

	<ul style="list-style-type: none"> • Klik tombol "Simpan" 				
Pengujian Pengubahan Transaksi	<ul style="list-style-type: none"> • Klik Menu Transaksi • Klik tombol "Edit" • Memilih transaksi yang akan diubah • Ubah data • Klik tombol "Simpan" 	'1' '13-01-2014' '20-01-2014' 'Paket Silver' 'Sudarman' '-' 'Rp.55.550.000' 'Belum' 'Belum'	Database transaksi diperbaharui dan muncul message "Data berhasil diubah"	Database transaksi diperbaharui dan muncul message "Data berhasil diubah"	Data tran diper munc "Dat diuk
Pengujian Hapus Transaksi	<ul style="list-style-type: none"> • Klik Menu Transaksi • Memilih transaksi yang akan dihapus • Klik tombol "Hapus" 	-	Transaksi terpilih terhapus dari database dan muncul message "Data berhasil dihapus"	Transaksi terpilih terhapus dari database dan muncul message "Data berhasil dihapus"	Tran terp teri data munc "Dat diha
Pengujian Filter Transaksi	<ul style="list-style-type: none"> • Klik Menu Transaksi • Masukkan inputan 	-	Dataview langsung diperbaharui sesuai dengan inputan	Dataview langsung diperbaharui sesuai dengan inputan	Data lang dipe sesu inpu

BAB VI

PENUTUP

6.1. Kesimpulan

Berdasarkan analisis, desain dan implementasi perangkat lunak dan pembahasan pada bab-bab sebelumnya, maka dapat ditarik beberapa kesimpulan yaitu:

1. Sistem informasi *Wedding Organizer* berbasis Web telah berhasil dibangun.
2. Sistem informasi ini berhasil membantu dan mempermudah perencanaan pernikahan bagi customer dalam pemilihan paket pernikahan dengan cepat dan mudah.

6.2. Saran

Saran yang dapat diambil dari proses analisis sampai pada implementasi SIWO di tugas akhir ini adalah sebagai berikut:

1. Menambahkan fungsionalitas pada aplikasi web, agar fungsionalitas SIWO tidak hanya sebatas *update data*, tetapi juga customer bisa langsung berkomunikasi langsung dengan admin melalui chatting sehingga mempermudah untuk berkomunikasi dengan waktu yang efisien.
2. Dalam pengembangan lebih lanjut, sebaiknya diterapkan sistem informasi versi mobile, sehingga dapat diakses sekaligus oleh pengguna mobile.

Daftar Pustaka

Kadir Abdul. 2002, *Dasar Pemrograman Web Dinamis Menggunakan PHP*. Yogyakarta: Andi Offset

Kadir Abdul. 2009, *Membuat Aplikasi Web dengan PHP+Database mySQL*. Yogyakarta: Penerbit Andi

Agung, Gregorius, dan Jubilee Enterprise. 2004, *Macromedia Dreamweaver MX for Creative Webmaster*. Jakarta: PT ELEX Media Komputindo

Madcoms. 2004, *Aplikasi Program PHP & MySQL untuk Membuat Website Interaktif*. Yogyakarta: Penerbit Andi

AL-Bahra Bin Ladjamudin 2011, *Analisis Dan Desain Sistem Informasi*. Yogyakarta: Penerbit Andi

Arvina, I. 2011. *Sistem Informasi Perancangan E-commerce di Wulan Wedding Organizer*. Universitas Andalas, Padang

Bunafit Nugroho. 2003, *Latihan Membuat Aplikasi Web PHP dan MySQL dengan Dreamweaver*. Yogyakarta: Penerbit Andi

Hartono, Jogianto, 2005, *Analisis & Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Graha Ilmu

Puspitosari H.2011, *Pemograman Web Database dengan PHP dan MY SQL*, Yogyakarta: Skripta Media Creative

Michael. 2011. *Perancangan Website Wedding Organizer Sausalito*, Institut Teknologi Sepuluh Nopember, Surabaya

Pandey R., Dwivedi S. 2011, *Ontology Description Using Owl To Support Semantic Web Applications*. England: Addison Wesley

Pamela, J. M., & Elisabeth, D. 2011. *Documentary tools in everyday life*. Germany: The Wedding Planner Journal

Simarmata, Janner. 2006. *Pengenalan Basis Data*. Bandung: Penerbit Informatika

Sucianti, & Intan, P. S. 2010. *Nadysa Wedding Organizer*, Teknik Informatika Telkom Bandung

Waljiyanto, 2003, *Analisis dan Pemodelan Data*, Graha Ilmu, Yogyakarta

Wella Caterina, 2012, *Pengembangan Sistem Pendukung Cerdas Untuk Perencanaan Wisata Berbasis Web Dengan Menggunakan Teknologi Web Service*: Universitas Atma Jaya Yogyakarta

Yoanda. 2006. *Perancangan Program aplikasi Sistem Penunjang Keputusan Pemilihan Paket Pernikahan Hotel pada I.S. event & Wedding Organizer*, Universitas Atma Jaya Yogyakarta



LAMP IRAN

SKPL

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

**PEMBANGUNAN SISTEM INFORMASI WEDDING ORGANIZER
BERBASIS WEB**

(SIWO)

Untuk :

Tugas Akhir (Skripsi)

Dipersiapkan oleh:

Maria Stevani N.P

090705961

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

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
		SKPL-SIWO		1/43
		Revisi		

DAFTAR PERUBAHAN

Revisi	Deskripsi
---------------	------------------

A	
B	
C	
D	
E	
F	

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

Daftar Halaman Perubahan

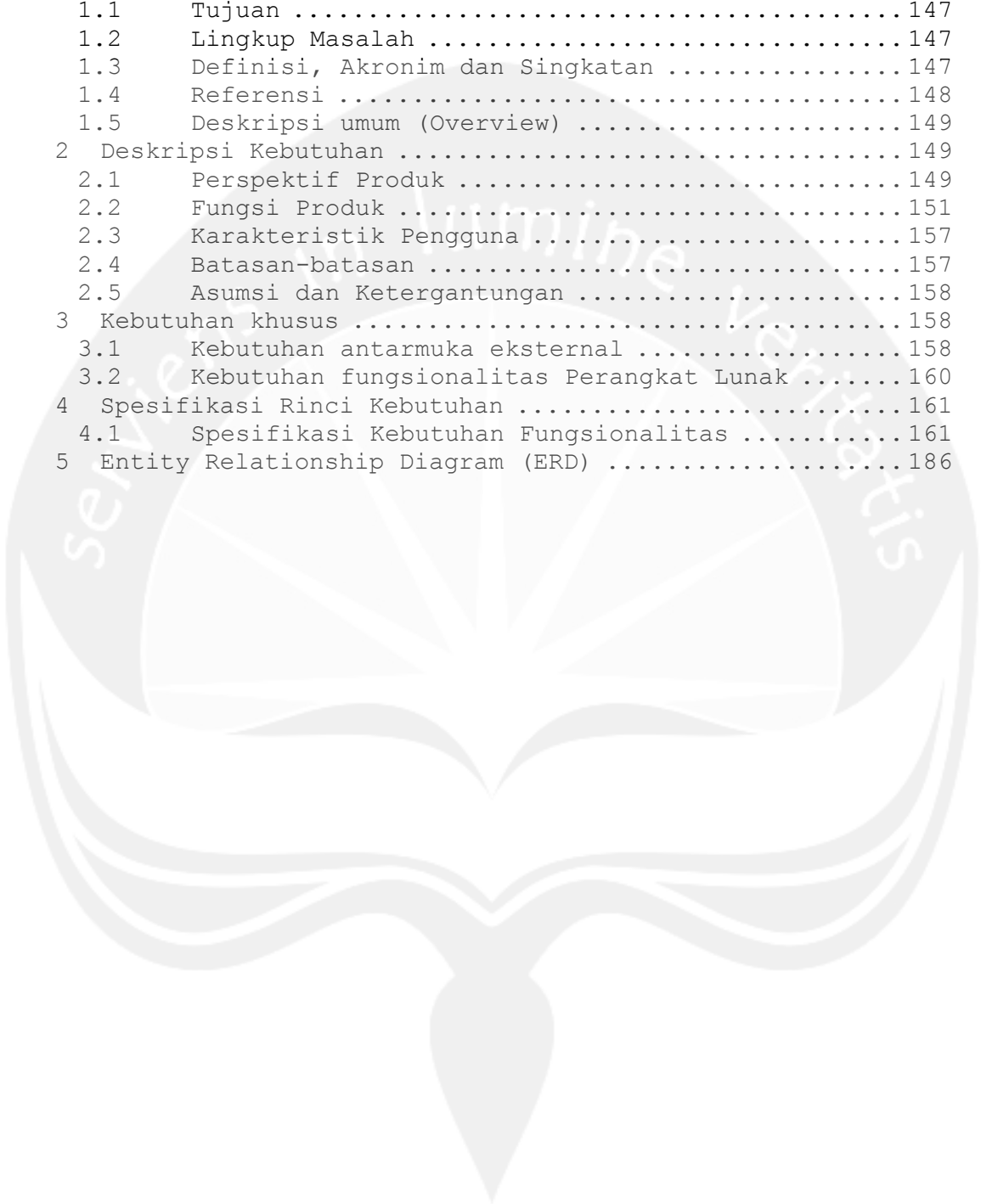
Halaman	Revisi	Halaman	Revisi



Daftar Isi

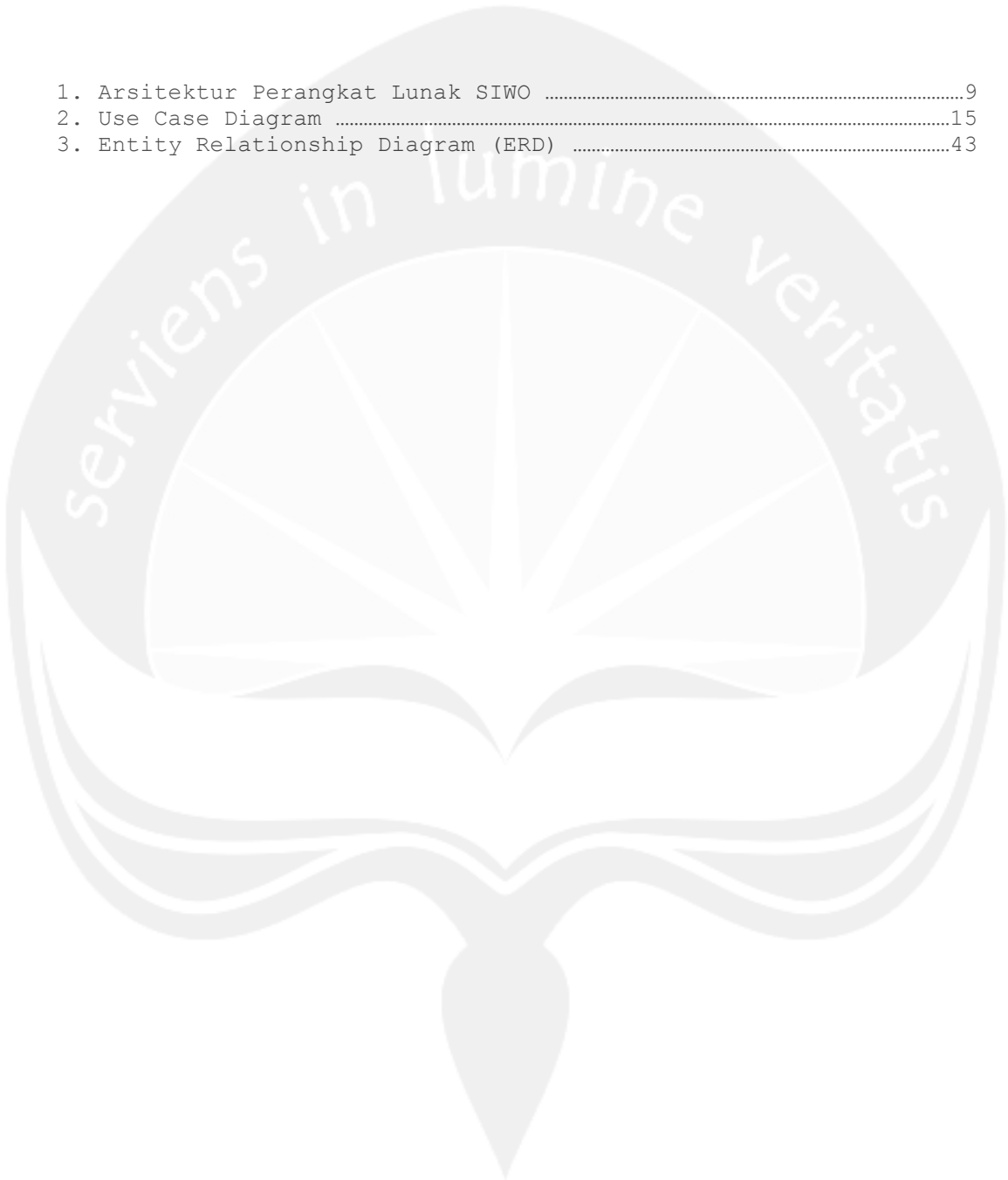
1	Pendahuluan	147
---	-------------------	-----

1.1	Tujuan	147
1.2	Lingkup Masalah	147
1.3	Definisi, Akronim dan Singkatan	147
1.4	Referensi	148
1.5	Deskripsi umum (Overview)	149
2	Deskripsi Kebutuhan	149
2.1	Perspektif Produk	149
2.2	Fungsi Produk	151
2.3	Karakteristik Pengguna	157
2.4	Batasan-batasan	157
2.5	Asumsi dan Ketergantungan	158
3	Kebutuhan khusus	158
3.1	Kebutuhan antarmuka eksternal	158
3.2	Kebutuhan fungsionalitas Perangkat Lunak	160
4	Spesifikasi Rinci Kebutuhan	161
4.1	Spesifikasi Kebutuhan Fungsionalitas	161
5	Entity Relationship Diagram (ERD)	186



Daftar Gambar

1. Arsitektur Perangkat Lunak SIWO	9
2. Use Case Diagram	15
3. Entity Relationship Diagram (ERD)	43



1 Pendahuluan

1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak *SIWO* ('Sistem Informasi Wedding Organizer') untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak dan perangkat keras, dan pengguna) performansi (kemampuan perangkat lunak dari segi kecepatan, tempat penyimpanan yang dibutuhkan, serta keakuratan), dan atribut (*feature-feature* tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-*SIWO* ini juga mendefinisikan batasan perancangan perangkat lunak.

1.2 Lingkup Masalah

Perangkat Lunak *SIWO* ini dibangun dengan tujuan untuk :

4. Menangani pencarian informasi mengenai berbagai macam hal dalam perencanaan dan pelaksanaan acara pernikahan,
5. Menawarkan macam-macam paket untuk pernikahan dan,
6. Menangani reservasi untuk pemesanan, dan berjalan pada web.

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- <i>SIWO</i> -XXX	Kode yang merepresentasikan kebutuhan pada

	<i>SIWO</i> ('Sistem Informasi Wedding Organizer) dimana XXX merupakan nomor fungsi produk.
<i>SIWO</i>	Sistem informasi untuk memberikan informasi mengenai paket pernikahan.
Administrator	Bertugas mengelola data "master" dari <i>SIWO</i> .
Wedding Organizer	Orang atau sekumpulan orang yang dapat membantu untuk memastikan lancarnya acara pesta pernikahan sesuai rencana yang ditentukan
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk Network global yang terdiri dari computer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai computer dan puluhan layanan informasi termasuk e-mail, FTP, dan World Wide Web.

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Bennet Simon, McRobb Steve, Farmer Ray, *Object-Oriented System Analysis and Design Using UML*, McGraw-Hill Companies, 2002.
2. Boggs Wendy, Boggs Michael, *Mastering UML with Rational Rose 2002*, SYBEX Inc, 2002.

1.5 Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas 3 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 SIWO yang akan dibangun, mencakup perspektif produk yang akan dibangun, fungsi produk perangkat lunak, karakteristik pengguna, batasan dalam penggunaan perangkat lunak dan asumsi yang dipakai dalam pembangunan perangkat lunak SIWO tersebut.

Bagian ketiga berisi penjelasan secara lebih rinci mengenai aplikasi web tentang jasa *SIWO* yang akan dibangun.

2 Deskripsi Kebutuhan

2.1 Perspektif Produk

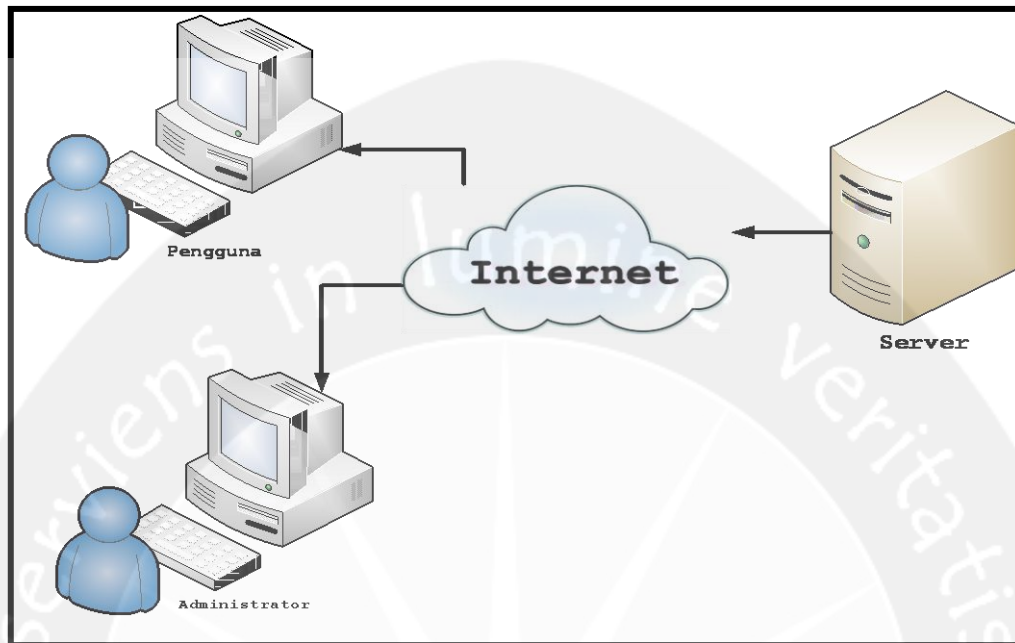
SIWO merupakan perangkat lunak berbasis web yang dibangun untuk membantu calon pengantin dalam perencanaan pernikahan. Sistem ini pada intinya menangani pencarian fasilitas dalam acara pernikahan seperti: gedung, katering, dekorasi, fotografer, video shooting, tradisi pernikahan, dll. Selain itu disediakan juga layanan bagi calon pengantin untuk memperoleh pilihan sistem paket yang disesuaikan dengan budget yang dimiliki calon pengantin. Nantinya sistem ini diharapkan dapat mempermudah calon pengantin dalam pencarian informasi dan reservasi. Aplikasi ini juga memiliki hak akses administrator. Pada hak administrator, memiliki hak untuk mengelola database maupun data di luar database. Sehingga

administrator bisa selalu mengupdate informasi-informasi yang berkaitan dengan informasi pernikahan.

Selain itu, di luar hak akses administrator terdapat user yang dapat melakukan reservasi paket pernikahan. Dalam menentukan paket pernikahan yang dipilih, user dapat menggunakan layanan yang disediakan oleh sistem. Layanan ini akan menampilkan seluruh data yang telah disediakan sesuai dengan budget user. User hanya tinggal memilih paket yang sudah ada atau dapat juga memilih *customize* sebagai pilihan. Apabila memilih *customize*, customer hanya dapat merubah pilihan yang sudah ditetapkan saja.

Perangkat lunak SIWO ini berjalan pada platform situs website dalam pembangunannya akan menggunakan bahasa pemrograman PHP dan database MySql sebagai penyimpanan data. Pengguna akan berinteraksi dengan sistem melalui antarmuka GUI (Graphical WO Interface). Pada sistem ini, seperti terlihat pada gambar 1, arsitektur perangkat lunak yang digunakan berupa *client server*, di mana semua data disimpan di server. User dapat mengakses data yang ada di server tersebut secara on-line.

Untuk seluruh data yang akan diolah disimpan di dalam database server. Sehingga jika ada pencarian data, maka data yang diinginkan akan dicari ke database server yang selanjutnya dikirimkan ke client yang merequest melalui web server.



Gambar 1. Arsitektur Perangkat lunak SIWO

2.2 Fungsi Produk

Fungsi produk perangkat lunak *SIWO* adalah sebagai berikut :

A. Pengelolaan Sistem (Admin)

1. Fungsi Login (**SKPL-SIWO-A-001**).

Login adalah fungsi yang digunakan oleh pihak Administrator untuk bisa masuk ke dalam sistem yang akan digunakan dengan hak akses sebagai administrator.

2. Fungsi Pengelolaan Paket (**SKPL-SIWO-A-002**)

Merupakan fungsi yang digunakan untuk mengelola Pengelolaan Paket Pernikahan oleh administrator.

d. Tambah Paket (**SKPL-SIWO-A-002-01**)

Merupakan fungsi yang digunakan untuk menambahkan paket pernikahan.

e. Ubah Paket (**SKPL-SIWO-A-002-02**)

Merupakan fungsi yang digunakan untuk mengubah paket pernikahan.

f. Hapus Paket (**SKPL-SIWO-A-002-03**)

Merupakan fungsi yang digunakan untuk menghapus paket pernikahan.

d. Filter Paket (**SKPL-SIWO-A-002-04**)

Merupakan fungsi yang digunakan untuk menyaring data paket pernikahan yang ingin dicari.

3. Fungsi Pengelolaan Gedung (**SKPL-SIWO-A-003**)

Merupakan fungsi yang digunakan untuk mengelola data gedung oleh administrator.

Fungsi pengelolaan Gedung meliputi:

d. Tambah Gedung (**SKPL-SIWO-A-003-01**)

Merupakan fungsi yang digunakan untuk menambahkan data gedung.

e. Ubah Gedung (**SKPL-SIWO-A-003-02**)

Merupakan fungsi yang digunakan untuk mengubah data gedung.

f. Hapus Gedung (**SKPL-SIWO-A-003-03**)

Merupakan fungsi yang digunakan untuk menghapus data gedung.

d. Filter Gedung (**SKPL-SIWO-A-003-04**)

Merupakan fungsi yang digunakan untuk menyaring data gedung yang ingin dicari.

4. Fungsi Pengelolaan Katering (**SKPL-SIWO-A-004**)

Merupakan fungsi yang digunakan untuk mengelola data katering oleh administrator.

Fungsi pengelolaan data katering meliputi:

d. Tambah Katering (**SKPL-SIWO-A-004-01**)

Merupakan fungsi yang digunakan untuk menambahkan data katering yang baru.

e. Ubah Katering (**SKPL-SIWO-A-004-02**)

Merupakan fungsi yang digunakan untuk mengubah data katering.

f. Hapus Katering (**SKPL-SIWO-A-004-03**)

Merupakan fungsi yang digunakan untuk menghapus data katering.

d. Filter Katering (**SKPL-SIWO-A-004-04**)

Merupakan fungsi yang digunakan untuk menyaring data katering yang ingin dicari.

5. Fungsi Pengelolaan Wisata (**SKPL-SIWO-A-005**)

Merupakan fungsi yang digunakan untuk mengelola data wisata oleh administrator.

Fungsi pengelolaan data Wisata meliputi:

b. Tambah Wisata (**SKPL-SIWO-A-005-01**)

Merupakan fungsi yang digunakan untuk menambahkan data wisata yang baru.

b. Ubah Wisata (**SKPL-SIWO-A-005-02**)

Merupakan fungsi yang digunakan untuk mengubah data wisata.

c. Hapus Wisata (**SKPL-SIWO-A-005-03**)

Merupakan fungsi yang digunakan untuk menghapus data wisata.

d. Filter Wisata (**SKPL-SIWO-A-005-04**)

Merupakan fungsi yang digunakan untuk menyaring data wisata yang ingin dicari.

6. Fungsi Pengelolaan Tradisi (**SKPL-SIWO-A-006**)

Merupakan fungsi yang digunakan untuk mengelola data tradisi oleh administrator.

Fungsi pengelolaan Tradisi meliputi:

a. Tambah Tradisi (**SKPL-SIWO-A-006-01**)

Merupakan fungsi yang digunakan untuk menambahkan data tradisi yang baru.

b. Ubah Tradisi (**SKPL-SIWO-A-006-02**)

Merupakan fungsi yang digunakan untuk mengubah data tradisi.

c. Hapus Tradisi (**SKPL-SIWO-A-006-03**)

Merupakan fungsi yang digunakan untuk menghapus data tradisi.

d. Filter Tradisi (**SKPL-SIWO-A-006-04**)

Merupakan fungsi yang digunakan untuk menyaring data tradisi yang ingin dicari.

7. Fungsi Pengelolaan Mobil (**SKPL-SIWO-A-007**)

Merupakan fungsi yang digunakan untuk mengelola data mobil oleh administrator.

Fungsi pengelolaan Data Mobil meliputi:

a. Tambah Mobil (**SKPL-SIWO-A-007-01**)

Merupakan fungsi yang digunakan untuk menambahkan data mobil yang baru.

b. Ubah Mobil (**SKPL-SIWO-A-007-02**)

Merupakan fungsi yang digunakan untuk mengubah data mobil.

c. Hapus Mobil (**SKPL-SIWO-A-007-03**)

Merupakan fungsi yang digunakan untuk menghapus data mobil.

d. Filter Mobil (**SKPL-SIWO-A-007-04**)

Merupakan fungsi yang digunakan untuk menyaring data mobil yang ingin dicari.

8. Fungsi Pengelolaan Undangan (**SKPL-SIWO-A-008**)

Merupakan fungsi yang digunakan untuk mengelola data undangan oleh administrator.

Fungsi pengelolaan Data Undangan meliputi:

a. Tambah Undangan (**SKPL-SIWO-A-008-01**)

Merupakan fungsi yang digunakan untuk menambahkan data undangan yang baru.

b. Ubah Undangan (**SKPL-SIWO-A-008-02**)

Merupakan fungsi yang digunakan untuk mengubah undangan.

c. Hapus Undangan (**SKPL-SIWO-A-008-03**)

Merupakan fungsi yang digunakan untuk menghapus undangan.

d. Filter Undangan (**SKPL-SIWO-A-008-04**)

Merupakan fungsi yang digunakan untuk menyaring data undangan yang ingin dicari.

9. Fungsi Kelola Admin (**SKPL-SIWO-009**)

Merupakan fungsi yang digunakan oleh administrator untuk mengubah data-data administrator dan hanya bisa dilakukan oleh administrator.

10. Fungsi Cetak Laporan Transaksi (**SKPL-SIWO-010**)

Merupakan fungsi yang digunakan oleh administrator untuk mencetak laporan transaksi customer.

B. User

1. Fungsi Registrasi Customer (**SKPL-SIWO-B-001**)

Merupakan fungsi agar customer dapat memasuki sistem. Jika telah melakukan pemesanan, maka customer harus melakukan registrasi untuk ke tahap selanjutnya.

2. Fungsi Login Customer (**SKPL-SIWO-B-002**).

Merupakan fungsi yang digunakan oleh customer agar bisa masuk ke dalam sistem untuk melakukan reservasi paket dan transaksi.

3. Fungsi Kelola Data Customer (**SKPL-SIWO-B-003**)

Merupakan fungsi yang digunakan customer untuk mengelola data-data pribadi.

Fungsi kelola data customer meliputi:

a. Ubah Data Customer (**SKPL-SIWO-B-003-01**)

Merupakan fungsi yang digunakan untuk mengubah data customer.

4. Fungsi Reservasi Paket (**SKPL-SIWO-004**).

Merupakan fungsi untuk menangani pemesanan paket pernikahan yang dilakukan customer. Di sini customer harus sudah memilih data paket pernikahan yang diinginkan. Kemudian customer diwajibkan untuk registrasi untuk ke tahap transaksi.

5. Fungsi Transaksi (**SKPL-SIWO-B-005**)

Merupakan fungsi yang digunakan customer untuk melakukan transaksi pembayaran pada paket pernikahan yang telah dipilih.

2.3 Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak *SIWO* adalah sebagai berikut :

1. Memahami pengoperasian PC.
2. Mengerti tentang internet.
3. Memahami penggunaan *SIWO*.

2.4 Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak *SIWO* tersebut adalah :

1. Kebijakan Umum

Berpedoman pada tujuan dari pengembangan perangkat lunak *SIWO*.

2. Keterbatasan perangkat keras

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

2.5 Asumsi dan Ketergantungan

Sistem ini dapat dijalankan pada komputer yang menggunakan sistem operasi windows.

3 Kebutuhan khusus

3.1 Kebutuhan antarmuka eksternal

Kebutuhan antar muka eksternal pada perangkat lunak *SIWO* meliputi kebutuhan antarmuka pemakai, antarmuka perangkat keras, antarmuka perangkat lunak, antarmuka komunikasi.

3.1.1 Antarmuka pemakai

Pengguna berinteraksi dengan antarmuka yang ditampilkan dalam bentuk form (GUI) yang tersedia. Pengguna memberikan inputan keyboard dan mouse sesuai dengan apa yang diminta oleh aplikasi tersebut.

3.1.2 Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak *SIWO* adalah:

1. PC yang terdiri dari prosesor 2.10 GHz, memori primer minimal 2 GB, kapasitas media penyimpanan sekunder server yaitu 1000 GB.

2. Keyboard digunakan untuk mengenali input yang dilakukan pengguna untuk menginputkan data berupa karakter dan teks.
3. Mouse digunakan untuk mengenali input yang dilakukan oleh pengguna yang berkaitan dengan event click.
4. Monitor untuk menampilkan form(GUI) serta halaman web kepada pengguna.

3.1.3 Antarmuka perangkat lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak *SIWO* adalah sebagai berikut :

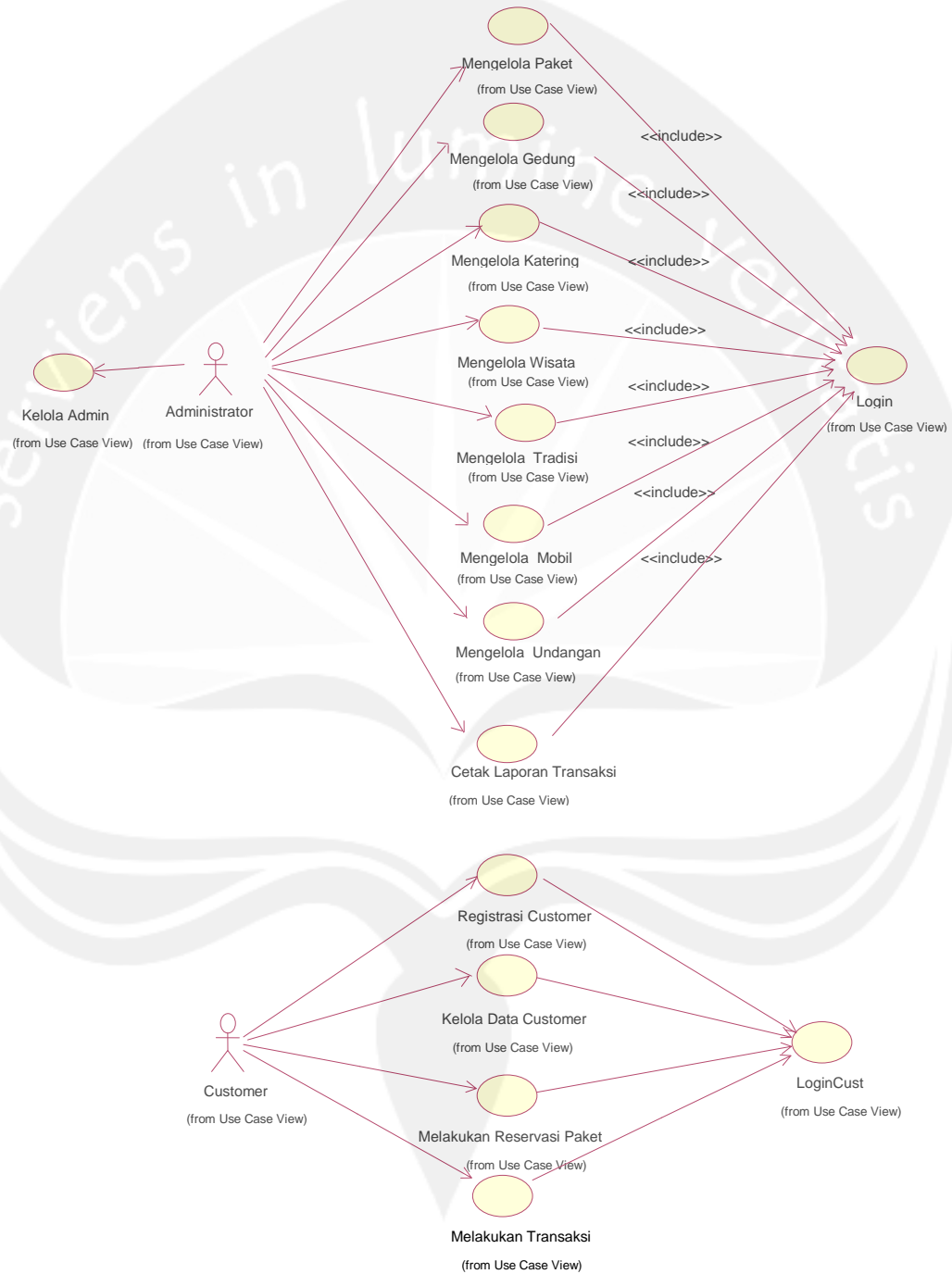
- a. Sistem Operasi : Microsoft Windows 7 Ultimate 32-bit
- b. PHP
- c. MySql
- d. Web Browser (Mozilla Firefox)

3.1.4 Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak *SIWO* menggunakan protocol HTTP. Sebab perangkat lunak ini digunakan untuk mendukung model sistem *client-server* dengan media komunikasi internet.

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

A. Pengelolaan Sistem (Admin)

4.1.1 Use case Spesification : Login

1. Brief Description

Use Case login ini digunakan oleh Administrator untuk memperoleh akses ke dalam sistem. Login di dasarkan pada sebuah nama dan password yang berupa rangkaian karakter.

2. Primary Actor

1. Administrator

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 username tidak sesuai

E-2 username tidak ada

5. Sistem memberikan akses ke aktor

6. Use Case ini selesai

5. Alternative Flow

None

6. Error Flow

E-1 Password atau username tidak sesuai

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

7. PreConditions

None

8. PostConditions

1. Aktor memasuki sistem dan dapat menggunakan fungsi fungsi pada sistem.

4.1.2 Use case Spesification : Pengelolaan Paket

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data Paket Pernikahan. Aktor dapat melakukan tambah Paket, ubah Paket, hapus Paket dan filter Paket.

2. Primary Actor

1. Administrator

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih login untuk melakukan pengelolaan data Paket.
2. Sistem memberikan pilihan untuk melakukan tambah Paket, ubah Paket, hapus Paket dan filter Paket.
3. Aktor memilih untuk melakukan tambah Paket
 - A-1 Aktor memilih untuk melakukan ubah Paket
 - A-2 Aktor memilih untuk melakukan hapus Paket

A-3 Aktor memilih untuk melakukan filter Paket

4. Aktor meminta sistem untuk menyimpan data Paket yang telah diinputkan.
5. Sistem mengecek Paket yang telah diinputkan.
6. Sistem menyimpan data Paket ke database.
7. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk melakukan ubah Paket

1. Aktor mencari data Paket yang ingin diubah
2. Sistem menampilkan data Paket yang ingin diubah
E-1 Data Paket yang mau diubah tidak ada
3. Data Paket yang telah di pilih lalu diubah
4. Berlanjut ke Basic Flow langkah ke 7

A-2 Aktor memilih untuk melakukan hapus Paket

1. Aktor mencari data Paket yang ingin dihapus
2. Sistem menghapus data Paket yang dipilih aktor
3. Data Paket yang telah di pilih dihapus
4. Berlanjut ke Basic Flow langkah ke 7

A-3 Aktor memilih untuk melakukan filter Paket

1. Aktor mengetik data yang ingin dicari
2. Sistem menampilkan data Paket
3. Berlanjut ke Basic Flow langkah ke 7

6. Error Flow

E-1 Data Paket yang ingin diubah tidak ada

1. Sistem memberikan pesan peringatan bahwa data yang ingin diubah tidak ada
2. Kembali ke Alternatif Flow A-1 Langkah ke 1

7. PreConditions

1. Use Case Pengelolaan Paket telah dilakukan
2. Aktor telah memasuki sistem

8. PostConditions

1. Data Paket di database telah terupdate.

4.1.3 Use case Spesification : Pengelolaan Gedung

1. Brief Description

Use Case ini digunakan oleh administrator untuk mengelola data Gedung pada paket pernikahan. Admin dapat melakukan tambah Gedung, ubah Gedung, hapus Gedung dan filter Gedung.

2. Primary Actor

Administrator

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data Gedung.
2. Sistem memberikan pilihan untuk melakukan tambah Gedung, ubah Gedung, hapus Gedung dan filter Gedung.
3. Aktor memilih untuk melakukan tambah Gedung

- A-1 Aktor memilih untuk melakukan ubah Gedung
- A-2 Aktor memilih untuk melakukan hapus Gedung
- A-3 Aktor memilih untuk melakukan filter Gedung
- 4. Aktor menginputkan data Gedung.
- 5. Aktor meminta sistem untuk menyimpan data Gedung yang telah diinputkan.
- 6. Sistem mengecek data Gedung yang telah diinputkan.
- 7. Sistem menyimpan data Gedung ke database.
- 8. Use Case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan ubah Gedung
 - 1. Aktor mencari data Gedung yang ingin diubah
 - 2. Sistem menampilkan data Gedung yang ingin diubah
 - E-1 Data yang mau diubah tidak ada
 - 3. Data yang telah di pilih lalu diubah
 - 4. Berlanjut ke Basic Flow langkah ke 8
- A-2 Aktor memilih untuk melakukan hapus Gedung
 - 1. Aktor mencari data Gedung yang ingin dihapus
 - 2. Sistem menghapus data Gedung yang dipilih
 - 3. Data yang telah di pilih telah dihapus
 - 4. Berlanjut ke Basic Flow langkah ke 8
- A-3 Aktor memilih untuk melakukan filter Gedung
 - 1. Aktor mengetik data yang ingin dicari
 - 2. Sistem menampilkan data Gedung
 - 3. Berlanjut ke Basic Flow langkah ke 8

6. Error Flow

E-1 Data yang mau diubah tidak ada

1. Kembali ke Alternatif Flow Langkah A-1 langkah ke 1

E-2 Data Gedung yang ingin dihapus tidak ada

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

7. PreConditions

1. Use Case Pengelolaan Gedung telah dilakukan
2. Aktor telah memasuki sistem

8. PostConditions

1. Data Gedung di database telah terupdate

4.1.4 Use case Spesification : Pengelolaan Katering

1. Brief Description

Use Case ini digunakan oleh administrator untuk mengelola data Katering. Admin dapat melakukan tambah Katering, ubah Katering, hapus Katering dan filter Katering.

2. Primary Actor

Administrator

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data Katering.

2. Sistem memberikan pilihan untuk melakukan tambah data Katering, ubah data Katering, hapus data Katering dan filter Katering
3. Aktor memilih untuk melakukan tambah data Katering
 - A-1 Aktor memilih untuk melakukan ubah data Katering
 - A-2 Aktor memilih untuk melakukan hapus data Katering
 - A-3 Aktor memilih untuk melakukan filter data Katering
4. Aktor menginputkan data Katering
5. Aktor meminta sistem untuk menyimpan data Katering yang telah diinputkan
6. Sistem mengecek data Katering yang telah diinputkan
7. Sistem menyimpan data Katering ke database
8. Use Case selesai

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan ubah data Katering
1. Aktor mencari data Katering yang ingin diubah
 2. Sistem menampilkan data Katering yang ingin diubah
 - E-1 Data yang mau diubah tidak ada
 3. Data yang telah di pilih lalu diubah
 4. Berlanjut ke Basic Flow langkah ke 7
- A-2 Aktor memilih untuk melakukan hapus data Katering
1. Aktor mencari data Katering yang ingin dihapus

2. Sistem menampilkan data Katering yang ingin dihapus

E-2 Data yang mau dihapus tidak ada

3. Data yang telah di pilih telah dihapus

4. Berlanjut ke Basic Flow langkah ke 7

A-3 Aktor memilih untuk melakukan filter Katering

1. Aktor mengetik data yang ingin dicari

2. Sistem menampilkan data Katering

3. Berlanjut ke Basic Flow langkah ke 8

6. Error Flow

E-1 Data yang mau diubah tidak ada

1. Kembali ke Alternatif Flow Langkah A-1 Langkah ke 1

E-2 Data Katering yang ingin dihapus tidak ada

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

7. PreConditions

1. Use Case Pengelolaan Katering telah dilakukan

2. Aktor telah memasuki system

8. PostConditions

1. Data Katering di database telah terupdate

4.1.5 Use case Spesification : Pengelolaan Tradisi

1. Brief Description

Use Case ini digunakan oleh administrator untuk mengelola data Tradisi. Pengelola dapat melakukan tambah data Tradisi, ubah data Tradisi, hapus data Tradisi, dan filter Tradisi.

2. Primary Actor

Administrator

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data Tradisi.
2. Sistem memberikan pilihan untuk melakukan tambah data Tradisi, ubah data Tradisi, hapus data Tradisi, dan filter Tradisi.
3. Aktor memilih untuk melakukan tambah data Tradisi
 - A-1 Aktor memilih untuk melakukan ubah data Tradisi
 - A-2 Aktor memilih untuk melakukan hapus data Tradisi
 - A-3 Aktor memilih untuk melakukan filter data Tradisi
4. Aktor menginputkan data Tradisi
5. Aktor meminta sistem untuk menyimpan data Tradisi yang telah diinputkan
6. Sistem mengecek data Tradisi yang telah diinputkan
7. Sistem menyimpan data Tradisi ke database
8. Use Case selesai

5. Alternative Flow

A-1 Aktor memilih untuk melakukan ubah data Tradisi

1. Aktor mencari data Tradisi yang ingin diubah
2. Sistem menampilkan data Tradisi yang ingin diubah

E-1 Data yang mau diubah tidak ada

3. Data yang telah di pilih diubah
4. Berlanjut ke Basic Flow langkah ke 7

A-2 Aktor memilih untuk melakukan hapus data Tradisi

1. Aktor mencari data Tradisi yang ingin dihapus
2. Sistem menampilkan data Tradisi yang ingin dihapus

E-2 Data yang mau dihapus tidak ada

3. Data yang telah di pilih dihapus
4. Berlanjut ke Basic Flow langkah ke 7

A-3 Aktor memilih untuk melakukan filter Tradisi

1. Aktor menginputkan data Tradisi yang ingin dicari
2. Sistem menampilkan data Tradisi yang dicari
3. Berlanjut ke Basic Flow langkah ke 8

6. Error Flow

E-1 Data yang mau diubah tidak ada

1. Sistem memberikan pesan peringatan bahwa data yang mau diubah tidak ada
2. Kembali ke Basic Flow Langkah A-1 Langkah ke 1

E-2 Data Tradisi yang ingin dihapus tidak ada

1. Sistem memberikan pesan peringatan bahwa data yang ingin dihapus tidak ada
2. Kembali ke Alternative Flow A-2 Langkah ke 1

7. PreConditions

1. Use Case Pengelolaan Tradisi telah dilakukan
2. Aktor telah memasuki sistem

8. PostConditions

1. Data Tradisi di database telah terupdate

4.1.6 Use case Spesification : Pengelolaan Wisata

1. Brief Description

Use Case ini digunakan oleh administrator untuk mengelola data Wisata. Admin dapat melakukan tambah data Wisata, ubah data Wisata, hapus data Wisata dan filter Wisata.

2. Primary Actor

Administrator

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data Wisata
2. Sistem memberikan pilihan untuk melakukan tambah data Wisata, ubah data Wisata, hapus data Wisata, filter data Wisata
3. Aktor memilih untuk melakukan tambah data Wisata
A-1 Aktor memilih untuk melakukan ubah data Wisata

A-2 Aktor memilih untuk melakukan hapus data Wisata

A-3 Aktor memilih untuk melakukan filter data Wisata

4. Aktor menginputkan data Wisata
5. Aktor meminta sistem untuk menyimpan data Wisata yang telah diinputkan
6. Sistem mengecek data Wisata yang telah diinputkan
7. Sistem menyimpan data Wisata ke database
8. Use Case selesai

5. Alternative Flow

A-1 Aktor memilih untuk melakukan ubah data Wisata

1. Aktor mencari data Wisata yang ingin diubah
2. Sistem menampilkan data Wisata yang ingin diubah

E-1 Data yang mau diubah tidak ada

3. Data yang telah di pilih diubah
4. Berlanjut ke Basic Flow langkah ke 7

A-2 Aktor memilih untuk melakukan hapus data Wisata

1. Aktor mencari data Wisata yang ingin dihapus
2. Sistem menampilkan data Wisata yang ingin dihapus

3. Data yang telah di pilih lalu dihapus
4. Berlanjut ke Basic Flow langkah ke 7

A-3 Aktor memilih untuk melakukan filter Tradisi

1. Aktor menginputkan data Tradisi yang ingin dicari

2. Sistem menampilkan data Tradisi yang dicari

3. Berlanjut ke Basic Flow langkah ke 8

6. Error Flow

E-1 Data yang mau diubah tidak ada

1. Kembali ke Alternatif Flow A-1 Langkah ke 1

E-2 Data Wisata yang dicari tidak ditemukan

1. Sistem memberikan pesan peringatan bahwa data yang dicari tidak ditemukan

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

7. PreConditions

1. Use Case Pengelolaan Data Wisata telah dilakukan

2. Aktor telah memasuki system

8. PostConditions

1. Data Wisata di database telah terupdate

4.1.7 Use case Spesification : Pengelolaan Mobil

1. Brief Description

Use Case ini digunakan oleh administrator untuk mengelola data Mobil. Pengelola dapat melakukan tambah mobil, ubah mobil, hapus mobil dan filter mobil.

2. Primary Actor

Administrator

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data Mobil.
2. Sistem memberikan pilihan untuk melakukan tambah Mobil, ubah Mobil, hapus Mobil, dan filter Mobil
3. Aktor memilih untuk melakukan tambah Mobil
 - A-1 Aktor memilih untuk melakukan ubah Mobil
 - A-2 Aktor memilih untuk melakukan hapus Mobil
 - A-3 Aktor memilih untuk melakukan filter Mobil
4. Aktor menginputkan data Mobil.
5. Aktor meminta sistem untuk menyimpan data Mobil yang telah diinputkan.
6. Sistem mengecek data Mobil yang telah diinputkan.
 - E-1 Data Mobil yang diinputkan aktor salah
7. Sistem menyimpan data Mobil ke database
8. Use Case selesai

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan ubah Mobil
 1. Aktor mencari data Mobil yang ingin diubah
 2. Sistem menampilkan data Mobil yang ingin diubah
 - E-1 Data yang mau diubah tidak ada
 3. Data yang telah di pilih diubah
 4. Berlanjut ke Basic Flow langkah ke 8
- A-2 Aktor memilih untuk melakukan hapus Mobil
 1. Aktor mencari data Mobil yang ingin dihapus
 2. Sistem menampilkan data Mobil yang ingin dihapus
 - E-2 Data yang mau dihapus tidak ada

3. Data yang telah di pilih dihapus

4. Berlanjut ke Basic Flow langkah ke 8

A-3 Aktor memilih untuk melakukan filter Mobil

1. Aktor menginputkan data Mobil yang ingin dicari

2. Sistem menampilkan data Mobil yang dicari

3. Berlanjut ke Basic Flow langkah ke 8

6. Error Flow

E-1 Data yang mau diubah tidak ada

1. Sistem memberikan pesan peringatan bahwa data yang mau diubah tidak ada

2. Kembali ke Basic Flow Langkah ke 4

E-2 Data Mobil yang ingin dihapus tidak ada

1. Sistem memberikan pesan peringatan bahwa data yang ingin dihapus tidak ada

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

7. PreConditions

1. Use Case Pengelolaan Data Mobil telah dilakukan

2. Aktor telah memasuki sistem.

8. PostConditions

1. Data Mobil di database telah terupdate.

4.1.8 Use case Spesification : Pengelolaan Undangan

1. Brief Description

Use Case ini digunakan oleh administrator untuk mengelola data Undangan. Pengelola dapat melakukan tambah Undangan, ubah Undangan, hapus Undangan.

2. Primary Actor

Administrator

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data Undangan.
2. Sistem memberikan pilihan untuk melakukan tambah Undangan, ubah Undangan, hapus Undangan.
3. Aktor memilih untuk melakukan tambah Undangan
 - A-1 Aktor memilih untuk melakukan ubah data Undangan
 - A-2 Aktor memilih untuk melakukan hapus data Undangan
4. Aktor menginputkan data Undangan.
5. Aktor meminta sistem untuk menyimpan data Undangan yang telah diinputkan.
6. Sistem mengecek data Undangan yang telah diinputkan.
 - E-1 Data Undangan yang diinputkan aktor salah
7. Sistem menyimpan data Undangan ke database.
8. Use Case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan ubah Undangan

1. Aktor mencari data Undangan yang ingin diubah
2. Sistem menampilkan data Undangan yang ingin diubah

E-1 Data yang mau diubah tidak ada

3. Data yang telah di pilih diubah
4. Berlanjut ke Basic Flow langkah ke 8

A-2 Aktor memilih untuk melakukan hapus Undangan

1. Aktor mencari data Undangan yang ingin dihapus
2. Sistem menampilkan data Undangan yang ingin dihapus

E-2 Data yang mau dihapus tidak ada

3. Data yang telah di pilih dihapus
4. Berlanjut ke Basic Flow langkah ke 8

6. Error Flow

E-1 Data Undangan yang mau diubah tidak ada

1. Sistem memberikan pesan peringatan bahwa data yang mau diubah tidak ada.
2. Kembali ke Basic Flow Langkah ke 4

E-2 Data Undangan yang ingin dihapus tidak ada

1. Sistem memberikan pesan peringatan bahwa data yang ingin dihapus tidak ada.
2. Kembali ke Alternative Flow A-2 Langkah ke 1

7. PreConditions

1. Use Case Pengelolaan Data Undangan telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Data Undangan di database telah terupdate.

4.1.9 Use Case Spesification : Cetak Laporan Transaksi

1. Brief Description

Use Case ini digunakan administrator untuk melakukan cetak laporan transaksi customer.

2. Primary Actor

Administrator

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika administrator memilih untuk melakukan cetak laporan transaksi customer.
2. Sistem menampilkan laporan transaksi customer.
3. Administrator melakukan cetak laporan transaksi.
4. Use Case selesai

5. Alternative Flow

-

6. Error Flow

None

7. PreConditions

1. Administrator melihat laporan Transaksi.

8. PostConditions

1. Administrator telah mencetak laporan transaksi.

B. User

4.1.10 Use Case Specification : Registrasi Customer

1. Brief Description

Use Case ini digunakan customer untuk masuk ke dalam sistem informasi sehingga customer bisa melakukan transaksi pemesanan paket pernikahan. Di sini customer harus menginputkan data diri untuk keperluan registrasi sebagai member.

2. Primary Actor

Customer

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika customer memilih untuk melakukan registrasi.
2. Customer melakukan input data pribadi.
3. Aktor meminta sistem untuk menyimpan data customer yang telah diinputkan
4. Sistem menyimpan data-data customer di database
5. Use Case selesai

5. Error Flow

E-1 Data customer yang diinputkan salah

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

6. PreConditions

1. Customer telah melakukan registrasi

7. PostConditions

1. Customer dapat melihat paket pernikahan dan melakukan transaksi.

4.1.11 Use case Spesification : Login Customer

1. Brief Description

Use Case login ini digunakan oleh Customer untuk memperoleh akses ke dalam sistem dan melakukan reservasi paket. Login di dasarkan pada email dan password.

2. Primary Actor

1. Customer

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika customer memilih untuk melakukan login
2. Sistem menampilkan antarmuka untuk login
3. Customer memasukkan email dan password
4. Sistem memeriksa email dan password yang diinputkan customer

E-1 Password atau email tidak sesuai

5. Sistem memberikan akses ke aktor

6. Use Case ini selesai

5. Alternative Flow

None

6. Error Flow

E-1 Password atau email tidak sesuai

1. Sistem menampilkan peringatan bahwa email atau password tidak sesuai

2. Kembali ke Basic Flow langkah ke 3

7. PreConditions

None

8. PostConditions

1. Customer memasuki sistem dan dapat menggunakan fungsi fungsi pada sistem.

4.1.12 Use Case Spesification : Kelola Data Customer

1. Brief Description

Use Case ini digunakan customer untuk mengelola data-data customer. Customer dapat mengubah data-data pribadi mereka.

2. Primary Actor

Customer

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika customer memilih untuk mengelola data customer.
2. Sistem memberikan pilihan untuk melakukan ubah data
3. Customer memilih untuk melakukan ubah data
4. Customer melakukan ubah data customer
5. Customer meminta sistem untuk menyimpan data yang telah diubah
6. Sistem menyimpan data-data customer di database
7. Use Case selesai

5. Alternative Flow

-

6. Error Flow

-

7. PreConditions

1. Customer telah mengelola data

8. PostConditions

1. Data Customer telah terupdate.

4.1.13 Use case Spesification : Reservasi Paket

1. Brief Description

Use Case ini digunakan customer untuk reservasi paket pernikahan. Di sini customer harus memilih paket pernikahan.

2. Primary Actor

Customer

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika customer memilih untuk melakukan pemesanan.
2. Customer memilih paket pernikahan yang ada
3. Sistem memberikan langkah-langkah yang harus diisi atau dipilih oleh customer
4. Sistem menyimpan data-data pemesanan customer di database
5. Use Case selesai

5. Error Flow

-

6. PreConditions

1. Customer telah memilih paket pernikahan

7. PostConditions

1. Customer dapat melihat detail data paket pernikahan yang sudah dipesan.

4.1.14 Use Case Specification : Transaksi

1. Brief Description

Use Case ini digunakan customer untuk melakukan transaksi paket pernikahan, tampil paket pernikahan, dan cetak paket pernikahan.

2. Primary Actor

Customer

3. Supporting Actor

None

4. Basic Flow

3. Use Case ini dimulai ketika customer memilih untuk melakukan transaksi paket pernikahan
4. Sistem memberikan pilihan untuk melakukan transaksi paket pernikahan, tampil paket pernikahan atau cetak paket pernikahan
3. Customer memilih untuk melakukan transaksi paket pernikahan

A-1 Customer memilih untuk tampil paket pernikahan

A-2 Customer memilih untuk melakukan cetak paket pernikahan

4. Customer melakukan transaksi paket pernikahan
5. Sistem menampilkan transaksi paket pernikahan
6. Use Case selesai

5. Alternative Flow

A-1 Customer memilih untuk melakukan tampil paket pernikahan

1. Sistem menampilkan paket pernikahan customer
2. Berlanjut ke Basic Flow langkah ke 5

A-2 Customer memilih untuk melakukan cetak paket pernikahan

1. Sistem menampilkan paket pernikahan customer
2. Sistem mencetak paket pernikahan customer
3. Berlanjut ke Basic Flow langkah ke 5

6. Error Flow

None

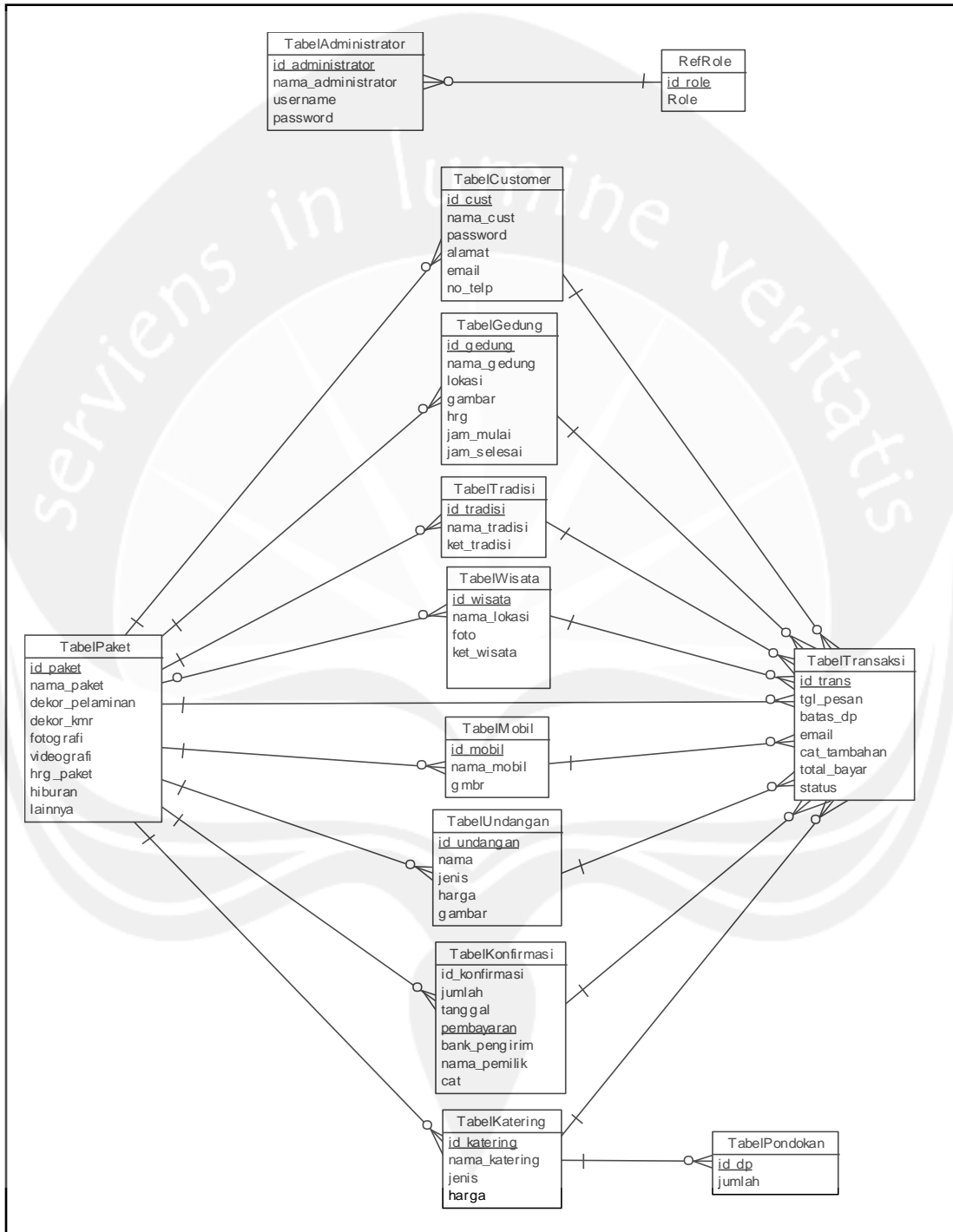
7. PreConditions

1. Customer melakukan transaksi

8. PostConditions

1. Customer telah melakukan transaksi.

5 Entity Relationship Diagram (ERD)



Gambar 4. Entity Relationship Diagram

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

('Sistem Informasi Wedding Organizer')

(SIWO)

Untuk :


Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Maria Stevani N.P / 5961

Program Studi Teknik Informatika - Fakultas Teknologi
Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
		DPPL-SIWO		1/71
		Revisi		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

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

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi



Daftar Isi

1	Pendahuluan.....	6
1.1	Tujuan.....	6
1.2	Ruang Lingkup.....	6
1.3	Definisi dan Akronim.....	6
1.4	Referensi.....	7
2	Perancangan Rinci.....	7
2.1	Sequence Diagram.....	7
2.1.1	Login Administrator.....	7
2.1.2	Ubah Password.....	8
2.1.3	Pengelolaan Data Paket.....	8
2.1.3.1	Tambah Paket.....	8
2.1.3.2	Ubah Paket.....	9
2.1.3.3	Hapus Paket.....	9
2.1.4	Pengelolaan Data Gedung.....	10
2.1.4.1	Tambah Gedung.....	10
2.1.4.2	Ubah Gedung.....	10
2.1.4.3	Hapus Gedung.....	11
2.1.5	Pengelolaan Katering.....	11
2.1.5.1	Tambah Katering.....	11
2.1.5.2	Ubah Katering.....	12
2.1.5.3	Hapus Katering.....	12
2.1.6	Pengelolaan Wisata.....	13
2.1.6.1	Tambah Wisata.....	13
2.1.6.2	Ubah Wisata.....	13
2.1.6.3	Hapus Wisata.....	14
2.1.7	Pengelolaan Tradisi.....	14
2.1.7.1	Tambah Tradisi.....	14
2.1.7.2	Ubah Tradisi.....	15
2.1.7.3	Hapus Tradisi.....	15
2.1.8	Pengelolaan Mobil.....	16
2.1.8.1	Tambah Mobil.....	16
2.1.8.2	Ubah Mobil.....	16
2.1.8.3	Hapus Mobil.....	17
2.1.9	Pengelolaan Undangan.....	17
2.1.9.1	Tambah Undangan.....	17
2.1.9.2	Ubah Undangan.....	18
2.1.9.3	Hapus Undangan Mobil.....	18
2.1.10	Registrasi Customer.....	19
2.1.10	Login Customer.....	19
2.1.11	Pengelolaan Data Customer.....	20
2.1.11.1	Ubah Data.....	20
2.1.12	Reservasi Paket.....	21
2.1.13	Transaksi.....	21

Daftar Gambar

Gambar 2.1	Sequence Diagram : Login.....	7
Gambar 2.2	Sequence Diagram : Ubah Password.....	8
Gambar 2.3	Sequence Diagram : Tambah Paket.....	8
Gambar 2.4	Sequence Diagram : Ubah Paket.....	9
Gambar 2.5	Sequence Diagram : Hapus Paket.....	9
Gambar 2.6	Sequence Diagram : Tambah Gedung.....	10
Gambar 2.7	Sequence Diagram : Ubah Gedung.....	10
Gambar 2.8	Sequence Diagram : Hapus Gedung.....	11
Gambar 2.9	Sequence Diagram : Tambah Katering.....	11
Gambar 2.10	Sequence Diagram : Ubah Katering.....	12
Gambar 2.11	Diagram Sequence : Hapus Katering.....	12
Gambar 2.12	Diagram Sequence : Tambah Wisata.....	13
Gambar 2.13	Diagram Sequence : Ubah Wisata.....	13
Gambar 2.14	Diagram Sequence : Hapus Wisata.....	14
Gambar 2.15	Diagram Sequence : Tambah Tradisi.....	14
Gambar 2.16	Diagram Sequence : Ubah Tradisi.....	15
Gambar 2.17	Diagram Sequence : Hapus Tradisi.....	15
Gambar 2.18	Diagram Sequence : Tambah Mobil.....	16
Gambar 2.19	Diagram Sequence : Ubah Mobil.....	16
Gambar 2.20	Diagram Sequence : Hapus Mobil.....	17
Gambar 2.21	Diagram Sequence : Tambah Undangan.....	17
Gambar 2.22	Diagram Sequence : Ubah Undangan.....	18
Gambar 2.23	Diagram Sequence : Hapus Undangan.....	18
Gambar 2.24	Diagram Sequence : Registrasi Customer.....	19
Gambar 2.25	Diagram Sequence : Login Customer.....	19
Gambar 2.26	Diagram Sequence : Ubah Data.....	20
Gambar 2.27	Diagram Sequence : Reservasi Paket.....	20
Gambar 2.28	Diagram Sequence : Transaksi.....	21
Gambar 3.1	Rancangan Arsitektur SIWO.....	22
Gambar 3.2	Class Diagram.....	23
Gambar 3.3	Physical Data Model	51
Gambar 4.1	Antarmuka Login.....	52
Gambar 4.2	Antarmuka Halaman Administrator.....	53
Gambar 4.3	Antarmuka Pengelolaan Paket.....	53
Gambar 4.4	Antarmuka Pengelolaan Gedung.....	54
Gambar 4.5	Antarmuka Pengelolaan Katering.....	55
Gambar 4.6	Antarmuka Pengelolaan Tradisi.....	56
Gambar 4.7	Antarmuka Pengelolaan Mobil.....	57
Gambar 4.8	Antarmuka Pengelolaan Undangan.....	58
Gambar 4.9	Antarmuka Pengelolaan Wisata.....	59
Gambar 4.10	Antarmuka Pengelolaan Data Customer.....	60
Gambar 4.11	Antarmuka Pengelolaan Data Transaksi.....	61
Gambar 4.12	Antarmuka Laporan Transaksi.....	62
Gambar 4.13	Antarmuka Home Page Web.....	63
Gambar 4.14	Antarmuka Paket Web.....	63
Gambar 4.15	Antarmuka Gedung Web.....	64
Gambar 4.16	Antarmuka Katering.....	65
Gambar 4.17	Antarmuka Mobil Web.....	66
Gambar 4.18	Antarmuka Wisata Web.....	67

Gambar 4.19 Antarmuka Undangan Web.....	68
Gambar 4.20 Antarmuka Registrasi Customer Web.....	69
Gambar 4.21 Antarmuka Reservasi Paket Web.....	70
Gambar 4.22 Antarmuka Transaksi Web.....	71



6 Pendahuluan

6.1 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.

6.2 Ruang Lingkup

Perangkat Lunak *SIWO* ini dibangun dengan tujuan untuk :

1. Menangani beberapa hal dalam perencanaan dan pelaksanaan acara pernikahan,
2. Menawarkan macam-macam paket untuk pernikahan dan,
3. Menangani reservasi untuk pemesanan, dan berjalan pada web.

Dan berjalan pada lingkungan dengan platform Windows XP.

6.3 Definisi dan Akronim

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak disebut juga Software Design Description (SDD) merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan.
SIWO	Sistem informasi untuk memberikan informasi mengenai paket pernikahan.

6.4 Referensi

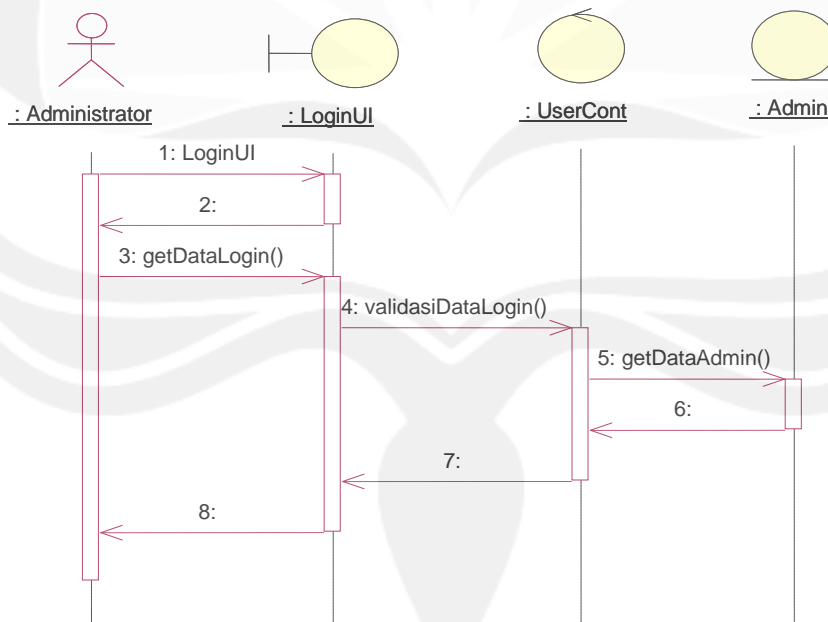
Referensi yang digunakan pada perangkat lunak tersebut adalah:

3. Bennet Simon, McRobb Steve, Farmer Ray, *Object-Oriented System Analysis and Design Using UML*, McGraw-Hill Companies, 2002.
4. Boggs Wendy, Boggs Michael, *Mastering UML with Rational Rose 2002*, SYBEX Inc, 2002.

2 Perancangan Rinci

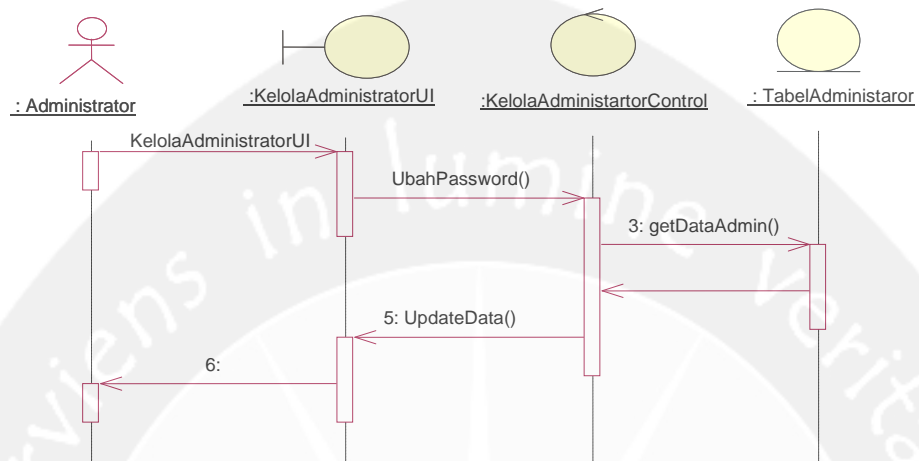
2.1 Sequence Diagram

2.1.1 Login Administrator



Gambar 2.1 Sequence Diagram : Login Administrator

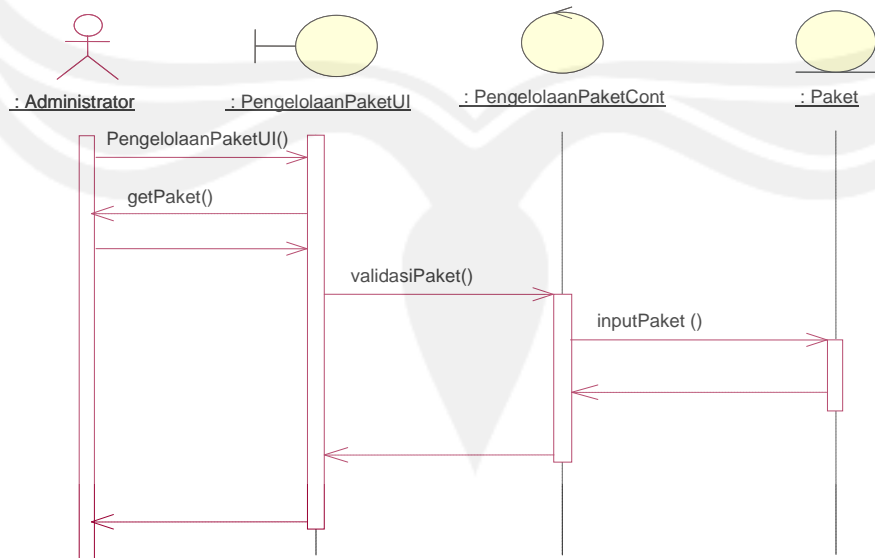
2.1.2 Kelola Administrator



Gambar 2.2 Sequence Diagram : Kelola Administrator

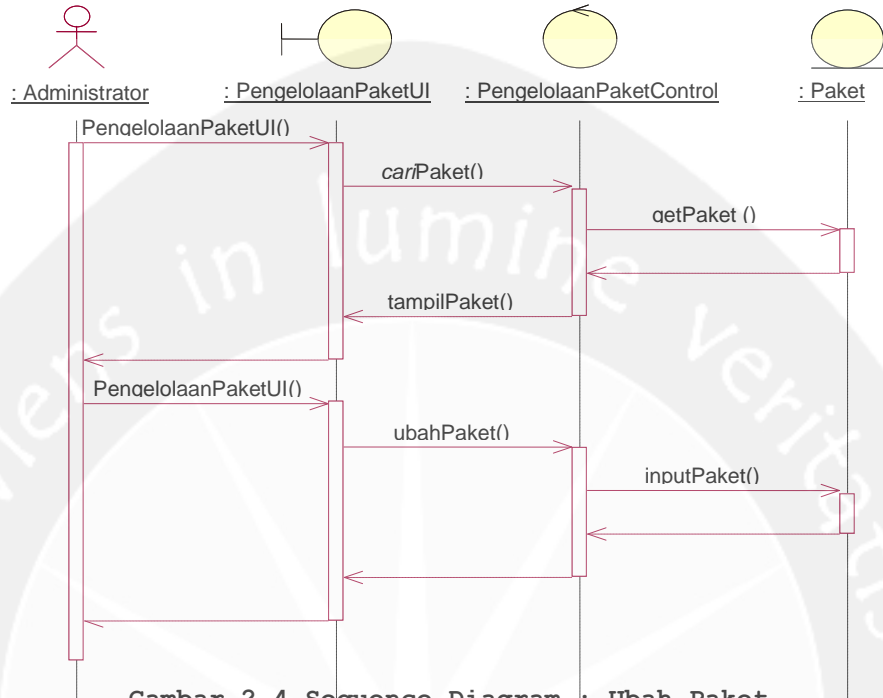
2.1.3 Pengelolaan Paket

2.1.3.1 Tambah Paket



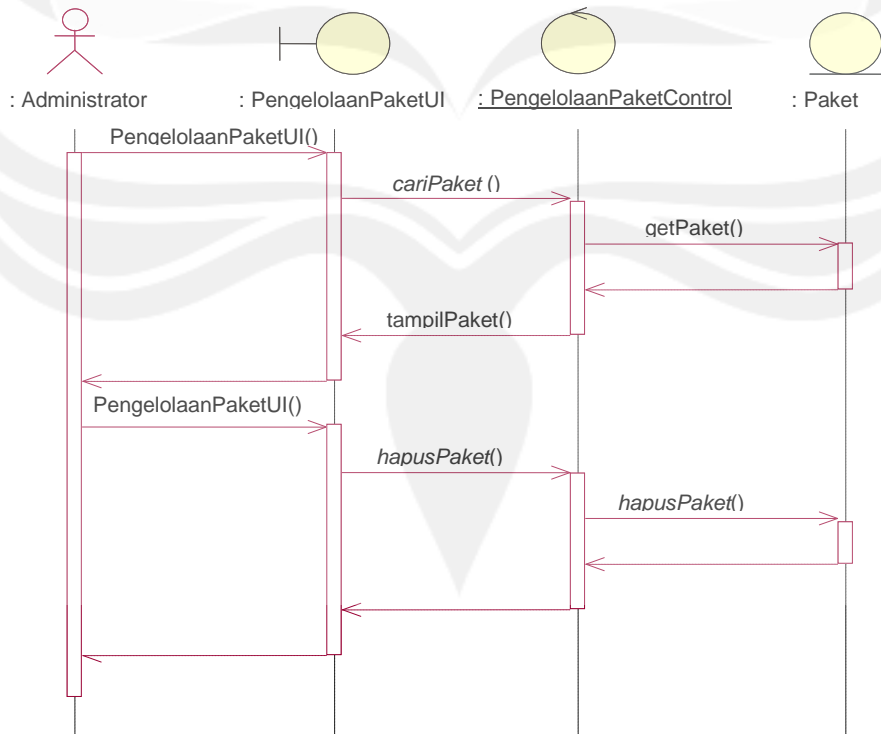
Gambar 2.3 Sequence Diagram : Tambah Paket

2.1.3.2 Ubah Paket



Gambar 2.4 Sequence Diagram : Ubah Paket

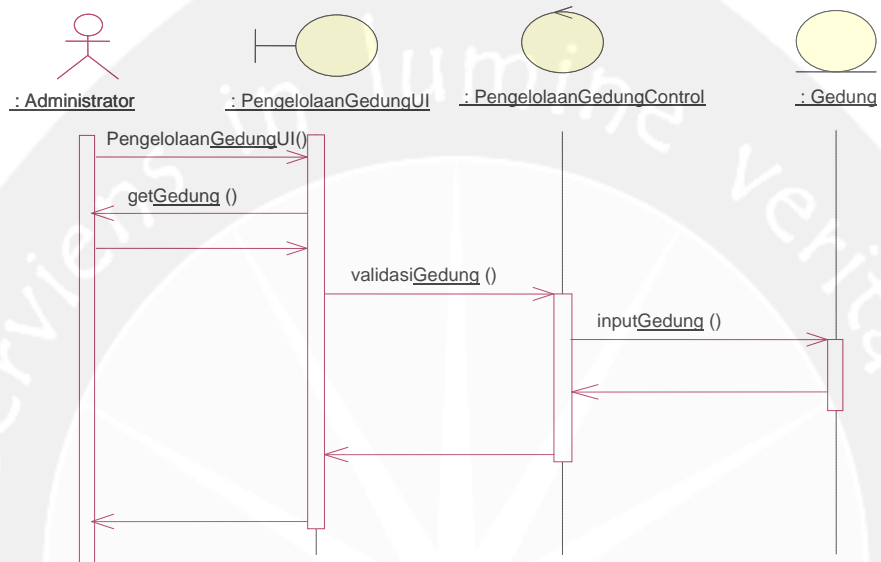
2.1.3.3 Hapus Paket



Gambar 2.5 Sequence Diagram : Hapus Paket

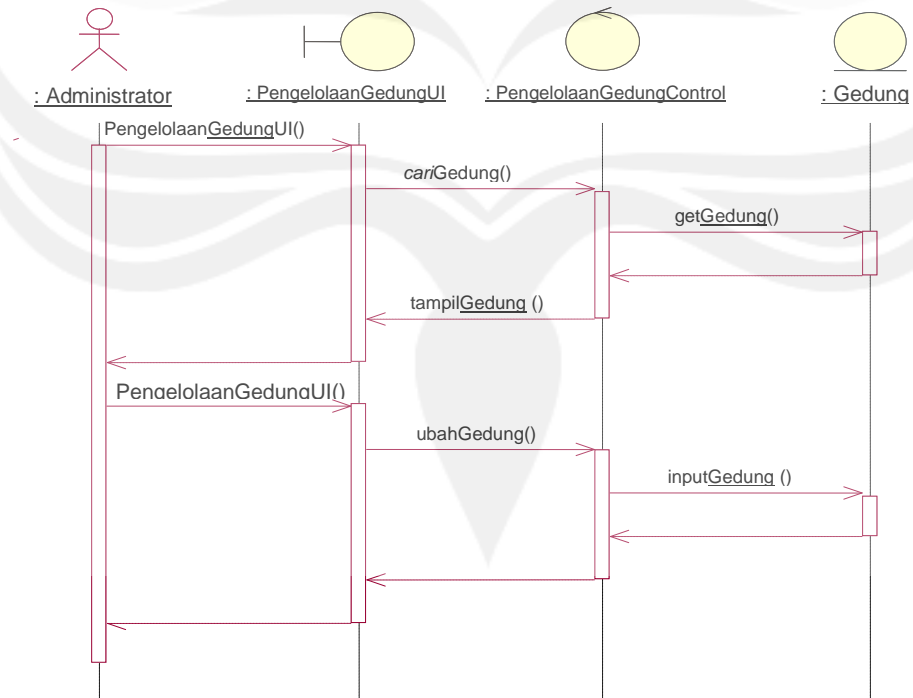
2.1.4 Pengelolaan Gedung

2.1.4.1 Tambah Gedung

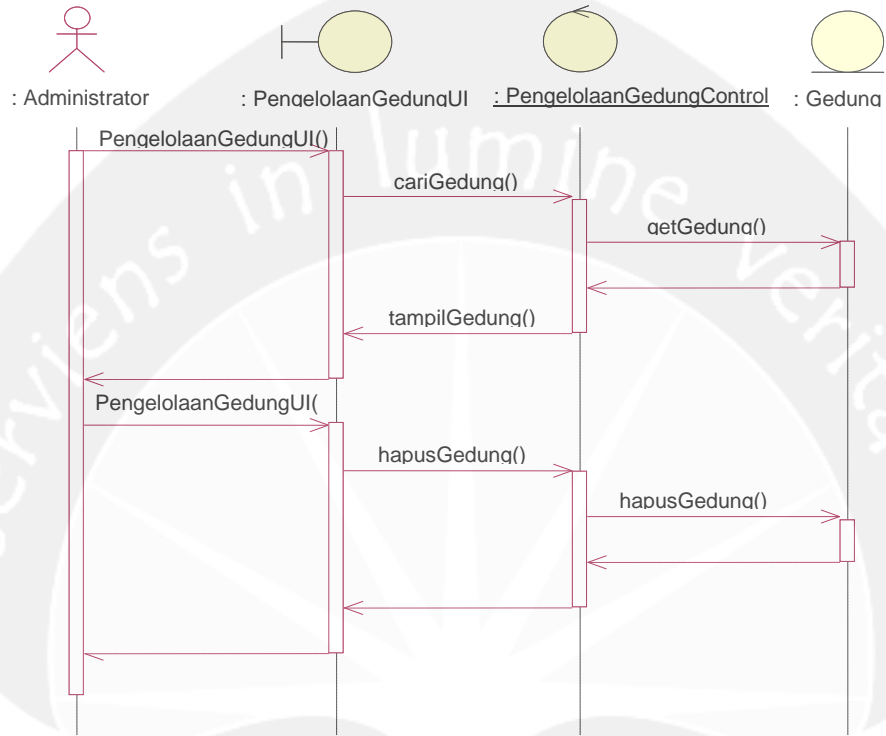


Gambar 2.6 Sequence Diagram : Tambah Gedung

2.1.4.2 Ubah Gedung



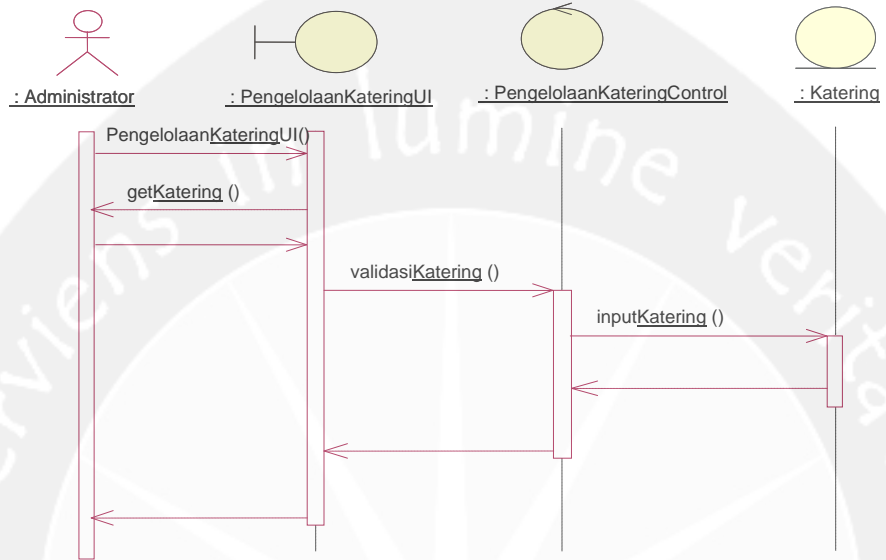
Gambar 2.7 Sequence Diagram : Ubah Gedung
2.1.4.3 Hapus Gedung



Gambar 2.8 Sequence Diagram : Hapus Gedung

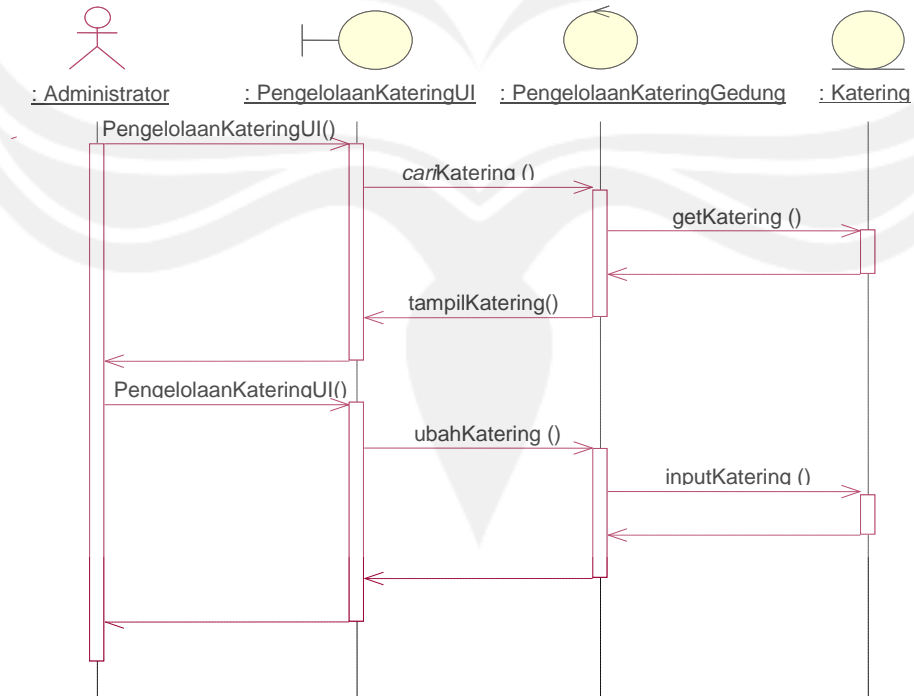
2.1.5 Pengelolaan Katering

2.1.5.1 Tambah Katering



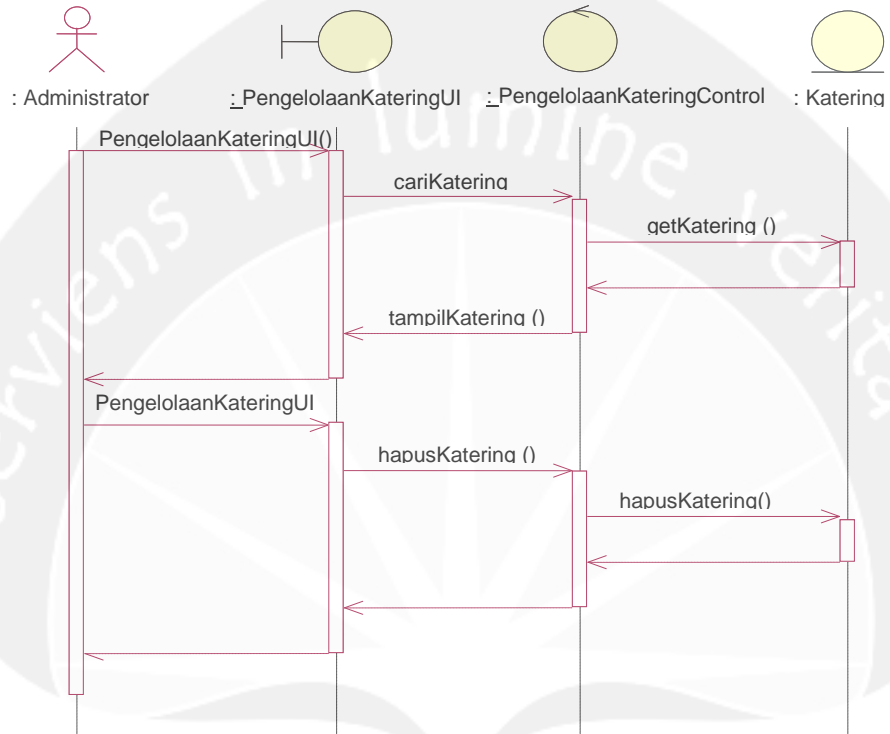
Gambar 2.9 Diagram Sequence : Tambah Katering

2.1.5.2 Ubah Katering



Gambar 2.10 Diagram Sequence : Ubah Katering

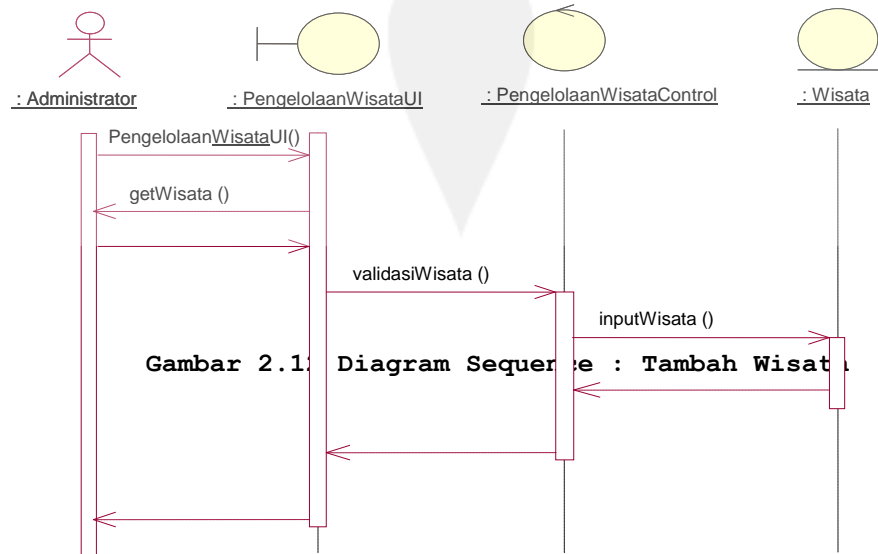
2.1.5.3 Hapus Katering



Gambar 2.11 Diagram Sequence : Hapus Katering

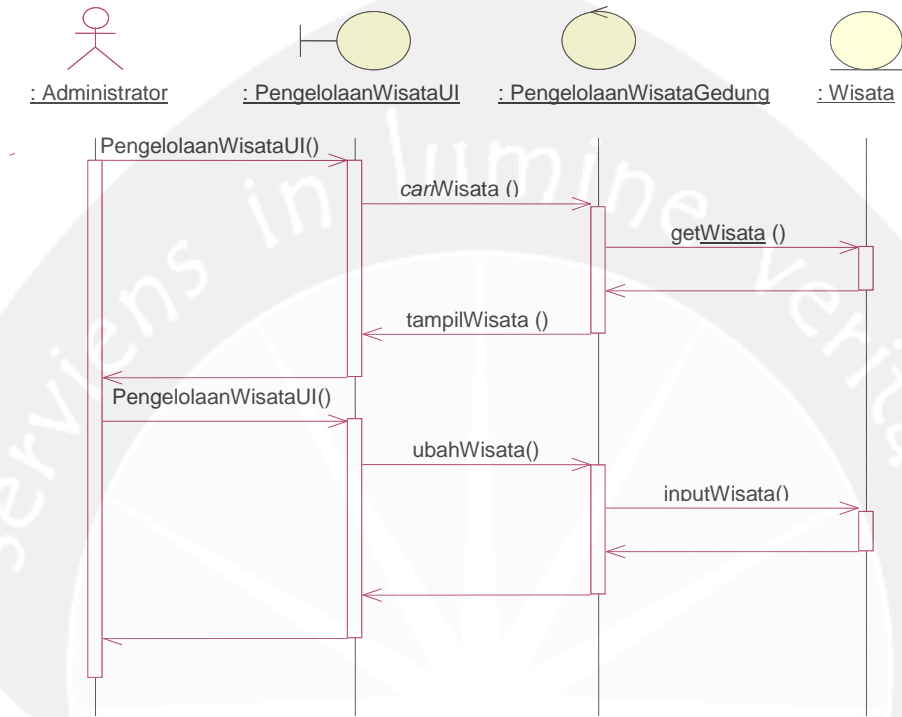
2.1.6 Pengelolaan Wisata

2.1.6.1 Tambah Wisata



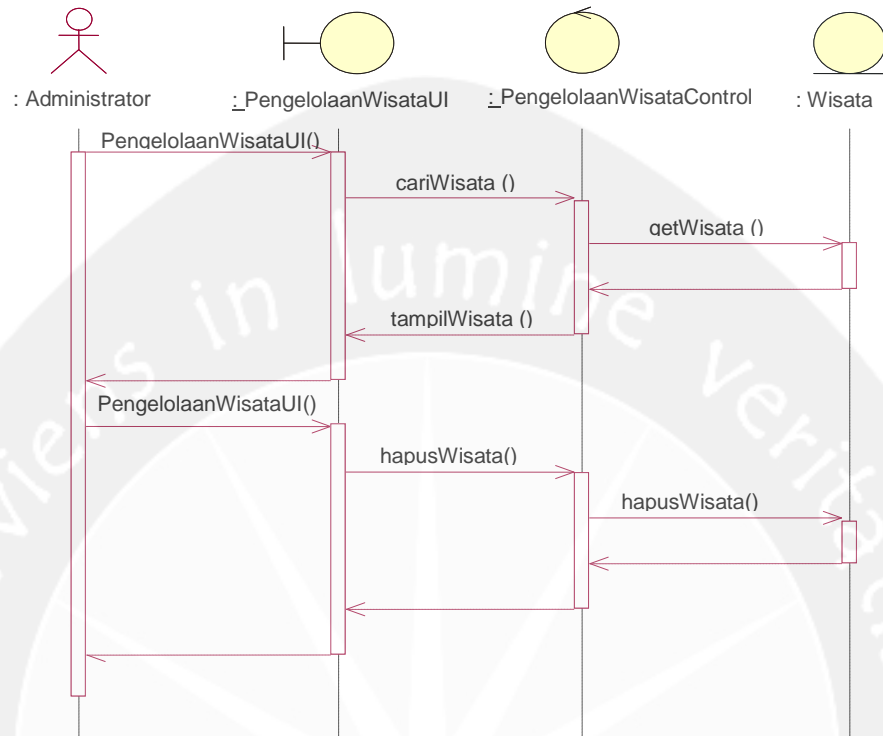
Gambar 2.11 Diagram Sequence : Tambah Wisata

2.1.6.2 Ubah Wisata



Gambar 2.13 Diagram Sequence : Ubah Wisata

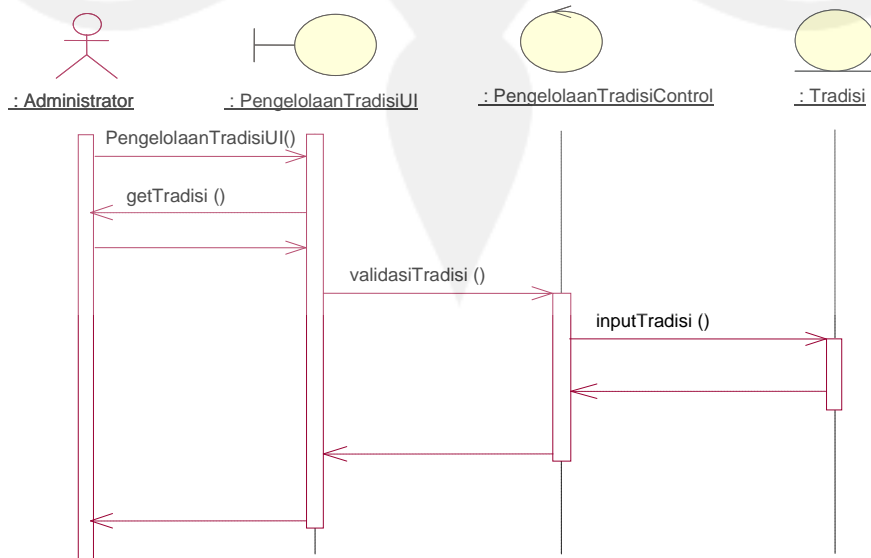
2.1.6.3 Hapus Wisata



Gambar 2.14 Diagram Sequence : Hapus Wisata

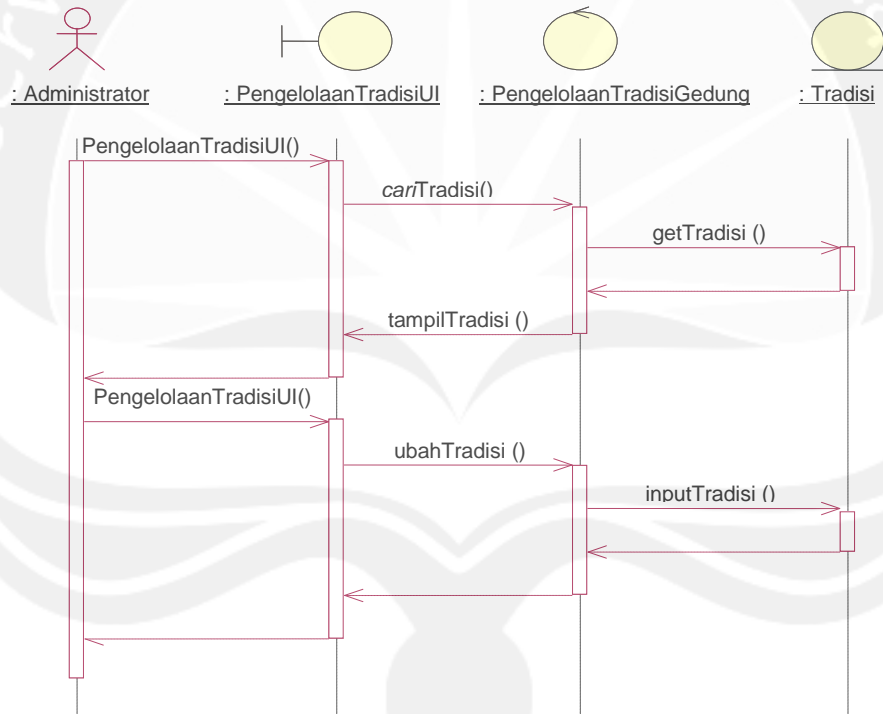
2.1.7 Pengelolaan Tradisi

2.1.7.1 Tambah Tradisi



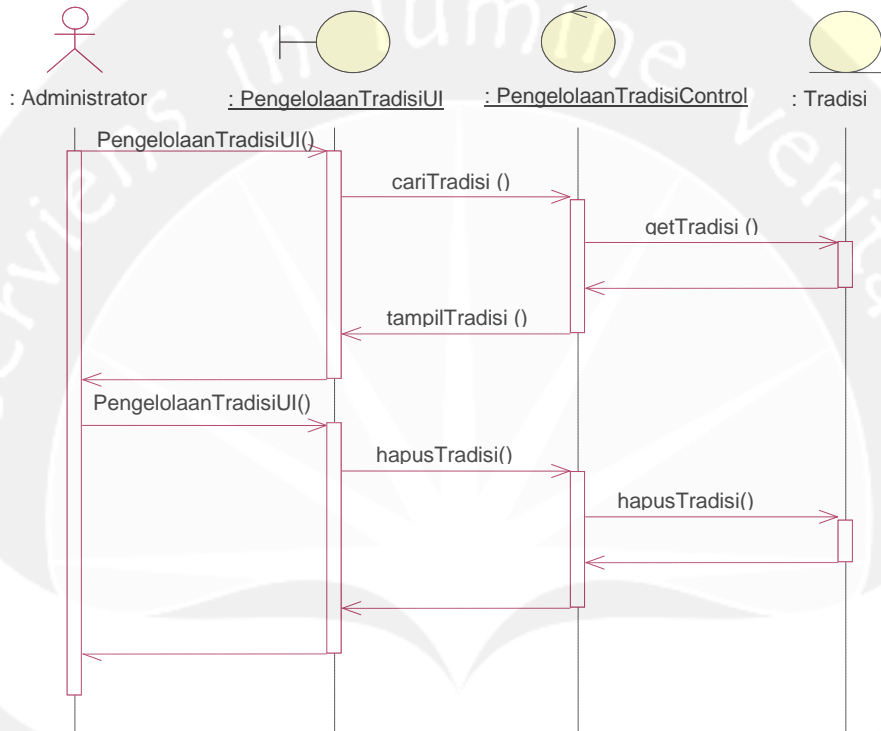
Gambar 2.15 Diagram Sequence : Tambah Tradisi

2.1.7.2 Ubah Tradisi



Gambar 2.16 Diagram Sequence : Ubah Tradisi

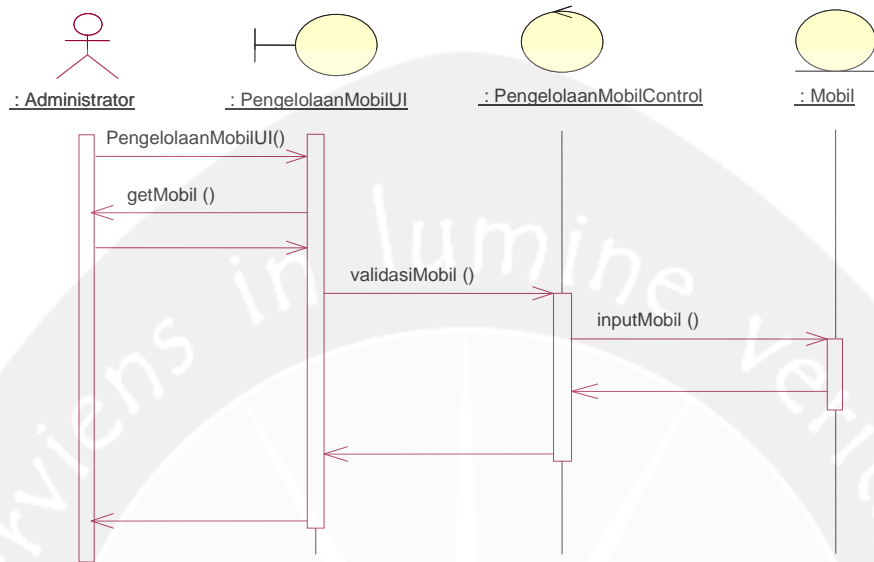
2.1.7.3 Hapus Tradisi



Gambar 2.17 Diagram Sequence : Hapus Tradisi

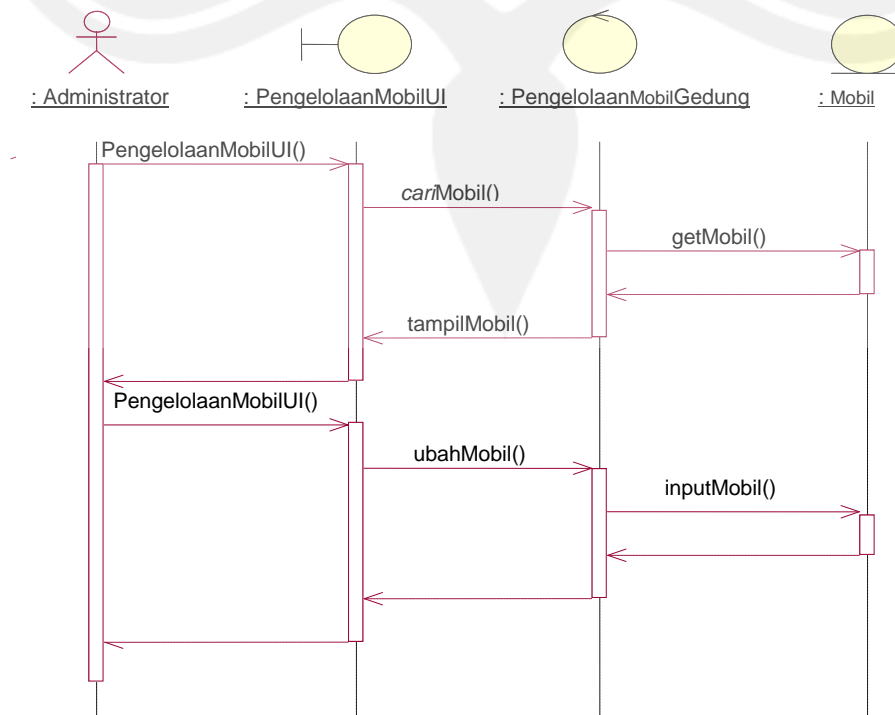
2.1.8 Pengelolaan Mobil

2.1.8.1 Tambah Mobil



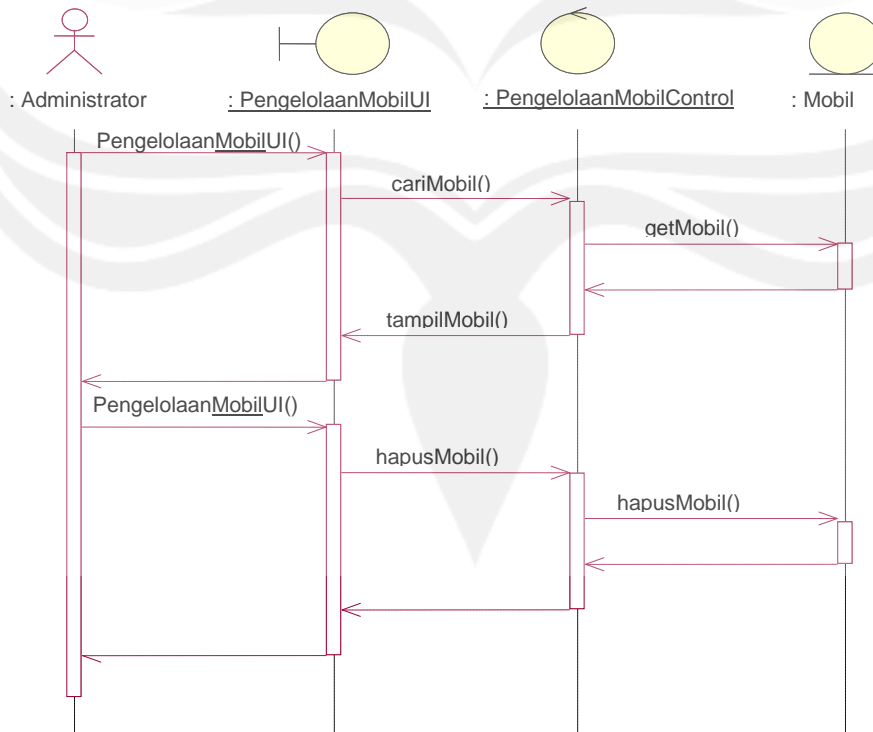
Gambar 2.18 Diagram Sequence : Tambah Mobil

2.1.7.2 Ubah Mobil



Gambar 2.19 Diagram Sequence : Ubah Mobil

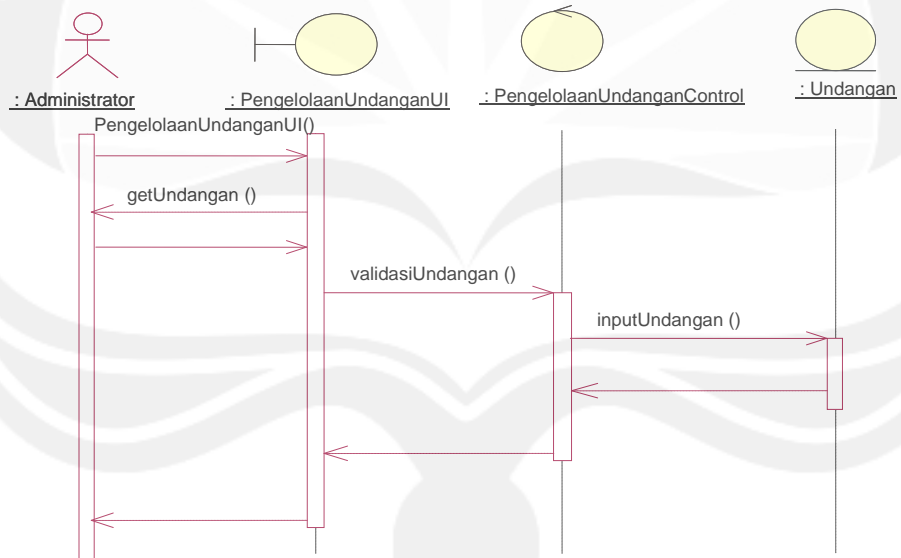
2.1.7.3 Hapus Mobil



Gambar 2.20 Diagram Sequence : Hapus Mobil

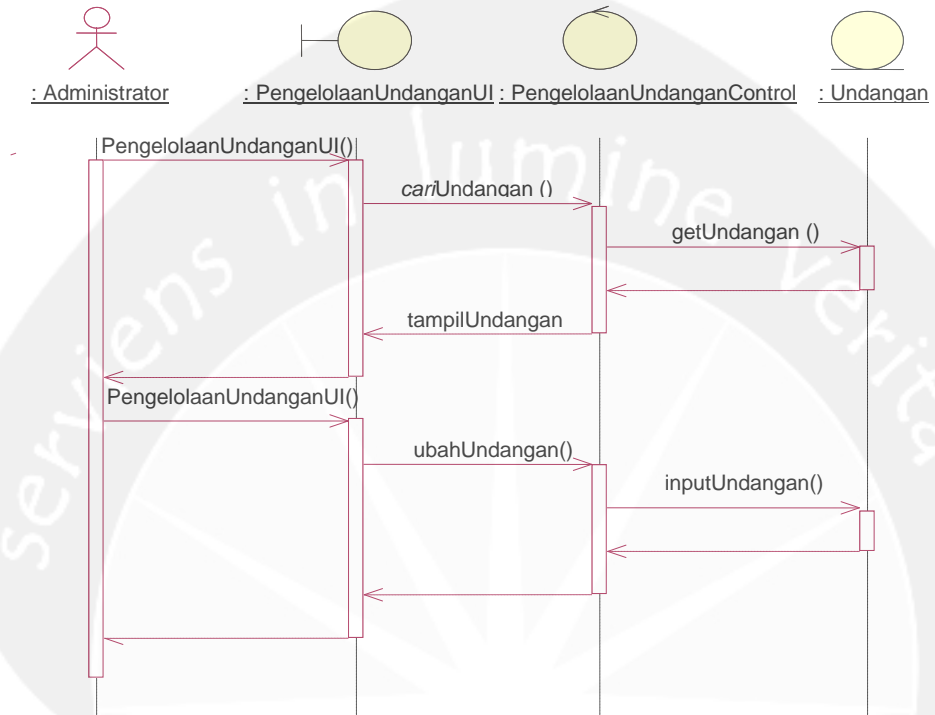
2.1.9 Pengelolaan Undangan

2.1.9.1 Tambah Undangan



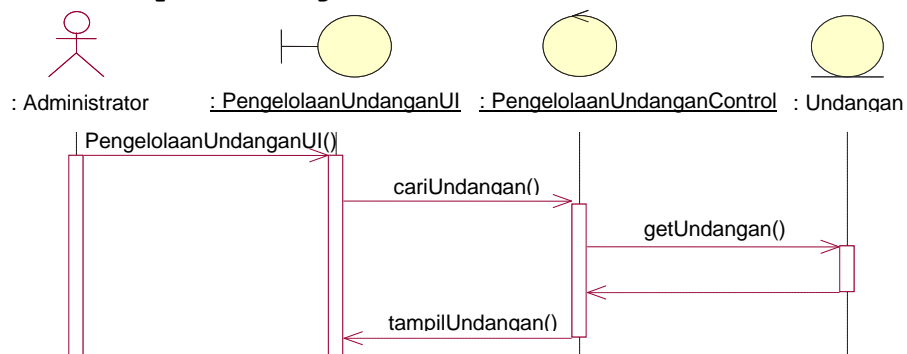
Gambar 2.21 Diagram Sequence : Tambah Undangan

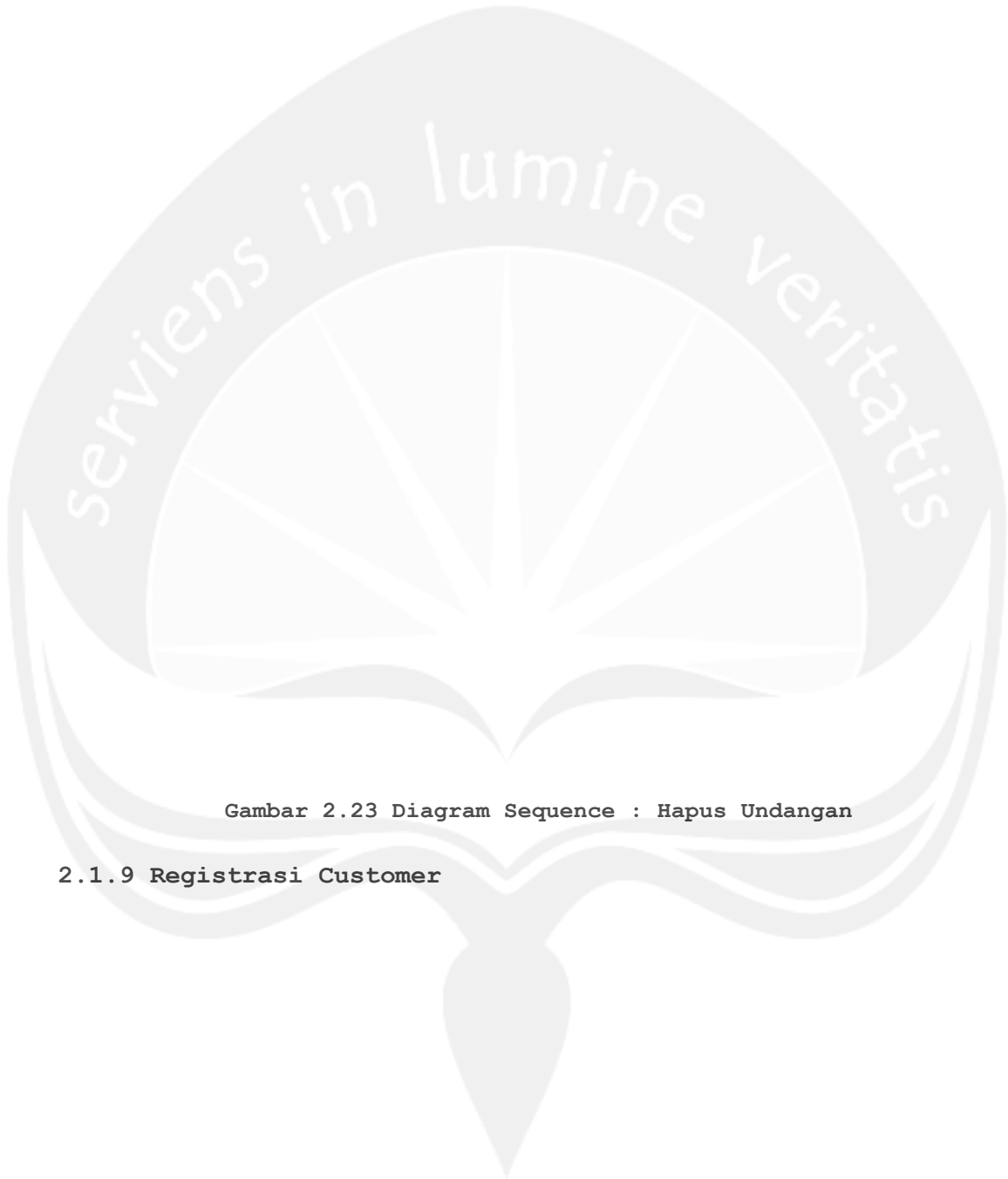
2.1.9.2 Ubah Undangan



Gambar 2.22 Diagram Sequence : Ubah Undangan

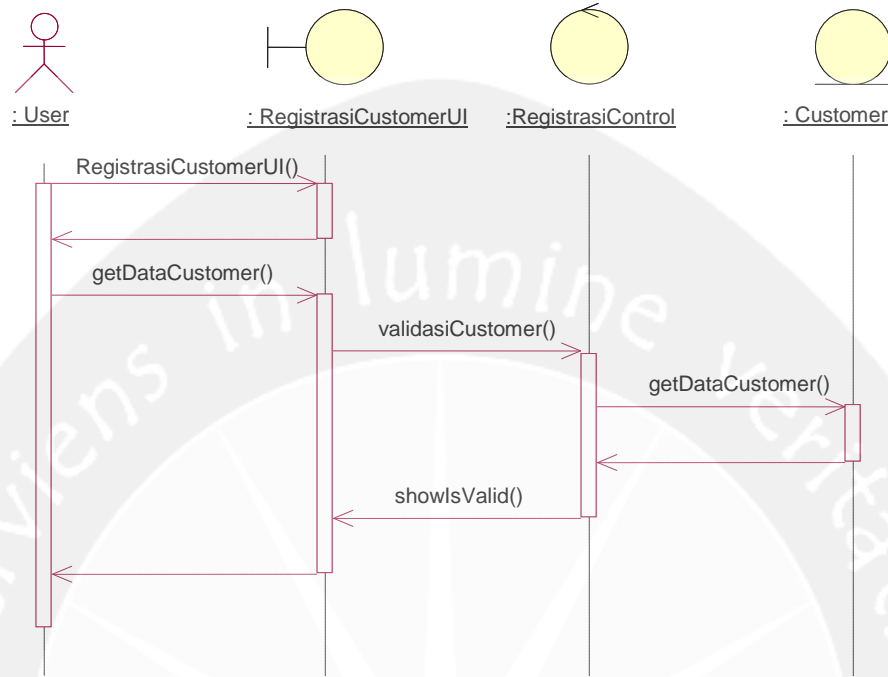
2.1.7.3 Hapus Undangan





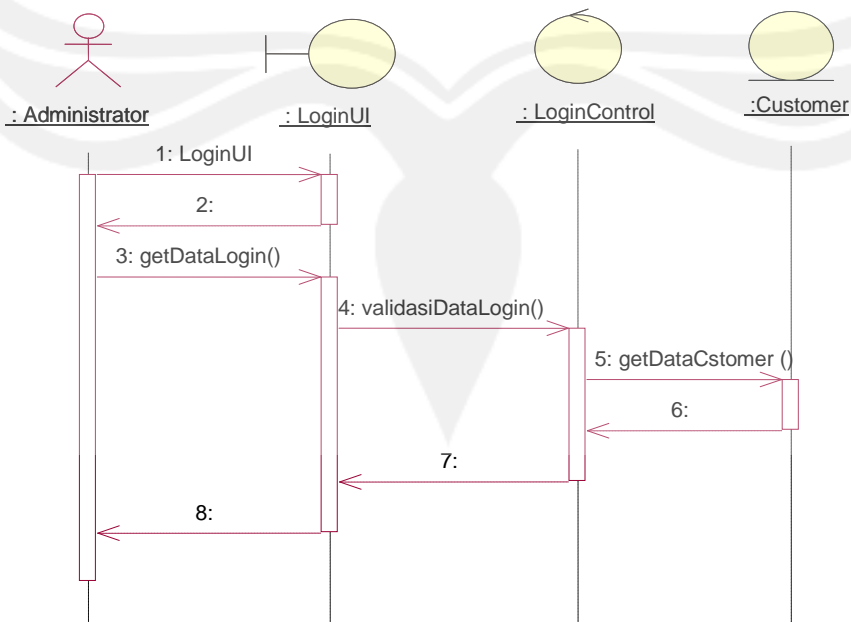
Gambar 2.23 Diagram Sequence : Hapus Undangan

2.1.9 Registrasi Customer



Gambar 2.24 Diagram Sequence : Registrasi Customer

2.1.10 Login Customer

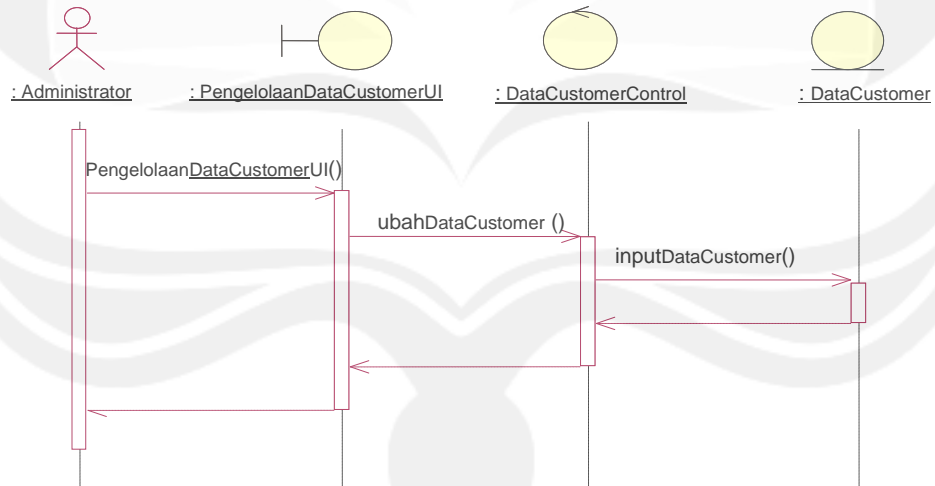


serviens in lumine veritatis

Gambar 2.25 Diagram Sequence : Login Customer

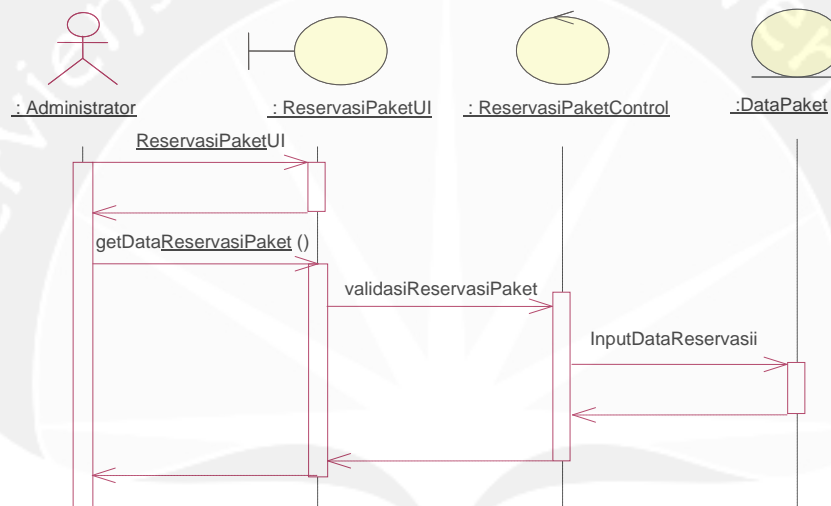
2.1.20 Pengelolaan Customer

2.1.10.1 Ubah Data Customer



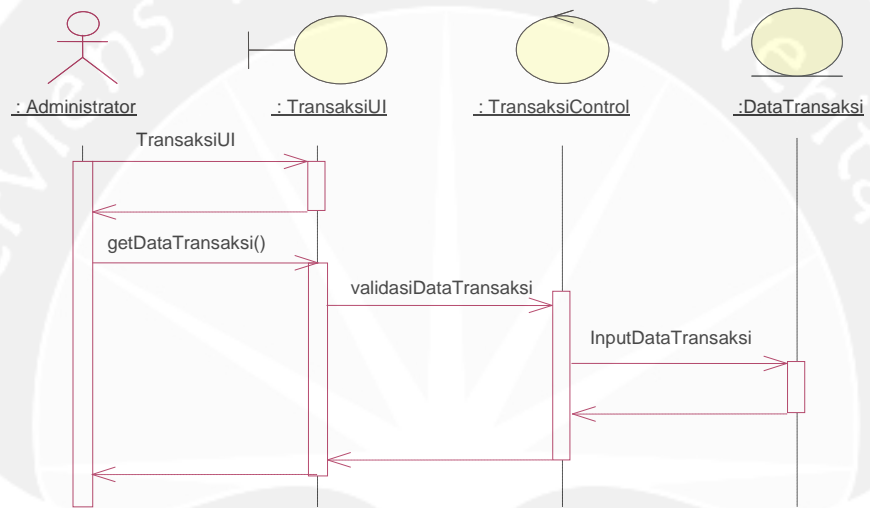
Gambar 2.26 Diagram Sequence : Ubah Data Customer

2.1.11 Reservasi Paket



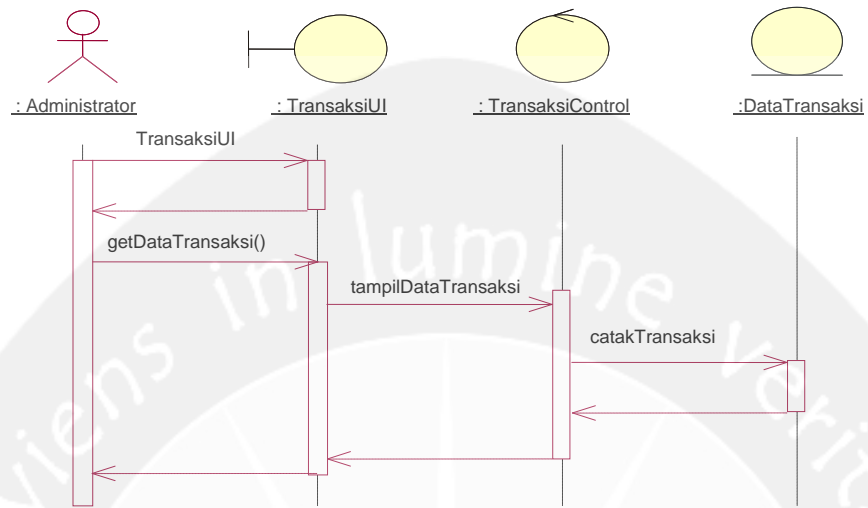
Gambar 2.27 Diagram Sequence : Reservasi Paket

2.1.12 Transaksi



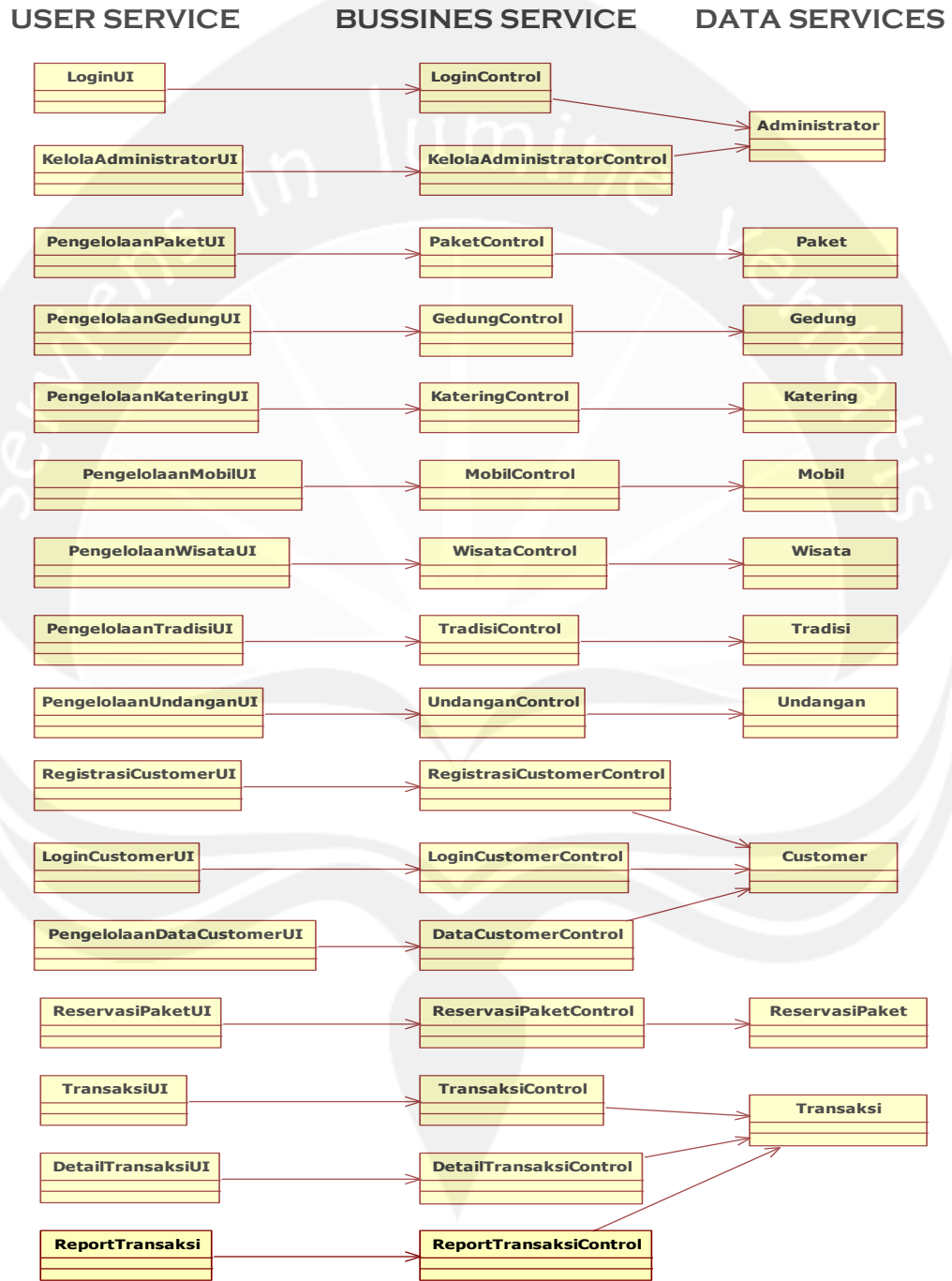
Gambar 2.28 Sequence Diagram : Transaksi

2.1.12 Cetak Laporan Transaksi



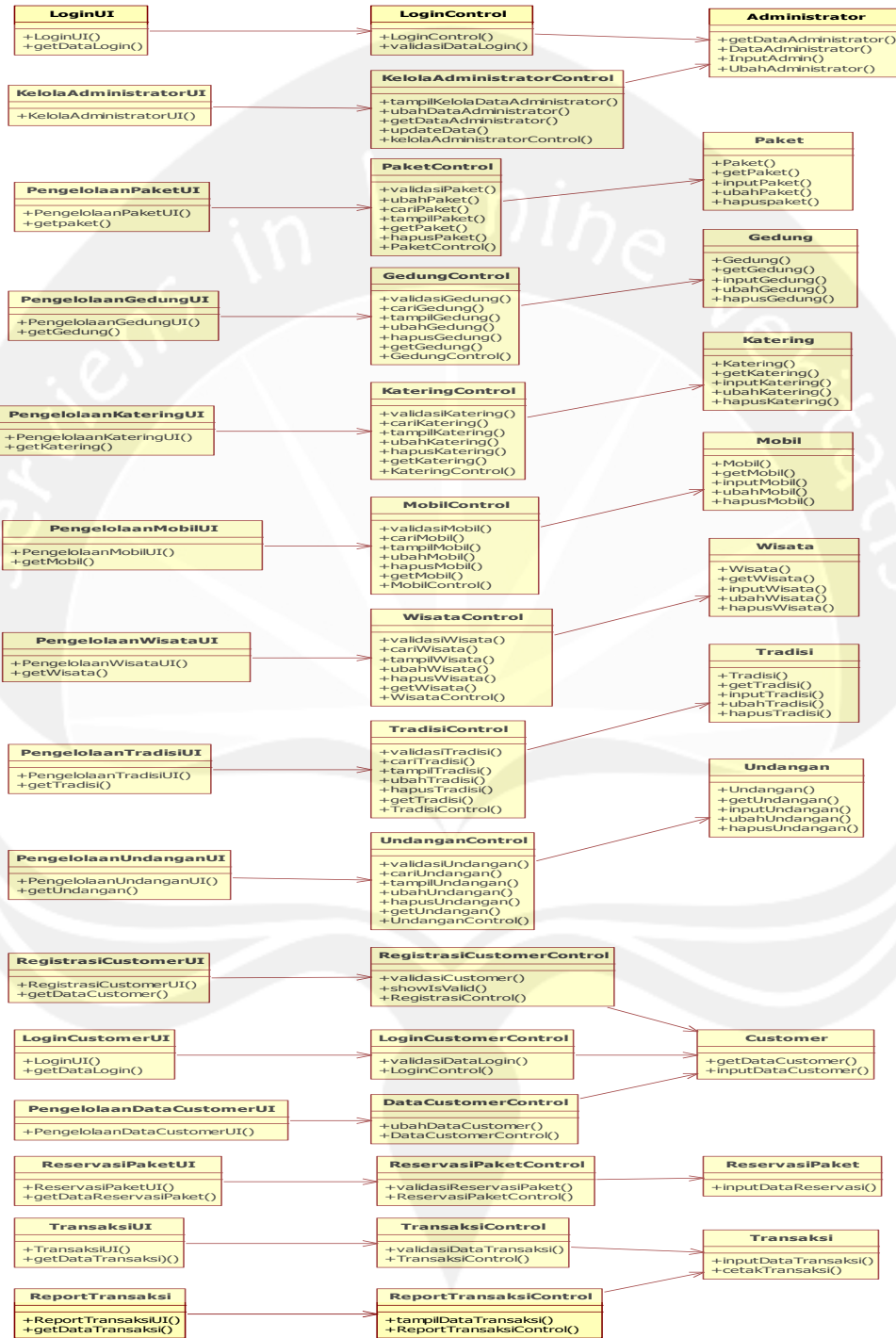
Gambar 2.30 Sequence Diagram : Cetak Laporan Transaksi

3.1 Rancangan Arsitektur Website



Gambar 3.1 Rancangan Arsitektur SIWO

3.2 Class Diagram



Gambar 3.2 Class Diagram

2.3.4. Spesifikasi Deskripsi Kelas Diagram

2.3.4.1. Spesifikasi Design Class Login (Desktop)

LoginUI	<<boundary>>
+LoginUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+getDataLogin() : void Operasi ini digunakan untuk mengambil data login yang diinputkan oleh user, yaitu nama dan password.	

2.3.4.2 Spesifikasi Design Class PengelolaanPaketUI

PengelolaanPaketUI	<<boundary>>
+PengelolaanPaketUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+tambahPaket() Operasi ini digunakan untuk mengisi data-data paket yang dimasukkan admin ke dalam <i>database</i> .	
+ubahpaket()	

Operasi ini digunakan untuk mengubah data-data paket pernikahan.

+savePaket()

Operasi ini digunakan untuk menyimpan data paket ke *database*.

+deletePaket()

Operasi ini digunakan untuk menghapus data paket dari *database*.

+filter()

Operasi ini digunakan untuk mencari dan menyaring data yang ingin dicari oleh administrator.

2.3.4.4. Spesifikasi Design Class PengelolaanPaket Control

PengelolaanPaketControl	<<control>>
+PengelolaanPaketControl() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	
+tambahPaket() Operasi ini digunakan untuk mengisi data-data paket yang dimasukkan admin ke dalam <i>database</i> .	
+getPaket()	

Operasi ini digunakan untuk mengambil data paket yang ada di dalam *database*.

+ubahPaket() : void

Operasi ini digunakan untuk mengubah data-data paket yang ada di dalam *database*.

+hapusPaket() : void

Operasi ini digunakan untuk menghapus data-data paket yang ada di dalam *database*.

+filter()

Operasi ini digunakan untuk mencari dan menyaring data yang ingin dicari oleh administrator.

2.3.4.5. Spesifikasi Design Class Paket

DataPaket	<<entity>>
<p>-idPaket : int</p> <p>Atribut ini digunakan untuk menyimpan sementara dari id paket yang bertipe integer</p> <p>-NamaPaket : varchar</p> <p>Atribut ini digunakan untuk menyimpan sementara dari nama paket yang bertipe varchar</p>	

DataPaket	<<entity>>
<p>-DekorasiPelaminan : text</p>	
<p>Atribut ini digunakan untuk menyimpan sementara dari dekorasi pelaminan yang bertipe text</p>	
<p>-DekorasiKamarPengantin : text</p>	
<p>Atribut ini digunakan untuk menyimpan sementara dari dekorasi kamar pengantin yang bertipe text</p>	
<p>-Fotografi : text</p>	
<p>Atribut ini digunakan untuk menyimpan sementara dari fotografi yang bertipe text</p>	
<p>-Videografi : text</p>	
<p>Atribut ini digunakan untuk menyimpan sementara dari videografi yang bertipe text</p>	
<p>-HrgPaket : int</p>	
<p>Atribut ini digunakan untuk menyimpan sementara dari harga paket yang bertipe integer</p>	
<p>-Hiburan : text</p>	
<p>Atribut ini digunakan untuk menyimpan sementara dari hiburan yang bertipe text</p>	
<p>-Lainnya : text</p>	
<p>Atribut ini digunakan untuk menyimpan sementara dari lainnya yang bertipe text</p>	

+DataPaket

+dataPaket()

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

+getPaket

Operasi ini digunakan untuk mengambil data paket yang ada di dalam *database*.

2.3.4.6. Spesifikasi Design Class PengelolaanGedungUI

PengeloalaanGedungUI	<<boundary>>
PengelolaanGedungUI()	
Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+tambahGedung() : void	
Operasi ini digunakan untuk mengisi data-data gedung yang dimasukkan admin ke dalam <i>database</i> .	
+getGedung()	
Operasi ini digunakan untuk mengambil data gedung yang ada di dalam <i>database</i> .	
+ubahGedung() : void	
Operasi ini digunakan untuk mengubah data-data gedung yang ada	

di dalam *database*.

+hapusGedung() : void

Operasi ini digunakan untuk menghapus data-data gedung yang ada di dalam *database*.

+filter()

Operasi ini digunakan untuk mencari dan menyaring data yang ingin dicari oleh administrator.

2.3.4.7 Spesifikasi Design Class PengelolaanGedung Control

PengelolaanGedungControl	<<control>>
<p>+PengelolaanGedungControl()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+tambahGedung()</p> <p>Operasi ini digunakan untuk mengisi data-data gedung yang dimasukkan admin ke dalam <i>database</i>.</p> <p>+getGedung()</p> <p>Operasi ini digunakan untuk mengambil data gedung yang ada di dalam <i>database</i>.</p> <p>+ubahGedung() : void</p> <p>Operasi ini digunakan untuk mengubah data-data gedung yang</p>	

ada di dalam *database*.

+hapusGedung() : void

Operasi ini digunakan untuk menghapus data-data gedung yang ada di dalam *database*.

+filter()

Operasi ini digunakan untuk mencari dan menyaring data yang ingin dicari oleh administrator.

2.3.4.8 Spesifikasi Design Class Gedung

DataGedung	<<entity>>
------------	------------

-idGedung : int

Atribut ini digunakan untuk menyimpan sementara dari id Gedung yang bertipe integer

-namaGedung : varchar

Atribut ini digunakan untuk menyimpan sementara dari nama gedung yang bertipe varchar

-lokasi : varchar

Atribut ini digunakan untuk menyimpan sementara dari lokasi yang bertipe varchar

-idPaket : int

Atribut ini digunakan untuk menyimpan sementara dari id paket yang bertipe integer

-gambar : varchar

Atribut ini digunakan untuk menyimpan sementara dari gambar yang bertipe varchar

-harga : int

Atribut ini digunakan untuk menyimpan sementara dari harga yang bertipe integer

+DataGedung

+DataGedung()

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

+getDataGedung() : DataGedung

Operasi ini digunakan untuk mengambil data Gedung yang ada di dalam database.

2.3.4.9 Spesifikasi Design Class PengelolaanData KateringUI

PengelolaanDataKateringUI	<<boundary>>

```

PengelolaanDataKateringUI ()

PengelolaanKateringUI ()

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

+tambahKatering() : void

Operasi ini digunakan untuk mengisi data-data katering yang
dimasukkan admin ke dalam database.

+getKatering()

Operasi ini digunakan untuk mengambil data katering yang ada
di dalam database.

+ubahKatering() : void

Operasi ini digunakan untuk mengubah data-data katering yang
ada di dalam database.

+hapusKatering() : void

Operasi ini digunakan untuk menghapus data-data katering
yang ada di dalam database.

+filter()

Operasi ini digunakan untuk mencari dan menyaring data yang
ingin dicari oleh administrator.

```

2.3.4.10 Spesifikasi Design Class PengelolaanKatering Control

PengelolaanDataKateringControl	<<control>>
---------------------------------------	--------------------------------

+PengelolaanKateringControl()

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

+tambahKatering()

Operasi ini digunakan untuk mengisi data-data katering yang dimasukkan admin ke dalam *database*.

+validasiKatering() : void

Operasi ini digunakan untuk memvalidasi data-data yang dari *form* PengelolaanKateringUI.

+getKatering()

Operasi ini digunakan untuk mengambil data katering yang ada di dalam *database*.

+ubahKatering() : void

Operasi ini digunakan untuk mengubah data-data katering yang ada di dalam *database*.

+hapusKatering() : void

Operasi ini digunakan untuk menghapus data-data katering yang ada di dalam *database*.

+filter()

Operasi ini digunakan untuk mencari dan menyaring data yang ingin dicari oleh administrator.

2.3.4.11 Spesifikasi Design Class Katering

DataKatering	<<entity>>
<p>-idKatering : int</p> <p>Atribut ini digunakan untuk menyimpan sementara dari id Katering yang bertipe integer</p> <p>-namaKatering : varchar</p> <p>Atribut ini digunakan untuk menyimpan sementara dari nama katering yang bertipe varchar</p> <p>-data : int</p> <p>Atribut ini digunakan untuk menyimpan sementara dari data yang bertipe integer</p> <p>-idPaket : int</p> <p>Atribut ini digunakan untuk menyimpan sementara dari id paket yang bertipe integer</p>	

-jenis : char

Atribut ini digunakan untuk menyimpan sementara dari jenis yang bertipe char

-harga : int

Atribut ini digunakan untuk menyimpan sementara dari harga yang bertipe integer

+DataKatering()

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

+getDataKatering() : DataKatering

Operasi ini digunakan untuk mengambil data katering yang ada di dalam *database*.

2.3.4.12 Spesifikasi Design Class PengelolaanWisataUI

PengeloalaanWisataUI	<<boundary>>
PengelolaanWisataUI()	
Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+tambahWisata ()	
Operasi ini digunakan untuk mengisi data-data wisata yang dimasukkan admin ke dalam <i>database</i> .	

+getWisata ()

Operasi ini digunakan untuk mengambil data wisata yang ada di dalam *database*.

+ubahWisata ()

Operasi ini digunakan untuk mengubah data-data wisata yang ada di dalam *database*.

+hapusWisata ()

Operasi ini digunakan untuk menghapus data-data wisata yang ada di dalam *database*.

+filter()

Operasi ini digunakan untuk mencari dan menyaring data yang ingin dicari oleh administrator.

2.3.4.13 Spesifikasi Design Class PengelolaanWisata Control

PengelolaanDataWisataControl

<<control>>

+PengelolaanDataWisataControl()

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

+tambahWisata ()

Operasi ini digunakan untuk mengisi data-data wisata yang dimasukkan admin ke dalam *database*.

```
+getWisata ()
```

Operasi ini digunakan untuk mengambil data wisata yang ada di dalam *database*.

```
+ubahWisata () : void
```

Operasi ini digunakan untuk mengubah data-data wisata yang ada di dalam *database*.

```
+hapusWisata () : void
```

Operasi ini digunakan untuk menghapus data-data wisata yang ada di dalam *database*.

```
+filter()
```

Operasi ini digunakan untuk mencari dan menyaring data yang ingin dicari oleh administrator.

2.3.4.14 Spesifikasi Design Class Wisata

DataTarifPengiriman	<<entity>>
<pre>-idWisata : int</pre> <p>Atribut ini digunakan untuk menyimpan sementara dari id Wisata yang bertipe integer</p> <pre>-namaWisata : varchar</pre> <p>Atribut ini digunakan untuk menyimpan sementara dari nama wisata yang bertipe varchar</p> <pre>-fotoWisata : varchar</pre>	

Atribut ini digunakan untuk menyimpan sementara dari foto wisata yang bertipe varchar

-ketWisata : text

Atribut ini digunakan untuk menyimpan sementara dari keterangan wisata yang bertipe text

-idPaket : int

Atribut ini digunakan untuk menyimpan sementara dari id paket yang bertipe integer

+DataWisata ()

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

+getDataWisata() : DataWisata

Operasi ini digunakan untuk mengambil data wisata yang ada di dalam *database*.

2.3.4.15 Spesifikasi Design Class PengelolaanMobilUI

PengelolaanMobilUI	<<boundary>>
+ PengelolaanDataMobilUI()	
Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+tambahMobil ()	

Operasi ini digunakan untuk mengisi data-data mobil yang dimasukkan admin ke dalam *database*.

```
+getMobil ()
```

Operasi ini digunakan untuk mengambil data mobil yang ada di dalam *database*.

```
+ubahMobil ()
```

Operasi ini digunakan untuk mengubah data-data mobil yang ada di dalam *database*.

```
+hapusMobil ()
```

Operasi ini digunakan untuk menghapus data-data mobil yang ada di dalam *database*.

```
+filter()
```

Operasi ini digunakan untuk mencari dan menyaring data yang ingin dicari oleh administrator.

2.3.4.16 Spesifikasi Design Class PengelolaanMobil Control

PengelolaanMobilControl	<<control>>
+ PengelolaanMobilControl ()	
Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	
+tambahMobil ()	

Operasi ini digunakan untuk mengisi data-data mobil yang dimasukkan admin ke dalam *database*.

+getMobil ()

Operasi ini digunakan untuk mengambil data mobil yang ada di dalam *database*.

+ubahMobil ()

Operasi ini digunakan untuk mengubah data-data mobil yang ada di dalam *database*.

+hapusMobil ()

Operasi ini digunakan untuk menghapus data-data mobil yang ada di dalam *database*.

+filter()

Operasi ini digunakan untuk mencari dan menyaring data yang ingin dicari oleh administrator.

2.3.4.17 Spesifikasi Design Class Mobil

DataMobil	<<entity>>
-idMobil : int	
Atribut ini digunakan untuk menyimpan sementara dari id tradisi yang bertipe integer	
-NamaMobil : varchar	
Atribut ini digunakan untuk menyimpan sementara dari nama	

<p>tradisi yang bertipe varchar</p> <p>-idPaket : int</p> <p>Atribut ini digunakan untuk menyimpan sementara dari id paket yang bertipe integer</p> <p>-gambar : varchar</p> <p>Atribut ini digunakan untuk menyimpan sementara dari gambar mobil yang bertipe varchar</p>
<p>+DataMobil()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+getDataMobil()</p> <p>Operasi ini digunakan untuk mengambil data mobil yang ada di dalam <i>database</i>.</p>

2.3.4.18 Spesifikasi Design Class PengelolaanTradisiUI

PengelolaanTradisiUI	<<boundary>>
<p>+ PengelolaanTradisiUI()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+tambahTradisi ()</p> <p>Operasi ini digunakan untuk mengisi data-data tradisi yang</p>	

dimasukkan admin ke dalam *database*.

+getTradisi ()

Operasi ini digunakan untuk mengambil data tradisi yang ada di dalam *database*.

+ubahTradisi ()

Operasi ini digunakan untuk mengubah data-data tradisi yang ada di dalam *database*.

+hapusTradisi ()

Operasi ini digunakan untuk menghapus data-data tradisi yang ada di dalam *database*.

+filter()

Operasi ini digunakan untuk mencari dan menyaring data yang ingin dicari oleh administrator.

2.3.4.19 Spesifikasi Design Class PengelolaanTradisi Control

PengelolaanDataTradisiControl	<<control>>
+ PengelolaanDataTradisiControl()	
Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	
+tambahTradisi ()	
Operasi ini digunakan untuk mengisi data-data tradisi yang	

dimasukkan admin ke dalam *database*.

```
+getTradisi ()
```

Operasi ini digunakan untuk mengambil data tradisi yang ada di dalam *database*.

```
+ubahTradisi () : void
```

Operasi ini digunakan untuk mengubah data-data tradisi yang ada di dalam *database*.

```
+hapusTradisi () : void
```

Operasi ini digunakan untuk menghapus data-data tradisi yang ada di dalam *database*.

```
+filter()
```

Operasi ini digunakan untuk mencari dan menyaring data yang ingin dicari oleh administrator.

2.3.4.20 Spesifikasi Design Class Tradisi

DataTradisi	<<entity>>
-idTradisi : int	
Atribut ini digunakan untuk menyimpan sementara dari id tradisi yang bertipe integer	
-NamaTradisi : varchar	
Atribut ini digunakan untuk menyimpan sementara dari nama tradisi yang bertipe varchar	

-KetTradisi : text

Atribut ini digunakan untuk menyimpan sementara dari keterangan tradisi yang bertipe text

+DataTradisi ()

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

+getDataTradisi ()

Operasi ini digunakan untuk mengambil data tradisi yang ada di dalam *database*.

2.3.4.21 Spesifikasi Design Class PengelolaanUndanganUI

PengelolaanUndanganUI

<<boundary>>

+ PengelolaanUndanganUI ()

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

+tambahUndangan ()

Operasi ini digunakan untuk mengisi data-data undangan yang dimasukkan admin ke dalam *database*.

+getUndangan ()

Operasi ini digunakan untuk mengambil data undangan yang ada di dalam *database*.

+ubahUndangan ()

Operasi ini digunakan untuk mengubah data-data undangan yang ada di dalam *database*.

+hapusUndangan ()

Operasi ini digunakan untuk menghapus data-data undangan yang ada di dalam *database*.

+filter()

Operasi ini digunakan untuk mencari dan menyaring data yang ingin dicari oleh administrator.

2.3.4.19 Spesifikasi Design Class PengelolaanUndangan Control

PengelolaanUndanganControl	<<control>>
<pre>+ PengelolaanUndanganControl() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +tambahUndangan () Operasi ini digunakan untuk mengisi data-data undangan yang dimasukkan admin ke dalam database. +getUndangan () Operasi ini digunakan untuk mengambil data undangan yang ada di dalam database. +ubahUndangan () : void</pre>	

Operasi ini digunakan untuk mengubah data-data undangan yang ada di dalam *database*.

+hapusUndangan () : void

Operasi ini digunakan untuk menghapus data-data undangan yang ada di dalam *database*.

+filter()

Operasi ini digunakan untuk mencari dan menyaring data yang ingin dicari oleh administrator.

2.3.4.20 Spesifikasi Design Class Undangan

DataUndangan	<<entity>>
<p>-idUndangan : int</p> <p>Atribut ini digunakan untuk menyimpan sementara dari id undangan yang bertipe integer</p> <p>-NamaUndangan : varchar</p> <p>Atribut ini digunakan untuk menyimpan sementara dari nama undangan yang bertipe varchar</p> <p>-Jenis : char</p> <p>Atribut ini digunakan untuk menyimpan sementara dari jenis yang bertipe char</p> <p>-Harga : int</p> <p>Atribut ini digunakan untuk menyimpan sementara dari harga</p>	

<p>yang bertipe integer</p> <p>-Gambar : varchar</p> <p>Atribut ini digunakan untuk menyimpan sementara dari gambar yang bertipe varchar</p>
<p>+DataUndangan ()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+getDataUndangan ()</p> <p>Operasi ini digunakan untuk mengambil data undangan yang ada di dalam <i>database</i>.</p>

2.3.4.24 Spesifikasi Design Class RegistrasiCustomerUI

RegistrasiCustomerUI	<<boundary>>
<p>+ RegistrasiCustomerUI ()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+getDataLogin() : void</p> <p>Operasi ini digunakan untuk mengambil data login yang diinputkan oleh customer, yaitu nama dan password.</p>	

2.3.4.25 Spesifikasi Design Class RegistrasiCustomer Control

RegistrasiCustomerControl	<<control>>
<pre>+ RegistrasiCustomerControl ()</pre> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <pre>+getCustomer ()</pre> <p>Operasi ini digunakan untuk mengambil data customer yang ada di dalam <i>database</i>.</p>	

2.3.4.26 Spesifikasi Design Class RegistrasiCustomer

RegistrasiCutomer	<<entity>>
<pre>-idCustomer : int</pre> <p>Atribut ini digunakan untuk menyimpan sementara dari id customer yang bertipe integer.</p> <pre>-NamaCustomer : varchar</pre> <p>Atribut ini digunakan untuk menyimpan sementara dari nama customer yang bertipe varchar.</p> <pre>-Password : varchar</pre> <p>Atribut ini digunakan untuk menyimpan sementara dari password yang bertipe varchar.</p> <pre>-Alamat : varchar</pre>	

Atribut ini digunakan untuk menyimpan sementara dari alamat yang bertipe varchar.

-Email : varchar

Atribut ini digunakan untuk menyimpan sementara dari email yang bertipe varchar.

-Telp : varchar

Atribut ini digunakan untuk menyimpan sementara dari telepon yang bertipe varchar

+DataCustomer ()

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

+getDataCustomer ()

Operasi ini digunakan untuk mengambil data customer yang ada di dalam *database*.

2.3.4.27 Spesifikasi Design Class PengelolaanData CustomerUI

PengelolaanDataCustomerUI	<<boundary>>
+ PengelolaanDataCustomerUI ()	
Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	
+ubahData () : void	

Operasi ini digunakan untuk mengubah data-data customer yang ada di dalam *database*.

2.3.4.28 Spesifikasi Design Class PengelolaanData CustomerControl

PengelolaanDataCustomerControl	<<control>>
+ PengelolaanDataCustomerControl() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	
+ubahData () : void Operasi ini digunakan untuk mengubah data-data customer yang ada di dalam <i>database</i> .	
+getData () : DataCustomer Operasi ini digunakan untuk mengambil data customer yang ada di dalam <i>database</i> .	

2.3.4.29 Spesifikasi Design Class PengelolaanData Customer

PengelolaanDataCustomer	<<entity>>
-idCustomer : int Atribut ini digunakan untuk menyimpan sementara dari id	

customer yang bertipe integer.

-NamaCustomer : varchar

Atribut ini digunakan untuk menyimpan sementara dari nama customer yang bertipe varchar.

-Password : varchar

Atribut ini digunakan untuk menyimpan sementara dari password yang bertipe varchar.

-Alamat : varchar

Atribut ini digunakan untuk menyimpan sementara dari alamat yang bertipe varchar.

-Email : varchar

Atribut ini digunakan untuk menyimpan sementara dari email yang bertipe varchar.

-Telp : varchar

Atribut ini digunakan untuk menyimpan sementara dari telepon yang bertipe varchar

+DataCustomer ()

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

+getDataCustomer ()

Operasi ini digunakan untuk mengambil data customer yang ada di dalam *database*.

2.3.4.30 Spesifikasi Design Class ReservasiPaketUI

ReservasiPaketUI	<<boundary>>
<pre>+ReservasiPaketUI() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +displayReservasiPaket() : void Operasi ini digunakan untuk menampilkan halaman reservasi yang ada dalam sistem ke dalam <i>form</i> ini.</pre>	

2.3.4.31 Spesifikasi Design Class Reservasi PaketControl

ReservasiPaketControl	<<control>>
<pre>+ReservasiPaketControl() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +detilReservasi() Operasi ini digunakan untuk menampilkan detil paket yang telah dipesan oleh customer.</pre>	

2.3.4.32 Spesifikasi Design Class ReservasiPaket

ReservasiPaket	<<entity>>
<p>-idTransaksi : int</p> <p>Atribut ini digunakan untuk menyimpan sementara dari id transaksi yang bertipe integer</p> <p>-tglPemesanan : date</p> <p>Atribut ini digunakan untuk menyimpan sementara dari tanggal pemesanan yang bertipe date</p> <p>-tglPelaksanaan : date</p> <p>Atribut ini digunakan untuk menyimpan sementara dari tanggal pelaksanaan yang bertipe date</p> <p>-jamPelaksanaan : time</p> <p>Atribut ini digunakan untuk menyimpan sementara dari jam pelaksanaan yang bertipe time</p> <p>-jamMulai : time</p> <p>Atribut ini digunakan untuk menyimpan sementara dari jam selesai yang bertipe time</p> <p>batasDp : date</p> <p>Atribut ini digunakan untuk menyimpan sementara dari batas DP yang bertipe date</p> <p>-idPaket : integer</p> <p>Atribut ini digunakan untuk menyimpan sementara dari id</p>	

paket yang bertipe integer

-katering : text

Atribut ini digunakan untuk menyimpan sementara dari katering yang bertipe text

-idObjekWisata : int

Atribut ini digunakan untuk menyimpan sementara dari id objek wisata yang bertipe integer

-idGedung : int

Atribut ini digunakan untuk menyimpan sementara dari id gedung yang bertipe integer

-idTradisi : int

Atribut ini digunakan untuk menyimpan sementara dari id tradisi yang bertipe int

-idMobil: int

Atribut ini digunakan untuk menyimpan sementara dari id mobil yang bertipe integer

-idCustomer : int

Atribut ini digunakan untuk menyimpan sementara dari idCustomer yang bertipe int

-catTambahhan : text

Atribut ini digunakan untuk menyimpan sementara dari catatan tambahan yang bertipe text

-totalBayar : int

Atribut ini digunakan untuk menyimpan sementara dari total bayar yang bertipe integer

-status : char

Atribut ini digunakan untuk menyimpan sementara dari status yang bertipe char

2.3.4.33 Spesifikasi Design Class TransaksiUI

TransaksiUI	<<boundary>>
+TransaksiUI() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	
+displayTransaksi() : void Operasi ini digunakan untuk menampilkan halaman transaksi yang ada dalam sistem ke dalam <i>form</i> ini.	

2.3.4.34 Spesifikasi Design Class TransaksiControl

TransaksiControl	<<control>>
+TransaksiControl() Default konstruktor, digunakan untuk inisialisasi semua	

atribut dari kelas ini.

```
+detilTransaksi ()
```

Operasi ini digunakan untuk menampilkan detil paket transaksi yang telah dipesan oleh customer.

2.3.4.35 Spesifikasi Design Class Transaksi

TransaksiPaket	<<entity>>
<pre>-idTransaksi : int</pre> <p>Atribut ini digunakan untuk menyimpan sementara dari id transaksi yang bertipe integer</p>	
<pre>-tglPemesanan : date</pre> <p>Atribut ini digunakan untuk menyimpan sementara dari tanggal pemesanan yang bertipe date</p>	
<pre>-tglPelaksanaan : date</pre> <p>Atribut ini digunakan untuk menyimpan sementara dari tanggal pelaksanaan yang bertipe date</p>	
<pre>-jamPelaksanaan : time</pre> <p>Atribut ini digunakan untuk menyimpan sementara dari jam pelaksanaan yang bertipe time</p>	
<pre>-jamMulai : time</pre> <p>Atribut ini digunakan untuk menyimpan sementara dari jam</p>	

selesai yang bertipe time

batasDp : date

Atribut ini digunakan untuk menyimpan sementara dari batas DP yang bertipe date

-idPaket : integer

Atribut ini digunakan untuk menyimpan sementara dari id paket yang bertipe integer

-katering : text

Atribut ini digunakan untuk menyimpan sementara dari katering yang bertipe text

-idObjekWisata : int

Atribut ini digunakan untuk menyimpan sementara dari id objek wisata yang bertipe integer

-idGedung : int

Atribut ini digunakan untuk menyimpan sementara dari id gedung yang bertipe integer

-idTradisi : int

Atribut ini digunakan untuk menyimpan sementara dari id tradisi yang bertipe int

-idMobil: int

Atribut ini digunakan untuk menyimpan sementara dari id mobil yang bertipe integer

-idCustomer : int

Atribut ini digunakan untuk menyimpan sementara dari idCustomer yang bertipe int

-catTambahhan : text

Atribut ini digunakan untuk menyimpan sementara dari catatan tambahan yang bertipe text

-totalBayar : int

Atribut ini digunakan untuk menyimpan sementara dari total bayar yang bertipe integer

-status : char

Atribut ini digunakan untuk menyimpan sementara dari status yang bertipe char

3. Perancangan Data

3.1. Dekomposisi Data

3.1.1. Deskripsi Entitas Admin

Nama	Tipe	Panjang	Keterangan
ID_ADMIN	Integer	-	ID administrator,

			Primary Key
NAMA_ADMIN	Varchar	50	Nama administrator

3.1.2. Deskripsi Entitas Paket

Nama	Tipe	Panjang	Keterangan
ID_PAKET	Integer	11	Id Paket, Primary key
NAMA_PAKET	Varchar	50	Nama Paket
DEKORASI_PELAMINAN	Text	-	Dekorasi Pelaminan
DEKORASI_KMR	Text	-	Dekorasi Kamar
FOTOGRAFI	Text	-	Fotografi
VIDEOGRAFI	Text	-	Videografi
HARGA_PAKET	Integer	11	Harga Paket
HIBURAN	Text	-	Hiburan
LAINNYA	Text	-	Lainnya

3.1.3. Deskripsi Entitas Gedung

Nama	Tipe	Panjang	Keterangan
ID_GEDUNG	Integer	11	ID GEDUNG, Primary Key
NAMA_GEDUNG	Varchar	50	Nama gedung
LOKASI	Varchar	128	Lokasi gedung
ID_PAKET	Integer	11	Id paket
GAMBAR	Varchar	128	Gambar gedung
HARGA	Integer	11	Harga gedung

3.1.4. Deskripsi Entitas Katering

Nama	Tipe	Panjang	Keterangan
ID_KATERING	Integer	11	ID katering
NAMA_KATERING	Varchar	50	Nama katering
DATA	Integer	11	Data katering
ID_PAKET	Integer	11	ID Paket
JENIS	Char	1	Jenis katering
HARGA	Integer	11	Harga katering

3.1.5. Deskripsi Entitas Data Wisata

Nama	Tipe	Panjang	Keterangan
ID_WISATA	Integer	-	ID Tarif, Primary Key
NAMA_LOKASI	Varchar	-	Nama Lokasi
FOTO_LOKASI	Varchar	90	Foto Lokasi
KETERANGAN	Text	-	Keterangan
ID_PAKET	Integer	20	ID paket

3.1.6. Deskripsi Entitas Data Tradisi

Nama	Tipe	Panjang	Keterangan
ID_TRADISI	Integer	11	ID Tradisi, Primary Key
NAMA_TRADISI	Varchar	50	Nama tradisi
KET_TRADISI	Text	-	Keterangan tradisi

3.1.7. Deskripsi Entitas Data Mobil

Nama	Tipe	Panjang	Keterangan
ID_MOBIL	Integer	11	ID_Mobil, primary key
NAMA_MOBIL	Varchar	128	Nama mobil
ID_PAKET	Integer	11	ID paket
GAMBAR	Varchar	128	Gambar mobil

3.1.8. Deskripsi Entitas Data Undangan

Nama	Tipe	Panjang	Keterangan
ID_UNDANGAN	Integer	11	ID_Undangan, primary key
NAMA	Varchar	128	Nama undangan
JENIS	Char	1	Jenis Undangan
HARGA	Integer	11	Harga Undangan
GAMBAR	Varchar	128	Gambar Undangan

3.1.9. Deskripsi Entitas Data Pondokan

Nama	Tipe	Panjang	Keterangan
ID_PONDOKAN	Integer	11	ID_Pondokan, primary key
ID_KATERING	Integer	11	ID katering
ID_TRANSAKSI	Integer	11	ID transaksi
JUMLAH	Integer	11	Jumlah porsi

3.1.10. Deskripsi Entitas Data Customer

Nama	Tipe	Panjang	Keterangan
ID_CUSTOMER	Integer	11	ID Customer, Primay Key
NAMA_CUSTOMER	Varchar	50	Nama Customer
PASSWORD	Varchar	128	Password
ALAMAT	Varchar	128	Alamat Customer
EMAIL	Varchar	50	Email Customer
NO_TELP	Varchar	20	No telp Customer

3.1.11. Deskripsi Entitas Data Transaksi

Nama	Tipe	Panjang	Keterangan
ID_TRANSAKSI	Integer	11	ID_Transaksi, primary key
TGL_PEMESANAN	Date	-	Tgl pemesanan
DP_AWAL	Date	-	Dp awal
ID_PAKET	Integer	11	ID paket
KATERING	Text	-	Katering
ID_WISATA	Integer	11	ID wisata
ID_GEDUNG	Integer	11	ID gedung
ID_TRADISI	Integer	11	ID tradisi
ID_MOBIL	Integer	11	ID mobil
ID_CUSTOMER	Integer	11	ID Customer
CAT_TAMBAHAN	Text	-	Cat tambahan
TOTAL_BAYAR	Integer	11	Total bayar

Gambar 3.3 Physical Data Model

4. Perancangan Antarmuka Desktop

4.1. Sketsa UI dan Deskripsinya

4.1.1. Antarmuka Halaman Login

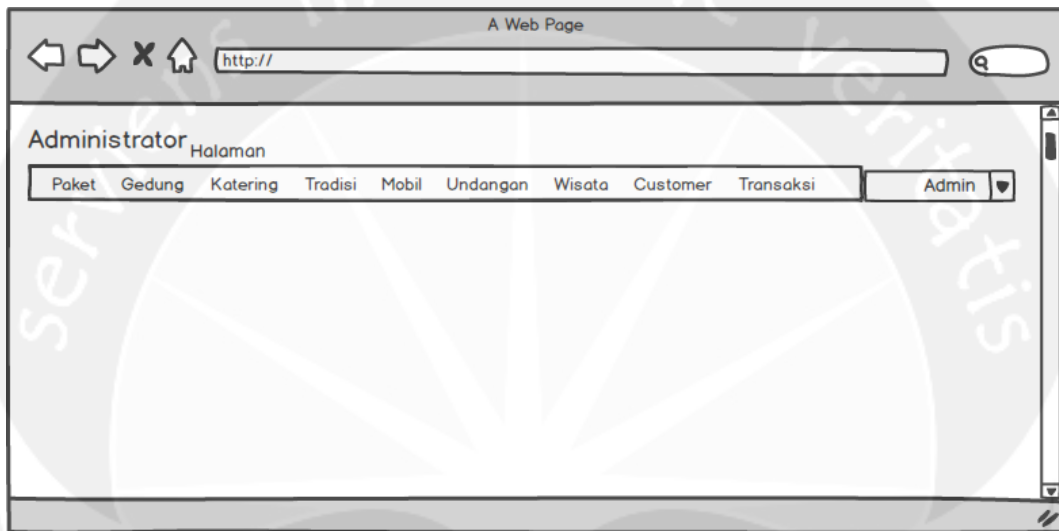


Gambar 4.1 Rancangan Antarmuka Login

Antarmuka ini digunakan administrator untuk melakukan proses login ke dalam sistem. Untuk mendapat akses masuk ke dalam sistem, admin harus menginputkan username dan password dengan benar pada textbox yang telah disediakan. Pada saat tombol sign in diklik, sistem akan mengecek username dan password yang diinputkan dengan username dan password yang telah tersimpan di *database*. Jika username

dan password benar maka admin akan masuk ke dalam sistem mendapatkan hak akses dan dapat melakukan pengelolaan data, sebaliknya jika username dan password salah atau tidak cocok maka akan diberikan pesan peringatan.

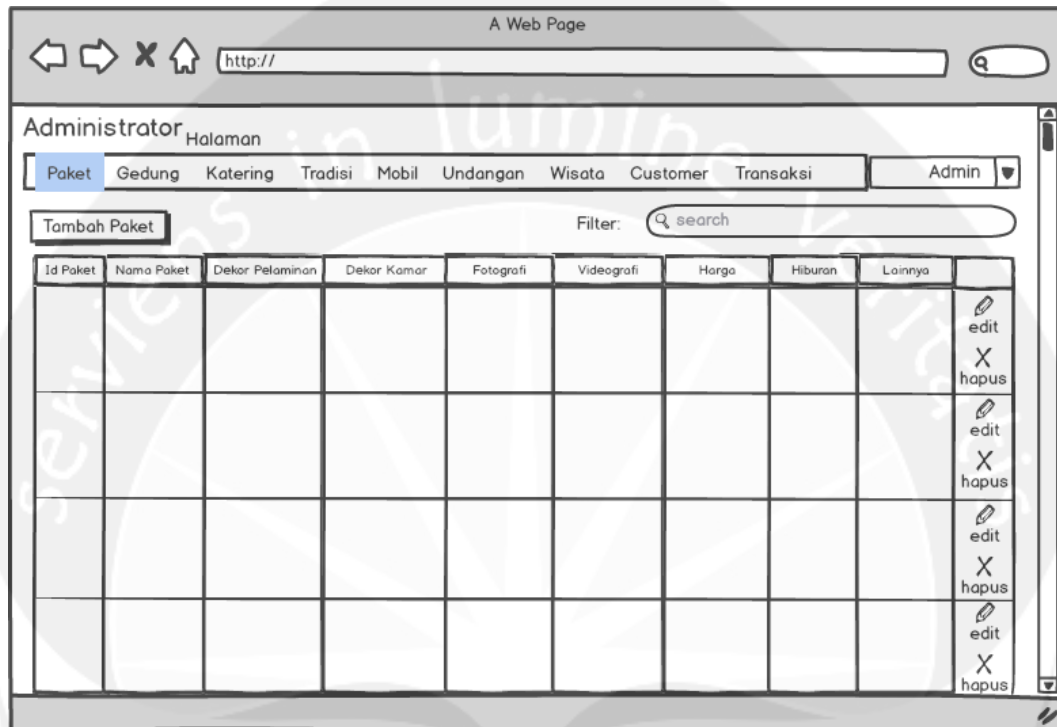
4.1.2 Antarmuka Halaman Administrator



Gambar 4.2 Rancangan Antarmuka Halaman Administrator

Antarmuka ini digunakan oleh admin untuk mengelola semua data yang ada di dalam *Wedding Organizer*.

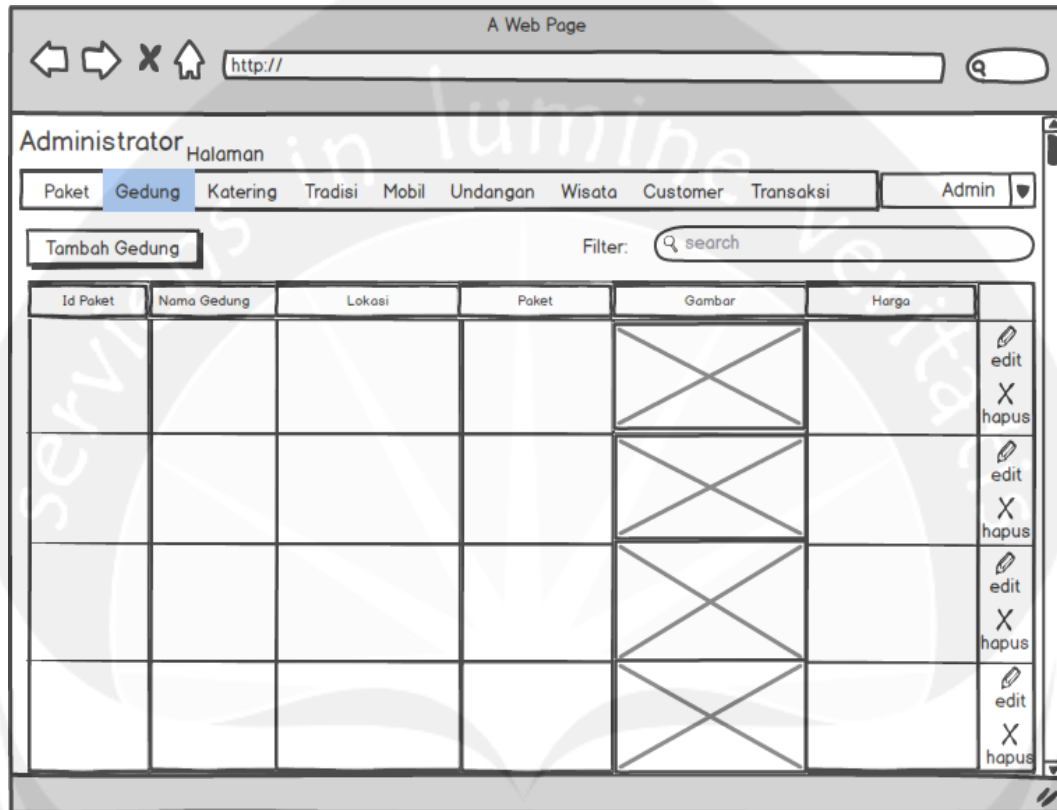
4.1.3 Antarmuka Pengelolaan Paket



Gambar 4.3 Rancangan Antarmuka Pengelolaan Paket

Antarmuka ini digunakan untuk melakukan pengelolaan data paket, termasuk di dalamnya tambah paket baru, ubah paket, dan hapus paket. Pada *form* ini terdapat id paket, nama paket, dekorasi pelaminan, dekorasi kamar pengantin, fotografi, videografi, harga paket, hiburan dan lainnya yang akan diinputkan oleh Admin. Terdapat 4 buah *Button* yaitu tambah paket yang digunakan untuk menambahkan paket yang baru ke dalam *database*. *Button* ubah digunakan untuk mengubah data. *Button* hapus untuk menghapus data. *Button* simpan digunakan untuk menyimpan data baru yang telah diinputkan. Terdapat *filter searchBox* yang digunakan untuk mencari data yang diinginkan.

4.1.4 Antarmuka Pengelolaan Gedung

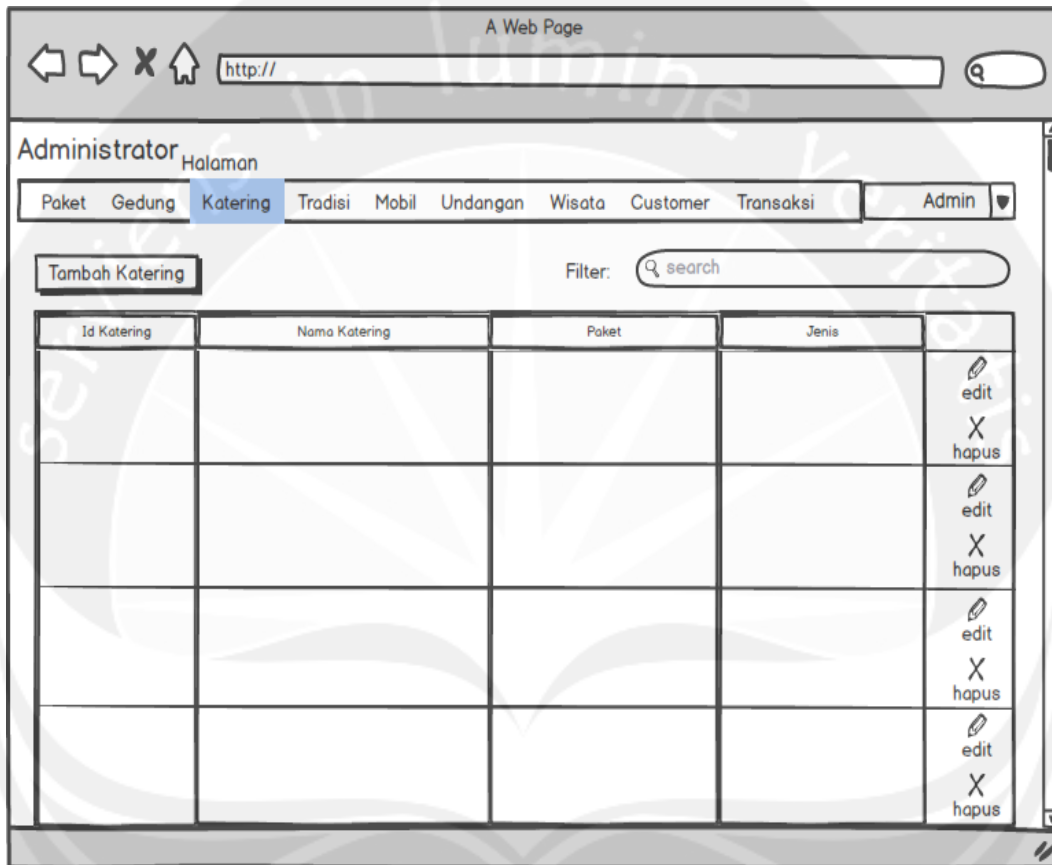


Gambar 4.4 Rancangan Antarmuka Pengelolaan Gedung

Antarmuka ini digunakan untuk melakukan pengelolaan data gedung, termasuk di dalamnya tambah gedung baru, ubah gedung, dan hapus gedung. Pada *form* ini terdapat id gedung, nama gedung, lokasi, paket, gambar dan harga. Kemudian, terdapat 4 buah *Button* yaitu *Button* tambah gedung yang digunakan untuk menambahkan gedung yang baru ke dalam *database*. *Button* ubah digunakan untuk mengubah. *Button* hapus untuk menghapus data. *Button* simpan digunakan untuk menyimpan data baru yang telah

diinputkan. Terdapat *filter searchBox* yang digunakan untuk mencari data yang diinginkan.

4.1.5 Antarmuka Pengelolaan Katering

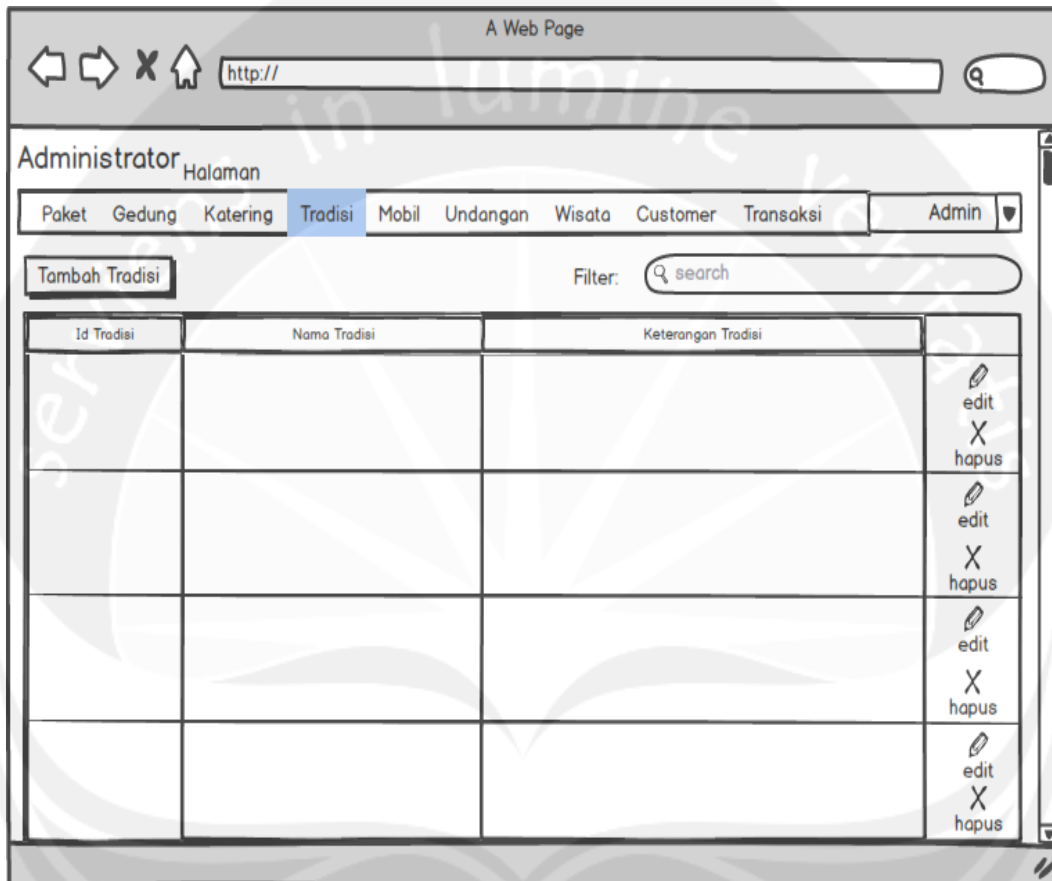


Gambar 4.5 Rancangan Antarmuka Pengelolaan Katering

Antarmuka ini digunakan untuk melakukan pengelolaan data katering, termasuk di dalamnya tambah katering baru, ubah katering, dan hapus katering. Pada *form* ini terdapat id katering, nama katering, paket dan jenis. Kemudian, terdapat 4 buah *Button* yaitu *Button* tambah katering yang digunakan untuk menambahkan katering yang baru. *Button* ubah digunakan untuk mengubah data. *Button* hapus untuk menghapus data. *Button*

simpan digunakan untuk menyimpan data baru. Terdapat *filter* *searchBox* yang digunakan untuk mencari data yang diinginkan.

4.1.6 Antarmuka Pengelolaan Tradisi

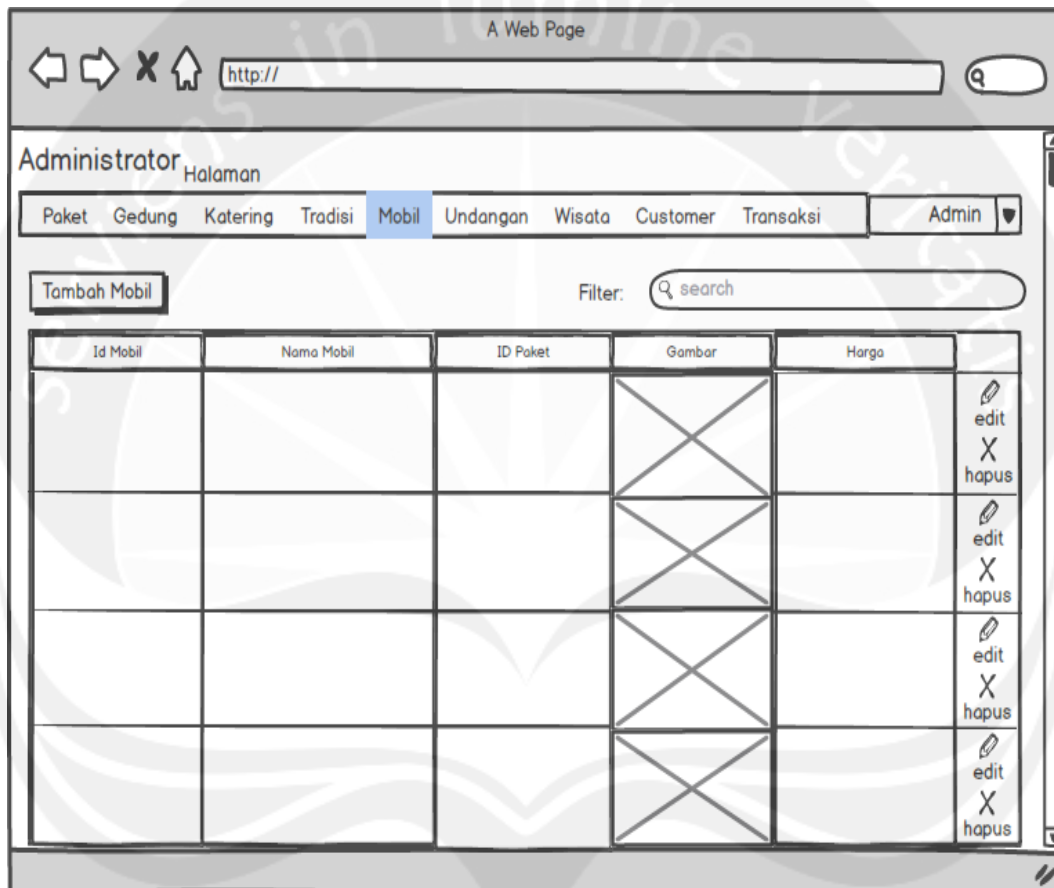


Gambar 4.6 Rancangan Antarmuka Pengelolaan Tradisi

Antarmuka ini digunakan untuk melakukan pengelolaan data tradisi, termasuk di dalamnya tambah tradisi baru, ubah tradisi, dan hapus tradisi. Pada *form* ini terdapat id tradisi, nama tradisi, dan keterangan tradisi. Kemudian, terdapat 4 buah *Button* yaitu *Button* tambah tradisi yang digunakan untuk menambahkan tradisi yang baru ke dalam *database*. *Button* ubah digunakan untuk mengubah data. *Button* hapus untuk menghapus

data. *Button* simpan digunakan untuk menyimpan data baru yang telah diinputkan oleh Admin. Terdapat *filter searchBox* yang digunakan untuk mencari data yang diinginkan.

4.1.7 Antarmuka Pengelolaan Mobil

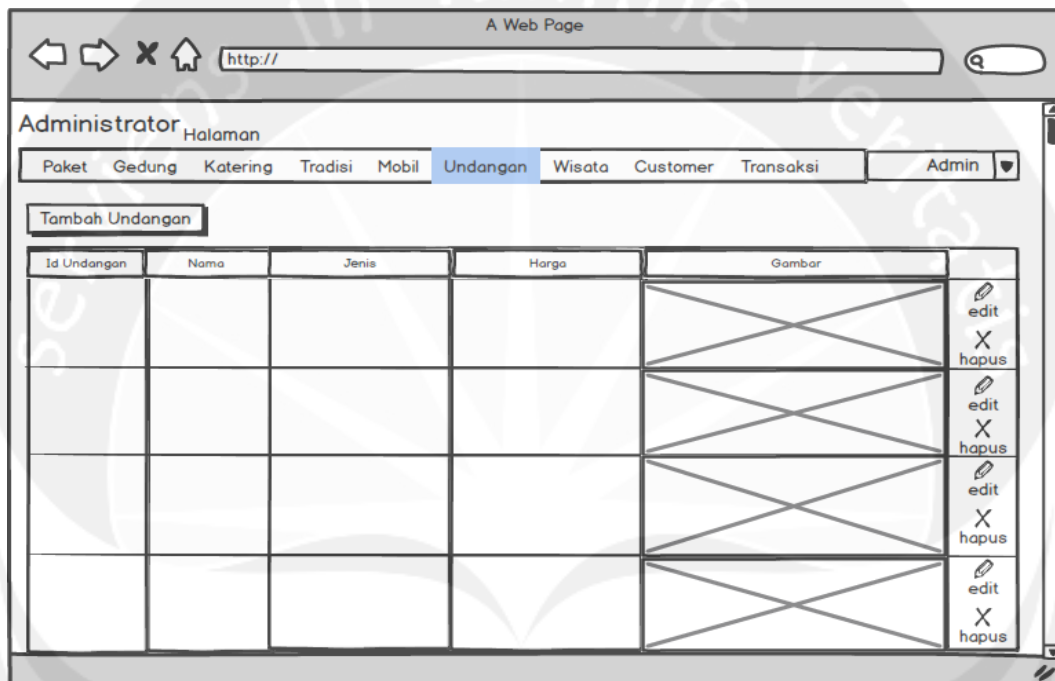


Gambar 4.7 Rancangan Antarmuka Pengelolaan Mobil

Antarmuka ini digunakan untuk melakukan pengelolaan data mobil, termasuk di dalamnya tambah mobil baru, ubah mobil, dan hapus mobil. Pada *form* ini terdapat id mobil, nama mobil, id paket dan gambar mobil. Kemudian, terdapat 4 buah *Button* yaitu *Button* tambah mobil yang digunakan untuk menambahkan mobil yang baru ke dalam *database*. *Button* ubah digunakan untuk

mengubah data yang diinginkan. *Button* hapus untuk menghapus data yang sudah diinputkan. *Button* simpan digunakan untuk menyimpan data baru. Terdapat *filter searchBox* yang digunakan untuk mencari data yang diinginkan.

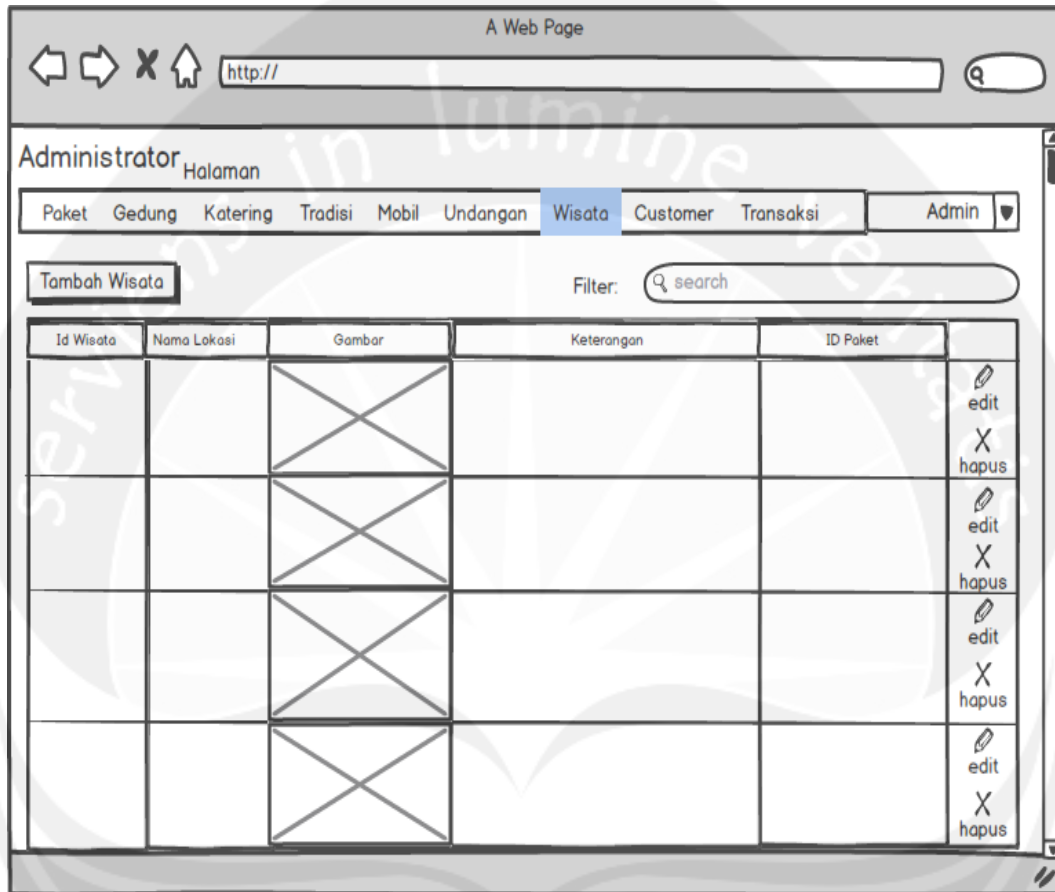
4.1.8 Antarmuka Pengelolaan Undangan



Gambar 4.8 Rancangan Antarmuka Pengelolaan Mobil

Antarmuka ini digunakan untuk melakukan pengelolaan data undangan, termasuk di dalamnya tambah undangan baru, ubah undangan, dan hapus undangan. Pada *form* ini terdapat id undangan, nama undangan, jenis, harga dan gambar undangan. Kemudian, terdapat 4 buah *Button* yaitu *Button* tambah undangan yang digunakan untuk menambahkan undangan yang baru ke dalam *database*. *Button* ubah digunakan untuk mengubah data yang diinginkan. *Button* hapus untuk menghapus data yang sudah diinputkan. *Button* simpan digunakan untuk menyimpan data baru.

4.1.9 Antarmuka Pengelolaan Wisata

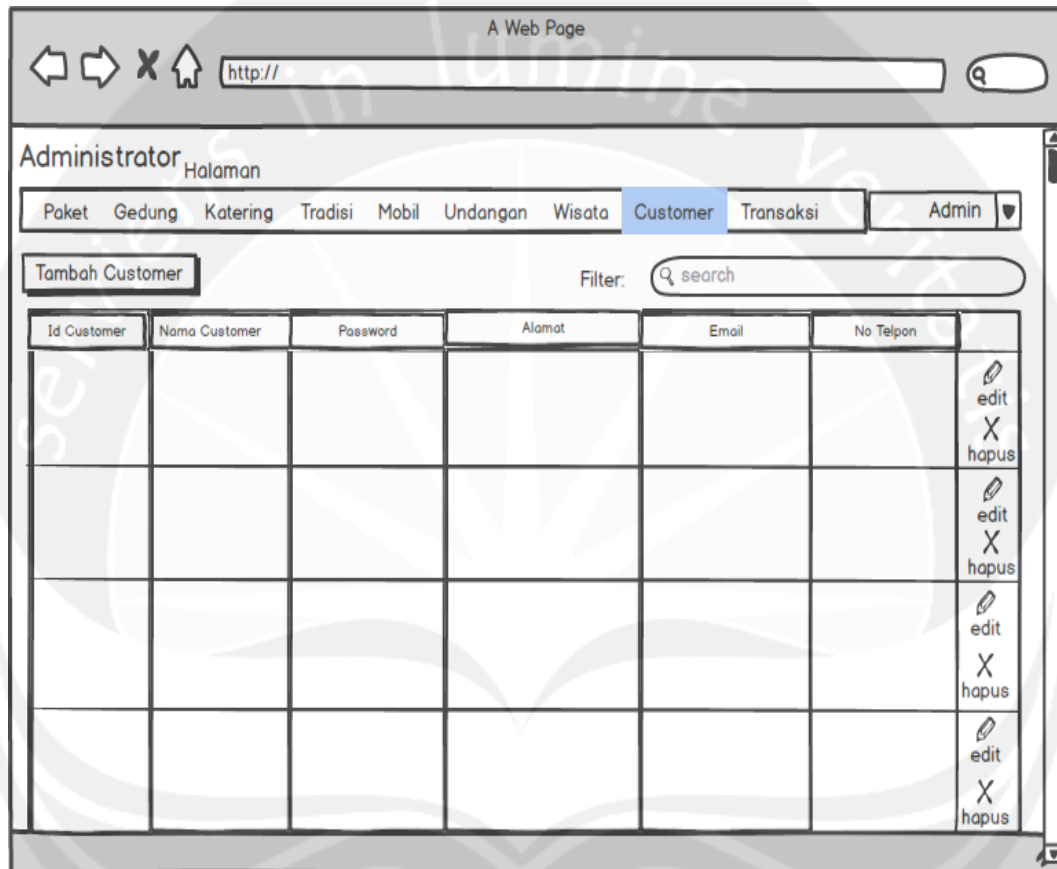


Gambar 4.9 Rancangan Antarmuka Pengelolaan Wisata

Antarmuka ini digunakan untuk melakukan pengelolaan data Wisata, termasuk di dalamnya tambah Wisata baru, ubah Wisata, dan hapus Wisata. Pada *form* ini terdapat id Wisata, nama lokasi Wisata, gambar, keterangan Wisata, dan id paket. Kemudian, terdapat 4 buah *Button* yaitu *Button* tambah Wisata yang digunakan untuk menambahkan Wisata yang baru ke dalam *database*. *Button* ubah digunakan untuk mengubah data yang diinginkan. *Button* hapus untuk menghapus data. *Button* simpan

digunakan untuk menyimpan data baru. Terdapat *filter searchBox* yang digunakan untuk mencari data yang diinginkan.

4.1.9 Antarmuka Pengelolaan Data Customer

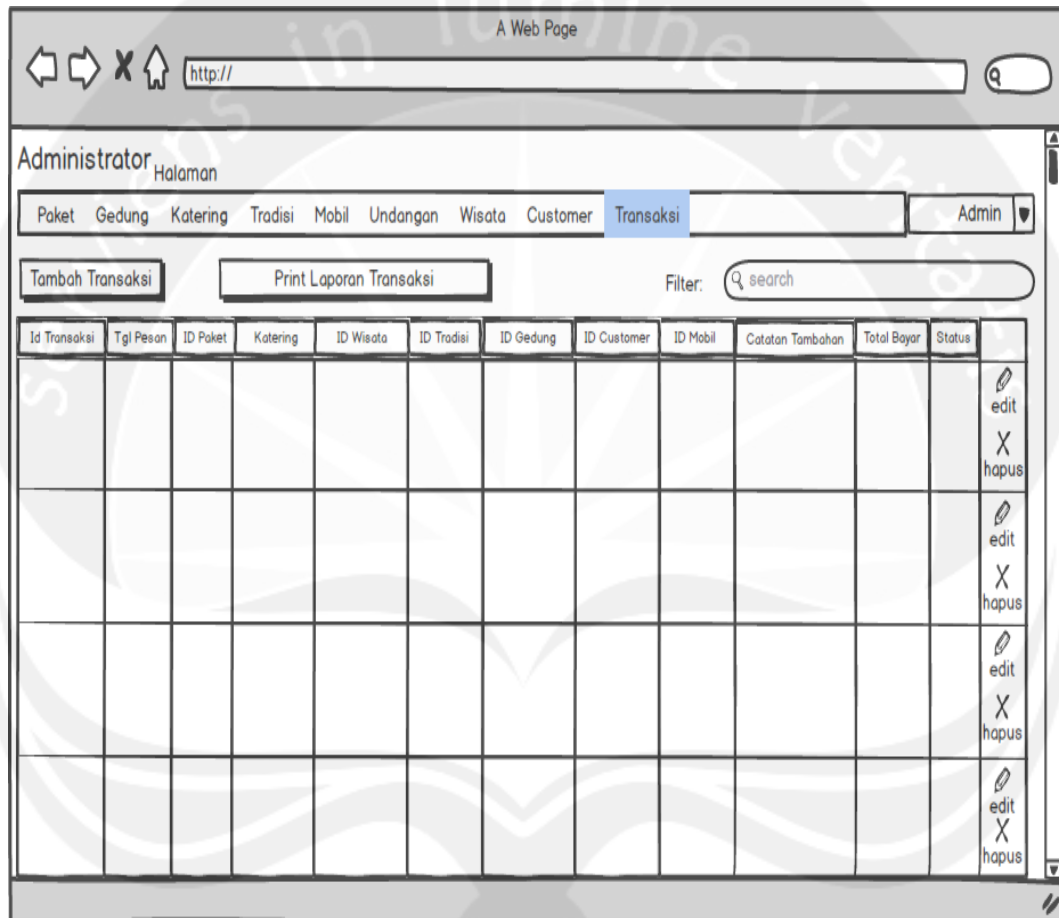


Gambar 4.10 Rancangan Antarmuka Pengelolaan Data Customer

Antarmuka ini digunakan untuk melakukan pengelolaan data customer, termasuk di dalamnya tambah customer baru, ubah customer, dan hapus customer. Pada *form* ini terdapat id customer, nama customer, password, alamat, email, dan telpon. Kemudian, terdapat 4 buah *Button* yaitu *Button* tambah customer yang digunakan untuk menambahkan customer yang baru. *Button* ubah digunakan untuk mengubah data. *Button* hapus untuk

menghapus data. *Button* simpan digunakan untuk menyimpan data baru. Terdapat *filter searchBox* yang digunakan untuk mencari data yang diinginkan.

4.1.10 Antarmuka Pengelolaan Data Transaksi



Gambar 4.11 Rancangan Antarmuka Pengelolaan Data Transaksi

Antarmuka ini digunakan untuk melakukan pengelolaan data transaksi, termasuk di dalamnya tambah transaksi baru, ubah transaksi, dan hapus transaksi. Pada *form* ini terdapat id transaksi, tanggal pesan, id paket, katering, id Wisata, id tradisi, id gedung, id customer, id mobil, catatan tambahan, total bayar, dan status. Kemudian, terdapat 4 buah *Button*

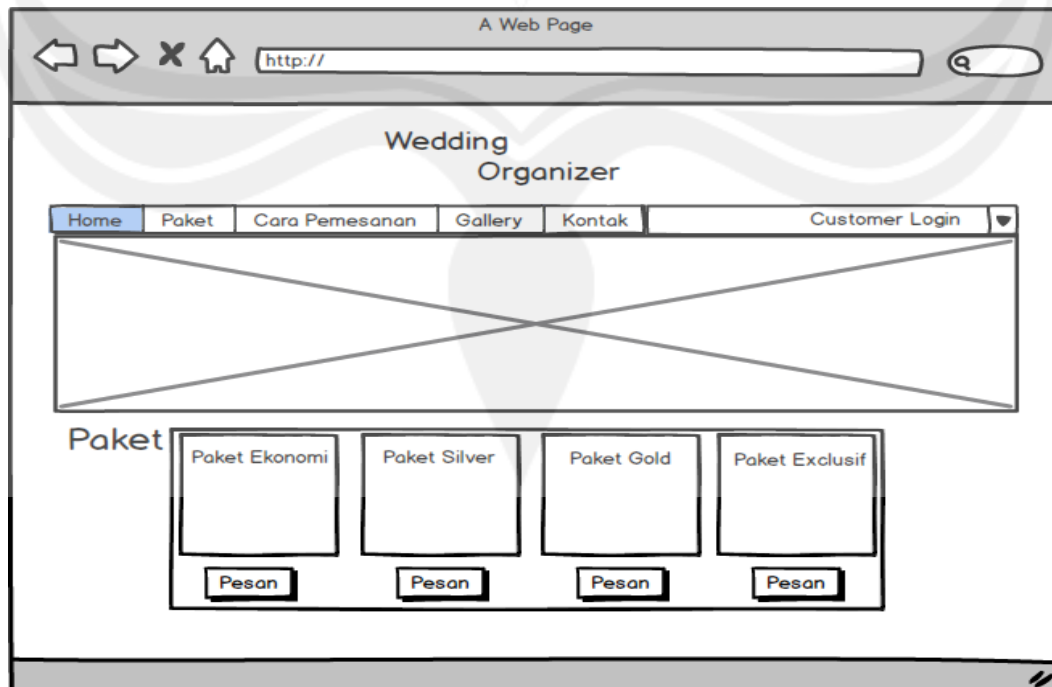
yaitu *Button* tambah transaksi yang digunakan untuk menambahkan transaksi yang baru. *Button* ubah digunakan untuk mengubah data. *Button* hapus untuk menghapus data. *Button* simpan digunakan untuk menyimpan data baru. Terdapat *filter searchBox* yang digunakan untuk mencari data yang diinginkan. *Button* print laporan transaksi untuk mencetak laporan transaksi customer.

4.1.11 Antarmuka Laporan Transaksi

Laporan Transaksi									
Kode	Customer	Paket	Pemesanan	Batas DP	Pelaksanaan	Harga Paket	Tambahan	Total Bayar	Status
E001022014	Sudarman Handoko	Ekonomi	20-01-2014	27-01-2014	18-02-2014	Rp 30.000.000	Rp 3.150.000	Rp 33.150.000	Lunas

Gambar 4.12 Rancangan Antarmuka Laporan Transaksi

4.1.12 Antarmuka Home Page Web



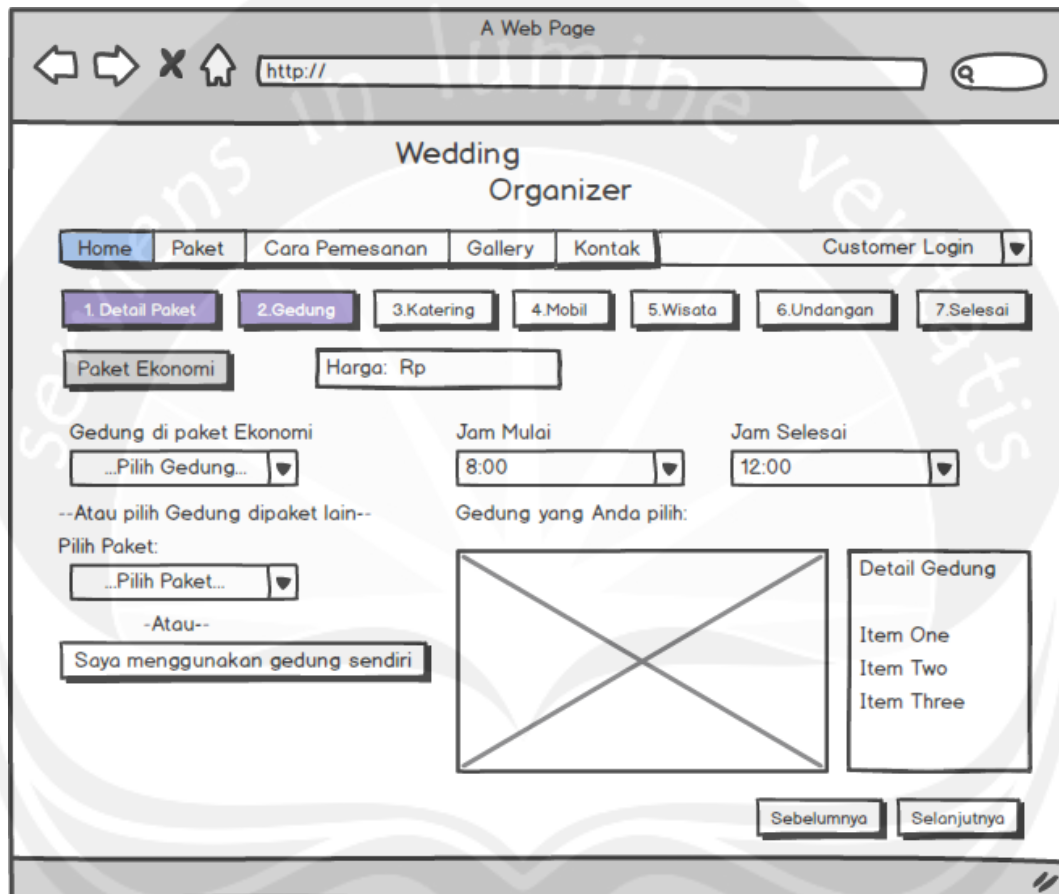
Gambar 4.13 Rancangan Antarmuka Home Page Web
Antarmuka ini digunakan untuk menampilkan halaman utama
dari sistem informasi ini.

4.1.13 Antarmuka Paket Web



Gambar 4.14 Rancangan Antarmuka Paket Web
Antarmuka ini digunakan untuk menampilkan halaman paket.
Berisi *combobox* yang meminta customer memilih tradisi yang
diinginkan. *Date chooser* yang digunakan untuk memilih tanggal
pernikahan. *Button* "sebelumnya" untuk kembali pada home page
awal. *Button* "selanjutnya" untuk melanjutkan ke langkah
selanjutnya, yaitu ke halaman gedung.

4.1.14 Antarmuka Gedung Web

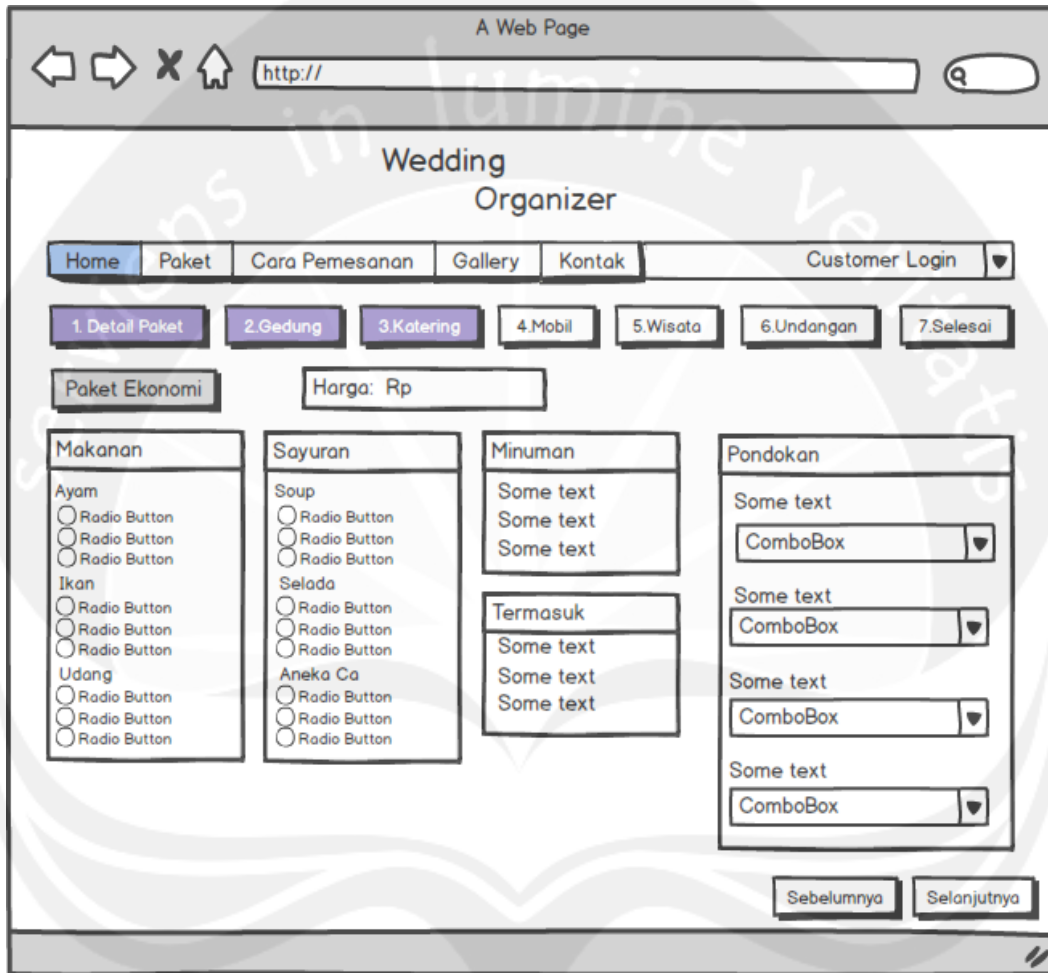


Gambar 4.15 Rancangan Antarmuka Gedung Web

Antarmuka ini digunakan untuk menampilkan halaman gedung. Berisi *combobox* yang meminta customer memilih gedung yang diinginkan pada paket awal atau bisa juga memilih gedung pada paket lain dan juga *comboBox* untuk jam pemakaian gedung. *Button* "saya menggunakan gedung sendiri" jika customer tidak ingin menggunakan gedung. *List* digunakan untuk menampilkan detail nama dan alamat gedung. *Button* "sebelumnya" kembali

pada halaman detail paket. *Button* "selanjutnya" untuk melanjutkan ke langkah selanjutnya, yaitu ke halaman katering.

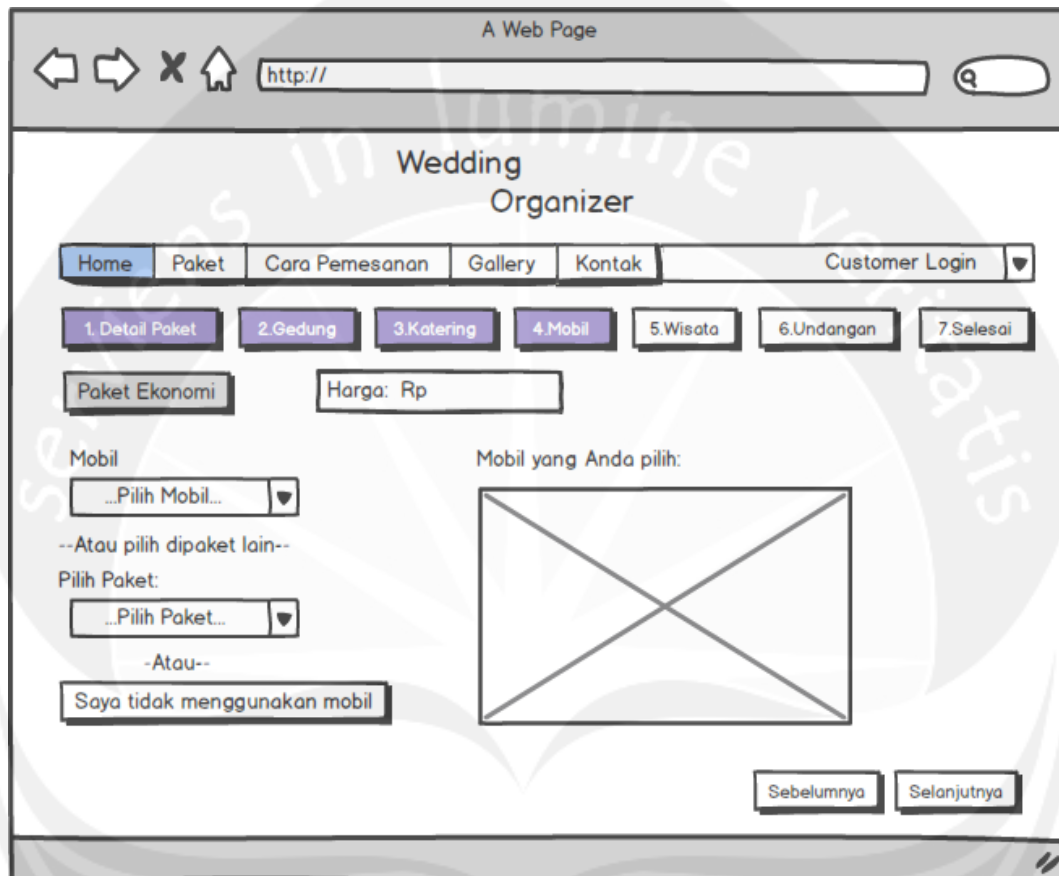
4.1.15 Antarmuka Katering Web



Gambar 4.16 Rancangan Antarmuka Katering Web

Antarmuka ini digunakan untuk menampilkan halaman katering. Berisi *radioButton* yang meminta customer untuk mengisi tiap menu makanan yang tersedia dalam pilihan. *ComboBox* pada menu pondokan jika customer ingin menambah porsi dari tiap menu di pondokan. *Button* "sebelumnya" kembali pada halaman gedung. *Button* "selanjutnya" untuk melanjutkan ke langkah selanjutnya, yaitu ke halaman mobil.

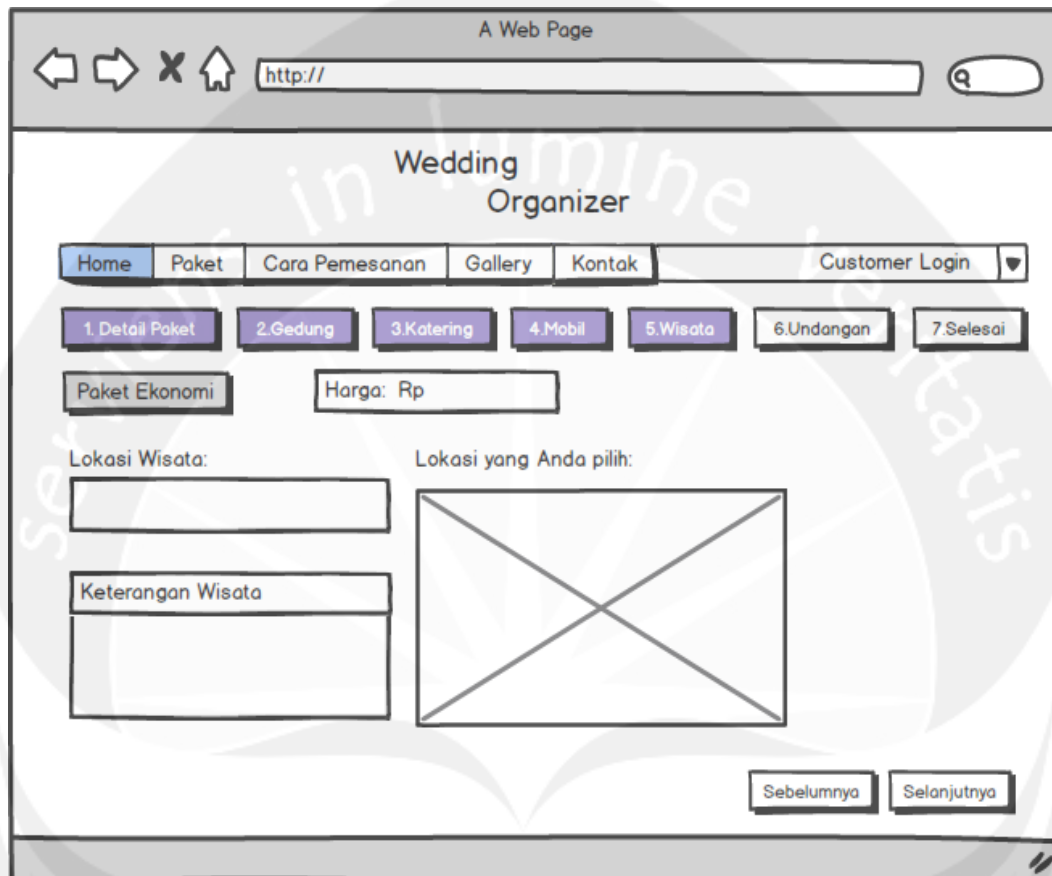
4.1.16 Antarmuka Mobil Web



Gambar 4.17 Rancangan Antarmuka Mobil Web

Antarmuka ini digunakan untuk menampilkan halaman mobil. Berisi *comboBox* yang meminta customer untuk memilih mobil yang tersedia dalam pilihan atau bisa juga memilih mobil yang ada pada paket lain. *Button* jika customer tidak ingin menggunakan mobil dari paket yang ada. *Button* "sebelumnya" kembali pada halaman katering. *Button* "selanjutnya" untuk melanjutkan ke langkah selanjutnya, yaitu ke halaman wisata.

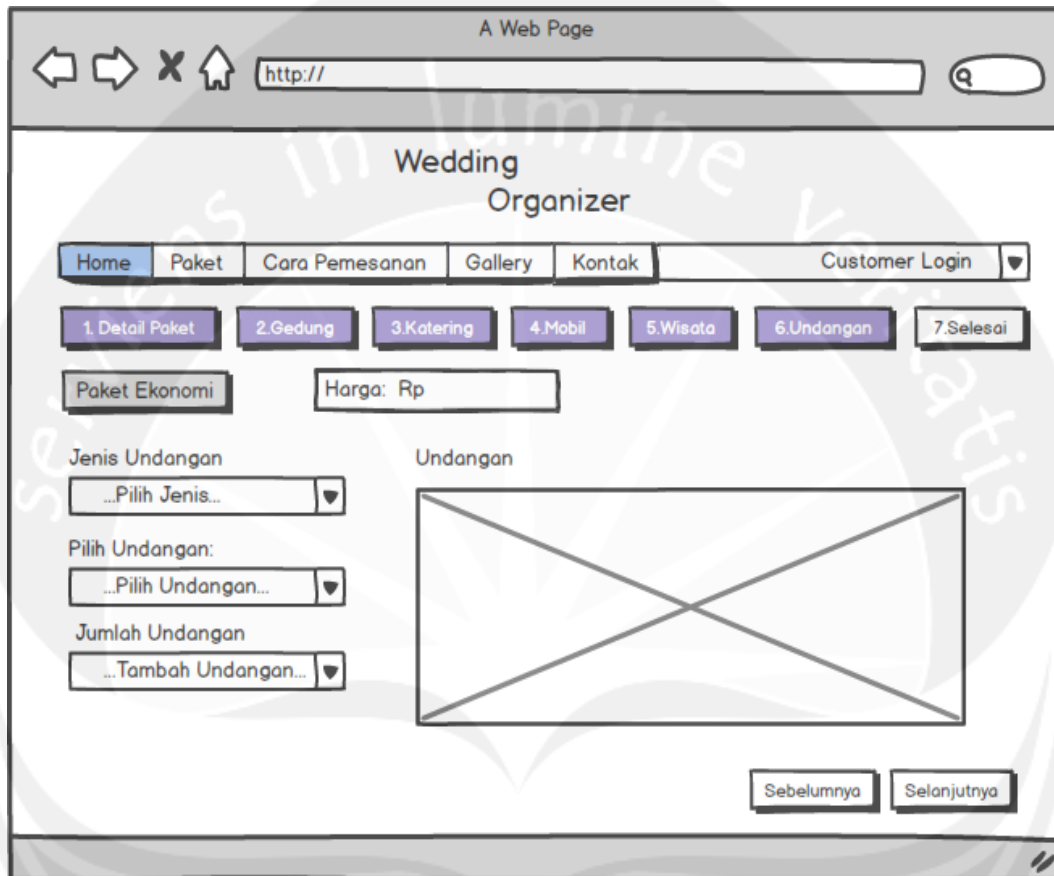
4.1.17 Antarmuka Wisata Web



Gambar 4.18 Rancangan Antarmuka Wisata Web

Antarmuka ini digunakan untuk menampilkan halaman wisata. Berisi keterangan wisata yang merupakan bonus dalam paket yang telah dipilih. *Button* "sebelumnya" kembali pada halaman mobil. *Button* "selanjutnya" untuk melanjutkan ke langkah selanjutnya, yaitu ke halaman Undangan.

4.1.18 Antarmuka Undangan Web



Gambar 4.19 Rancangan Antarmuka Undangan Web

Antarmuka ini digunakan untuk menampilkan halaman Undangan. Berisi *comboBox* yang meminta customer untuk memilih jenis undangan, pilih undangan, dan jumlah undangan yang tersedia dalam pilihan. *Button* "sebelumnya" kembali pada halaman mobil manten. *Button* "selanjutnya" untuk melanjutkan ke langkah selanjutnya, yaitu ke halaman Selesai.

4.1.19 Antarmuka Registrasi Customer Web

A Web Page

http://

Wedding Organizer

Home Paket Gallery Kontak Cara Pemesanan Customer Login

1. Detail Paket 2. Gedung 3. Katering 4. Mobil 5. Wisata 6. Undangan 7. Selesai

Login member

Email Address

Password

Login

Register

Sebelumnya Selanjutnya

A Web Page

http://

Wedding Organizer

Home Paket Cara Pemesanan Gallery Kontak Customer Login

1. Detail Paket 2. Gedung 3. Katering 4. Mobil 5. Wisata 6. Undangan 7. Selesai

Daftar member baru

Nama

Alamat

Telepon

Email

Password

Daftar

Login member

Email Address

Password

Login

Sebelumnya Selanjutnya

Gambar 4.20 Rancangan Antarmuka Registrasi Customer Web

Antarmuka ini digunakan customer untuk registrasi. Berisi *textfield* yang meminta customer untuk mengisi nama, password, alamat, telepon, dan email. *Button* "daftar" untuk memasukan data customer ke dalam *database* agar terdaftar sebagai member. *Textfield* email dan password untuk memasukkan alamat email dan password customer. *Button* "login" untuk masuk ke dalam member area. *Button* "sebelumnya" kembali pada halaman Undangan.

4.1.20 Antarmuka Reservasi Paket Web

A Web Page

Wedding Organizer

Home Paket Cara Pemesanan Gallery Kontak Customer Login

1. Detail Paket 2. Gedung 3. Katering 4. Mobil 5. Wisata 6. Undangan 7. Selesai

Detil Transaksi

Some text

Some text

Some text

Some text

Some text

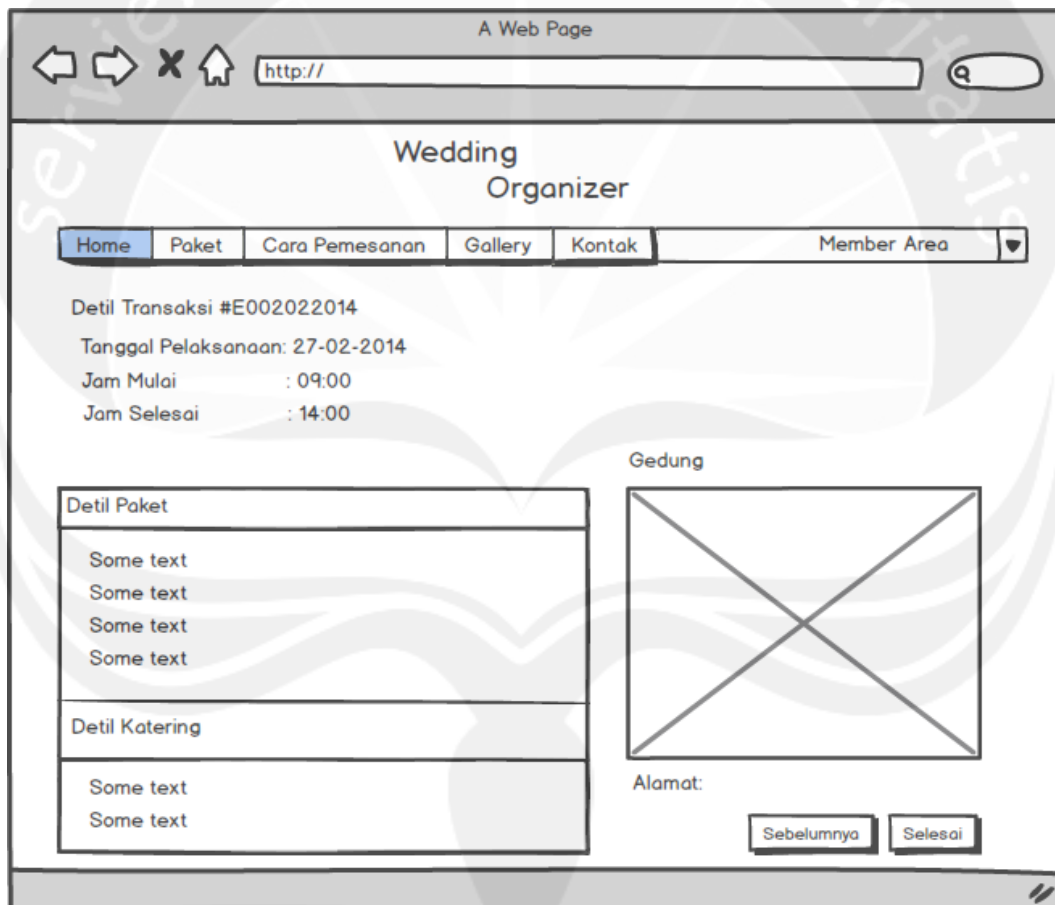
Catatan yang ingin ditambahkan:

Sebelumnya Selesai

Gambar 4.20 Rancangan Antarmuka Reservasi Paket Web

Antarmuka ini digunakan untuk menampilkan detail transaksi. *dialogBox* digunakan apabila customer ingin menambahkan catatan tambahan dan juga terdapat link untuk mengakses halaman lainnya. *Button* "sebelumnya" kembali pada halaman catering. *Button* "selesai" untuk mengakhiri proses pemesanan dan melangkah ke tahap transaksi.

4.1.21 Antarmuka Transaksi Web



Gambar 4.22 Rancangan Antarmuka Transaksi Web

Antarmuka ini digunakan untuk menampilkan detail informasi transaksi dan juga terdapat link untuk mengakses halaman lainnya.

