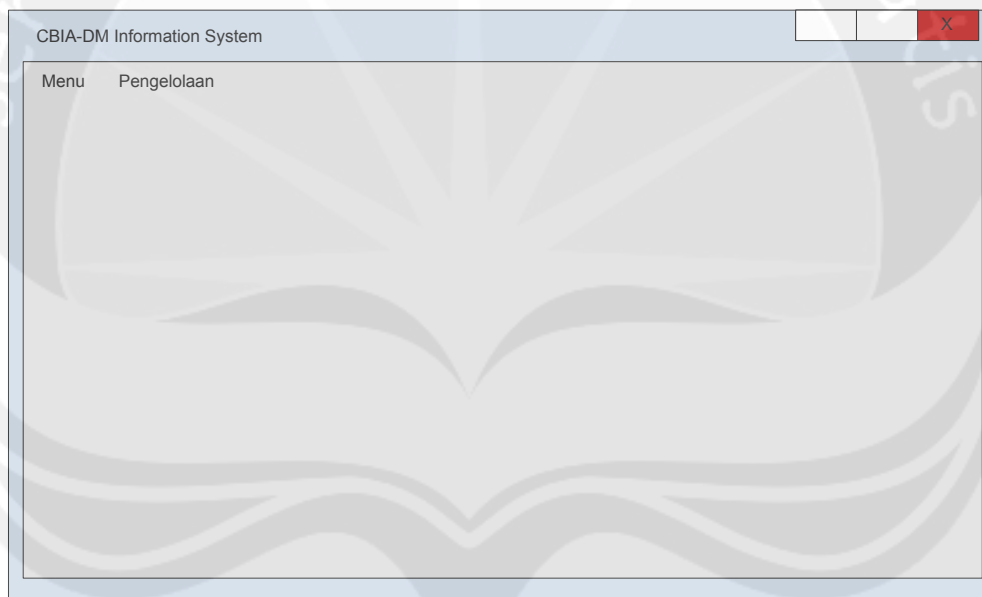
UPDATE ACCOUNT

Program Studi Teknik Informatika	DPPL – GMIS	81/ 88
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

```
SET PASSWORD = @PASSWORD  
WHERE IDACCOUNT = @IDACCOUNT
```

Dengan menggunakan password baru sebagai parameter dan idaccount sebagai parameter *identifier* yang di-update-kan pada tabel ACCOUNT. Data password diambil dari inputan user sedangkan idaccount diambil dari data login user.

4.3 Form Utama (User)



Gambar 4.3 Rancangan Antarmuka Form Utama (User)

Form ini yang akan tampil setelah user melakukan login. Pada contoh gambar di atas user login sebagai pembuat sistem jadi semua opsi dari menu bar: Menu (user), Pengelolaan (Control dan Administrator), dan Laporan (Control) dapat dilihat.

4.4 Form Pengelolaan Account (Administrator)

The screenshot shows a web application window titled "CIBIA-DM Information System -----Pengelolaan Akun-----". The interface is divided into several sections. On the left, there is a "Menu" section with "Pengelolaan" selected. Below it is the "Akun Baru" form, which includes input fields for "Username", "Nama", "ID Peran", and "Peran", along with a "Pilih" dropdown and a "Simpan" button. To the right of the "Akun Baru" form is a "Picture" label. Below the "Akun Baru" form is a "Tabel Referensi Peran" table. On the right side of the window, there is a "Pencarian Akun" section with a "Kata Kunci" input field, a "berdasar" dropdown menu, and a "Cari" button. Below this is a "Tabel Akun" table. At the bottom right, there are three buttons: "Hapus", "Refresh", and "Keluar".

Gambar 4.4 Rancangan Antarmuka Pengelolaan Account (Administrator)

Form ini ditujukan untuk bagian administrator agar dapat melakukan pengelolaan terhadap account pengguna CBIADMIS. Pada pengelolaan ini user dapat melakukan *add data*, *edit data*, *delete data*, *display data*, dan *search data*. Form dengan menu ini hanya dapat diakses oleh user yang login sebagai administrator.

4.5 Form Pengelolaan Anggota (Control)

The screenshot shows a web-based control form for member management. The interface includes a menu bar with 'Pengelolaan' selected, a search section for finding members, and a form for adding new members with various personal data fields.

Gambar 4.5 Rancangan Antarmuka Pengelolaan Anggota (Control)

Form ini ditujukan untuk bagian operasional agar dapat melakukan pengelolaan terhadap anggota. Pada pengelolaan ini user dapat melakukan *add data*, *edit data*, *delete data*, *display data*, dan *search data*. Namun, pada form ini user hanya dapat melakukan *add*, *display* dan *search* anggota. Untuk dapat melakukan *edit* ataupun *delete* pada anggota tertentu, user harus memilihnya terlebih dahulu dari table dan melakukannya pada form baru. Form dengan menu ini hanya dapat diakses oleh user yang login sebagai operasional.

4.6 Form Pasien (Control)

Diet Pasien	
Tanggal Diberikan	: 9/5/2011 12:00:00 AM
Pagi	: 550 Kalori
Selingan Siang	: 100 Kalori
Siang	: 550 Kalori
Selingan Sore	: 100 Kalori
Malam	: 550 Kalori

Terapi Pasien	
Tanggal Diberikan	: 6/5/2011 3:45:00 PM
Nama Obat	: Bioplasenton
Aturan Pakai	: 3x sehari

Gambar 4.6 Rancangan Antarmuka Pasien (Control)

Form ini yang diakses oleh Control untuk melakukan pengelolaan Anggota seperti: *Edit* dan *Delete* data akun pasien/anggota yang bersangkutan. Tentu saja form ini hanya dapat diakses jika user telah login ke dalam sistem sebagai control. Pada form ini user dapat juga melakukan pengelolaan terhadap Diet, Terapi, Check Up, serta Daily Check yang dimiliki oleh pasien. Pada form ini data-data tersebut akan ditampilkan sebagai display seperti rekam medis yang menyertai anggota.

Untuk melakukan pengelolaan lebih lanjut untuk Diet, Terapi, Check Up, serta Daily Check yang dimiliki anggota dapat dilakukan pada form berikutnya, yang akan muncul jika tombol ditekan.

4.7 Form Pengelolaan Diet (Control)

The screenshot shows a window titled "Diet Ignatius" with a standard Windows-style title bar. The window is divided into several sections:

- Data Baru Pasien:** A form for entering new patient diet data. It includes fields for "ID Diet", "Tanggal Diberikan" (set to "Monday, May 9, 2011"), and five rows for meal types: "Pagi", "Selingan Siang", "Siang", "Selingan Siang", and "Malam". Each row has a text input field and a "Kalori" label. A "Simpan" button is at the bottom of this section.
- Pencarian Diet:** A search section with a "Kata Kunci" input field, a "berdasar:" label, and a "Cari" button.
- Rekam Diet Pasien:** A large, empty rectangular area intended for displaying patient diet records.
- Buttons:** "Refresh" and "Tutup" buttons are located at the bottom right of the window.

Gambar 4.7 Rancangan Antarmuka Pengelolaan Diet (Control)

Pada form pengelolaan Diet ini Control dapat melakukan pemasukkan data diet pasien yang baru, *edit* data diet, *display*, serta *search*. *Delete* atau penghapusan tidak diperkenankan karena ini menyangkut *history* anggota yang bersangkutan dan hal ini sangat krusial. Sebenarnya pengeditan pun sangat diantisipasi bahkan jika memang dilakukan pengeditan, user akan ditanyai oleh sistem apakah benar-benar ingin mengedit.

4.8 Form Pengelolaan Terapi (Control)

Terapi Ignatius

Terapi Baru Pasien

ID Terapi : ID Terapi
Tanggal Diberikan : Friday, May 6, 2011
Nama Obat :
Aturan Pakai :

Simpan

Pencarian Terapi

Kata Kunci: [] berdasar: [] Cari

Rekam Terapi Pasien

Refresh Tutup

Gambar 4.8 Rancangan Antarmuka Pengelolaan Terapi (Control)

Pada form pengelolaan Terapi ini, user dapat melakukan pengelolaan seperti *add*, *edit*, *display*, serta *search* data terapi yang dimiliki oleh pasien yang bersangkutan. Sama seperti pengelolaan pada data Diet, pengelolaan pada data Terapi ini juga tidak diperkenankan melakukan *delete* terhadap data yang ada karena pengeloaan ini berfungsi sebagai rekam medis anggota yang bahkan *edit* data pun juga sangat diantisipasi. Sama seperti form pengelolaan diet, form ini hanya dapat diakses user yang login dengan peran Control.

4.9 Form Pengelolaan Check Up (Control)

Check Up Ignatius

Check Up Baru Pasien

ID Ceck Up : ID Check Up

Tanggal Periksa :

GDP : mg/dL

GDPP : mg/dL

GDS : %

HBA1C : mg/dL

LDL : mg/dL

HDL : mg/dL

Kolesterol Total : mg/dL

Trigliseric : mg/dL

Ureum : mg/dL

Kreatinin : mg/dL

Tek. Darah Atas : mmHg

Tek. Darah Bawah : mmHg

Cek Mata :

Cek Jantung :

Simpan

Pencarian Check Up

Kata Kunci : berdasar : Cari

Rekam Check Up Pasien

Refresh

Tutup

Gambar 4.9 Rancangan Antarmuka Pengelolaan Check Up (Control)

Pada form pengelolaan perawatan ini Control dapat melakukan pengelolaan terhadap data Check Up yang dimiliki oleh anggota yang bersangkutan. Control dapat melakukan: *add*, *edit*, *display*, atau pun *search* Check Up tertentu. Sama halnya pada form-form sebelumnya, penghapusan tidak diperkenankan. Edit pun juga sangat diantisipasi.

SKPL

SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK

CBIADMIS

(CBIA-DM Information System)

Untuk:


Kelompok CBIA-DM

Dipersiapkan oleh:

Reinardus Kristyanto / 070705198

Program Studi Teknik Informatika - Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		SKPL-CBIADMIS		1/51
		Revisi		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh	RK							
Diperik sa oleh	IW, SP							
Disetuj ui oleh	IW, SP							

Daftar Halaman Perubahan

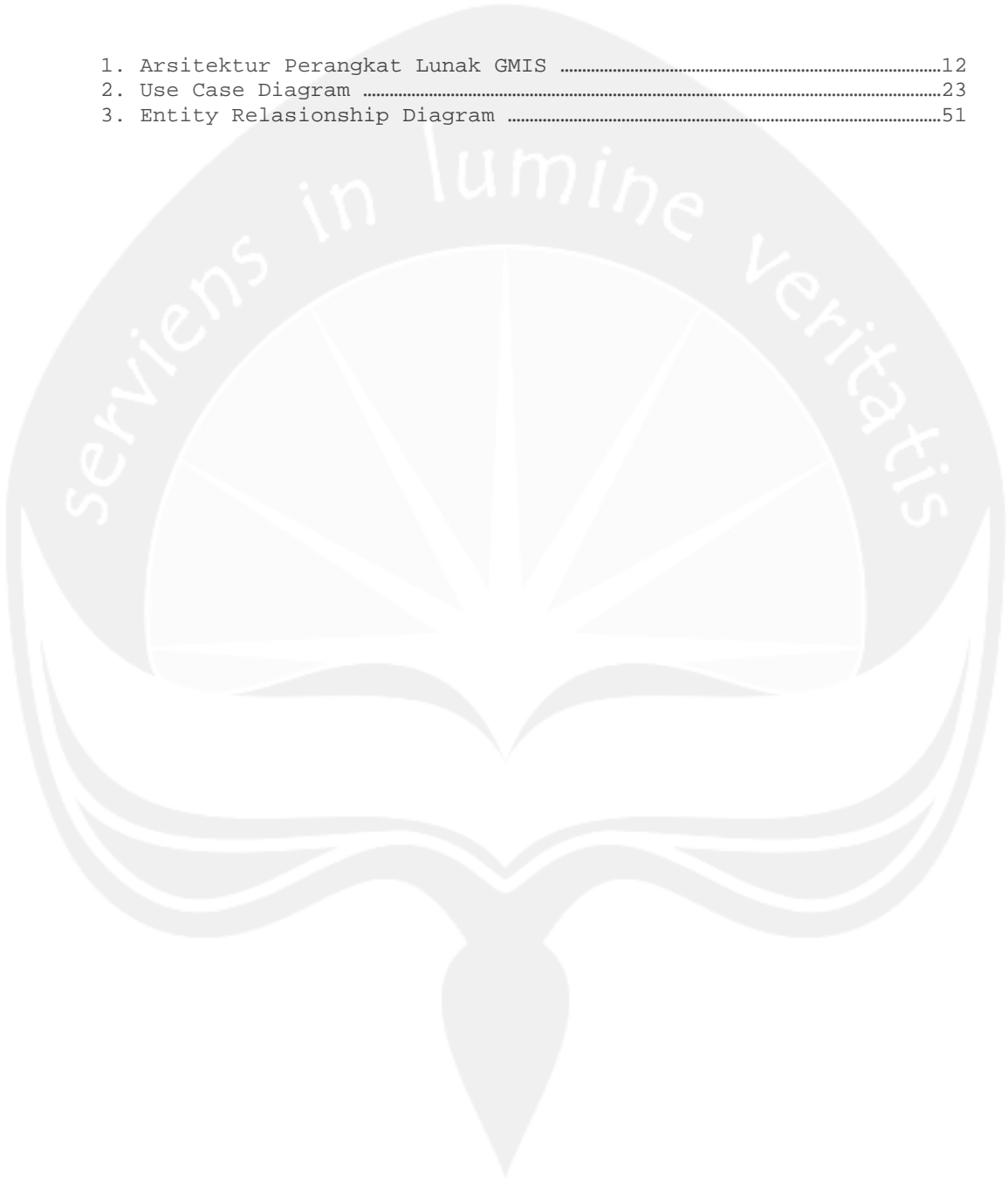
Halaman	Revisi	Halaman	Revisi

Daftar Isi

1	Pendahuluan	6
1.1	Tujuan	6
1.2	Lingkup Masalah.....	6
1.3	Definisi, Akronim dan Singkatan.....	7
1.4	Referensi	9
1.5	Deskripsi umum (Overview)	10
2	Deskripsi Kebutuhan	11
2.1	Perspektif produk.....	11
2.2	Fungsi Produk.....	12
2.3	Karakteristik Pengguna	20
2.4	Batasan-batasan.....	21
2.5	Asumsi dan Ketergantungan.....	21
3	Kebutuhan khusus	21
3.1	Kebutuhan antarmuka eksternal.....	21
3.2	Kebutuhan fungsionalitas Perangkat Lunak.....	23
4	Spesifikasi Rinci Kebutuhan.....	24
4.1	Spesifikasi Kebutuhan Fungsionalitas.....	24
5	Entity Relationship Diagram (ERD).....	51

Daftar Gambar

1. Arsitektur Perangkat Lunak GMIS	12
2. Use Case Diagram	23
3. Entity Relationship Diagram	51



1 Pendahuluan

1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak CBIADMIS (CBIA-DM Information System) untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak dan perangkat keras, dan pengguna), tempat penyimpanan yang dibutuhkan, serta keakuratan), dan atribut (*feature-feature* tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-CBIADMIS ini juga mendefinisikan batasan perancangan perangkat lunak.

1.2 Lingkup Masalah

Perangkat Lunak CBIADMIS dikembangkan dengan tujuan untuk:

1. Menangani login.
2. Menangani logout.
3. Menangani pengelolaan data account.
4. Menangani ubah password.
5. Menangani pengelolaan data anggota.
6. Menangani pengelolaan check up.
7. Menangani pengelolaan daily check.
8. Menangani pengelolaan terapi.
9. Menangani pengelolaan diet.
10. Menangani pengelolaan SMS.
11. Menangani pelaporan list yang belum melakukan check-up.
12. Menangani scheduled and alert SMS.
13. Menangani autorespond SMS.

14. Menangani registrasi anggota.

15. Menangani unregistrasi anggota.

Perangkat Lunak ini berbasisan desktop yang menangani pengelolaan layanan sms. Perangkat Lunak dibangun dan berjalan pada lingkungan dengan platform Windows 7.

1.3 Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
SKPL	Merupakan spesifikasi kebutuhan dari perangkat lunak yang akan dikembangkan.
SKPL-CBIADMIS-XXX	Kode yang merepresentasikan kebutuhan pada CBIADMIS (CBIA-DM Information System) dimana XXX merupakan nomor fungsi produk.
CBIADMIS	Perangkat lunak yang menangani pengelolaan kelompok penderita Diabetes Mellitus dengan basis CBIA (CBIA-DM) melalui sms yang berbasis desktop.
CBIA	Metode pembelajaran yang menuntut keaktifan dari si pembelajar.
DM	Singkatan dari Diabetes Mellitus. Nama penyakit peningkatan kadar gula dalam darah yang disebabkan kurangnya insulin.
CBIA-DM	Metode pembelajaran CBIA yang diterapkan untuk para penderita DM.
Gula Darah Puasa	Gula Darah Puasa (Fasting Plasma Glucose) adalah kadar gula darah pada saat puasa. Dengan satuan mg/dL.

GDPP	Gula Darah Post Prandial (Post Prandial Glucose) adalah kadar gula darah pada saat 2 jam sesudah makan. Dengan satuan mg/dL.
GDS	Gula Darah Sewaktu (Random Blood Glucose) adalah kadar gula darah random. Dengan satuan mg/dL.
HbA1C	Kadar gula darah dalam darah. Ditunjukkan dalam prosentase (%).
LDL	Nilai atas kadar lipoprotein dalam darah. Satuan yang digunakan adalah mg/dL.
HDL	Nilai bawah kadar lipoprotein dalam darah. Satuan yang digunakan adalah mg/dL.
Cholesterol Total	Total kolesterol yang terdapat dalam darah. Dengan satuan mg/dL.
Trigliserid	Kadar lemak yang ada dalam darah. Satuan yang digunakan adalah mg/dL.
Ureum	Hasil metabolisme protein, terdapat dalam darah. Satuan yang digunakan adalah mg/dL.
Kreatinin	Merupakan zat racun dalam darah, produk sisa dari perombakan keratin fosfat yang terjadi dalam otot. Satuan yang digunakan mg/dL.

Role	Peran yang disediakan oleh sistem untuk masing-masing user.
Previllege	Wewenang yang dimiliki oleh user yang diberikan oleh sistem.
User	Semua pengguna komputer yang berhubungan langsung dengan sistem.
Administrator	User yang memiliki role sebagai pengelola akun dalam sistem.
Anggota	User yang memilki role sebagai anggota penderita DM dalam sistem.
Control	User yang memiliki role sebagai pengawas yang melakukan pengawasan terhadap keteraturan check-up anggota serta dapat melakukan pengiriman sms manual peringatan lanjutan.
Desktop	Istilah yang digunakan saat user menggunakan komputer yang tidak memiliki koneksi ke internet.
Account	Istilah bahasa Inggris yang digunakan untuk menyebut kata akun.

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Reinardus Kristyanto. *Spesifikasi Kebutuhan Perangkat Lunak GMIS*, Universitas Atma Jaya Yogyakarta, 2010.
2. Reinardus Kristyanto. *Spesifikasi Kebutuhan Perangkat Lunak Soft*, Universitas Atma Jaya Yogyakarta, 2010.

3. Reinardus Kristyanto. *Spesifikasi Kebutuhan Perangkat Lunak DC-CRIS*, Universitas Atma Jaya Yogyakarta, 2010.
4. Bennet Simon, McRobb Steve, Farmer Ray, *Object-Oriented System Analysis and Design Using UML*, McGraw-Hill Companies, 2002.
5. Boggs Wendy, Boggs Michael, *Mastering UML with Rational Rose 2002*, SYBEX Inc, 2002.
6. Ariyanto, dkk. , *Pengembangan Aplikasi Sistem Informasi Akademik Berbasis SMS dengan Java*, Penerbit Salemba Infotek, 2005.

1.5 Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas 5 bagian utama. Bagian utama berisi penjelasan mengenai dokumen SKPL tersebut yang mencakup tujuan pembuatan SKPL, ruang lingkup masalah dalam pengembangan perangkat lunak tersebut, definisi, referensi dan deskripsi umum tentang dokumen SKPL ini.

Bagian kedua berisi penjelasan umum tentang perangkat lunak CBIADMIS yang akan dikembangkan, mencakup perspektif produk yang akan dikembangkan, fungsi produk perangkat lunak, karakteristik pengguna, batasan dalam penggunaan perangkat lunak dan asumsi yang dipakai dalam pengembangan perangkat lunak CBIADMIS tersebut.

Bagian ketiga berisi penjelasan secara lebih rinci tentang kebutuhan perangkat lunak CBIADMIS yang akan dikembangkan yang meliputi kebutuhan antarmuka eksternal serta kebutuhan fungsionalitas perangkat lunak dari sistem tersebut.

Program Studi Teknik Informatika	SKPL – CBIADMIS	10/ 51
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Bagian keempat berisi rincian spesifikasi kebutuhan dari sistem yang terdiri dari spesifikasi kebutuhan fungsionalitas dimana merupakan penjabaran dari use case diagram yang terdapat pada bagian tiga sebelumnya. Di bagian ini fungsionalitas dari setiap use case dijelaskan secara detail satu-persatu.

Bagian kelima berisi penggambaran Entity Relationship Diagram (ERD) yang merupakan gambaran sistem secara garis besar dilihat dari sudut pandang relasi antar entitas yang ada dalam CBIADMIS tersebut.

2 Deskripsi Kebutuhan

2.1 Perspektif produk

CBIADMIS merupakan perangkat lunak yang dikembangkan untuk membantu pengelolaan sebuah kelompok penderita DM yang dibina secara CBIA (CBIA-DM) dengan basis desktop yang memanfaatkan system layanan sms untuk keperluan komunikasi kelompok tersebut. Sistem ini menangani pengelolaan data account, pengelolaan ubah password, pengelolaan data anggota, pengelolaan pendaftaran anggota, pengelolaan pemberhentian anggota, pengelolaan sms, pengelolaan autorespond sms, pengelolaan pengiriman scheduled dan alert sms, pengelolaan pengiriman sms manual, serta pengelolaan pelaporan list yang belum melakukan check-up.

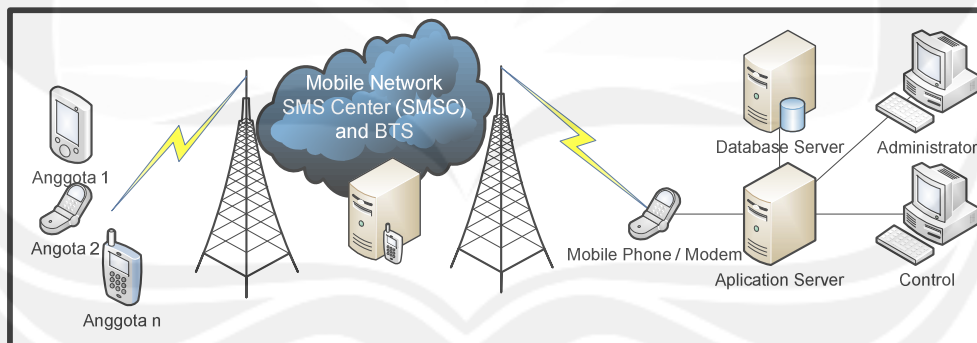
Perangkat lunak CBIADMIS ini berjalan pada platform Windows yang berbasis desktop, dan dibuat menggunakan bahasa pemrograman C#. Sedangkan untuk lingkungan pemrogramannya menggunakan Microsoft Visual Studio 2005.

Program Studi Teknik Informatika	SKPL – CBIADMIS	11/ 51
----------------------------------	-----------------	--------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

Pengguna akan berinteraksi dengan sistem melalui antarmuka GUI (Graphical User Interface). Pada sistem ini, seperti terlihat pada gambar 1, arsitektur perangkat lunak yang digunakan berupa 3 layer application, yaitu *User Interface Layer (Boundary)*, *Management Layer (Control)*, dan *Data Access Object Layer (DAO)*, di mana kemudian semua data disimpan di database server. Karena sistem ini berbasis desktop maka user dapat mengakses data pada database server secara langsung melalui application server.

Inputan data yang dimasukkan akan disimpan dalam database server, sehingga jika ada pencarian data, maka data yang diinginkan akan dicari ke database server yang selanjutnya dikirimkan ke client yang me-request langsung melalui application server.



Gambar 1. Arsitektur Perangkat lunak CBIADMIS.

2.2 Fungsi Produk

Fungsi produk perangkat lunak CBIADMIS adalah sebagai berikut:

1. Fungsi Login (SKPL-CBIADMIS-001)

Merupakan fungsi yang digunakan oleh seluruh pengguna sistem untuk dapat masuk dalam sistem

sehingga dapat mengakses fungsi yang akan digunakan.

2. Fungsi *Logout* (**SKPL-CBIADMIS-002**)

Merupakan fungsi yang dapat digunakan oleh seluruh pengguna sistem untuk dapat keluar dari dalam sistem sehingga dapat login sebagai user yang berbeda.

3. Fungsi *Pengelolaan Account* (**SKPL-CBIADMIS-003**)

Merupakan fungsi yang digunakan untuk mengelola data user untuk keperluan login ke dalam sistem.

Fungsi *Pengelolaan Account* mencakup:

a. Fungsi *Add Account* (**SKPL-CBIADMIS-003-01**)

Merupakan fungsi yang digunakan untuk menambahkan data account user yang baru.

b. Fungsi *Edit Account* (**SKPL-CBIADMIS-003-02**)

Merupakan fungsi yang digunakan untuk mengubah data account user.

c. Fungsi *Delete Account* (**SKPL-CBIADMIS-003-03**)

Merupakan fungsi yang digunakan untuk menghapus data account user.

d. Fungsi *Display Account* (**SKPL-CBIADMIS-003-04**)

Merupakan fungsi yang digunakan untuk menampilkan data account user.

e. Fungsi *Search Account* (**SKPL-CBIADMIS-003-05**)

Merupakan fungsi yang digunakan untuk menampilkan data account user tertentu.

4. Fungsi *Ubah Password* (**SKPL-CBIADMIS-004**)

Program Studi Teknik Informatika	SKPL – CBIADMIS	13/ 51
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Merupakan fungsi yang digunakan untuk melakukan pengelolaan ubah password. Informasi yang dikelola adalah perubahan password oleh user yang bersangkutan.

5. Fungsi *Pengelolaan Anggota* (**SKPL-CBIADMIS-005**)

Merupakan fungsi yang digunakan untuk melakukan pengelolaan data anggota. Informasi yang dikelola mencakup nama, umur, jenis kelamin, tingkat pendidikan, status, pekerjaan, serta lama menderita.

Fungsi *Pengelolaan Anggota* mencakup:

a. Fungsi *Add Anggota* (**SKPL-CBIADMIS-005-01**)

Merupakan fungsi yang digunakan untuk menambahkan data anggota baru ke dalam database.

b. Fungsi *Edit Anggota* (**SKPL-CBIADMIS-005-02**)

Merupakan fungsi yang digunakan untuk mengubah data anggota yang telah tersimpan dalam database dan menggantinya dengan informasi yang baru.

c. Fungsi *Delete Anggota* (**SKPL-CBIADMIS-005-03**)

Merupakan fungsi yang digunakan untuk menghapus data-data anggota yang tersimpan dalam database.

d. Fungsi *Display Anggota* (**SKPL-CBIADMIS-005-04**)

Merupakan fungsi yang digunakan untuk menampilkan data-data anggota yang tersimpan dalam database.

e. Fungsi *Search Anggota* (**SKPL-CBIADMIS-005-05**)

Merupakan fungsi yang digunakan untuk menampilkan data-data anggota tertentu yang tersimpan dalam database.

6. Fungsi *Pengelolaan Check Up* (**SKPL-CBIADMIS-006**)

Merupakan fungsi yang digunakan untuk mengelola data detil kondisi klinis dari para anggota. Informasi yang dikelola mencakup tekanan darah, GPP, GDS, HbA1C, LDL, HDL, kolesterol total, Triglisericid, Ureum, Kreatinin, ditambah pula dengan hasil check jantung dan mata.

Fungsi *Pengelolaan Check Up* mencakup:

a. Fungsi *Add Check Up* (**SKPL-CBIADMIS-006-01**)

Merupakan fungsi yang digunakan untuk menambahkan data kondisi klinis yang baru.

b. Fungsi *Edit Check Up* (**SKPL-CBIADMIS-006-02**)

Merupakan fungsi yang digunakan untuk mengubah data kondisi klinis yang ada pada database.

c. Fungsi *Display Check Up* (**SKPL-CBIADMIS-006-03**)

Merupakan fungsi yang digunakan untuk menampilkan data kondisi klinis yang telah tersimpan pada database.

d. Fungsi *Search Check Up* (**SKPL-CBIADMIS-006-04**)

Merupakan fungsi yang digunakan untuk menampilkan data kondisi klinis tertentu yang telah tersimpan dalam database.

7. Fungsi *Pengelolaan Daily Check* (**SKPL-CBIADMIS-007**)

Merupakan fungsi yang digunakan untuk mengelola data detil kondisi fisik dari para anggota. Informasi yang dikelola mencakup perawatan kaki, olahraga, serta penjalanan diet dan terapi keseharian.

Fungsi *Pengelolaan Daily Check* mencakup:

a. Fungsi *Add Daily Check* (**SKPL-CBIADMIS-007-01**)

Merupakan fungsi yang digunakan untuk menambahkan data kondisi fisik yang baru.

b. Fungsi *Edit Daily Check* (**SKPL-CBIADMIS-007-02**)

Merupakan fungsi yang digunakan untuk mengubah data kondisi fisik yang ada pada database.

c. Fungsi *Display Daily Check* (**SKPL-CBIADMIS-007-03**)

Merupakan fungsi yang digunakan untuk menampilkan data kondisi fisik yang telah tersimpan pada database.

d. Fungsi *Search Daily Check* (**SKPL-CBIADMIS-007-04**)

Merupakan fungsi yang digunakan untuk menampilkan data kondisi fisik tertentu yang telah tersimpan dalam database.

8. Fungsi *Pengelolaan Terapi* (**SKPL-CBIADMIS-008**)

Merupakan fungsi yang digunakan untuk mengelola data detil terapi dari para anggota. Informasi

yang dikelola mencakup nama obat dan aturan pakai.

Fungsi Pengelolaan Terapi mencakup:

a. Fungsi *Add Terapi* (**SKPL-CBIADMIS-008-01**)

Merupakan fungsi yang digunakan untuk menambahkan data terapi yang baru.

b. Fungsi *Edit Terapi* (**SKPL-CBIADMIS-008-02**)

Merupakan fungsi yang digunakan untuk mengubah data terapi yang ada pada database.

c. Fungsi *Display Terapi* (**SKPL-CBIADMIS-008-03**)

Merupakan fungsi yang digunakan untuk menampilkan data terapi yang telah tersimpan pada database.

d. Fungsi *Search Terapi* (**SKPL-CBIADMIS-008-04**)

Merupakan fungsi yang digunakan untuk menampilkan data terapi tertentu yang telah tersimpan dalam database.

9. Fungsi *Pengelolaan Diet* (**SKPL-CBIADMIS-009**)

Merupakan fungsi yang digunakan untuk mengelola data detil diet dari para anggota. Informasi yang dikelola mencakup pagi, selingan siang, siang, selingan sore, dan malam.

Fungsi Pengelolaan Diet mencakup:

a. Fungsi *Add Diet* (**SKPL-CBIADMIS-009-01**)

Merupakan fungsi yang digunakan untuk menambahkan data diet yang baru.

b. Fungsi *Edit Diet* (**SKPL-CBIADMIS-009-02**)

Merupakan fungsi yang digunakan untuk mengubah data diet yang ada pada database.

c. Fungsi *Display Diet* (**SKPL-CBIADMIS-009-03**)

Merupakan fungsi yang digunakan untuk menampilkan data diet yang telah tersimpan pada database.

d. Fungsi *Search Diet* (**SKPL-CBIADMIS-009-04**)

Merupakan fungsi yang digunakan untuk menampilkan data diet tertentu yang telah tersimpan dalam database.

10. Fungsi *Pengelolaan SMS* (**SKPL-CBIADMIS-010**)

Merupakan fungsi yang digunakan untuk mengelola data sms yang masuk maupun yang keluar.

Fungsi Pengelolaan SMS mencakup:

a. Fungsi *New SMS* (**SKPL-CBIADMIS-010-01**)

Merupakan fungsi yang digunakan untuk membuat sms baru yang kemudian akan dikirimkan.

b. Fungsi *Inbox* (**SKPL-CBIADMIS-010-02**)

Merupakan fungsi yang digunakan untuk mengelola inbox (kotak masuk).

c. Fungsi *Sent Item* (**SKPL-CBIADMIS-010-03**)

Merupakan fungsi yang digunakan untuk mengelola pesan-pesan yang telah terkirim.

d. Fungsi *Draft* (**SKPL-CBIADMIS-010-04**)

Merupakan fungsi yang digunakan untuk mengelola pesan-pesan yang akan dikirimkan.

e. Fungsi *Deleted* (**SKPL-CBIADMIS-010-05**)

Merupakan fungsi yang digunakan untuk mengelola pesan-pesan yang akan dihapus. Biasa juga disebut tempat sampah.

11. Fungsi *Pembuatan List yang Belum Check-up* (**SKPL-CBIADMIS-011**)

Merupakan fungsi yang digunakan untuk melakukan pendataan pasien-pasien yang belum melakukan check-up hingga pada waktu yang ditentukan.

12. Fungsi *Scheduled and Alert SMS* (**SKPL-CBIADMIS-012**)

Merupakan fungsi yang digunakan untuk mengirimkan pesan-pesan yang akan dikirimkan secara berkala baik berupa informasi maupun peringatan kepada anggota.

13. Fungsi *Autorespond SMS* (**SKPL-CBIADMIS-013**)

Merupakan fungsi yang digunakan untuk membalas secara otomatis pesan-pesan yang diterima system yang dikirim oleh anggota.

14. Fungsi *Registrasi Anggota* (**SKPL-CBIADMIS-014**)

Merupakan fungsi yang digunakan untuk melakukan pendaftaran keanggotaan ke dalam system yang dimana kemudian data-datanya disimpan ke dalam database.

15. Fungsi *Unregistrasi Anggota* (**SKPL-CBIADMIS-015**)

Merupakan fungsi yang digunakan untuk melakukan pemberhentian keanggotaan dari system.

2.3 Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak CBIADMIS adalah sebagai berikut:

1. Control

- a. Memahami penggunaan komputer terutama dengan sistem operasi Windows XP.
- b. Memahami alur login dari sebuah sistem.
- c. Memahami dasar-dasar pengelolaan.
- d. Memahami pembacaan berbagai laporan.
- e. Memahami aktivitas dari CBIA-DM.

2. Administrator

- a. Memahami penggunaan komputer terutama dengan sistem operasi Windows XP.
- b. Memahami tentang basisdata dan manajemennya terutama SQL Server 2005 serta penggunaannya.
- c. Memahami alur login dari sebuah sistem.
- d. Memahami dasar-dasar pengelolaan account.

3. Anggota

- a. Memahami penggunaan ponsel terutama untuk penggunaan SMS.
- b. Memahami alur login dari sebuah system yang berbasis SMS.
- c. Memahami pembacaan berbagai informasi yang diberikan oleh system yang berbasis SMS.

2.4 Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak CBIADMIS tersebut adalah:

1. Kebijaksanaan Umum

Berpedoman pada tujuan dari pengembangan perangkat lunak CBIADMIS.

2. Keterbatasan perangkat keras

Dapat diketahui kemudian setelah sistem ini berjalan (sesuai dengan kebutuhan).

2.5 Asumsi dan Ketergantungan

Untuk sistem ini yang berbasis desktop dapat dijalankan pada komputer yang berplatform Windows XP.

3 Kebutuhan khusus

3.1 Kebutuhan antarmuka eksternal

Kebutuhan antar muka eksternal pada perangkat lunak CBIADMIS meliputi kebutuhan antarmuka pemakai, antarmuka perangkat keras, dan antarmuka perangkat lunak.

3.1.1 Antarmuka pemakai

Pengguna berinteraksi dengan antarmuka yang ditampilkan dalam bentuk form-form.

3.1.2 Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak CBIADMIS adalah:

1. Perangkat komputer (PC, Laptop, dll).

2. Perangkat mobile (Smart Phone, Pocket PC, PDA, dll) yang bisa digunakan untuk SMS.

3.1.3 Antarmuka perangkat lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak GMIS pada sisi server adalah sebagai berikut:

1. Nama : SQL Server 2005
Sumber : Microsoft
Sebagai database management system (DBMS) yang digunakan untuk penyimpan data di sisi server.
2. Nama : Windows XP
Sumber : Microsoft.
Sebagai sistem operasi untuk perangkat komputer.
3. Nama : .NET Framework 2.0
Sumber : Microsoft.
Sebagai framework untuk membangun aplikasi.

Sementara pada sisi client tool-tool yang mendukung dalam pengoperasian CBIADMIS adalah sebagai berikut:

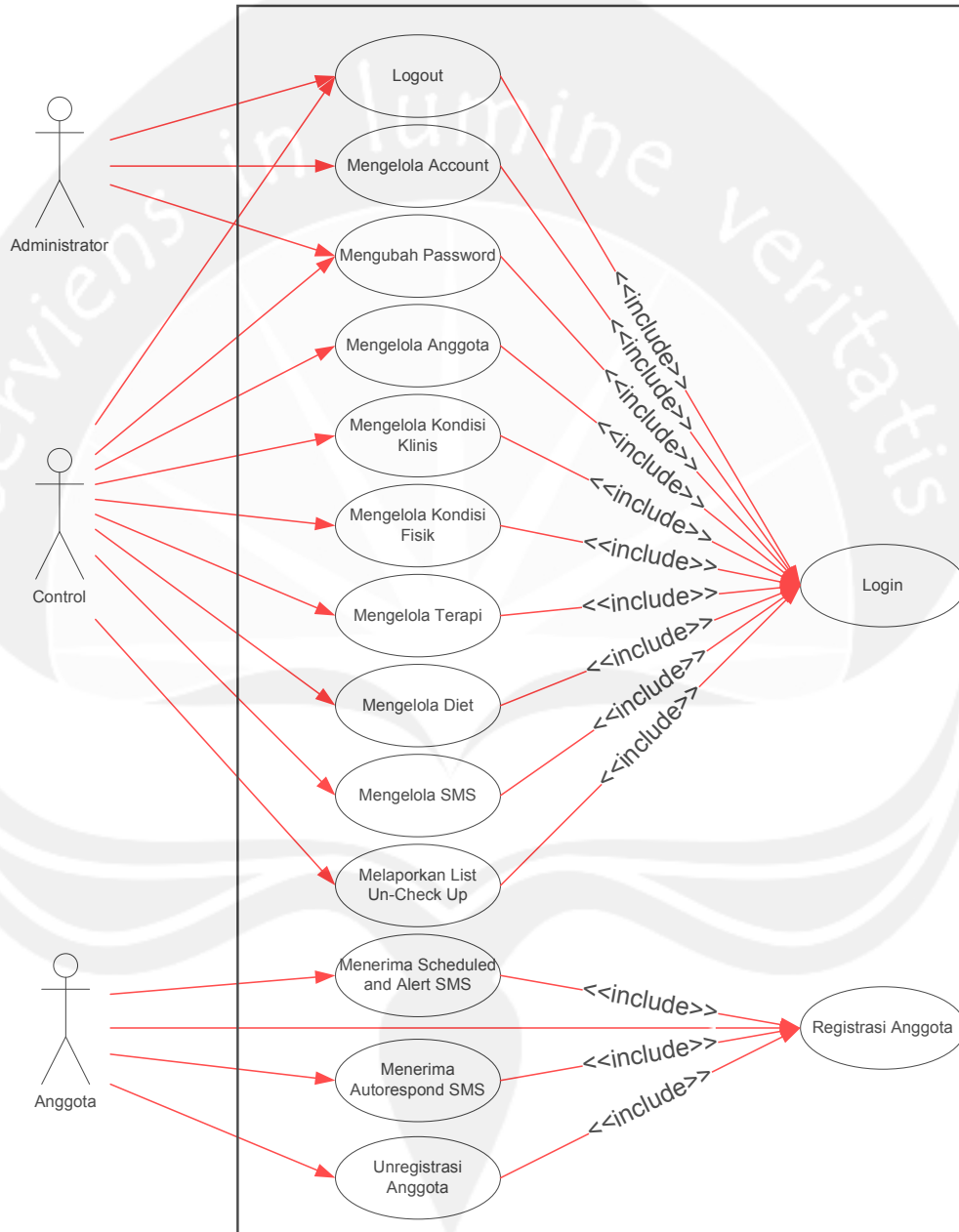
1. Nama : Windows XP
Sumber : Microsoft
Sebagai sistem operasi untuk desktop-user.

3.1.4 Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak CBIADMIS terhubung secara client-server dalam lingkup jaringan SMS dan SMS Gateway.

3.2 Kebutuhan fungsionalitas Perangkat Lunak

3.2.1 Use Case Diagram



Gambar 2. Use Case Diagram

4 Spesifikasi Rinci Kebutuhan

4.1 Spesifikasi Kebutuhan Fungsionalitas

4.1.1 Use case Specification: Login

1. Brief Description

Use Case ini digunakan oleh aktor untuk memperoleh akses ke sistem. Login didasarkan pada sebuah id unik yaitu username dari user dan password yang berupa rangkaian karakter.

2. Primary Actor

1. User

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan login.
2. Sistem menampilkan antarmuka untuk login.
3. Aktor memasukkan id dan password.
4. Sistem memeriksa id dan password yang diinputkan aktor.

E-1 Password atau id user tidak sesuai.

5. Sistem memberikan akses ke aktor.
6. Use Case ini selesai.

5. Alternative Flow

none

6. Error Flow

E-1 Password atau nama user tidak sesuai.

1. Sistem menampilkan peringatan bahwa id user atau password tidak sesuai.
2. Kembali ke Basic Flow langkah ke 3.

7. PreConditions

Username dan password user sudah terdaftar dalam basisdata sistem.

8. PostConditions

Program Studi Teknik Informatika	SKPL – CBIADMIS	24/ 51
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Aktor memasuki sistem dan dapat menggunakan fungsi-fungsi pada system sesuai dengan privilegenya.

4.1.2 Use case Spesification: Logout

1. Brief Description

Use Case ini digunakan oleh aktor untuk keluar dari sistem tanpa harus mematikan jalannya software.

2. Primary Actor

1. User

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan logout.
2. Sistem menampilkan antarmuka setelah logout.
3. Use Case ini selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Use Case Login sudah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Aktor memasuki keluar dari sistem tanpa harus mematikan jalannya aplikasi.

4.1.3 Use case Spesification : Mengelola Data Account

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola account dari user. Aktor dapat melakukan add data

account, edit data account, delete data account, display data account, atau search data account.

2. Primary Actor

1. Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan kelola data account.
2. Sistem memberikan pilihan untuk melakukan entry data account (tampilan awal), edit data account, delete data account, display data account, atau search data account tertentu.
3. Aktor memilih untuk melakukan entry data account.
 - A-1 Aktor memilih untuk melakukan edit data account.
 - A-2 Aktor memilih untuk melakukan delete data account.
 - A-3 Aktor memilih untuk melakukan display data account.
 - A-4 Aktor memilih untuk melakukan search data account tertentu.
4. Aktor menginputkan data account.
5. Aktor meminta sistem untuk menyimpan data account yang telah diinputkan.
6. Sistem mengecek data account yang telah diinputkan.
 - E-1 Data account yang diinputkan aktor salah.
7. Sistem menyimpan data account ke database.
8. Use Case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan edit data account.
1. Sistem meminta aktor untuk memasukkan nilai yang dapat menjadi identifikasi.

2. Aktor menginputkan nilai.

E-2 Data account tidak ditemukan.

E-3 Nilai yang telah diinputkan salah.

3. Sistem menampilkan data account.

4. Aktor mengedit data account yang sudah ditampilkan.

5. Aktor meminta sistem untuk menyimpan data account yang telah diedit.

6. Sistem melakukan pengecekan terhadap data account yang telah diedit.

E-4 Data account yang telah diinput salah.

7. Sistem menyimpan data account yang telah diedit ke database.

8. Berlanjut ke Basic Flow langkah ke 8.

A-2 Aktor memilih untuk melakukan delete data account.

1. Sistem meminta aktor menginputkan nilai yang dapat menjadi identifikasi.

2. Aktor menginputkan nilai.

E-5 Nilai yang telah diinputkan salah.

E-6 Data account tidak ditemukan.

3. Sistem menampilkan data account.

4. Aktor melakukan delete.

5. Sistem melakukan update ke database.

6. Berlanjut ke Basic Flow langkah ke 8.

A-3 Aktor memilih untuk melakukan display data account.

1. Sistem menampilkan data account.

2. Berlanjut ke Basic Flow langkah ke 8.

A-4 Aktor memilih untuk melakukan search data pegawai tertentu.

1. Sistem meminta aktor menginputkan nilai yang dapat menjadi identifikasi.

2. Aktor menginputkan nilai.

E-7 Nilai yang telah diinputkan salah.

E-8 Data account tidak ditemukan.

3. Sistem menampilkan data account.

4. Berlanjut ke Basic Flow langkah ke 8.

6. Error Flow

E-1 Data account yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.

2. Kembali ke Basic Flow Langkah ke 4.

E-2 Data account yang tidak ditemukan.

1. Sistem memberikan pesan peringatan bahwa data tidak ditemukan.

2. Kembali ke Alternative Flow A-1 Langkah ke 1.

E-3 Nilai yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa nilai yang diinput salah.

2. Kembali ke Alternative Flow A-1 Langkah ke 1.

E-4 Data account yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diedit salah.

2. Kembali ke Alternative Flow A-1 Langkah ke 4.

E-5 Nilai yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa nilai yang diinput salah.

2. Kembali ke Alternative Flow A-2 Langkah ke 1.

E-6 Data account tidak ditemukan.

1. Sistem memberikan pesan peringatan bahwa data yang diedit salah.

2. Kembali ke Alternative Flow A-2 Langkah ke 1.

E-7 Data account yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diedit salah.

2. Kembali ke Alternative Flow A-4 Langkah ke 1.

E-8 Data account tidak ditemukan.

1. Sistem memberikan pesan peringatan bahwa data tidak ditemukan.
2. Kembali ke Alternative Flow A-4 Langkah ke 1.

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Data account di database telah terupdate.

4.1.4 Use case Spesification: Ubah Password

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengubah password yang dimilikinya guna keperluan login sistem.

2. Primary Actor

1. User

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan kelola ubah password.
2. Sistem menampilkan meminta aktor untuk memasukkan password lama dan password baru.
3. Aktor memasukkan password lama dan password baru.
4. Sistem memeriksa password yang diinputkan aktor.
E-1 Password tidak sesuai.
5. Sistem memberikan pesan kepada aktor bahwa password telah diubah.
6. Use Case ini selesai.

5. Alternative Flow

none

6. Error Flow

E-1 Password tidak sesuai.

1. Sistem menampilkan peringatan bahwa password tidak sesuai.

Program Studi Teknik Informatika	SKPL – CBIADMIS	29/ 51
----------------------------------	-----------------	--------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

2. Kembali ke Basic Flow langkah ke 2.

7. PreConditions

1. Use Case Login sudah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Aktor memiliki password baru untuk login.

4.1.5 Use case Spesification : Mengelola Data Anggota

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data anggota yang terdaftar dalam sistem. Aktor dapat melakukan add data anggota, edit data anggota, display data anggota, atau search data anggota.

2. Primary Actor

1. Control

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan kelola data anggota.
2. Sistem memberikan pilihan untuk melakukan entry data anggota (tampilan awal), edit data anggota, delete data anggota, display data anggota, atau search data anggota tertentu.
3. Aktor memilih untuk melakukan entry data anggota.
 - A-1 Aktor memilih untuk melakukan edit data anggota.
 - A-2 Aktor memilih untuk melakukan display data anggota.
 - A-3 Aktor memilih untuk melakukan search data anggota tertentu.
4. Aktor menginputkan data anggota.

5. Aktor meminta sistem untuk menyimpan data anggota yang telah diinputkan.
6. Sistem mengecek data anggota yang telah diinputkan.
 - E-1 Data anggota yang diinputkan aktor salah.
7. Sistem menyimpan data anggota ke database.
8. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk melakukan edit data anggota.

1. Sistem meminta aktor untuk memasukkan nilai yang dapat menjadi identifikasi.
2. Aktor menginputkan nilai.
 - E-2 Data anggota tidak ditemukan.
 - E-3 Nilai yang telah diinputkan salah.

3. Sistem menampilkan data anggota.
4. Aktor mengedit data anggota yang sudah ditampilkan.
5. Aktor meminta sistem untuk menyimpan data anggota yang telah diedit.
6. Sistem melakukan pengecekan terhadap data anggota yang telah diedit.
 - E-4 Data anggota yang telah diinput salah.
7. Sistem menyimpan data anggota yang telah diedit ke database.
8. Berlanjut ke Basic Flow langkah ke 8.

A-2 Aktor memilih untuk melakukan display data anggota.

1. Sistem menampilkan data anggota.
2. Berlanjut ke Basic Flow langkah ke 8.

A-3 Aktor memilih untuk melakukan search data pegawai tertentu.

1. Sistem meminta aktor menginputkan nilai yang dapat menjadi identifikasi.
2. Aktor menginputkan nilai.

E-5 Nilai yang telah diinputkan salah.

E-6 Data anggota tidak ditemukan.

3. Sistem menampilkan data anggota.

4. Berlanjut ke Basic Flow langkah ke 8.

6. Error Flow

E-1 Data anggota yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.

2. Kembali ke Basic Flow Langkah ke 4.

E-2 Data anggota yang tidak ditemukan.

1. Sistem memberikan pesan peringatan bahwa data tidak ditemukan.

2. Kembali ke Alternative Flow A-1 Langkah ke 1.

E-3 Nilai yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa nilai yang diinput salah.

2. Kembali ke Alternative Flow A-1 Langkah ke 1.

E-4 Data anggota yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diedit salah.

2. Kembali ke Alternative Flow A-1 Langkah ke 4.

E-5 Data anggota yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diedit salah.

2. Kembali ke Alternative Flow A-3 Langkah ke 1.

E-6 Data anggota tidak ditemukan.

1. Sistem memberikan pesan peringatan bahwa data tidak ditemukan.

2. Kembali ke Alternative Flow A-3 Langkah ke 1.

7. PreConditions

1. Use Case Login telah dilakukan.

2. Aktor telah memasuki sistem.

8. PostConditions

1. Data anggota di database telah terupdate.

4.1.6 Use case Spesification : Mengelola Data Check Up

1. Brief Description

Use Case ini memungkinkan user untuk melakukan pengelolaan data kondisi klinis, termasuk didalamnya menambah data kondisi klinis baru, edit data kondisi klinis, display data kondisi klinis, dan search data kondisi klinis.

2. Primary Actor

1. Operasional

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data check up.
2. Sistem memberikan pilihan untuk melakukan entry data kondisi klinis, edit data kondisi klinis, display data kondisi klinis, atau display data kondisi klinis tertentu.
3. Aktor memilih untuk melakukan entry data kondisi klinis.
 - A-1 Aktor memilih untuk melakukan edit data kondisi klinis.
 - A-2 Aktor memilih untuk melakukan display data kondisi klinis.
 - A-3 Aktor memilih untuk melakukan search data kondisi klinis tertentu.
4. Aktor menginputkan data kondisi klinis
5. Aktor meminta sistem untuk menyimpan data kondisi klinis yang telah diinputkan
6. Sistem mengecek data kondisi klinis yang telah diinputkan

E-1 Data kondisi klinis yang diinputkan aktor salah

7. Sistem menyimpan data kondisi klinis ke database
8. Use Case selesai

5. Alternative Flow

A-1 Aktor memilih untuk melakukan edit data kondisi klinis

1. Sistem meminta aktor menginputkan nilai yang dapat menjadi identifikasi.
2. Aktor menginputkan nilai.

E-2 Nilai yang telah diinputkan salah.

E-3 Data tidak ditemukan.

3. Sistem menampilkan data kondisi klinis.
4. Aktor mengedit data kondisi klinis yang sudah ditampilkan.
5. Aktor meminta sistem untuk menyimpan data kondisi klinis yang telah diedit.
6. Sistem melakukan pengecekan terhadap data kondisi klinis yang telah diedit.

E-4 Data kondisi klinis yang telah diedit salah.

7. Sistem menyimpan data kondisi klinis yang telah diedit ke database.
8. Berlanjut ke Basic Flow langkah ke 8.

A-2 Aktor memilih untuk melakukan display data kondisi klinis

1. Sistem menampilkan data kondisi klinis.
2. Berlanjut ke Basic Flow langkah ke 8.

A-3 Aktor memilih untuk melakukan search data kondisi klinis tertentu.

1. Sistem meminta aktor menginputkan nilai yang dapat menjadi identifikasi.
2. Aktor menginputkan nilai.

E-5 Nilai yang telah diinputkan salah.

E-6 Data tidak ditemukan.

3. Sistem menampilkan data atau kondisi klinis.
4. Berlanjut ke Basic Flow langkah ke 8.

6. Error Flow

E-1 Data kondisi klinis yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
2. Kembali ke Basic Flow Langkah ke 4.

E-2 Nilai yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa nilai yang diinput salah.
2. Kembali ke Alternative Flow A-1 Langkah ke 1.

E-3 Data tidak ditemukan.

1. Sistem memberikan pesan peringatan bahwa data tidak ditemukan.
2. Kembali ke Alternative Flow A-1 Langkah ke 1.

E-4 Data kondisi klinis yang diinputkan aktor salah

1. Sistem memberikan pesan peringatan bahwa data yang diedit salah.
2. Kembali ke Alternative Flow A-1 Langkah ke 4.

E-5 Nilai yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa nilai yang diinput salah.
2. Kembali ke Alternative Flow A-3 Langkah ke 1.

E-6 Data tidak ditemukan.

1. Sistem memberikan pesan peringatan bahwa data tidak ditemukan.
2. Kembali ke Alternative Flow A-3 Langkah ke 1.

7. PreConditions

1. Use Case Login sudah dilakukan
2. Aktor telah memasuki sistem

8. PostConditions

1. Data kondisi klinis di database telah terupdate

4.1.7 Use case Spesification : Mengelola Data Daily Check Up

1. Brief Description

Use Case ini memungkinkan user untuk melakukan pengelolaan data kondisi fisik, termasuk didalamnya menambah data kondisi fisik baru, edit data kondisi fisik, display data kondisi fisik, dan search data kondisi fisik tertentu.

2. Primary Actor

1. Control

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data daily check up.
2. Sistem memberikan pilihan untuk melakukan entry data kondisi fisik, edit data kondisi fisik, atau display data kondisi fisik.
3. Aktor memilih untuk melakukan entry data kondisi fisik
 - A-1 Aktor memilih untuk melakukan edit data kondisi fisik
 - A-2 Aktor memilih untuk melakukan display data kondisi fisik.
 - A-3 Aktor memilih untuk melakukan display data kondisi fisik tertentu.
4. Aktor menginputkan data kondisi fisik
5. Aktor meminta sistem untuk menyimpan data kondisi fisik yang telah diinputkan
6. Sistem mengecek data kondisi fisik yang telah diinputkan
 - E-1 Data kondisi fisik yang diinputkan aktor salah
7. Sistem menyimpan data kondisi fisik ke database

8. Use Case selesai

5. Alternative Flow

A-1 Aktor memilih untuk melakukan edit data kondisi fisik

1. Sistem meminta aktor menginputkan nilai yang dapat menjadi identifikasi.

2. Aktor menginputkan nilai.

E-2 Nilai yang telah diinputkan salah.

E-3 Data tidak ditemukan.

3. Sistem menampilkan data atau profile kondisi fisik

4. Aktor mengedit data kondisi fisik yang sudah ditampilkan

5. Aktor meminta sistem untuk menyimpan data kondisi fisik yang telah diedit

6. Sistem melakukan pengecekan terhadap data kondisi fisik yang telah diedit

E-4 Data kondisi fisik yang telah diedit salah

5. Sistem meyimpan data kondisi fisik yang telah diedit ke database

6. Berlanjut ke Basic Flow langkah ke 8

A-2 Aktor memilih untuk melakukan display data kondisi fisik

1. Sistem menampilkan data atau profile kondisi fisik

2. Berlanjut ke Basic Flow langkah ke 8

A-3 Aktor memilih untuk melakukan display data kondisi fisik tertentu.

1. Sistem meminta aktor menginputkan nilai yang dapat menjadi identifikasi.

2. Aktor menginputkan nilai.

E-5 Nilai yang telah diinputkan salah.

E-6 Data tidak ditemukan.

3. Sistem menampilkan data kondisi fisik.
4. Berlanjut ke Basic Flow langkah ke 8.

6. Error Flow

E-1 Data kondisi fisik yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
2. Kembali ke Basic Flow Langkah ke 4.

E-2 Nilai yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa nilai yang diinput salah.
2. Kembali ke Alternative Flow A-1 Langkah ke 1.

E-3 Data tidak ditemukan.

1. Sistem memberikan pesan peringatan bahwa data tidak ditemukan.
2. Kembali ke Alternative Flow A-1 Langkah ke 1.

E-4 Data kondisi fisik yang diinputkan aktor salah

1. Sistem memberikan pesan peringatan bahwa data yang diinput salah.
2. Kembali ke Alternative Flow A-1 Langkah ke 4.

E-5 Nilai yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa nilai yang diinput salah.
2. Kembali ke Alternative Flow A-3 Langkah ke 1.

E-6 Data tidak ditemukan.

1. Sistem memberikan pesan peringatan bahwa data tidak ditemukan.
2. Kembali ke Alternative Flow A-3 Langkah ke 1.

7. PreConditions

1. Use Case Login sudah dilakukan
2. Aktor telah memasuki sistem

8. PostConditions

1. Data kondisi fisik di database telah terupdate

4.1.8 Use case Spesification : Mengelola Terapi

Program Studi Teknik Informatika	SKPL – CBIADMIS	38/ 51
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Brief Description

Use Case ini memungkinkan user untuk melakukan pengelolaan data terapi, termasuk didalamnya menambah data terapi, edit data terapi, delete data terapi, display data terapi, atau search terapi.

2. Primary Actor

1. Control

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data terapi.
2. Sistem memberikan pilihan untuk melakukan entry data terapi, edit data terapi, display data terapi, atau search terapi.
3. Aktor memilih untuk melakukan entry data terapi.
 - A-1 Aktor memilih untuk melakukan edit data terapi.
 - A-2 Aktor memilih untuk melakukan display data pemeliharaan mobil.
 - A-3 Aktor memilih untuk melakukan search data pemeliharaan mobil tertentu.
4. Aktor menginputkan data terapi.
5. Aktor meminta sistem untuk menyimpan data terapi yang telah diinputkan.
6. Sistem mengecek data terapi yang telah diinputkan.
 - E-1 Data terapi yang diinputkan aktor salah.
7. Sistem menyimpan data terapi ke database.
8. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk melakukan edit data terapi.

1. Sistem meminta aktor menginputkan nilai yang dapat menjadi identifikasi.

Program Studi Teknik Informatika	SKPL – CBIADMIS	39/ 51
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Aktor menginputkan nilai.

E-2 Nilai yang telah diinputkan salah.

E-3 Data tidak ditemukan.

3. Sistem menampilkan data terapi.

4. Aktor mengedit data terapi yang sudah ditampilkan.

5. Aktor meminta sistem untuk menyimpan data terapi yang telah diedit.

6. Sistem melakukan pengecekan terhadap data terapi yang telah diedit.

E-4 Data terapi yang telah diedit salah

7. Sistem menyimpan data terapi yang telah diedit ke database

8. Berlanjut ke Basic Flow langkah ke 8

A-2 Aktor memilih untuk melakukan display data terapi.

1. Sistem menampilkan data atau profile terapi.

2. Berlanjut ke Basic Flow langkah ke 8.

A-3 Aktor memilih untuk melakukan display data terapi tertentu.

1. Sistem meminta aktor menginputkan nilai yang dapat menjadi identifikasi.

2. Aktor menginputkan nilai.

E-5 Nilai yang telah diinputkan salah.

E-6 Data tidak ditemukan.

3. Sistem menampilkan data terapi.

4. Berlanjut ke Basic Flow langkah ke 8.

6. Error Flow

E-1 Data terapi yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.

2. Kembali ke Basic Flow Langkah ke 4.

E-2 Nilai yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa nilai yang diinput salah.
 2. Kembali ke Alternative Flow A-1 Langkah ke 1.
- E-3 Data tidak ditemukan.
1. Sistem memberikan pesan peringatan bahwa data tidak ditemukan.
 2. Kembali ke Alternative Flow A-1 Langkah ke 1.
- E-4 Data terapi yang diinputkan aktor salah
1. Sistem memberikan pesan peringatan bahwa data yang diedit salah.
 2. Kembali ke Alternative Flow A-1 Langkah ke 4.
- E-5 Nilai yang diinputkan aktor salah.
1. Sistem memberikan pesan peringatan bahwa nilai yang diinput salah.
 2. Kembali ke Alternative Flow A-3 Langkah ke 1.
- E-6 Data tidak ditemukan.
1. Sistem memberikan pesan peringatan bahwa data tidak ditemukan.
 2. Kembali ke Alternative Flow A-3 Langkah ke 1.

7. PreConditions

1. Use Case Login sudah dilakukan
2. Aktor telah memasuki sistem

8. PostConditions

1. Data terapi di database telah terupdate

4.1.9 Use case Spesification : Mengelola Diet

1. Brief Description

Use Case ini memungkinkan user untuk melakukan pengelolaan data diet, termasuk didalamnya menambah data diet, edit data diet, display data diet, atau search diet.

2. Primary Actor

1. Control

3. Supporting Actor

Program Studi Teknik Informatika	SKPL – CBIADMIS	41/ 51
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data diet.
2. Sistem memberikan pilihan untuk melakukan entry data diet, edit data diet, display data diet, atau search diet.
3. Aktor memilih untuk melakukan entry data diet.
 - A-1 Aktor memilih untuk melakukan edit data diet.
 - A-2 Aktor memilih untuk melakukan display data pemeliharaan mobil.
 - A-3 Aktor memilih untuk melakukan search data pemeliharaan mobil tertentu.
4. Aktor menginputkan data diet.
5. Aktor meminta sistem untuk menyimpan data diet yang telah diinputkan.
6. Sistem mengecek data diet yang telah diinputkan.
 - E-1 Data diet yang diinputkan aktor salah.
7. Sistem menyimpan data diet ke database.
8. Use Case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan edit data diet.
 1. Sistem meminta aktor menginputkan nilai yang dapat menjadi identifikasi.
 2. Aktor menginputkan nilai.
 - E-2 Nilai yang telah diinputkan salah.
 - E-3 Data tidak ditemukan.
 3. Sistem menampilkan data diet.
 4. Aktor mengedit data diet yang sudah ditampilkan.
 5. Aktor meminta sistem untuk menyimpan data diet yang telah diedit.

6. Sistem melakukan pengecekan terhadap data diet yang telah diedit.

E-4 Data diet yang telah diedit salah

7. Sistem menyimpan data diet yang telah diedit ke database

8. Berlanjut ke Basic Flow langkah ke 8

A-2 Aktor memilih untuk melakukan display data diet.

1. Sistem menampilkan data atau profile diet.

2. Berlanjut ke Basic Flow langkah ke 8.

A-3 Aktor memilih untuk melakukan display data diet tertentu.

1. Sistem meminta aktor menginputkan nilai yang dapat menjadi identifikasi.

2. Aktor menginputkan nilai.

E-5 Nilai yang telah diinputkan salah.

E-6 Data tidak ditemukan.

3. Sistem menampilkan data diet.

4. Berlanjut ke Basic Flow langkah ke 8.

6. Error Flow

E-1 Data diet yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.

2. Kembali ke Basic Flow Langkah ke 4.

E-2 Nilai yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa nilai yang diinput salah.

2. Kembali ke Alternative Flow A-1 Langkah ke 1.

E-3 Data tidak ditemukan.

1. Sistem memberikan pesan peringatan bahwa data tidak ditemukan.

2. Kembali ke Alternative Flow A-1 Langkah ke 1.

E-4 Data diet yang diinputkan aktor salah

1. Sistem memberikan pesan peringatan bahwa data yang diedit salah.

2. Kembali ke Alternative Flow A-1 Langkah ke 4.
- E-5 Nilai yang diinputkan aktor salah.
1. Sistem memberikan pesan peringatan bahwa nilai yang diinput salah.
 2. Kembali ke Alternative Flow A-3 Langkah ke 1.
- E-6 Data tidak ditemukan.
1. Sistem memberikan pesan peringatan bahwa data tidak ditemukan.
 2. Kembali ke Alternative Flow A-3 Langkah ke 1.

7. PreConditions

1. Use Case Login sudah dilakukan
2. Aktor telah memasuki sistem

8. PostConditions

1. Data diet di database telah terupdate

4.1.10 Use case Spesification : Mengelola SMS

1. Brief Description

Use Case ini memungkinkan user untuk melakukan pengelolaan SMS, termasuk didalamnya new SMS, inbox, sent item, draft, atau deleted.

2. Primary Actor

1. Control

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan SMS.
2. Sistem memberikan pilihan untuk melakukan new SMS, inbox, sent item, atau draft.
3. Aktor memilih untuk melakukan new SMS.
 - A-1 Aktor memilih untuk melakukan inbox.
 - A-2 Aktor memilih untuk melakukan sent item.
 - A-3 Aktor memilih untuk melakukan draft.
4. Aktor melakukan new SMS.

5. Aktor memasukkan SMS baru dan nomor tujuan.
6. Aktor meminta sistem untuk mengirimkan new SMS yang telah diinputkan.
7. Sistem mengecek apakah nomor tujuan sudah diinputkan.
 - E-1 Nomor tujuan belum diinputkan.
8. Sistem mengirim SMS ke nomor tujuan dan menyimpan SMS tersebut ke sent item pada database.
9. Use Case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan inbox
 1. Sistem menampilkan data-data SMS pada inbox.
 2. Aktor mengelola inbox.
 3. Berlanjut ke Basic Flow langkah ke 9.
- A-2 Aktor memilih untuk melakukan sent item.
 1. Sistem menampilkan data-data SMS pada sent item.
 2. Aktor mengelola sent item.
 3. Berlanjut ke Basic Flow langkah ke 9.
- A-3 Aktor memilih untuk melakukan draft.
 1. Sistem menampilkan data-data SMS pada draft.
 2. Aktor mengelola draft.
 3. Berlanjut ke Basic Flow langkah ke 9.

6. Error Flow

- E-1 Nomor tujuan belum diinputkan
 1. Sistem memberikan pesan peringatan bahwa nomor tujuan belum diinputkan.
 2. Sistem meminta aktor untuk menginputkan nomor tujuan.
 3. Aktor menginputkan nomor tujuan.
 4. Kembali ke Basic Flow langkah ke 8.

7. PreConditions

1. Use Case Login sudah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Data SMS (Inbox, Sent, Draft) di database telah terupdate.

4.1.11 Use case Spesification : Melaporkan List Un-Check Up

1. Brief Description

Use Case ini memungkinkan user untuk membuat laporan yang intinya berisi data-data anggota yang belum melakukan check up yang seharusnya dilakukan oleh mereka pada waktu tertentu sesuai dengan jadwal dan kebutuhan mereka.

2. Primary Actor

1. Control

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melaporkan List Un-Check Up.
2. Sistem memberikan pilihan untuk display data Un-Check Up pada tahun (dan bulan) tertentu.
3. Aktor memilih tahun (dan bulan).
4. Aktor meminta sistem untuk menampilkan data KIR sesuai dengan pilihannya.
5. Sistem menampilkan data KIR pada tahun (dan bulan) tersebut ke layar display.
6. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Use Case Login sudah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Data KIR tertampil di layar display

4.1.12 Use case Spesification : Scheduled and Alert SMS

1. Brief Description

Use Case ini memungkinkan user untuk menerima scheduled and alert SMS yang dikirimkan sebagai pengingat.

2. Primary Actor

1. Anggota

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan scheduled and alert SMS.
2. Sistem mengirimkan scheduled and alert SMS pada saat tertentu.
3. Aktor menerima scheduled and alert SMS.
4. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Use Case Registrasi Anggota sudah dilakukan.
2. Aktor terdaftar dalam system.

8. PostConditions

1. Aktor menerima Scheduled and Alert SMS.

4.1.13 Use case Spesification : Autorespond SMS

1. Brief Description

Use Case ini memungkinkan user untuk mendapatkan balasan SMS secara otomatis dari sistem.

2. Primary Actor

1. Anggota

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mengirimkan SMS ke sistem.
2. Sistem menerima SMS dari user.
3. Sistem membaca keyword dari SMS.
E-1 Keyword SMS salah.
4. Sistem untuk membalas SMS sesuai dengan pilihannya.
5. Aktor menerima balasan SMS dari sistem.
6. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

E-1 Keyword SMS salah.

1. Sistem mengirimkan SMS peringatan bahwa keyword SMS salah.
2. Kembali ke Basic Flow Langkah ke 1.

7. PreConditions

1. Fungsi Pendaftaran Anggota pada Use Case Registrasi sudah dilakukan.
2. Aktor telah terdaftar dalam sistem.

8. PostConditions

1. Aktor mendapatkan balasan SMS dari system secara otomatis.
2. Database terupdate.

4.1.14 Use case Spesification : Registrasi Anggota

1. Brief Description

Use Case ini memungkinkan user untuk melakukan registrasi anggota.

Program Studi Teknik Informatika	SKPL – CBIADMIS	48/ 51
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Primary Actor

1. Anggota

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan registrasi anggota.
2. Aktor menginputkan data anggota baru.
3. Aktor meminta sistem untuk menyimpan data anggota baru yang telah diinputkan.
4. Sistem mengecek data anggota yang telah diinputkan.
E-1 Data anggota yang diinputkan aktor salah.
5. Sistem menyimpan data anggota ke database.
6. Sistem mengirimkan pesan pada anggota bahwa registrasi berhasil.
7. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

- E-1 Data anggota yang diinputkan aktor salah.
1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
 2. Kembali ke Basic Flow Langkah ke 2.

7. PreConditions

1. Fungsi Pendaftaran Anggota pada Use Case Registrasi sudah dilakukan.
2. Aktor telah terdaftar dalam sistem.

8. PostConditions

1. Data anggota di database telah terupdate

4.1.15 Use case Spesification : Unregistrasi Anggota

1. Brief Description

Program Studi Teknik Informatika	SKPL – CBIADMIS	49/ 51
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Use Case ini memungkinkan user untuk melakukan pemberhentian anggota.

2. Primary Actor

1. Anggota

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan unregistrasi anggota.
2. Aktor meminta sistem untuk melakukan unregistrasi.
3. Sistem melakukan unregistrasi anggota.
4. Sistem menyimpan mengirimkan pesan ke anggota bahwa unregistrasi berhasil.
5. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

none

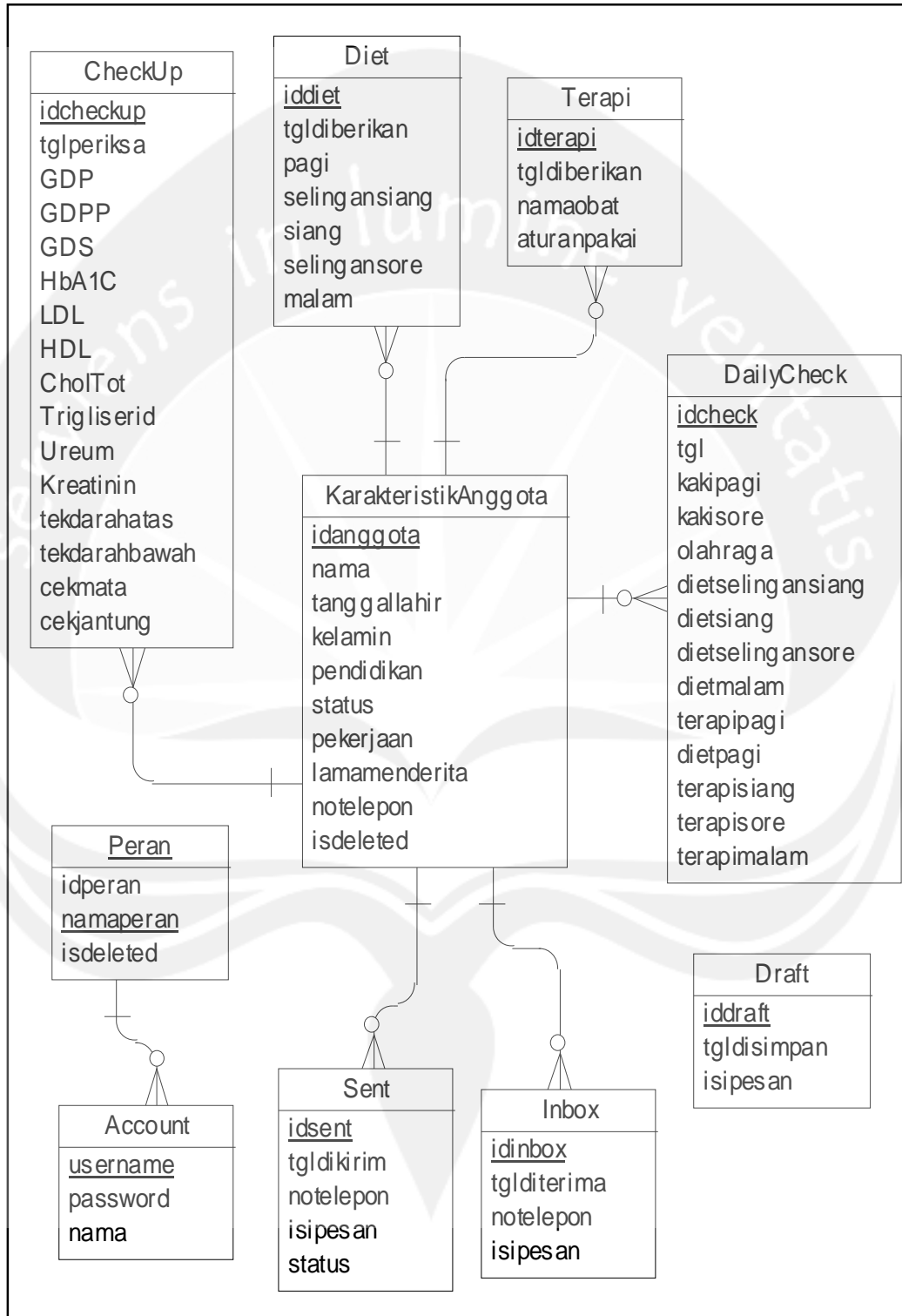
7. PreConditions

1. Fungsi Pendaftaran Anggota pada Use Case Registrasi sudah dilakukan.
2. Aktor telah terdaftar dalam sistem.

8. PostConditions

1. Data anggota di database telah terupdate

5 Entity Relationship Diagram (ERD)



Gambar 3. Entity Relationship Diagram