

**DPPL-04**

## **DESKRIPSI PERANCANGAN PERANGKAT LUNAK**

### **Website Rakit-Komputer**

untuk:

Rakit Komputer

Dipersiapkan oleh:


Kelompok 4

- Arva Adwitya S
- Bramantya Purbaya
- Fariz Muhammad R
- Sya Raihan Heggi

Program Studi Informatika

Fakultas Informatika

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

	<b>Prodi S1- Informatika Universitas Telkom</b>	<b>Nomor Dokumen</b>		<b>Halaman</b>
		<b><i>DPPL-04</i></b>		<i>1/&lt;jml #</i>
		<b>Revisi</b>	<i>&lt;nomor revisi&gt;</i>	<i>Tgl: 21/04/2020</i>



## DAFTAR PERUBAHAN

Revisi	Deskripsi
<b>A</b>	Penambahan usecase diagram baru, dan robustness analysis
<b>B</b>	
<b>C</b>	
<b>D</b>	
<b>E</b>	
<b>F</b>	
<b>G</b>	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh	24/04/2021							
Diperiksa oleh								
Disetujui oleh								

**Daftar Halaman Perubahan**

<b>Halaman</b>	<b>Revisi</b>	<b>Halaman</b>	<b>Revisi</b>

# Daftar Isi

1. Pendahuluan	8
1.1 Tujuan Penulisan Dokumen	8
1.2 Lingkup Masalah	8
1.3 Definisi dan Istilah	8
1.4 Referensi	9
1.5 Sistematika Pembahasan	9
2 Deskripsi Perancangan Global	11
2.1 Rancangan Lingkungan Implementasi	11
2.2 Deskripsi Arsitektural	11
2.3 Deskripsi Komponen	12
3 Perancangan Rinci	13
3.1 Realisasi Use Case	13
3.1.1 Use Case Login	25
3.1.1.1 Identifikasi Kelas Login	25
3.1.1.2 Sequence Diagram Login	25
3.1.1.3 Diagram Kelas Login	25
3.1.2 Use Case Input Pesanan	26
3.1.2.1 Identifikasi Kelas Input Pesanan	26
3.1.2.2 Sequence Diagram Input Pesanan	26
3.1.2.3 Diagram Kelas Input Pesanan	26
3.1.3 Use Case Edit Pesanan	27
3.1.3.1 Identifikasi Kelas Edit Pesanan	27
3.1.3.2 Sequence Diagram Edit Pesanan	27
3.1.3.3 Diagram Kelas Edit Pesanan	27
3.1.4 Use Case Hapus Pesanan	28
3.1.4.1 Identifikasi Kelas Hapus Pesanan	28
3.1.4.2 Sequence Diagram Hapus Pesanan	28
3.1.4.3 Diagram Kelas Hapus Pesanan	28
3.1.5 Use Case View Pesanan	29
3.1.5.1 Identifikasi Kelas View Pesanan	29
3.1.5.2 Sequence Diagram View Pesanan	29
3.1.5.3 Diagram Kelas View Pesanan	30
3.1.6 Use Case View Konsultasi	30
3.1.6.1 Identifikasi Kelas View Konsultasi	30
3.1.6.2 Sequence Diagram View Konsultasi	31
3.1.6.3 Diagram Kelas View Konsultasi	31
3.1.7 Use Case Input Konsultasi	32
3.1.7.1 Identifikasi Kelas Input Konsultasi	32
3.1.7.2 Sequence Diagram Input Konsultasi	32
3.1.7.3 Diagram Kelas Input Konsultasi	33

3.1.8	Use Case Hapus Konsultasi	33
3.1.8.1	Identifikasi Kelas Hapus Konsultasi	33
3.1.8.2	Sequence Diagram Hapus Konsultasi	34
3.1.8.3	Diagram Kelas Hapus Konsultasi	34
3.1.9	Use Case View Laporan Keuangan	35
3.1.9.1	Identifikasi Kelas View Laporan Keuangan	35
3.1.9.2	Sequence Diagram View Laporan Keuangan	35
3.1.9.3	Diagram Kelas View Laporan Keuangan	35
3.1.10	Use Case Edit Laporan Keuangan	36
3.1.10.1	Identifikasi Kelas Edit Laporan Keuangan	36
3.1.10.2	Sequence Diagram Edit Laporan Keuangan	36
3.1.10.3	Diagram Kelas Edit Laporan Keuangan	36
3.1.11	Use Case Edit Registrasi	37
3.1.11.1	Identifikasi Kelas Registrasi	37
3.1.11.2	Sequence Diagram Registrasi	37
3.1.11.3	Diagram Kelas Registrasi	37
3.1.12	Use Case Pembayaran	38
3.1.12.1	Identifikasi Kelas Pembayaran	38
3.1.12.2	Sequence Diagram Pembayaran	38
3.1.12.3	Diagram Kelas Pembayaran	39
3.1.13	Use Case Input Barang	39
3.1.13.1	Identifikasi Kelas Input Barang	39
3.1.13.2	Sequence Diagram Input Barang	39
3.1.13.3	Diagram Kelas Input Barang	40
3.1.14	Use Case Edit Barang	40
3.1.14.1	Identifikasi Kelas Edit Barang	40
3.1.14.2	Sequence Diagram Edit Barang	41
3.1.14.3	Diagram Kelas Edit Barang	41
3.1.15	Use Case Hapus Barang	41
3.1.15.1	Identifikasi Kelas Hapus Barang	41
3.1.15.2	Sequence Diagram Hapus Barang	42
3.1.15.3	Diagram Kelas Hapus Barang	42
3.1.16	Use Case View Barang	42
3.1.16.1	Identifikasi Kelas View Barang	42
3.1.16.2	Sequence Diagram View Barang	43
3.1.16.3	Diagram Kelas Hapus Barang	43
3.1.17	Use Case Edit Edit Profile	43
3.1.17.1	Identifikasi Kelas Edit Profile	43
3.1.17.2	Sequence Diagram Edit Profile	44
3.1.17.3	Diagram Kelas Edit Profile	44
3.2	Perancangan Detil Kelas	44

3.2.1	Kelas Barang	45
3.2.2	Kelas Pelanggan	45
3.2.3	Kelas Konsultasi	46
3.2.4	Kelas Konsultan	46
3.2.5	Kelas User	46
3.2.6	Kelas Admin	47
3.2.7	Kelas Laporan Keuangan	47
3.2.8	Kelas Manajer	48
3.2.9	Kelas Pesanan	48
3.3	Diagram Kelas Keseluruhan	49
3.4	Algoritma/Query	50
3.5	Statechart Diagram	64
3.6	Perancangan Antarmuka	67
3.7	Perancangan Representasi Persistensi Kelas	80
4	Matriks Keruntutan	81

# 1. Pendahuluan

## 1.1 Tujuan Penulisan Dokumen

Dokumen ini berisi penjelasan mengenai Deskripsi Perancangan Perangkat Lunak (DPPL) untuk Website Rakit-Komputer. Tujuan penulisan dokumen ini adalah untuk mendeskripsikan secara rinci mengenai perangkat lunak yang akan dibangun.

Dokumen ini akan digunakan untuk acuan pada proses pembangunan perangkat lunak dan evaluasi diakhir pembangunan perangkat lunak. Pengguna dari dokumen ini adalah tim pengembang dari perangkat lunak dan stakeholders yang terlibat dalam sistem ini. Dengan disusunnya dokumen DPPL ini, diharapkan pembangunan perangkat lunak akan lebih terkonsep dan tidak menimbulkan ambiguitas pada saat pembangunannya.

## 1.2 Lingkup Masalah

*Website Rakit-Komputer* yang akan dikembangkan dalam makalah ini membahas mengenai Website yang mengatur Jual-Beli dan Konsultasi yang akan dilakukan oleh Pengguna dokumen ini nantinya, pada *Website* ini akan terdiri dari Admin, Pengguna, Konsultan, dan Manajer yang nantinya terlibat dalam penjualan *spare-part* atau komponen dari komputer, dan Pelanggan dapat melakukan Konsultasi mengenai Komputer seperti apa yang ingin dibangun nantinya, serta nantinya pula memudahkan para manajer untuk mengecek penjualan yang terjadi pada Rakit-Komputer ini terutama dalam keuangan dan stok barang yang ada.

## 1.3 Definisi dan Istilah

Berikut ini adalah beberapa definisi, singkatan, dan akronim yang terdapat di dalam dokumen ini:

- **DPPL** : Dokumen yang mendeskripsikan dan menjabarkan secara tereperinci mengenai perancangan perangkat lunak yang akan dibangun.
- **SKPL** : Singkatan dari Spesifikasi Kebutuhan Perangkat Lunak atau sering juga disebut sebagai Software Requirements Spesification (SRS) merupakan spesifikasi dari perangkat lunak yang akan dikembangkan.
- **CodeIgniter/CI** : Merupakan salah satu Framework dalam pengembangan Website yang menggunakan sistem MVC (Model,View,Controller).



- **MySQL** : Perangkat lunak sistem manajemen basis data SQL yang bersifat open source.
- **DBMS** : Singkatan dari “Database Management System” yaitu sistem pengorganisasian dan sistem pengolahan basis data pada komputer

#### 1.4 Referensi

- Kelompok 3 IF-39-03. (2017). DPPL Sistem Informasi Citra Mart
- Kelompok 4 IF-42-03. (2018). SKPL Rakit-Komputer
- Muhardian A. (2018). *Tutorial Codeigniter #2: MVC dan Routing, Konsep dasar CI yang Harus Kamu Pahami* Retrieved From PetaniKode.com (<https://www.petanikode.com/codeigniter-mvc/>) diakses pada 18:52 21/04/2020.
- Wikipedia (2020), MySQL Retrieved From (<https://en.wikipedia.org/wiki/MySQL>) diakses pada 12.48 26/04/2020

#### 1.5 Sistematika Pembahasan

Dokumen DPPL ini berisikan deskripsi rancangan perangkat lunak sistem informasi perpustakaan yang akan dikembangkan berdasarkan dokumen SKPL. Pada dokumen DPPL ini akan dijelaskan rincian dari rancangan perangkat lunak sehingga dapat diimplementasikan. Dokumen ini secara garis besar terdiri dari empat bab dengan perincian sebagai berikut:

##### A. Pendahuluan

Pendahuluan berisi penjelasan tentang dokumen DPPL yang mencakup tujuan pembuatan dokumen ini, lingkup masalah yang diselesaikan oleh perangkat lunak yang dikembangkan, definisi, aturan penamaan dan penomoran, referensi, dan ikhtisar dokumen.

##### B. Deskripsi Perancangan Global Deskripsi

Perancangan Global berisi tentang rancangan dari perangkat lunak yang akan dibangun meliputi, rancangan lingkungan implementasi, deskripsi arsitektural, dan deskripsi komponen.

### **C. Perancangan Rinci**

Perancangan Rinci ini berisi tentang realisasi use case, perancangan detil kelas, deskripsi diagram kelas, algoritma/query, diagram statechart, perancangan antarmuka, dan perancangan representasi persistensi kelas.

### **D. Matriks Keterunutan**

Matriks keterunutan berisi tentang hal fungsional yang terdapat pada dokumen SKPL.

## 2 Deskripsi Perancangan Global

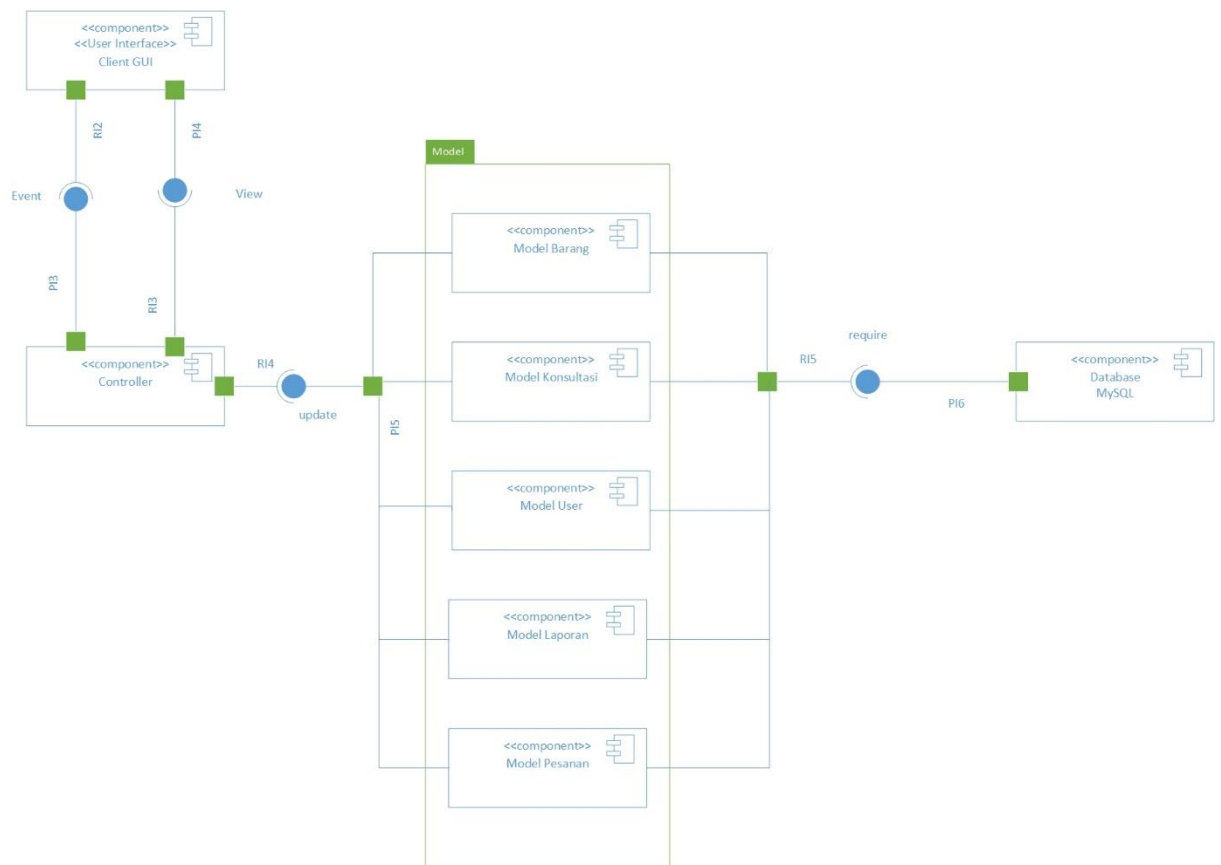
### 2.1 Rancangan Lingkungan Implementasi

Sistem ini diimplementasikan dalam lingkungan sebagai berikut :

- **Sistem Operasi** : Windows, Linux, Mac
- **Bahasa Pemrograman** : PHP, HTML, JavaScript, dan JQuery
- **DBMS** : MySQL
- **Development Tool**: PHPMyAdmin (XAMPP/LAMMP), Apache2 (XAMPP/LAMMP), dan Visual Studio Code/ Sublime Text

### 2.2 Deskripsi Arsitektural

Merupakan gambaran arsitektur atau komponen yang akan diterapkan pada Website Rakit-Komputer untuk mempermudah pengembang dalam mengembangkan atau mengimplementasikan perangkat lunak ini.



Gambar 1. Component Diagram

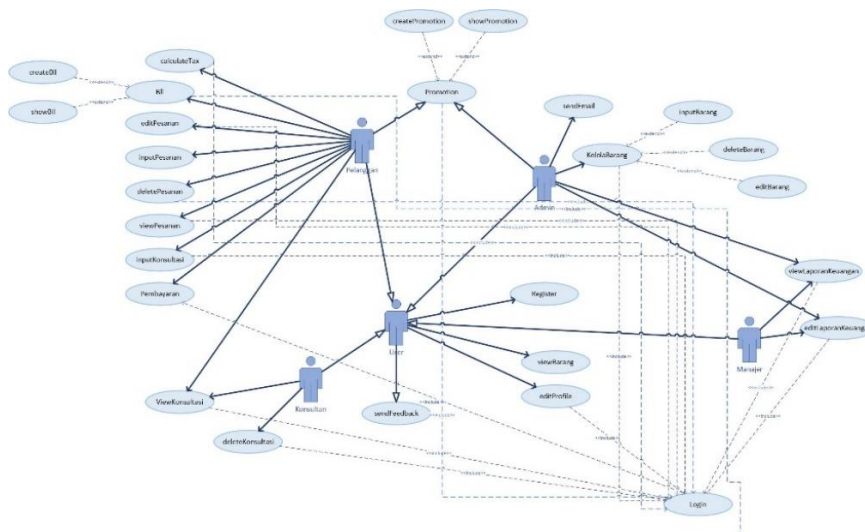
### 2.3 Deskripsi Komponen

No	Nama Komponen	Keterangan
1	Model Barang	Model yang digunakan untuk mengakses tabel yang berkaitan dengan Barang yang nantinya akan digunakan oleh Admin, Pelanggan, dan Konsultan.
2	Model Konsultasi	Model yang digunakan untuk mengakses table yang berkaitan dengan Konsultasi yang nantinya akan digunakan oleh Konsultan dan Pelanggan.
3	Model User	Model yang digunakan untuk mengakses tabel yang berkaitan dengan pengguna Website ini, dalam hal ini ada Pelanggan, Konsultan, Admin, dan Manajer secara fungsionalitas model ini akan mengakses tabel User dan mengakses pula tabel data dari pengguna Website.
4	Model Laporan	Model ini digunakan untuk mengakses data laporan yang digunakan untuk data kepada admin dan manager.
5	Model Pesanan	Model ini digunakan untuk mengakses data pesanan yang ada di database untuk digunakan oleh Admin, Pelanggan.
6	Controller	Controller disini berfungsi untuk mengarahkan model kepada view GUI yang dibutuhkan sesuai dengan kebutuhan dalam hal ini akan membantu dalam mengontrol Pesanan, Konsultasi, Barang , Laporan, serta Data dari Pengguna.
7	Client GUI	Client GUI ini terdiri dari serangkaian jenis View yang digunakan nantinya untuk laman Website, yang terdiri dari Menu Utama , Menu Login, Menu Register, Menu Admin, Menu Pelanggan, Menu Konsultan, dan Menu Manager dan di setiap masing-masing menu itu akan menampilkan sesuai dengan fungsionalitasnya seperti Admin dapat mengontrol Barang, Laporan, Pesanan, sementara Konsultan dapat mengakses Menu Utama dan Konsultasi, Pelanggan dapat mengakses Menu Utama, Konsultasi, dan Pesanan, sementara itu Manager dapat mengakses Menu Laporan.
8	Database MySQL	Database MySQL ini merupakan media penyimpanan data yang ada di database ini.

### 3 Perancangan Rinci

Dalam bab perancangan rinci akan dijelaskan mengenai realisasi use case, perancangan detil kelas, diagram kelas keseluruhan, algoritma/query, diagram statechart, perancangan antarmuka, dan perancangan representasi persistensi kelas

### 3.1 Realisasi Use Case



*Gambar 2.1.0 Use Case Diagram*

### 3.1.1 Usecase Scenario #1

Nama Use Case	Input Pesanan	
Deskripsi	Fungsi ini digunakan oleh user untuk menginputkan Pesanan ke sistem	
Pre-Kondisi	User telah memiliki akun dan sudah melakukan login	
Post-Kondisi	User telah menginputkan data pesanan dan disimpan ke database pesanan.	
Aktor	Pelanggan	
Skenario Utama		
	Aktor	Sistem
	1. Membuka Menu Pesanan	2. Menampilkan pengelolaan barang
	3. Klik input Pesanan	
		4. Menampilkan Form Pesanan
	5. Menginput data Pesanan	

		6. Sistem menyimpan data Pesanan
--	--	----------------------------------

### 3.1.2 Usecase Scenario #2

Nama Use Case	Input Barang	
Deskripsi	Admin telah login dan terdapat data barang yang belum disimpan oleh admin dalam database	
Pre-Kondisi	Data Barang tersimpan di database barang	
Post-Kondisi	Admin telah menginputkan data barang kedalam database	
Aktor	Admin	
Skenario Utama		
	Aktor	Sistem
	1. Membuka Menu barang	
		2. Menampilkan tampilan menu barang
	3. Memilih opsi input barang	
		4. Menampilkan tampilan form input barang
	5. Memasukan Data Barang	
	6. Menekan tombol simpan inputan	
		7. System akan memeriksa apakah inputan valid atau tidak 8. Jika inputan valid maka akan lanjut ke langkah 10 9. Jika tidak valid maka system akan menampilkan notifikasi “input gagal” dan kembali ke langkah 4 10. Inputan data barang akan disimpan dalam database 11. menampilkan data barang yang sudah diinputkan

### 3.1.3 Usecase Scenario #3

Nama Use Case	Edit Pesanan
Deskripsi	Fungsi ini digunakan oleh user untuk mengedit pesanan ke sistem
Pre-Kondisi	User telah memiliki akun dan sudah melakukan login, juga sudah melakukan pemesanan barang.

Post-Kondisi	Pesanan user telah diedit dari database pesanan	
Aktor	Pelanggan	
Skenario Utama		
	Aktor	Sistem
	1. Membuka Menu Pesanan	
		2. Menampilkan pengelolaan pesanan
	3. Klik button edit di salah satu pesanan	
		4. Menampilkan Form Pesanan
	5. Menginput data pesanan	
		6. Sistem menyimpan data Pesanan

#### 3.1.4 Usecase Scenario #4

Nama Use Case	Edit Barang	
Deskripsi	Admin telah login kemudian dapat melakukan edit data barang yang dilakukan	
Pre-Kondisi	Data Barang berada di database pesanan.	
Post-Kondisi	Admin sudah mengubah isi dari data barang yang dilakukan dan kemudian disimpan kembali di database barang.	
Aktor	Admin	
Skenario Utama		
	Aktor	Sistem
	1. Membuka menu list Barang lalu Memilih Barang yang	
		2. Menampilkan Menu detail barang
	3. Memilih opsi edit barang yang sedang dilakukan.	
		4. Menampilkan form yang berisi detail barang
	5. Mengganti data barang	
	6. Menekan tombol Simpan barang	

		7. System akan memeriksa apakah inputan valid atau tidak.
		8. Jika inputan valid maka akan lanjut ke langkah 10
		9. Jika tidak valid maka system akan menampilkan notifikasi “input gagal” dan kembali ke langkah 4
		10. Inputan data pesanan akan disimpan dalam database

#### 3.1.5 Usecase Scenario #5

Nama Use Case	Hapus Pesanan	
Deskripsi	Fungsi ini digunakan oleh user untuk menghapus pesanan dari sistem	
Pre-Kondisi	User telah memiliki akun dan sudah melakukan login, juga sudah melakukan pemesanan barang.	
Post-Kondisi	Pesanan user telah dihapus dari database pesanan	
Aktor	Pelanggan	
Skenario Utama		
	Aktor	Sistem
	1. Membuka Menu Pesanan	
		2. Menampilkan pengelolaan barang
	3. Klik button hapus di salah satu pesanan	
		4. Menampilkan dialog apakah akan menghapus pesanan
	5. Menekan salah satu tombol, ‘Ya’ atau ‘Tidak’	



		6. Apabila 'Ya', data terhapus dari pesanan sistem. Apabila 'Tidak', kembali ke nomor 1
--	--	---

### 3.1.6 Usecase Scenario #6

Nama Use Case	Hapus Barang	
Deskripsi	Admin telah login ke akunnya kemudian dapat mengedit barang yang salah satu opsinya adalah menghapus barang	
Pre-Kondisi	Data barang berada di database barang.	
Post-Kondisi	Data barang yang dimaksud sudah terhapus dari database barang.	
Aktor	Admin	
Skenario Utama		
	Aktor	Sistem
	1. Membuka Menu yang menampilkan barang yang dijual.	
	2. Memilih barang yang ingin diedit dengan mengklik button edit.	
		3. Menampilkan detail barang juga button edit dan button hapus barang.
	4. Mengklik tombol hapus barang.	
		5. Menampilkan konfirmasi untuk menghapus barang
	6. Menekan button 'Tidak' untuk membatalkan, 'Ya' untuk menghapus barang	
		7. Jika 'Ya', barang akan terhapus dari database, jika 'Tidak' maka kembali ke poin 2.

### 3.1.7 Usecase Scenario #7

Nama Use Case	View Barang	
Deskripsi	Pengguna tidak perlu login untuk menjalankan use case ini, list barang yang dijual tampil sebagai tampilan utama.	
Pre-Kondisi	Pengguna ingin melihat barang yang dijual	
Post-Kondisi	List barang yang dijual ditampilkan	
Aktor	Admin,Pelanggan,Konsultan	
Skenario Utama		
	Aktor	Sistem
	1. Pengguna membuka halaman utama website	
		2. Menampilkan list barang yang dijual beserta harganya
	3. Apabila ingin melihat detail barang, pengguna bisa mengklik button detail	
	4. Mengklik tombol hapus barang.	5. Apabila button detail di klik maka, sistem menampilkan detail barang seperti spesifikasi dan pilihan warna

### 3.1.8 Usecase Scenario #8

Nama Use Case	View Konsultasi	
Deskripsi	Digunakan untuk melihat konsultasi yang tersimpan di database	
Pre-Kondisi	User ingin melihat data konsultasi yang pernah terjadi.	
Post-Kondisi	Sistem menampilkan data konsultasi yang pernah terjadi	
Aktor	Pelanggan,Konsultan	
Skenario Utama		
	Aktor	Sistem
	1. Mengklik menu lihat konsultasi	
		2. Menampilkan list konsultasi yang pernah melakukan konsultasi
	3. Memilih salah satu konsultasi	
		4. Menampilkan detail konsultasi

### 3.1.9 Usecase Scenario #9

Nama Use Case	Input Konsultasi	
Deskripsi	Digunakan oleh pelanggan untuk melakukan konsultasi	
Pre-Kondisi	User melihat tampilan utama web	
Post-Kondisi	Konsultasi tersimpan di database	
Skenario Utama		
	Aktor	Sistem
	1. Mengklik menu 'konsultasi' di navigation bar	
		2. Menampilkan form konsultasi
	3. Mengisi form konsultasi lalu menekan button 'Kirim'	
		4. Menyimpan data konsultasi

### 3.1.10 Usecase Scenario #10

Nama Use Case	Hapus Konsultasi	
Deskripsi	Digunakan oleh Konsultan untuk menghapus konsultasi yang masuk	
Pre-Kondisi	Data konsultasi ditampilkan	
Post-Kondisi	Data konsultasi yang dipilih terhapus dari database	
Aktor	Konsultan	
Skenario Utama		
	Aktor	Sistem
	1. Membuka menu lihat konsultasi	
		2. Menampilkan list konsultasi yang pernah melakukan konsultasi
	3. Memilih salah satu konsultasi	
		4. Menampilkan detail konsultasi
	5. Menekan tombol hapus konsultasi	
		6. Konsultasi yang dipilih terhapus dari

		database lalu redirect ke nomor 2
--	--	-----------------------------------

### 3.1.11 Usecase Scenario #11

Nama Use Case	View Laporan Keuangan	
Deskripsi	Digunakan oleh manajer dan admin untuk melihat laporan keuangan (hasil penjualan, dsb.)	
Pre-Kondisi	Manajer atau admin ingin melihat laporan keuangan	
Post-Kondisi	Data laporan keuangan ditampilkan di layar	
Aktor	Manager, Admin	
Skenario Utama		
	Aktor	Sistem
	1. Manajer atau Admin harus sudah login, lalu memilih menu lihat laporan keuangan	
		2. Menampilkan laporan keuangan

### 3.1.12 Usecase Scenario #12

Nama Use Case	Edit Laporan Keuangan	
Deskripsi	Digunakan oleh manajer dan admin untuk mengedit laporan keuangan (hasil penjualan, dsb.)	
Pre-Kondisi	Data laporan keuangan ditampilkan	
Post-Kondisi	Data laporan keuangan yang sudah diedit tersimpan di database	
Aktor	Manager, Admin	
Skenario Utama		
	Aktor	Sistem
	1. Manajer atau Admin harus sudah login, lalu memilih menu lihat laporan keuangan	
		2. Menampilkan laporan keuangan
	3. Manajer atau admin mengklik data yang ingin diedit	4.
		5. Detail data yang ingin diedit tampil ke layar
	6. Manajer atau admin mengganti value data yang ingin diubah lalu	

	mengklik tombol 'Simpan'	
		7. Muncul dialog konfirmasi
	8. Mengklik 'Ya' atau 'Tidak'	
		9. Jika 'Ya', data baru akan tersimpan, jika 'Tidak' kembali ke nomor 6.

### 3.1.13 Usecase Scenario #13

Nama Use Case	Login	
Deskripsi	Digunakan oleh semua role untuk melakukan login	
Pre-Kondisi	User belum login sehingga hanya bisa melihat halaman utama	
Post-Kondisi	User sudah login dan bisa melihat halaman sesuai role mereka	
Aktor	Pelanggan, Admin, Konsultan, Manager	
Skenario Utama		
	Aktor	Sistem
	1. User membuka halaman utama website	
	2. User mengklik button login	
		3. Menampilkan menu login
	4. Mengisi form login (username dan password)	
		5. Mencocokkan dengan data user di database, apabila sama maka session login dimulai, apabila salah kembali ke nomor 3.

### 3.1.14 Usecase Scenario #14

Nama Use Case	View Pesanan	
Deskripsi	Pelanggan dapat melihat pesanan yang dilakukan	
Pre-Kondisi	Pelanggan ingin melihat pesanan yang di lakukan	
Post-Kondisi	Pelanggan berhasil melihat list pesanan yang sedang berlangsung yang dilakukan oleh dirinya.	
Aktor	Pelanggan	
Skenario Utama		
	Aktor	Sistem
	1. Berada di halmaan utaman lalu membuka Menu Pesanan	
		2. Menampilkan List Pesanan yang berlangsung

### 3.1.15 Usecase Scenario #15

Nama Use Case	Edit Profile	
Deskripsi	Aktor dapat mengubah profile yang dimiliki, untuk diisi dengan data yang aktual	
Pre-Kondisi	Aktor belum mengubah, atau mengisi profile yang dimiliki	
Post-Kondisi	Profile Aktor sudah diubah dan disimpan di database	
Aktor	Pelanggan, Manager, Admin, Konsultan	
Skenario Utama		
	Aktor	Sistem
	1. Membuka Menu Profile	
		2. Menampilkan Menu Profile
	3. Pilih Menu Edit Profile	
		4. Mengambil Data Diri dari Akun yang Login
		5. Mengembalikan dalam bentuk Form edit
	6. Mengisi Form Edit	
	7. Klik Save	

		8. Menyimpan ke Database
		9. Menampilkan Menu Pilihan Kembali
	10. Klik OK	
		11. Kembali Ke Menu Utama

#### 3.1.16 Usecase Scenario #16

Nama Use Case	Registrasi User	
Deskripsi	Mendaftarkan data user baru ke database oleh user yang belum terdaftar	
Pre-Kondisi	User telah memasuki halaman utama dan belum memiliki akun	
Post-Kondisi	Akun baru user telah terdaftar dan telah melakukan login	
Aktor	Pelanggan	
Skenario Utama		
	Aktor	Sistem
	1. Klik Sign Up	
		2. Menampilkan form registrasi akun user
	3. Mengisi form registrasi akun user 4. Klik Submit	
		4. Mengecek data – data pada form register 5. Jika data – data valid akan dilanjutkan ke langkah 7 6. Jika data – data tidak valid akan

		<p>menampilkan text “data yang diinputkan terdapat kesalahan” dan kembali ke langkah 2</p> <p>7. Menampilkan text “Akun Anda sudah terdaftar” dan menyimpan data user ke database</p> <p>8. Kembali ke halaman utama</p>
--	--	--

### 3.1.17 Usecase Scenario #17

Nama Use Case	Pembayaran	
Deskripsi	Pelanggan melakukan pembayaran pada transaksi yang dilakukan	
Pre-Kondisi	Aktor sudah login dan memiliki pesanan, kemudian ingin melakukan pembayaran	
Post-Kondisi	Aktor telah melakukan pembayaran	
Aktor	Pelanggan	
Skenario Utama		
	Aktor	Sistem
	1. Pelanggan sudah di menu pesanan lalu mengklik detail pesanan	
		2. Menampilkan informasi Pesanan dan jumlah harga
	3. Memilih opsi pembayaran 4. Klik bayar	
		5. Mengirimkan kode verifikasi
	6. Memasukkan kode verifikasi 7. Klik OK	
		<p>8. System akan memeriksa kode verifikasi</p> <p>9. Jika inputan valid maka akan lanjut ke langkah 11</p> <p>10. Jika tidak valid maka system akan menampilkan notifikasi “kode verifikasi salah” dan kembali ke langkah 6</p> <p>11. Data pembayaran diterima sistem dan dimasukkan ke database</p> <p>12. Menampilkan teks “Pembayaran berhasil”</p> <p>13. Kembali ke halaman utama</p>

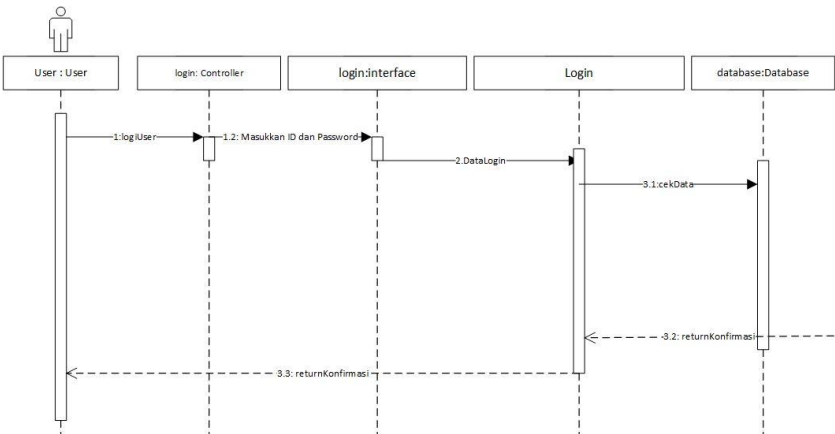


3.1.1 Use Case Login

3.1.1.1 Identifikasi Kelas Login

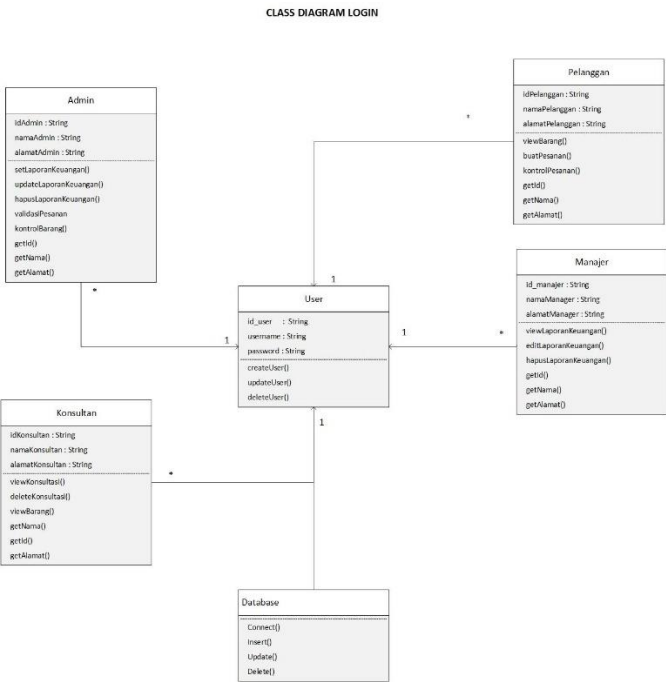
No	Nama Kelas Perancangan	Tipe Kelas
1	User	Entity
2	Admin	Entity
3	Manager	Entity
4	Pelanggan	Entity
5	Konsultan	Entity
6	Database	Entity

3.1.1.2 Sequence Diagram Login



Gambar 3.1.1 Sequence Diagram Login

3.1.1.3 Diagram Kelas Login



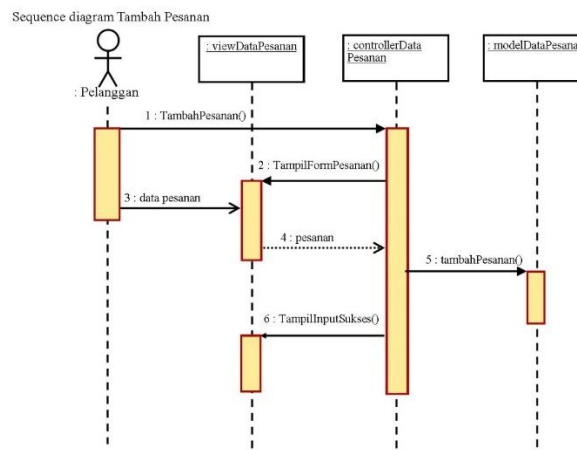
Gambar 2.1.2 Class Diagram Login

### 3.1.2 Use Case Input Pesanan

#### 3.1.2.1 Identifikasi Kelas Input Pesanan

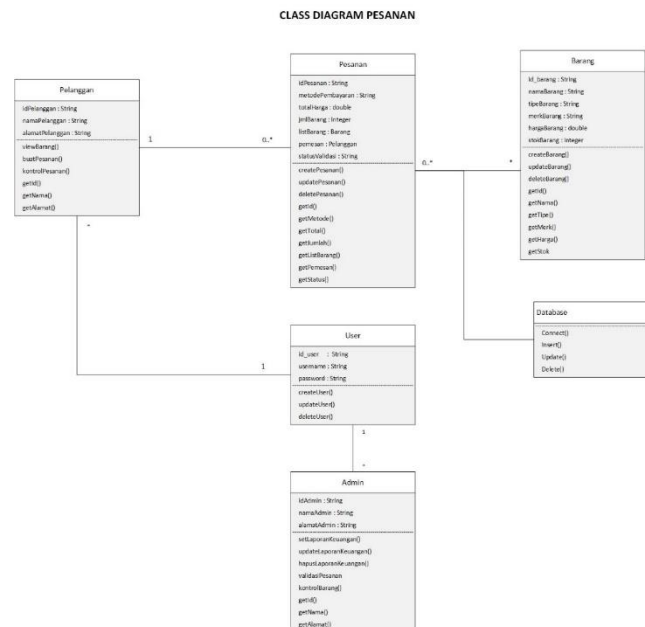
No	Nama Kelas Perancangan	Tipe Kelas
1	User	Entity
2	Pelanggan	Entity
3	Pesanan	Entity
4	Barang	Entity
5	Database	Entity

#### 3.1.2.2 Sequence Diagram Input Pesanan



Gambar 2.2.1 Sequence Diagram Input Pesanan

#### 3.1.2.3 Diagram Kelas Input Pesanan



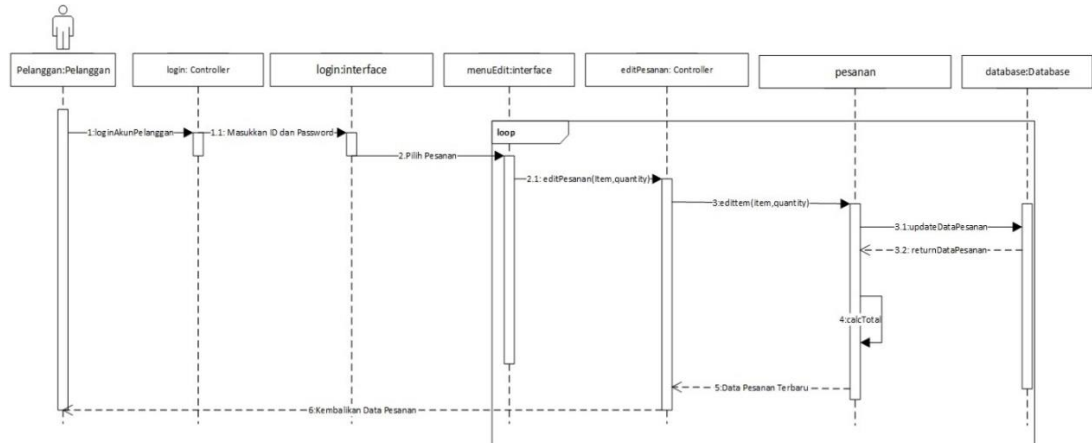
Gambar 2.2.2 Class Diagram Input Pesanan

### 3.1.3 Use Case Edit Pesanan

#### 3.1.3.1 Identifikasi Kelas Edit Pesanan

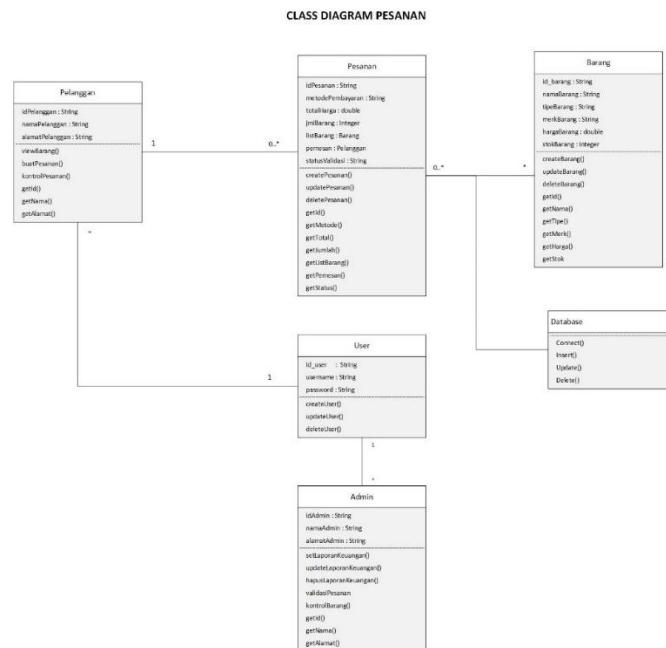
No	Nama Kelas Perancangan	Tipe Kelas
1	User	Entity
2	Pelanggan	Entity
3	Pesanan	Entity
4	Barang	Entity
5	Database	Entity

#### 3.1.3.2 Sequence Diagram Edit Pesanan



Gambar 2.3.1 Sequence Diagram Edit Pesanan

#### 3.1.3.3 Diagram Kelas Edit Pesanan



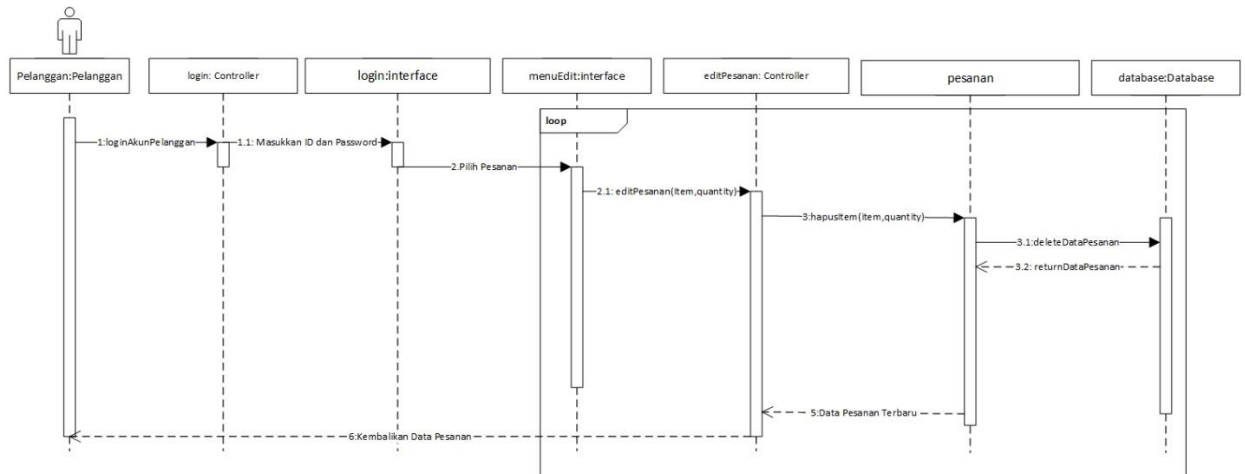
Gambar 2.3.2 Class Diagram Edit Pesanan

### 3.1.4 Use Case Hapus Pesanan

#### 3.1.4.1 Identifikasi Kelas Hapus Pesanan

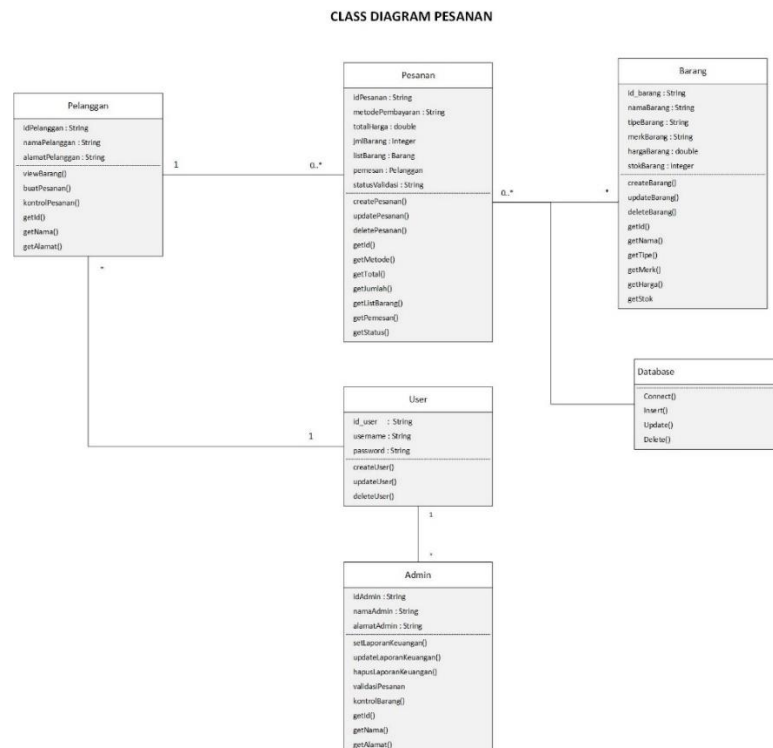
No	Nama Kelas Perancangan	Tipe Kelas
1	User	Entity
2	Pelanggan	Entity
3	Pesanan	Entity
4	Barang	Entity
5	Database	Entity

#### 3.1.4.2 Sequence Diagram Hapus Pesanan



Gambar 2.4.1 Sequence Diagram Hapus Pesanan

#### 3.1.4.3 Diagram Kelas Hapus Pesanan



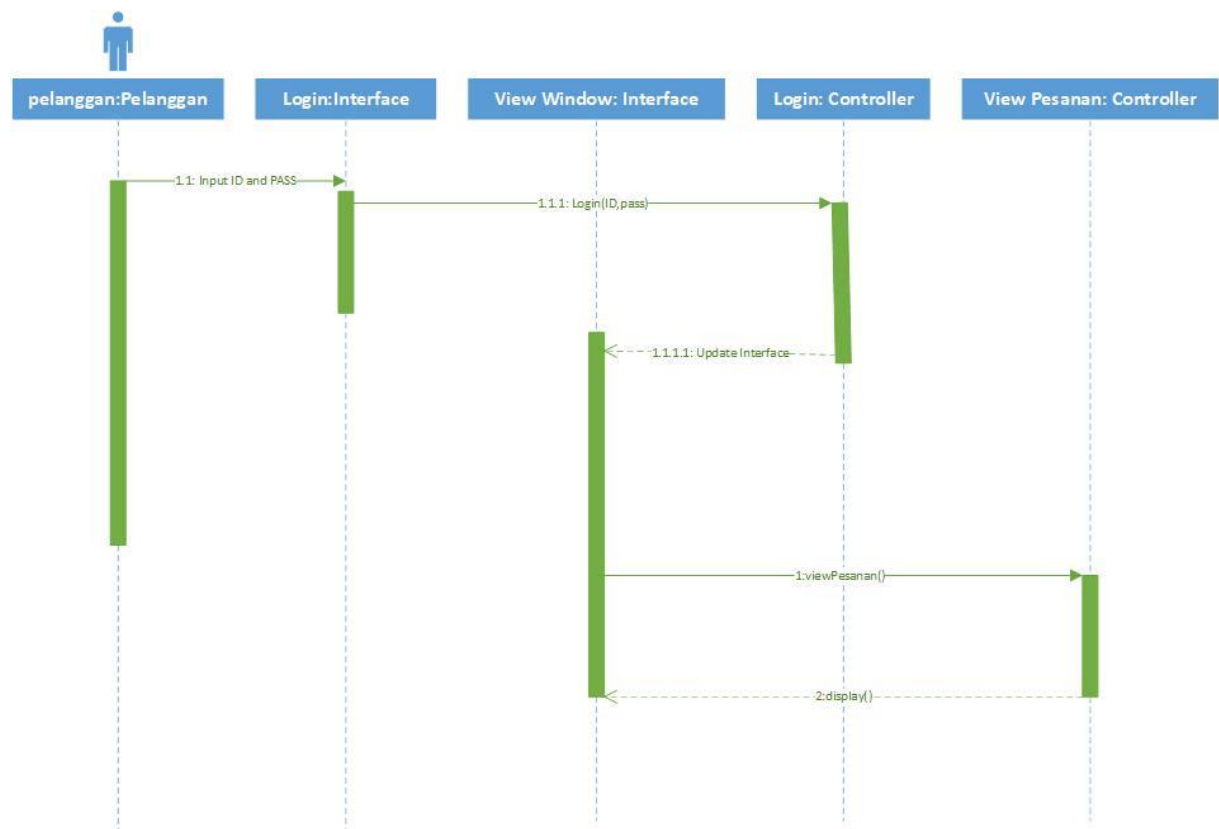
Gambar 2.4.2 Class Diagram Hapus Pesanna

### 3.1.5 Use Case View Pesanan

#### 3.1.5.1 Identifikasi Kelas View Pesanan

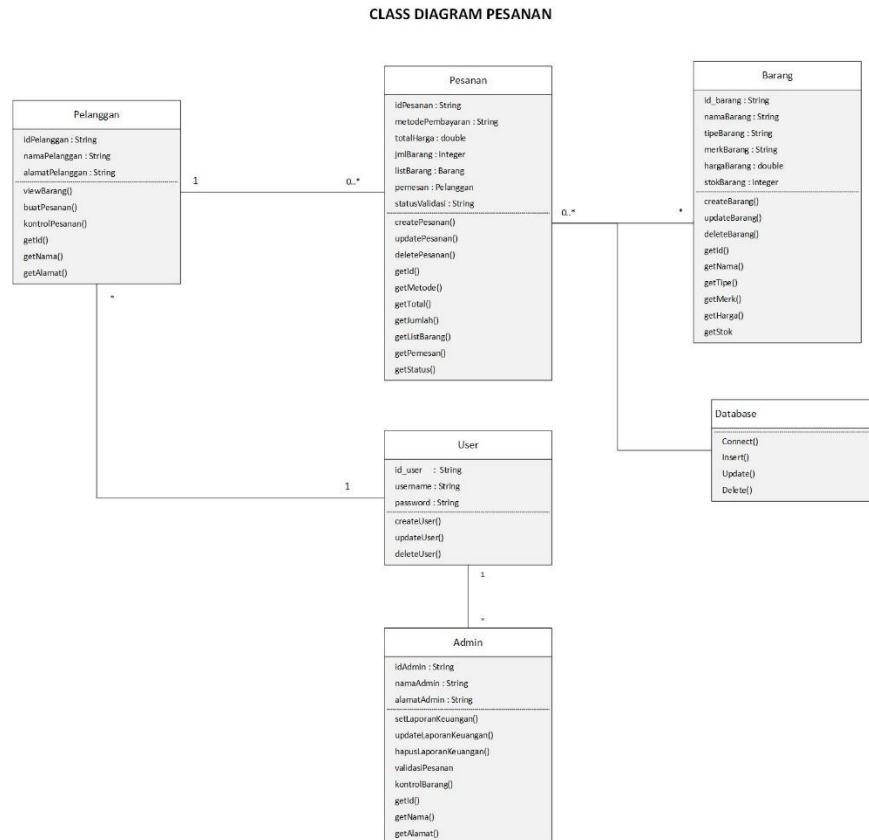
No	Nama Kelas Perancangan	Tipe Kelas
1	User	Entity
2	Pelanggan	Entity
3	Pesanan	Entity
4	Barang	Entity
5	Database	Entity

#### 3.1.5.2 Sequence Diagram View Pesanan



Gambar 2.5.1 Sequence Diagram View Pesanan

### 3.1.5.3 Diagram Kelas View Pesanan



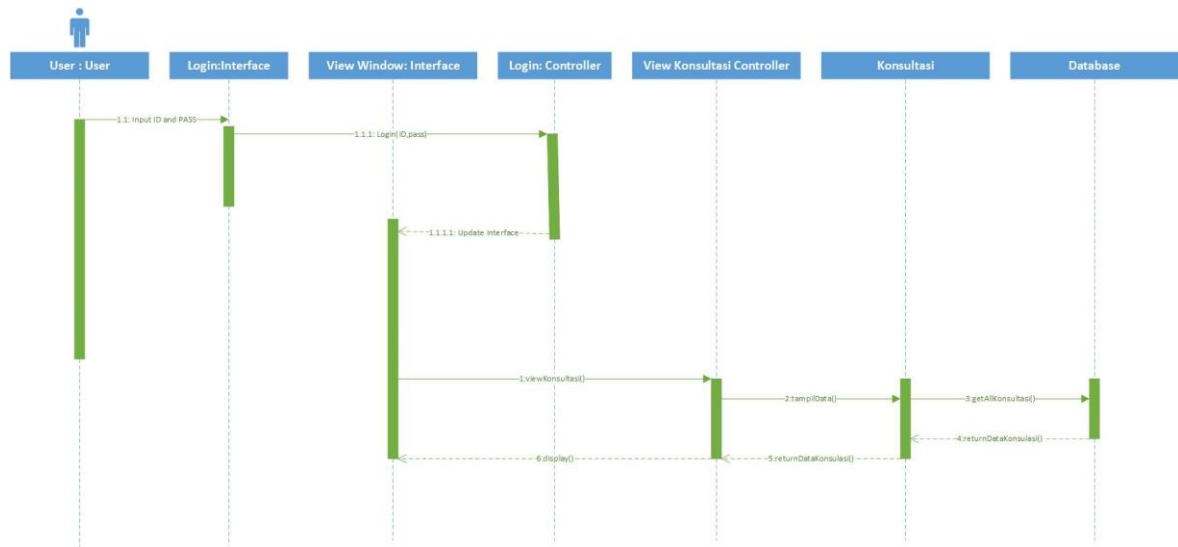
Gambar 2.5.2 Class Diagram View Pesanan

### 3.1.6 Use Case View Konsultasi

#### 3.1.6.1 Identifikasi Kelas View Konsultasi

No	Nama Kelas Perancangan	Tipe Kelas
1	User	Entity
2	Pelanggan	Entity
3	Konsultasi	Entity
4	Barang	Entity
5	Database	Entity

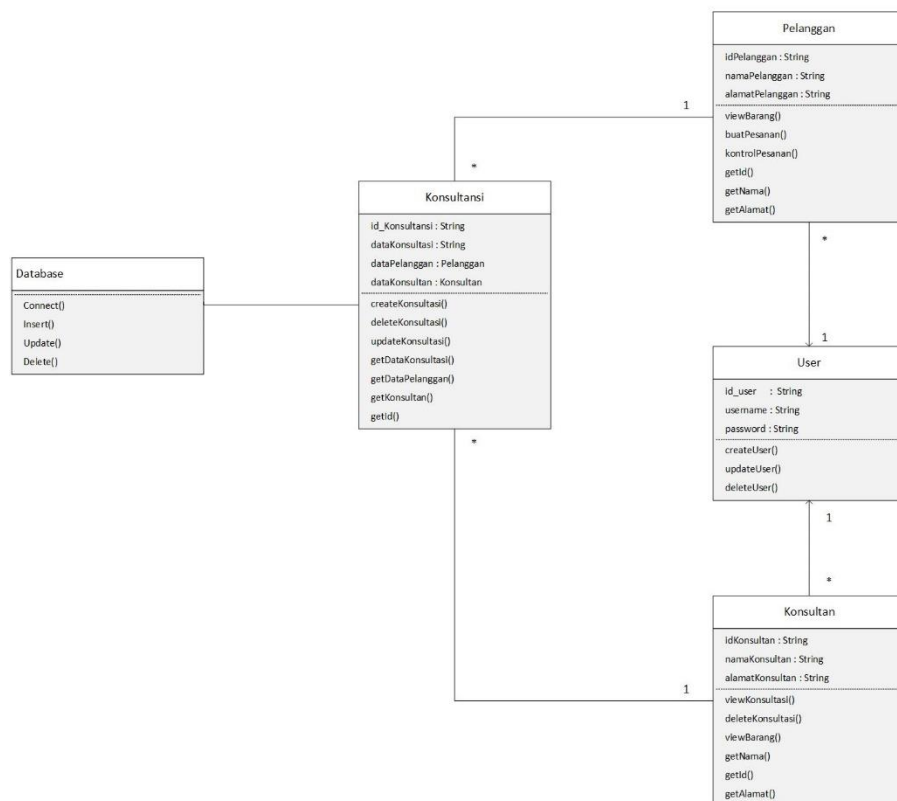
### 3.1.6.2 Sequence Diagram View Konsultasi



Gambar 2.6.1 Sequence Diagram Input Pesanan

### 3.1.6.3 Diagram Kelas View Konsultasi

#### CLASS DIAGRAM KONSULTASI



Gambar 2.6.2. Class Diagram View Konsultasi

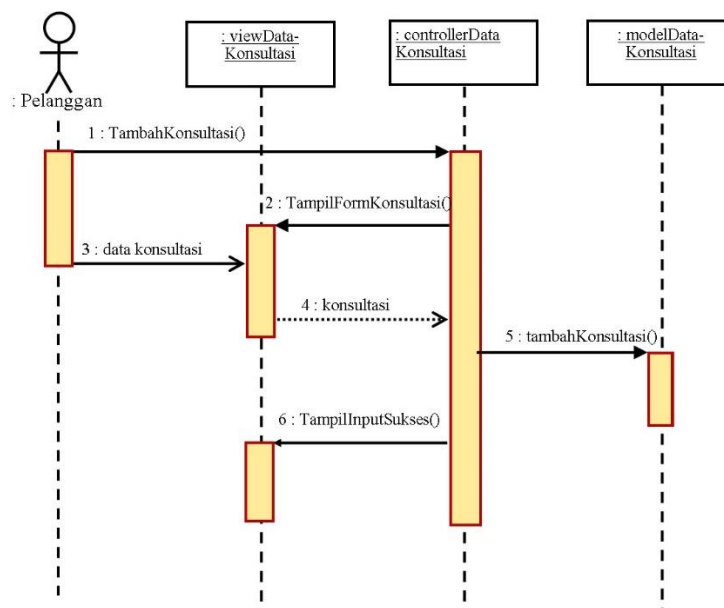
### 3.1.7 Use Case Input Konsultasi

#### 3.1.7.1 Identifikasi Kelas Input Konsultasi

No	Nama Kelas Perancangan	Tipe Kelas
1	User	Entity
2	Pelanggan	Entity
3	Konsultasi	Entity
4	Barang	Entity
5	Database	Entity

#### 3.1.7.2 Sequence Diagram Input Konsultasi

Sequence diagram Tambah Konsultasi

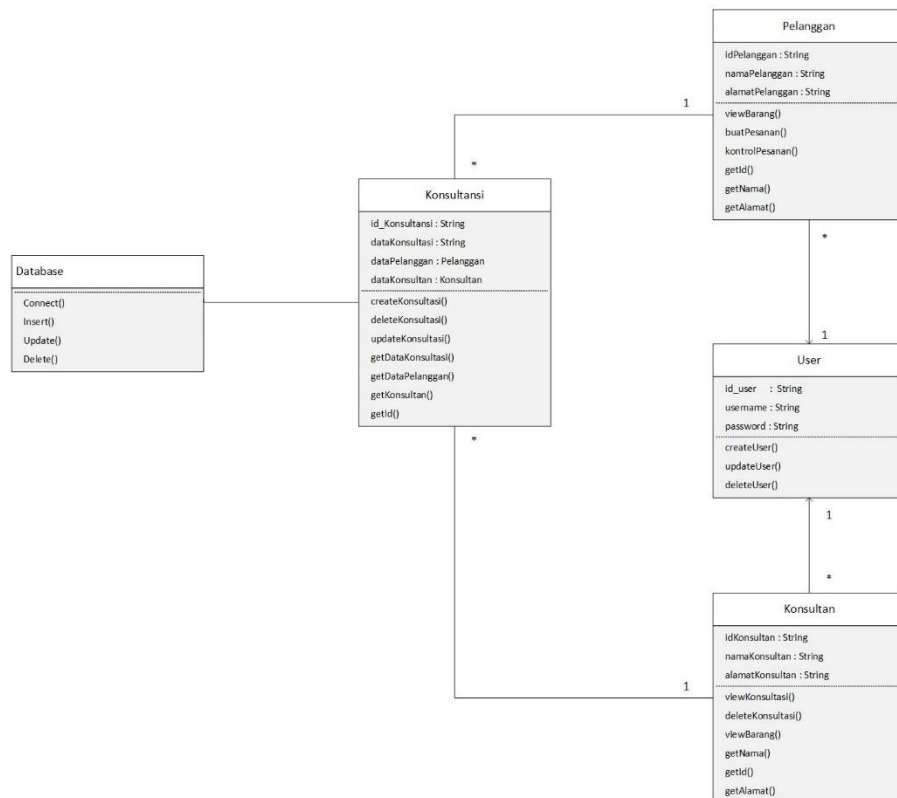


Gambar 2.7.1 Sequence Diagram Input Konsultasi



### 3.1.7.3 Diagram Kelas Input Konsultasi

#### CLASS DIAGRAM KONSULTASI



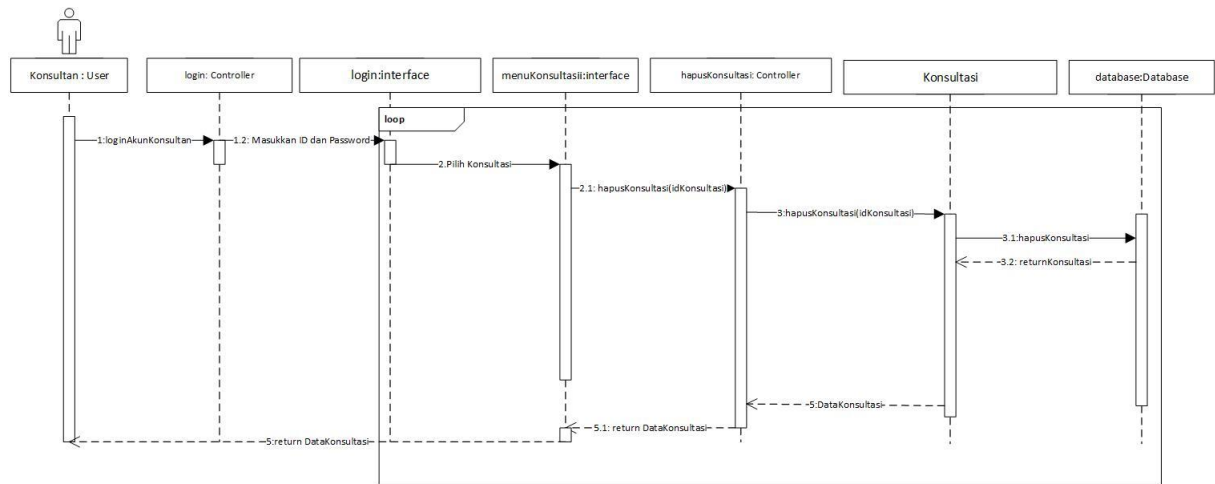
Gambar 2.7.2 Class Diagram Input Konsultasi

### 3.1.8 Use Case Hapus Konsultasi

#### 3.1.8.1 Identifikasi Kelas Hapus Konsultasi

No	Nama Kelas Perancangan	Tipe Kelas
1	User	Entity
2	Pelanggan	Entity
3	Konsultasi	Entity
4	Barang	Entity
5	Database	Entity

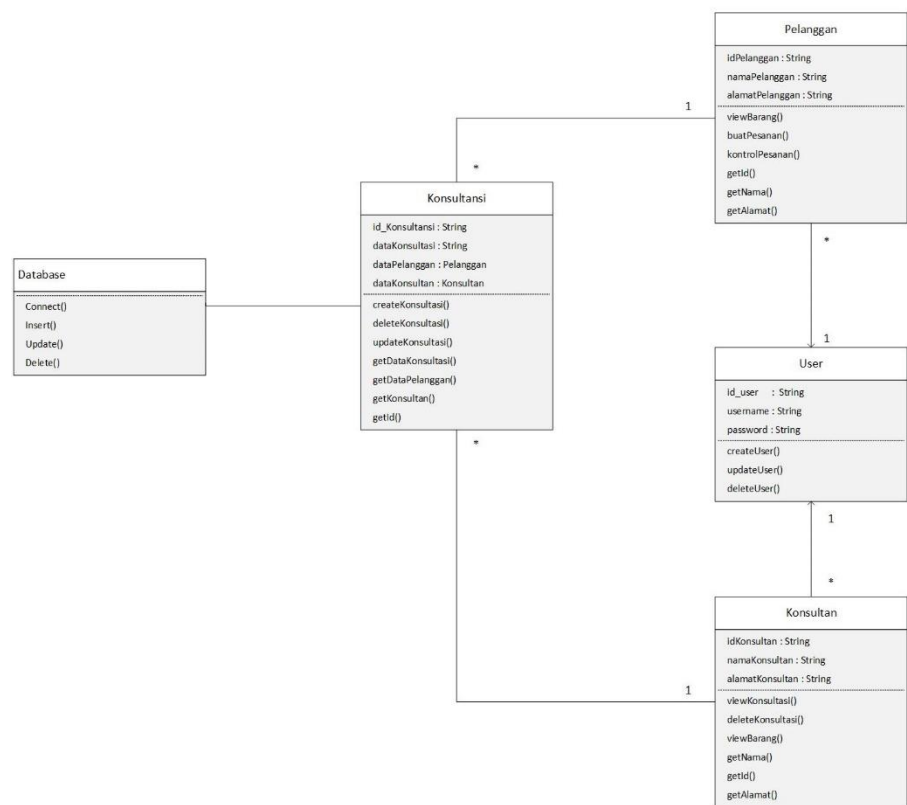
3.1.8.2 Sequence Diagram Hapus Konsultasi



Gambar 2.8.1 Sequence Diagram Hapus Konsultasi

3.1.8.3 Diagram Kelas Hapus Konsultasi

CLASS DIAGRAM KONSULTASI



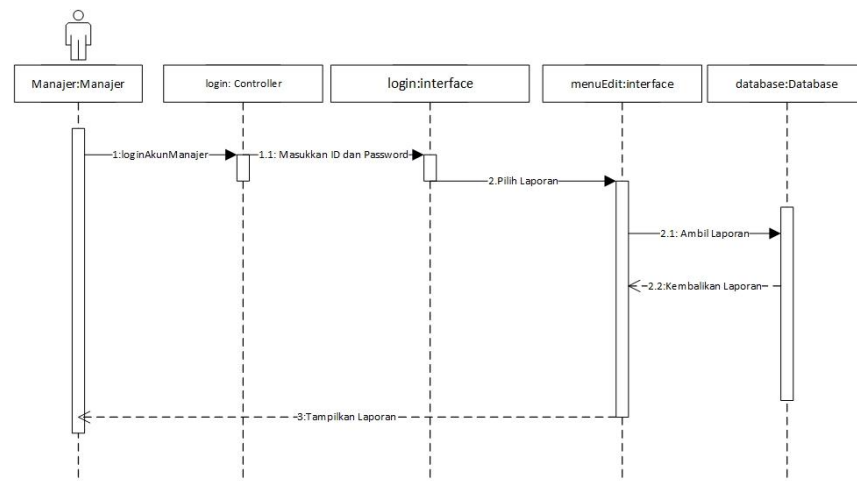
Gambar 2.8.2 Class Diagram Hapus

### 3.1.9 Use Case View Laporan Keuangan

#### 3.1.9.1 Identifikasi Kelas View Laporan Keuangan

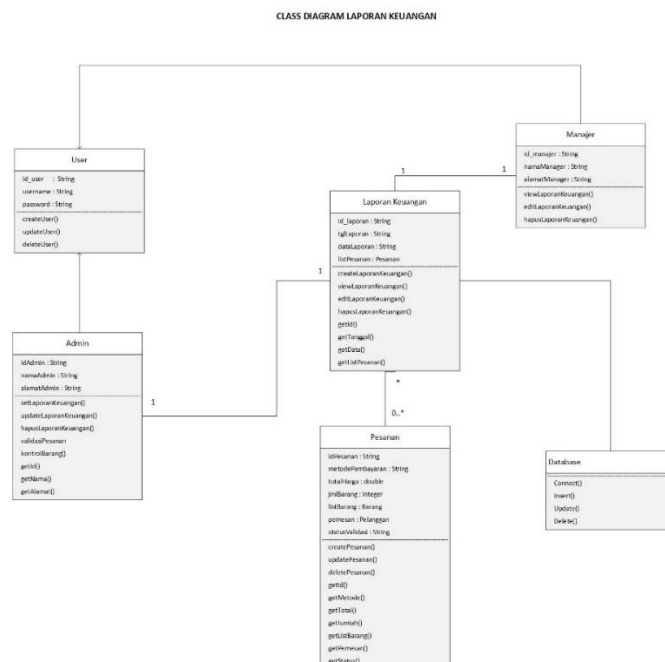
No	Nama Kelas Perancangan	Tipe Kelas
1	User	Entity
2	Admin	Entity
3	Manager	Entity
4	Laporan Keuangan	Entity
5	Pesanan	Entity
6	Database	Entity

#### 3.1.9.2 Sequence Diagram View Laporan Keuangan



Gambar 2.9.1 Sequence Diagram View Laporan Keuangan

#### 3.1.9.3 Diagram Kelas View Laporan Keuangan



Gambar 2.9.2. Class Diagram Laporan Keuangan

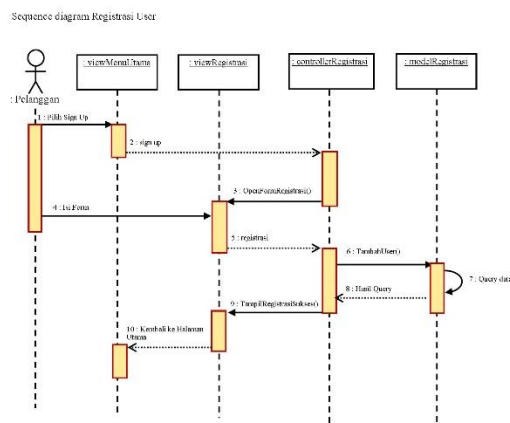


3.1.11 Use Case Edit Registrasi

3.1.11.1 Identifikasi Kelas Registrasi

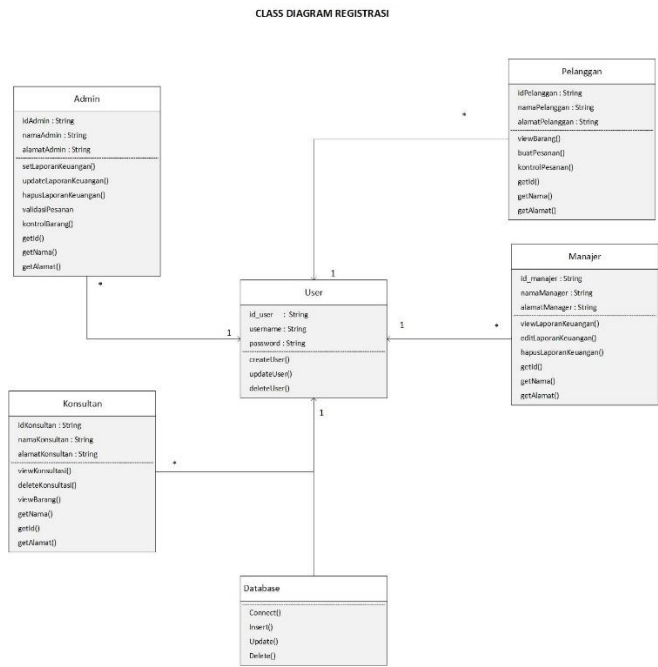
No	Nama Kelas Perancangan	Tipe Kelas
1	User	Entity
2	Admin	Entity
3	Manager	Entity
4	Pelanggan	Entity
5	Konsultan	Entity
6	Database	Entity

3.1.11.2 Sequence Diagram Registrasi



Gambar 2.11.1 Sequence Diagram Registrasi

3.1.11.3 Diagram Kelas Registrasi



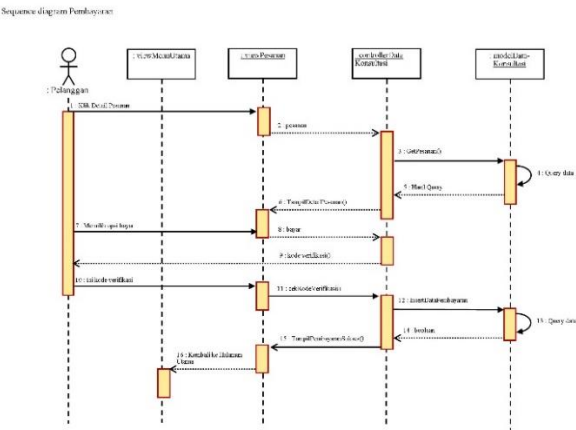
Gambar 2.11.2. Class Diagram Registrasi

3.1.12 Use Case Pembayaran

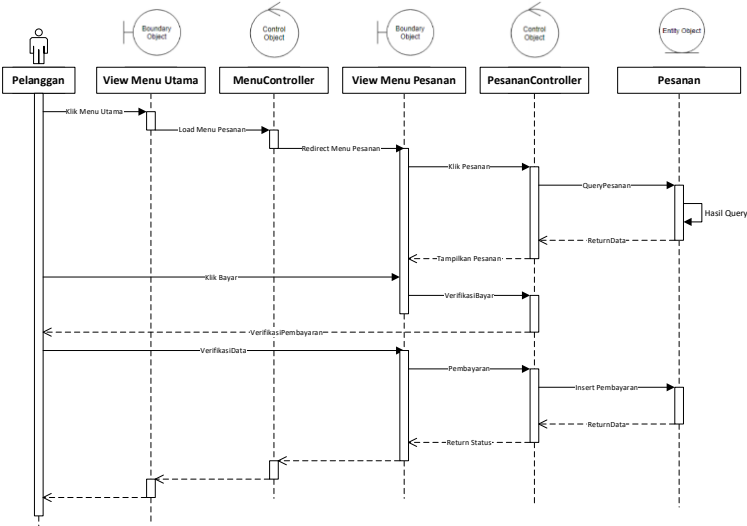
3.1.12.1 Identifikasi Kelas Pembayaran

No	Nama Kelas Perancangan	Tipe Kelas
1	User	Entity
2	Pelanggan	Entity
3	Pesanan	Entity
4	Barang	Entity
5	Database	Entity

3.1.12.2 Sequence Diagram Pembayaran

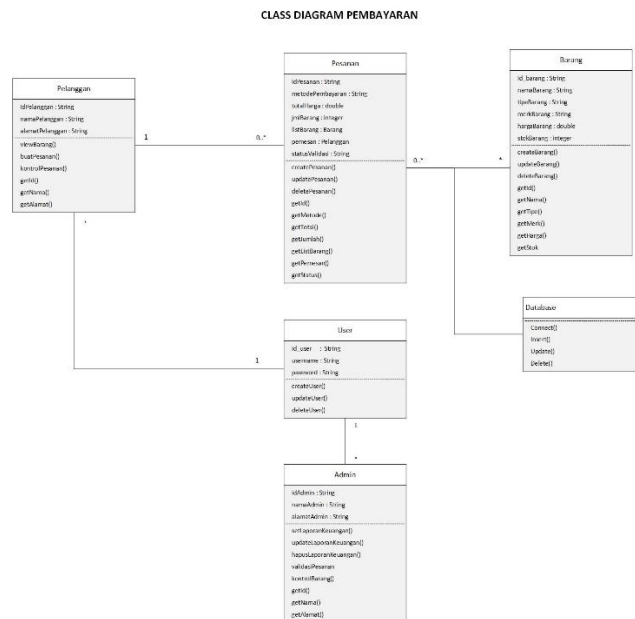


Gambar 2.12.1 Sequence Diagram Pembayaran



Gambar 2.12.2 Sequence Diagram Pembayaran (Robustness Analysis)

### 3.1.12.3 Diagram Kelas Pembayaran



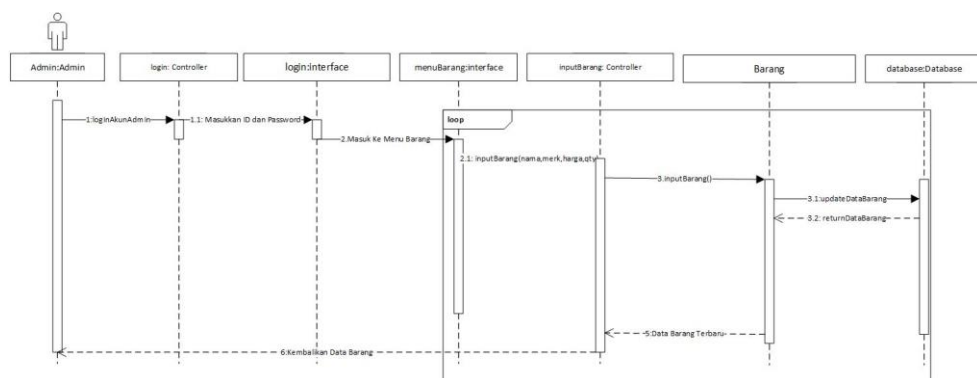
Gambar 2.12.3. Class Diagram Pembayaran

### 3.1.13 Use Case Input Barang

#### 3.1.13.1 Identifikasi Kelas Input Barang

No	Nama Kelas Perancangan	Tipe Kelas
1	User	Entity
2	Admin	Entity
3	Barang	Entity
4	Database	Entity

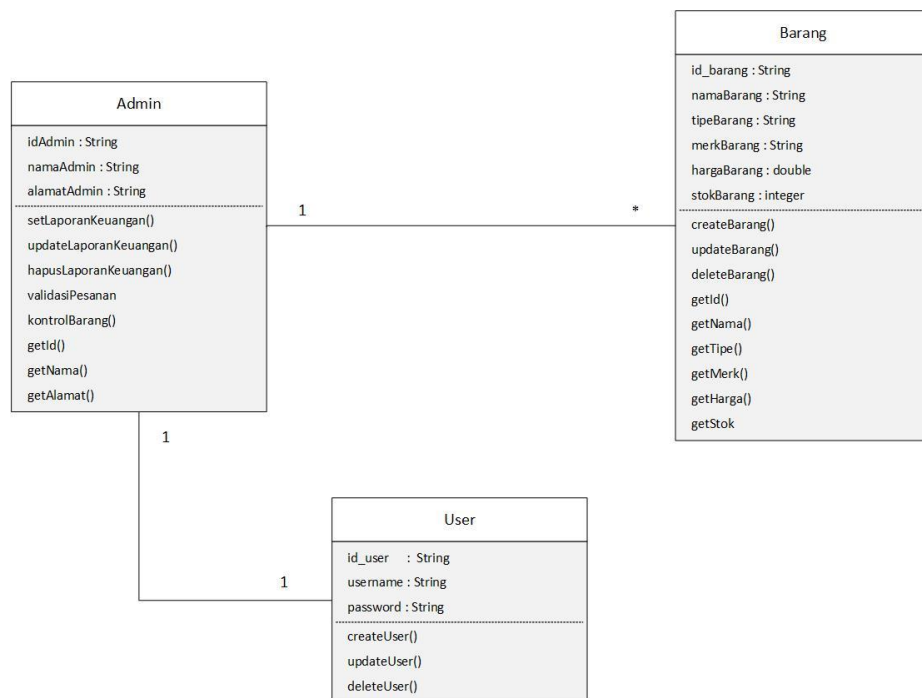
#### 3.1.13.2 Sequence Diagram Input Barang



Gambar 2.13.1 Sequence Diagram Input Barang

### 3.1.13.3 Diagram Kelas Input Barang

CLASS DIAGRAM BARANG



Gambar 2.13.2. Class Diagram Input Barang

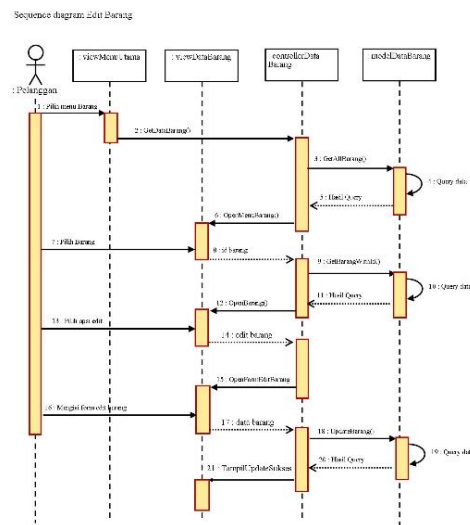
### 3.1.14 Use Case Edit Barang

#### 3.1.14.1 Identifikasi Kelas Edit Barang

No	Nama Kelas Perancangan	Tipe Kelas
1	User	Entity
2	Admin	Entity
3	Barang	Entity
4	Database	Entity

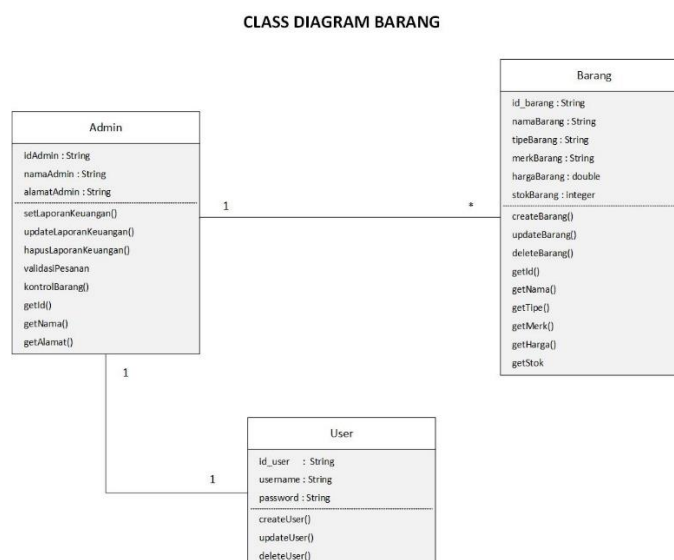


### 3.1.14.2 Sequence Diagram Edit Barang



*Gambar 2.14.1 Sequence Diagram Edit Barang*

### 3.1.14.3 Diagram Kelas Edit Barang



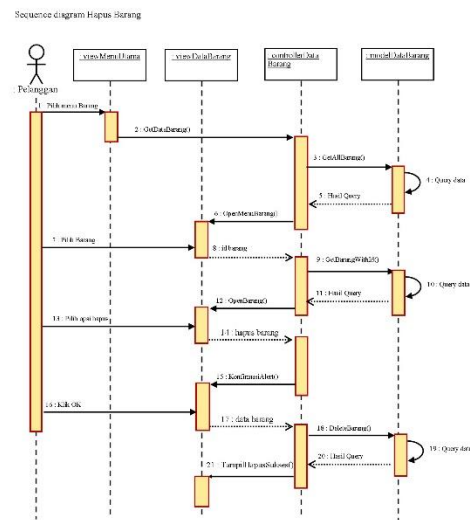
*Gambar 2.14.2. Class Diagram Edit Barang*

### 3.1.15 Use Case Hapus Barang

### 3.1.15.1 Identifikasi Kelas Hapus Barang

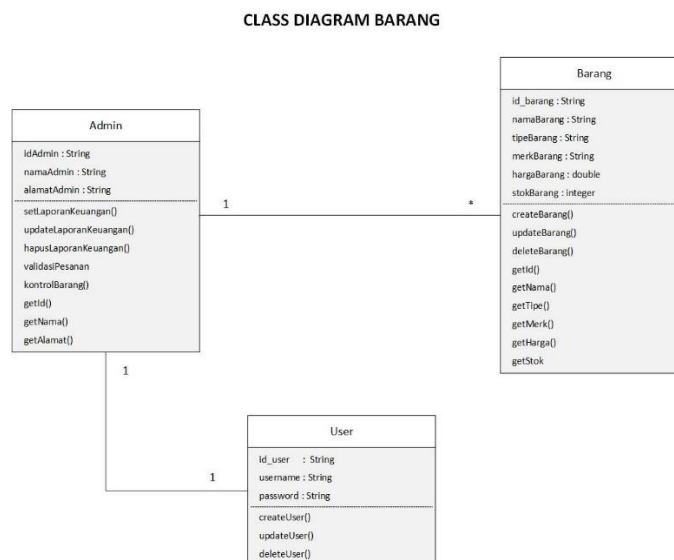
No	Nama Kelas Perancangan	Tipe Kelas
1	User	Entity
2	Admin	Entity
3	Barang	Entity
4	Database	Entity

### 3.1.15.2 Sequence Diagram Hapus Barang



Gambar 2.15.1 Sequence Diagram Hapus Barang

### 3.1.15.3 Diagram Kelas Hapus Barang



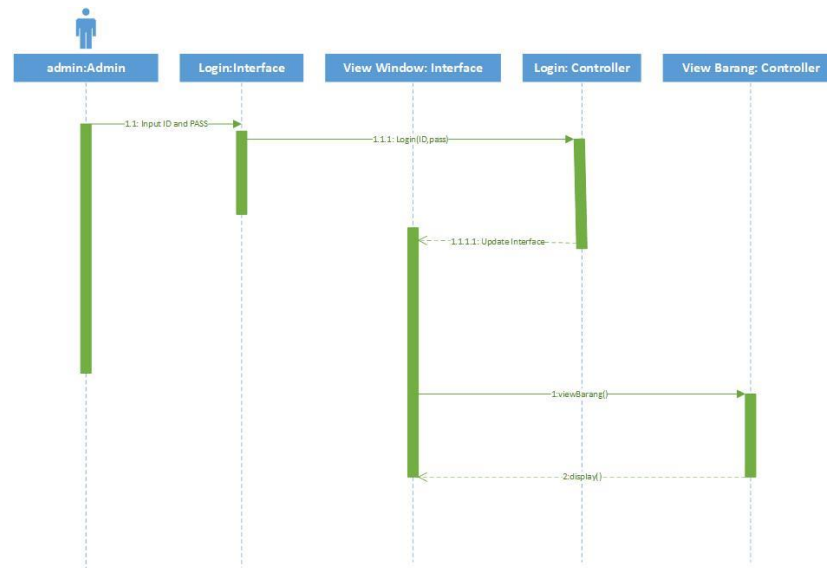
Gambar 2.15.2. Class Diagram Hapus Barang

### 3.1.16 Use Case View Barang

#### 3.1.16.1 Identifikasi Kelas View Barang

No	Nama Kelas Perancangan	Tipe Kelas
1	User	Entity
2	Admin	Entity
3	Barang	Entity
4	Database	Entity

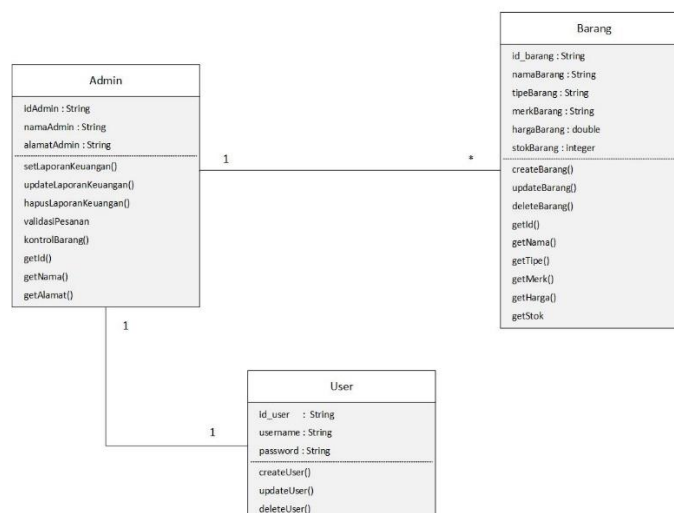
### 3.1.16.2 Sequence Diagram View Barang



Gambar 2.16.1 Sequence Diagram View Barang

### 3.1.16.3 Diagram Kelas Hapus Barang

CLASS DIAGRAM BARANG



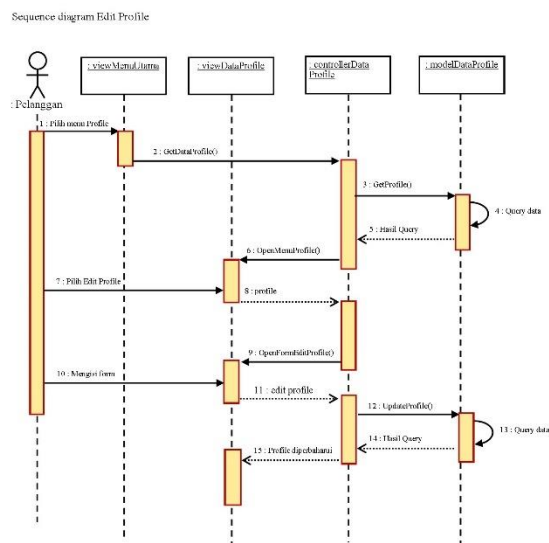
Gambar 2.16.2. Class Diagram View Barang

### 3.1.17 Use Case Edit Edit Profile

#### 3.1.17.1 Identifikasi Kelas Edit Profile

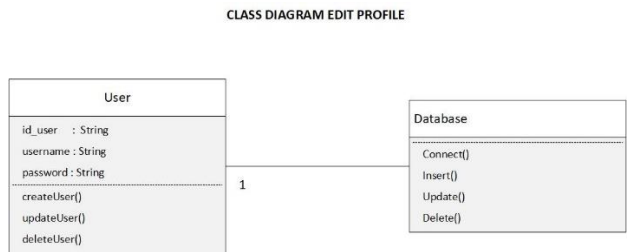
No	Nama Kelas Perancangan	Tipe Kelas
1	User	Entity
2	Database	Entity

3.1.17.2 Sequence Diagram Edit Profile



Gambar 2.17.1 Sequence Diagram Edit Profile

3.1.17.3 Diagram Kelas Edit Profile



Gambar 2.10.2. Class Diagram Edit Profile

3.2 Perancangan Detil Kelas

No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1	Barang	Barang
2	Pesanan	Pesanan
3	Pelanggan	Pelanggan
4	User	User
5	Admin	Admin
6	Laporan Keuangan	Laporan Keuangan
7	Manajer	Manajer

8	Konsultansi	Konsultansi
9	Konsultan	Konsultan

### 3.2.1 Kelas Barang

Nama Kelas : Barang

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
createBarang()	public	Operasi ini adalah operasi membuat objek Barang dan memasukkan semua keperlunya.
updateBarang()	public	Operasi yang dilakukan adalah mengupdate data baru
deleteBarang()	public	Operasi yang digunakan untuk menghapus barang.
getId()	public	Untuk mendapatkan id barang
getNama()	public	Untuk mendapatkan nama barang
getTipe()	public	Untuk mendapatkan tipe barang
getMerk()	public	Untuk mendapatkan merk barang
getHarga()	public	Untuk mendapatkan harga barang
getStok()	public	Untuk mendapatkan
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
Id_barang	private	String
namaBarang	private	String
tipeBarang	private	String
merkBarang	private	String
hargaBarang	private	Double
stokBarang	private	Integer

### 3.2.2 Kelas Pelanggan

Nama Kelas : Pelanggan

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
viewBarang()	public	Operasi digunakan melihat barang yang tersedia di Website
buatPesanan()	public	Operasi yang digunakan untuk membuat pesanan.
kontrolPesanan	public	Digunakan Untuk CRUD Pesanan
getId()	public	Untuk mendapatkan Id Pelanggan
getNama()	public	Untuk mendapatkan Nama Pelanggan
getAlamat()	public	Untuk mendapatkan Alamat Pelanggan
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
Id_pelanggan	private	String
namaPelanggan	private	String
alamatPelanggan	private	String

### 3.2.3 Kelas Konsultasi

Nama Kelas : Konsultasi

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
createKonsultasi()	public	Operasi digunakan untuk membuat Konsultasi
deleteKonsultasi()	public	Operasi digunakan untuk menghapus data konsultasi.
updateKonsultasi()	public	Digunakan Untuk Update Konsultasi
getId()	public	Untuk mendapatkan Id Konsultasi
getDataKonsultasi	public	Untuk mendapatkan data konsultasi
getDataPelanggan	public	Digunakan untuk mendapatkan Pelanggan siapa yang sedang melakukan konsultasi
getDataKonsultan	Public	Untuk mendapatkan Data dari Konsultan yang sedang menangani Konsultasi.
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
Id_konsultasi	private	String
dataKonsultasi	private	Konsultasi
dataPelanggan	private	Pelanggan
dataKonsultan	private	Konsultan

### 3.2.4 Kelas Konsultan

Nama Kelas : Konsultan

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
viewKonsultasi()	Public	Operasi yang digunakan untuk melihat konsultasi
createKonsultasi()	public	Operasi digunakan untuk membuat Konsultasi
viewBarang()	public	Operasi digunakan melihat barang yang tersedia di Website
getId()	public	Untuk mendapatkan Id Konsultan
getNama()	public	Untuk mendapatkan Nama Konsultan
getAlamat()	public	Digunakan untuk mendapatkan Alamat dari seorang Konsultan
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
Id_konsultan	private	String
namaKonsultan	private	String
alamatKonsultan	private	String

### 3.2.5 Kelas User

Nama Kelas : User

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
---------------------	---	-------------------

createUser()	Public	Digunakan untuk membuat user
updateUser()	public	Digunakan untuk mengupdate atribut user
deleteUser()	public	Digunakan untuk menghapus user dari sistem
<b><i>Nama Atribut</i></b>	<b><i>Visibility (private, public)</i></b>	<b><i>Tipe</i></b>
Id_user	private	String
Username	private	String
Password	private	String

### 3.2.6 Kelas Admin

Nama Kelas : Admin

<b><i>Nama Operasi</i></b>	<b><i>Visibility (private, public)</i></b>	<b><i>Keterangan</i></b>
setLaporanKeuangan()	Public	Digunakan untuk membuat user
updateLaporanKeuangan()	public	Digunakan untuk mengupdate atribut user
hapusLaporanKeuangan()	public	Digunakan untuk menghapus user dari sistem
validasiPesanan()	public	Digunakan untuk memvalidasi pesanan yang dilakukan oleh pelanggan
kontrolBarang()	public	Digunakan untuk mengontrol barang (CRUD)
getId()	public	Digunakan untuk mendapatkan id konsultan
getNama()	public	Digunakan untuk mendapatkan nama admin
getAlamat()	public	Digunakan untuk mendapatkan alamat admin
<b><i>Nama Atribut</i></b>	<b><i>Visibility (private, public)</i></b>	<b><i>Tipe</i></b>
Id_admin	private	String
Nama_admin	private	String
alamatAdmin	private	String

### 3.2.7 Kelas Laporan Keuangan

Nama Kelas : Laporan Keuangan

<b><i>Nama Operasi</i></b>	<b><i>Visibility (private, public)</i></b>	<b><i>Keterangan</i></b>
createLaporanKeuangan()	Public	Digunakan untuk membuat laporan keuangan baru
viewLaporanKeuangan()	public	Digunakan melihat laporan keuangan
editLaporanKeuangan()	public	Digunakan untuk mengubah isi dari laporan keuangan

hapusLaporanKeuangan()	public	Digunakan untuk menghapus Laporan Keuangan user dari sistem
getId()	public	Digunakan untuk mendapatkan id laporan
getTanggal()	public	Digunakan untuk mendapatkan tanggal pembuatan laporan keuangan
getData()	public	Digunakan data-data laporan keuangan
getListPesanan()	public	Digunakan untuk mendapatkan transaksi yang telah terjadi dalam periode laporan keuangan
<b><i>Nama Atribut</i></b>	<b><i>Visibility (private, public)</i></b>	<b><i>Tipe</i></b>
Id_laporan	private	String
tglLaporan	private	String
dataLaporan	private	String
listPesanan	private	Pesanan

### 3.2.8 Kelas Manajer

Nama Kelas : Manajer

<b><i>Nama Operasi</i></b>	<b><i>Visibility (private, public)</i></b>	<b><i>Keterangan</i></b>
viewLaporanKeuangan()	public	Digunakan melihat laporan keuangan
editLaporanKeuangan()	public	Digunakan untuk mengubah isi dari laporan keuangan
hapusLaporanKeuangan()	public	Digunakan untuk menghapus user dari sistem
getId()	public	Digunakan untuk mendapatkan id manajer
getNama()	public	Digunakan untuk mendapatkan nama manajer
getAlamat()	public	Digunakan untuk mendapatkan alamat manajer
<b><i>Nama Atribut</i></b>	<b><i>Visibility (private, public)</i></b>	<b><i>Tipe</i></b>
Id_manager	private	String
namaManager	private	String
alamatManager	private	String

### 3.2.9 Kelas Pesanan

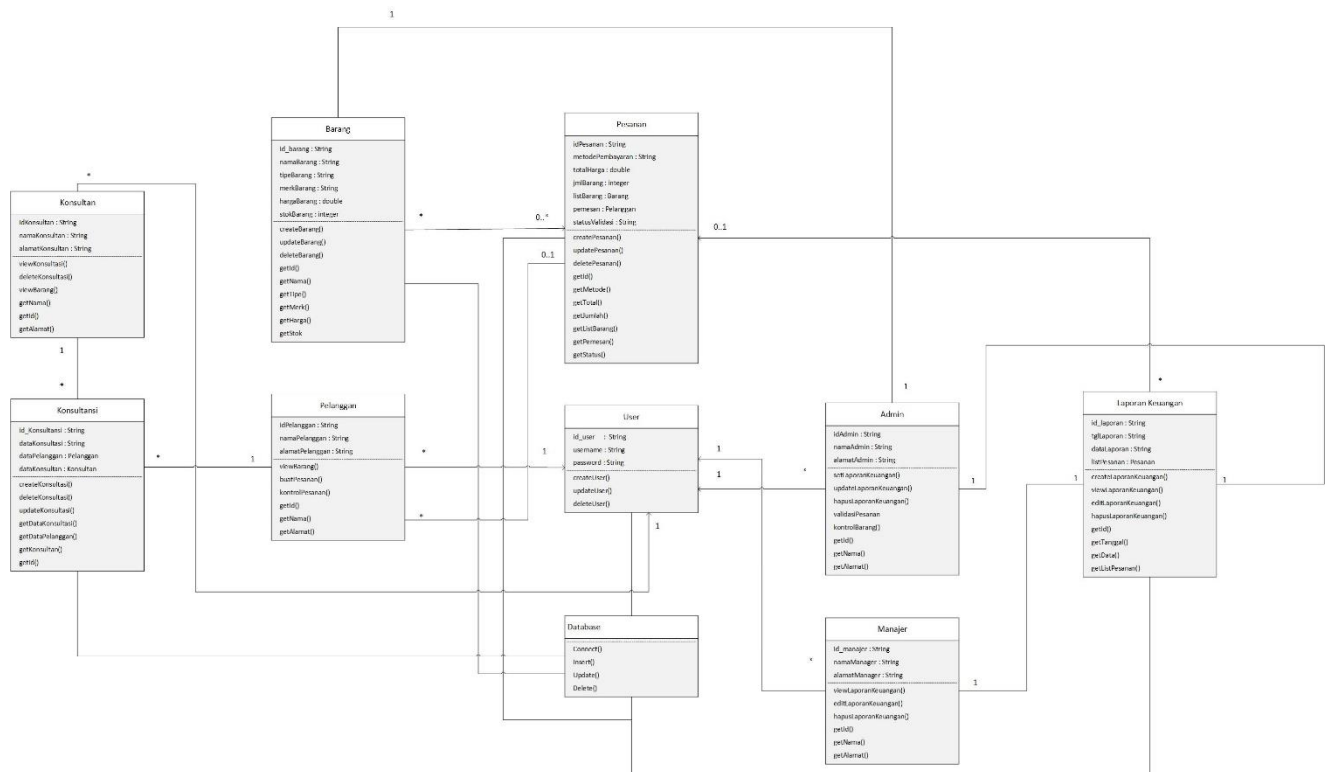
Nama Kelas : Pesanan

<b><i>Nama Operasi</i></b>	<b><i>Visibility (private, public)</i></b>	<b><i>Keterangan</i></b>
createPesanan()	public	Digunakan untuk membuat pesanan
updatePesanan()	public	Digunakan untuk mengubah isi pesanan



deletePesanan()	public	Digunakan untuk menghapus pesanan
getId()	public	Digunakan untuk mendapatkan id pesanan
getMetode()	public	Digunakan untuk mendapatkan metode pembayaran
getTotal()	public	Digunakan untuk mendapatkan total harga
getJumlah()	public	Digunakan untuk mendapatkan jumlah barang
getListBarang()	public	Digunakan untuk mendapatkan barang apa saja yang dimasukkan ke pesanan
getPemesan()	public	Digunakan untuk mendapatkan pemesan
getStatus()	public	Digunakan untuk mendapatkan status validasi
<b>Nama Atribut</b>	<b>Visibility (private, public)</b>	<b>Tipe</b>
Id_pesanan	private	String
metodePembayaran	private	String
totalHarga	private	double
jmlBarang	private	Integer
listBarang	private	Barang
pemesan	private	Pelanggan
statusValidasi	private	String

### 3.3 Diagram Kelas Keseluruhan



### 3.4 Algoritma/Query

Nama Kelas : Barang  
Nama Operasi : **Input Barang , Update Barang, Delete Barang, viewBarang**  
Algoritma : **inputBarang(), updateBarang(), deleteBarang(), viewBarangAll(), viewOneBarang();**

```
Function inputBarang(){
    Id_barang = unique_random
    Nama = input
    Tipe =input
    Merk = input
    Stok = input
    Harga = input
    If(input != null){
        If( not(Select * From barang where namaBarang = 'Nama' and Merk = 'Merk')){
            Data= array(
                Isi dengan spesifikasi tabel
            )
            (Insert into barang values(nama,tipe,merk,stok,harga)
            Redirect('Halaman Barang')
        }else {
            Output('Data Sudah Ada')
        }
    }else {
        Output('Data Tidak Boleh Kosong')
    }
}

Function updateBarang(idBarang : String){
    Id_barang = unique_random
    Nama = input
```

```

Tipe =input
Merk = input
Stok = input
Harga = input
If(input != null){
    If((Select * From barang where      namaBarang = 'Nama' and Merk = 'Merk')){
        Data= array(
            Isi dengan spesifikasi tabel
        )
        (UPDATE 'barang' SET field1 = new-value1, field2 = new-value2 [WHERE Clause])
        Redirect('Halaman Barang')
    }else {
        Output('Data Tidak Ada')
    }
}else {
    Output('Data Tidak Boleh Kosong')
}
}

Function deleteBarang(idBarang: String){
    If((Select * From barang where      namaBarang = 'Nama' and Merk = 'Merk')){
        (DELETE FROM `barang` [WHERE condition];
        Redirect('Halaman Barang'))
    }Else {
        Output('Data Tidak Ada')
    }
}

Function viewBarangAll(){
    Data = (Select Field 1, Field 2, ... From barang);
    Output(data);
}

Function viewOneBarang(idBarang = String){
    Data = (Select Field 1, Field 2, ... From barang where idbarang = 'idBarang');
    Output(data);
}

```

*Query* : Digunakan untuk methods pada Class Barang, yaitu CRUD data ke database

No Query	Query	Keterangan
Q-001	INSERT INTO barang VALUES (value_1,value_2,...);	Digunakan untuk memasukkan data kedalam database tabel barang/
Q-002	SELECT Field 1, Field 2, Field 3, FROM barang ;	Digunakan untuk mendapatkan Field tertentu dari database barang
Q-003	UPDATE 'barang' SET field1 = new-value1, field2 = new-value2 [WHERE Clause]	Digunakan untuk mengupdate data barang
Q-004	DELETE FROM `barang` [WHERE condition];	Digunakan untuk menghapus data dari tabel barang
Q-005	SELECT Field 1, Field 2, Field 3, ....	Digunakan untuk mengambil data tertentu dengan penkondisian tertentu pada tabel barang

	FROM barang where Field n = 'Kondisi';	
--	---	--

Nama Kelas : Pesanan  
 Nama Operasi : **Input Pesanan , Update Pesnana, Delete Pesanan viewPesanan**  
 Algoritma : **inputPesanan(), updatePesanan(), deletePesanan(), viewPesananAll(), viewOnePesan(), validasi()**

```

Function inputPesanan(){
    idPesanan= unique_random
    metodePembayaran = input
    totalHarga =input
    listBarang = input
    pemesan = input
    statusValidasi = false
    If(input != null){
        If( not(query checkin )){
            Data= array(
                Isi dengan spesifikasi tabel
            )
            (Insert into pesanan values(isi dengan array data)
            Redirect('Halaman Pesanan')
        }else {
            Output('Data Sudah Ada')
        }else {
            Output('Data Tidak Boleh Kosong')
        }
    }

Function updatePesanan(idPesanan : String){
    idPesanan= unique_random
  
```

```

    metodePembayaran = input
    totalHarga =input
    listBarang = input
    pemesan = input
    statusValidasi = false
    If(input != null){
        If((query checkin)){
            Data= array(
                Isi dengan spesifikasi tabel
            )
            (UPDATE 'pesanan' SET field1 = new-value1, field2 = new-value2 where)
            Redirect('Halaman Pesanan')
        }
        }else {
            Output('Data Tidak Ada')
        }
    }else {
        Output('Data Tidak Boleh Kosong')
    }
}

Function deletePesanna(idPesanan: String){
    If((query checkin)){
        (DELETE FROM `pesanan` [WHERE condition];
        Redirect('Halaman Pesanan')
    }Else {
        Output('Data Tidak Ada')
    }
}

Function viewPesananAll(){
    Data = (Select Field 1, Field 2, ... From pesanan);
    Output(data);
}

Function viewOnPesanan(idPesanan = String){
    Data = (Select Field 1, Field 2, ... From pesanan where idPesanan = 'idBarang');
    Output(data);
}

Function Validasi(idPesanan : String){
    Status = True
    If((query checkin)){
        Data= array(
            Isi dengan spesifikasi tabel
        )
        (UPDATE 'pesanan' SET field1 = new-value1, field2 = new-value2 where)
        Redirect('Halaman Pesanan')
    }else {
        Output('Data Tidak Ada')
    }
}
}

```

*Query* : Digunakan untuk methods pada Class Pesanan, yaitu CRUD data ke database

No Query	Query	Keterangan
----------	-------	------------

Q-006	INSERT INTO pesanan VALUES (value_1,value_2,...);	Digunakan untuk memasukkan data kedalam database tabel pesanan,.
Q-007	SELECT Field 1, Field 2, Field 3, .... FROM pesanan;	Digunakan untuk mendapatkan Field tertentu dari database pesanan
Q-008	SELECT Field 1, Field 2, Field 3, .... FROM pesanan where Field n = 'Kondisi';	Digunakan untuk mengambil data tertentu dengan penkondisian tertentu pada tabel pesanan
Q-009	UPDATE 'pesanan' SET field1 = new-value1, field2 = new-value2 [WHERE Clause]	Digunakan untuk mengupdate data pesanan
Q-010	DELETE FROM `pesanan` [WHERE condition];	Digunakan untuk menghapus data dari tabel pesanan

Nama Kelas : Pelanggan  
Nama Operasi : **Register, Login, editProfile.**  
Algoritma : **register(), login(), editProfile().**

```

Function register(){
    idPelanggan = unique_random
    namaPelanggan = input
    alamatPelanggan = input
    id_user = unique_random
    username = input
    password = input
    If(input != null){
        If( not((Select * from pelanggan where namaPelanggan = 'namaPelanggan' ) and (Select * from
user where username = 'username'))){
            DataUser= array(
                Isi dengan spesifikasi table user
            )
            (Insert into user values(isi dengan array dataUser)
            DataPelanggan= array(
                Isi dengan spesifikasi table pelanggan
            )
            (Insert into pelanggan values(isi dengan array DataPelanggan)
            Redirect('Halaman Pelanggan')
        } else {
            Output('Data Sudah Ada')
        }
    } else {

```

```

        Output('Data Tidak Boleh Kosong')
    }
}

Function login(){
    Username = input;
    Password = input
    If(input != null){
        If(select * from user where username = 'username' and password = 'password'){
            Session = (select * from user where username = 'username')
            Redirect('Halaman Pelanggan')
        }else {
            Output('Data Tidak Ditemukan')
        }
    }else {
        Output('Tidak Boleh Dikosongkan')
    }
}

Function editProfile(){
    idPelanggan = unique_random
    namaPelanggan = input
    alamatPelanggan = input
    id_user = unique_random
    username = input
    password = input
    If(input != null){
        If( not((Select * from pelanggan where namaPelanggan = 'namaPelanggan' ) and (Select * from
user where username = 'username'))){
            DataUser= array(
                Isi dengan spesifikasi table user
            )
            If ( not(query get data dengan data diri baru)){
                (update user set(isi dengan array dataUser)
                DataPelanggan= array(
                    Isi dengan spesifikasi table pelanggan
                )
                (update pelanggan set(isi dengan array DataPelanggan)
                Redirect('Halaman Pelanggan')
            }else {
                Output('Data Sudah Ada')
            }else {
                Output('Data Milik Orang Lain')
            }
        }else {
            Output('Data tidak Boleh Kosong')
        }
    }
}

```

*Query* : Digunakan untuk methods pada Class Pelanggan, yaitu CRUD data ke database

No Query	Query	Keterangan
Q-011	INSERT INTO user VALUES (value_1,value_2,...);	Digunakan untuk memasukkan data kedalam database tabel user.
Q-012	INSERT INTO pelanggan VALUES (value_1,value_2,...);	Digunakan untuk memasukkan data kedalam database tabel pelanggan.

Q-013	SELECT Field 1, Field 2, Field 3, .... FROM pesanan where Field n = 'Kondisi';	Digunakan untuk mengambil data tertentu dengan penkondisian tertentu pada tabel pesanan
Q-014	SELECT Field 1, Field 2, Field 3, .... FROM user;	Digunakan untuk mengambil data tertentu dengan penkondisian tertentu pada tabel user
Q-015	UPDATE 'pelanggan' SET field1 = new-value1, field2 = new- value2 [WHERE Clause]	Digunakan untuk mengupdate data pelanggan

Nama Kelas : Konsultan  
Nama Operasi : **Login, Edit Profile**  
Algoritma : **login() , editProfile()**

```

Function login(){
    Username = input;
    Password = input
    If(input != null){
        If((select * from konsultan where username = 'username' and password = 'password')&&
(select * from user where username = 'username' and password = 'password')){
            Session = (select * from user where username = 'username')
            Redirect('Halaman Konsultan')
        }else {
            Output('Data Tidak Ditemukan')
        }
    }else {
        Output('Tidak Boleh Dikосongkan')
    }
}

Function editProfile(){
    idKonsultan = unique_random
    namaKonsultan = input
    alamatKonsultan = input
    username = input
    password = input
    If(input != null){
        If( not((Select * from konsultan where namaKonsultan = 'namakonsultan' ) and (Select * from
user where username = 'username'))){
            DataUser= array(
                Isi dengan spesifikasi table user
            )
            If ( not(query get data dengan data diri baru)){
                (update user set(isi dengan array dataUser)
                DataAdmin= array(
                    Isi dengan spesifikasi table pelanggan
                )
                (update admin set(isi dengan array DataPelanggan)
                Redirect('Halaman Konsultan')
            }else {
                Output('Data Sudah Ada')
            }else {
                Output('Data Milik Orang Lain')
            }
        }else {
            Output('Data tidak Boleh Kosong')
        }
    }
}

```



```

    }
}

```

*Query* : Digunakan untuk methods pada Class Barang, yaitu CRUD data ke database

No Query	Query	Keterangan
Q-016	SELECT Field 1, Field 2, Field 3, .... FROM user where Field n = 'Kondisi';	Digunakan untuk mengambil data tertentu dengan penkondisian tertentu pada tabel user
Q-017	SELECT Field 1, Field 2, Field 3, .... FROM konsultan;	Digunakan untuk mendapatkan Field tertentu dari database konsultan
Q-018	SELECT Field 1, Field 2, Field 3, .... FROM konsultan where Field n = 'Kondisi';	Digunakan untuk mengambil data tertentu dengan penkondisian tertentu pada tabel konsultan
Q-019	Q-011	SELECT Field 1, Field 2, Field 3, .... FROM user ;

Nama Kelas : Admin  
 Nama Operasi : **Login,Edit Profile,insertKonsultan, deleteKonsultan**  
 Algoritma : **login() , editProfile(), insertKonsultan(), deleteKonsultan()**

```

Function login(){
    Username = input;
    Password = input
    If(input != null){
        If((select * from admin where username = 'username' and password = 'password') && (select
* from user where username = 'username' and password='password')){
            Session = (select * from admin where username = 'username')
            Redirect('Halaman Admin')
        }else {
            Output('Data Tidak Ditemukan')
        }
    }else {
        Output('Tidak Boleh Dikosongkan')
    }
}

Function editProfile(){
    idAdmin = unique_random
    namaAdmin = input
    alamatAdmin = input
    id_admin = unique_random
    username = input
    password = input
    If(input != null){
        If( not((Select * from admin where namaAdmin = 'namaAdmin' ) and (Select * from
admin where username = 'username'))){

```

```

        DataUr= array(
            Isi dengan spesifikasi table user
        )
        If ( not(query get data dengan data diri baru)){
            (update user set(isi dengan array dataUser)
            DataAdmin= array(
                Isi dengan spesifikasi table pelanggan
            )
            (update admin set(isi dengan array DataPelanggan)
            Redirect('Halaman Admin')
        }else {
            Output('Data Sudah Ada')
        }else {
            Output('Data Milik Orang Lain')
        }else {
            Output('Data tidak Boleh Kosong')
        }
    }

function insertKonsultan(){
    idKonsultan = unique_random
    namaKonsultan = input
    alamatKonsultan = input
    username = input
    password = input
    If(input != null){
        If( not((Select * from Konsultan where namaKonsultan = 'namaKonsultan' ) and (Select * from
konsultan where username = 'username'))){
            DataUser= array(
                Isi dengan spesifikasi table user
            )
            (Insert into user values(isi dengan array dataUser)
            DataKonsultan= array(
                Isi dengan spesifikasi table konsultan
            )

            (Insert into konsultan values(isi dengan array DataKonsultan)
            Redirect('Halaman Admin')
        }else {
            Output('Data Sudah Ada')
        }else {
            Output('Data Tidak Boleh Kosong')
        }
    }
}

function DeleteKonsultan(){
    idKonsultan = unique_random
    namaKonsultan = input
    alamatKonsultan = input
    username = input
    password = input
    If(input != null){
        dataKonsultan =(select * from konsultan where idKonsultan = 'idKonsultan')
        (delete From Konsultan where( isi dari input disini))
        Redirect('Halaman Admin');
    }Else {
        Output('Data Tidak ditemukan');
    }
}

```

```

    }

function insertManajer(){
    idManajer = unique_random
    namaManajer = input
    alamatManajer = input
    username = input
    password = input
    If(input != null){
        If( not((Select * from Manajer where namaManajer = 'namaManajer' ) and (Select * from
user where username = 'username'))){
            DataUser= array(
                Isi dengan spesifikasi table user
            )
            (Insert into user values(isi dengan array dataUser)
            DataManajer= array(
                Isi dengan spesifikasi table manajer
            )

            (Insert into manajer values(isi dengan array DataManajer)
            Redirect('Halaman Admin')
        }else {
            Output('Data Sudah Ada')
        }else {
            Output('Data Tidak Boleh Kosong')
        }
    }
}

function DeleteManajer(){
    idManajer= unique_random
    namaManajer = input
    alamatManajer= input
    username = input
    password = input
    If(input != null){
        dataManajer =(select * from Manajer where idManajer = 'idManajer')
        (delete From Manajer where( isi dari input disini))
        Redirect('Halaman Admin');
    }Else {
        Output('Data Tidak ditemukan');
    }
}

```

*Query* : Digunakan untuk methods pada Class Barang, yaitu CRUD data ke database

No Query	Query	Keterangan
Q-009	INSERT INTO admin VALUES (value_1,value_2,...);	Digunakan untuk memasukkan data kedalam database tabel admin.
Q-020	SELECT Field 1, Field 2, Field 3, FROM admin ;	Digunakan untuk mendapatkan Field tertentu dari database admin
Q-021	SELECT Field 1, Field 2, Field 3, FROM admin where Field n = 'Kondisi';	Digunakan untuk mengambil data tertentu dengan penkondisian tertentu pada tabel admin
Q-022	DELETE FROM `admin` [WHERE condition];	Digunakan untuk menghapus data dari tabel admin
Q-023	UPDATE 'admin' SET field1 = new-value1, field2 = new-value2 [WHERE Clause]	Digunakan untuk mengupdate data admin
Q-024	DELETE FROM `konsultan` [WHERE condition];	Digunakan untuk menghapus data dari tabel konsultan
Q-025	INSERT INTO konsultan VALUES (value_1,value_2,...);	Digunakan untuk memasukkan data kedalam database tabel konsultan.
Q-026	INSERT INTO Manajer VALUES (value_1,value_2,...);	Digunakan untuk memasukkan data kedalam database tabel Manajer.
Q-027	DELETE FROM `Manajer` [WHERE condition];	Digunakan untuk menghapus data dari tabel manajer

Nama Kelas : Manajer  
Nama Operasi : **Login,Edit Profile**  
Algoritma : **login() , editProfile()**

```

Function login(){
    Username = input;
    Password = input
    If(input != null){
        If((select * from Manajer where username ='username' and password='password')&&(select
* from user where username = 'username' and password = 'password')){
            Session = (select * from user where username = 'username')
            Redirect('Halaman Manajer )
        }else {
            Output('Data Tidak Ditemukan')
        }
    }else {
        Output('Tidak Boleh Dikosongkan')
    }
}

Function editProfile(){
    idManajer = unique_random
    namaManajer = input
    alamatManajer = input
    username = input
    password = input
    If(input != null){
        If( not((Select * from manajer where namaManajer = 'namaManajer' ) and (Select * from
user where username = 'username'))){

```

```

        DataManajer= array(
            Isi dengan spesifikasi table user
        )
        If ( not(query get data dengan data diri baru)){
            (update user set(isi dengan array dataUser)
            DataManajer= array(
                Isi dengan spesifikasi table Manajer
            )
            (update admin set(isi dengan array DataManajer)
            Redirect('Halaman Manajer')
        }else {
            Output('Data Sudah Ada')
        }else {
            Output('Data Milik Orang Lain')
        }else {
            Output('Data tidak Boleh Kosong')
        }
    }
}

```

*Query* : Digunakan untuk methods pada Class Barang, yaitu CRUD data ke database

No Query	Query	Keterangan
Q-016	SELECT Field 1, Field 2, Field 3, .... FROM user where Field n = 'Kondisi';	Digunakan untuk mengambil data tertentu dengan penkondisian tertentu pada tabel user
Q-017	SELECT Field 1, Field 2, Field 3, .... FROM Manajer;	Digunakan untuk mendapatkan Field tertentu dari database Manajer
Q-018	SELECT Field 1, Field 2, Field 3, .... FROM Manajer where Field n = 'Kondisi';	Digunakan untuk mengambil data tertentu dengan penkondisian tertentu pada tabel Manajer
Q-019	SELECT Field 1, Field 2, Field 3, .... FROM User;	Digunakan untuk mengambil data dari user

Nama Kelas : Konsultasi

Nama Operasi : **Create Konsultasi, Update Konsultasi, Delete Konsultasi, View Konsultasi**

Algoritma : **CreateKonsultasi(), UpdateKonsultasi(), DeleteKonsultasi(), ViewKonsultasi(), ViewOnKonsultasi()**

```

Function createKonsultasi{
    idKonsultasi : random_unique
    idPelanggan : Input
    IdKonsultan : Input
    DataKonsultasi : Input
    If(input != null){
        DataKonsultasi = array(Isi dengan semua data yang diperlukan dan didapatkan dari input)
    }
}

```

```

        (insert into konsultasi value(DataKonsultasi)
        Redirect(Halaman Konsultasi)
    }else {
        Output(Data Tidak Lengkap)
    }
}

Function updateKonsultasi(){
    idKonsultasi : input
    idPelanggan : Input
    IdKonsultan : Input
    DataKonsultasi : Input
    If((input != null) && (select * from konsultasi where idKonsultasi = 'idKonsultasi')){
        DataKonsultasi = array(Isi dengan semua data yang diperlukan dan didapatkan dari input)
        (update konsultasi set(DataKonsultasi)
        Redirect(Halaman Konsultasi)
    }else {
        Output(Data Tidak Lengkap)
    }
}

Function DeleteKonsultasi(){
    idKonsultasi : input
    If((input != null) && (select * from konsultasi where idKonsultasi = 'idKonsultasi')){
        Delete form Kosultasi where idKonsultasi = 'idKonsultasi';
        Redirect('Halaman Konsultasi');
    }else{
        Output('Data Tidak Ada');
    }
}

Function ViewKonsultasi(){
    dataKonsultasi = (select * from Konsultasi)
    output(dataKonsultasi);
}

Function ViewOnKonsultasi(){
    id = input
    dataKonsultasi = (select * from Konsultasi where IdKonsultasi ='id')
    output(dataKonsultasi);
}

```

*Query* : Digunakan untuk methods pada Class Barang, yaitu CRUD data ke database

No Query	Query	Keterangan
Q-020	INSERT INTO konsultasi VALUES (value_1,value_2,...);	Digunakan untuk memasukkan data kedalam database tabel konsultasi.
Q-021	SELECT Field 1, Field 2, Field 3, .... FROM konsultasi ;	Digunakan untuk mendapatkan Field tertentu dari database konsultasi

Q-022	SELECT Field 1, Field 2, Field 3, .... FROM konsultasi where Field n = 'Kondisi';	Digunakan untuk mengambil data tertentu dengan penkondisian tertentu pada tabel konsultasi
Q-023	DELETE FROM `konsultasi` [WHERE condition];	Digunakan untuk menghapus data dari tabel konsultasi
Q-024	UPDATE 'konsultasi' SET field1 = new-value1, field2 = new- value2 [WHERE Clause]	Digunakan untuk mengupdate data konsultasi

Nama Kelas : Laporan Keuangan

Nama Operasi : **Create Laporan, View Laporan, Edit Laporan , Delete Laporan**

Algoritma : **CreateLaporan(), ViewLaporan(), EditLaporan(), deleteLaporan(), ViewOnLaporan()**

```

Function createLaporan(){
    dataLaporan = input
    If(input != null){
        (insert into Laporan value(dataLaporan)
        Redirect(Halaman Laporan)
    }else {
        Output(Data Tidak Lengkap)
    }
}

Function updateLaporan(){
    dataLaporan = input
    If((input != null) && (select * from Laporan where idLaporan = 'dataLaporan[idLaporan]')){
        (update Laporan set(dataLaporan)
        Redirect(Halaman Laporan)
    }else {
        Output(Data Tidak Lengkap)
    }
}

Function DeleteLaporan(){
    idLaporan = input
    If((input != null) && (select * from Laporan where idLaporan = 'idLaporan')){
        Delete form Laporan where idLaporan = 'idLaporan';
        Redirect('Halaman Konsultasi');
    }else{
        Output('Data Tidak Ada');
    }
}

Function ViewLaporan(){
    dataLaporan = (select * from Laporan)
    output(Laporan);
}

Function ViewOnLaporan(){
    id = input
    dataLaporan = (select * from Laporan where idLaporan ='id')
    output(Laporan);
}

```

}

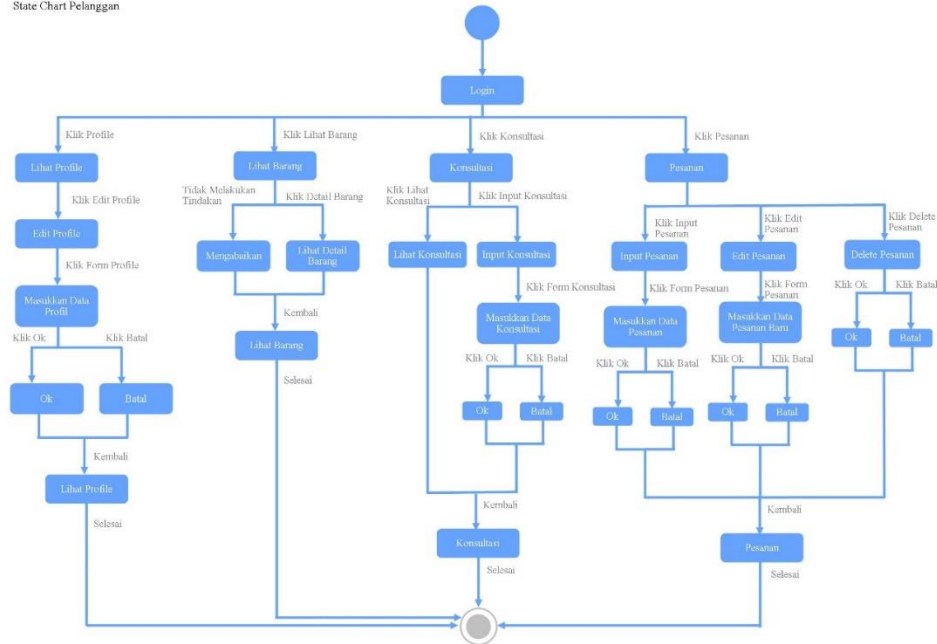
*Query* : Digunakan untuk methods pada Class Barang, yaitu CRUD data ke database

No Query	Query	Keterangan
Q-025	INSERT INTO laporankeuangan VALUES (value_1,value_2,...);	Digunakan untuk memasukkan data kedalam database tabel laporankeuangan.
Q-026	SELECT Field 1, Field 2, Field 3, FROM laporankeuangan;	Digunakan untuk mendapatkan Field tertentu dari database laporankeuangan
Q-027	SELECT Field 1, Field 2, Field 3, FROM laporankeuangan where Field n = 'Kondisi';	Digunakan untuk mengambil data tertentu dengan penkondisian tertentu pada tabel laporan keuangan pada tabel laporankeuangan
Q-028	DELETE FROM 'laporankeuangan' [WHERE condition];	Digunakan untuk menghapus data dari tabel laporankeuangan
Q-029	UPDATE 'laporankeuangan' SET field1 = new-value1, field2 = new- value2 [WHERE Clause]	Digunakan untuk mengupdate data laporankeuangan

### 3.5 Statechart Diagram

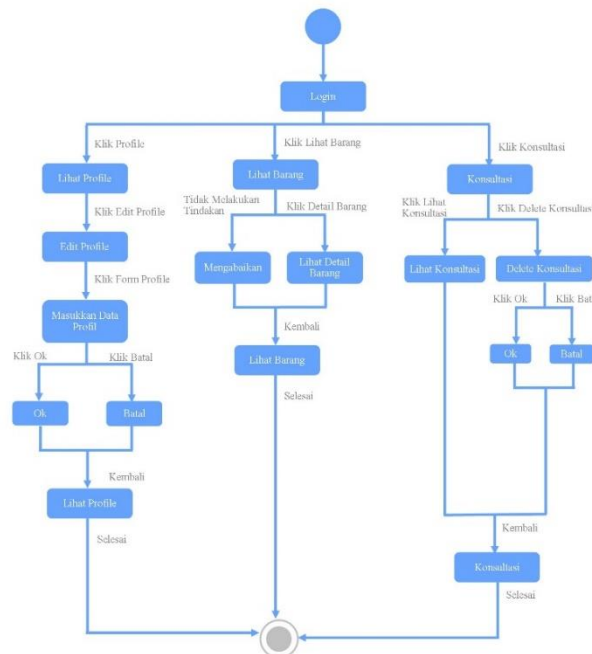


State Chart Pelanggan



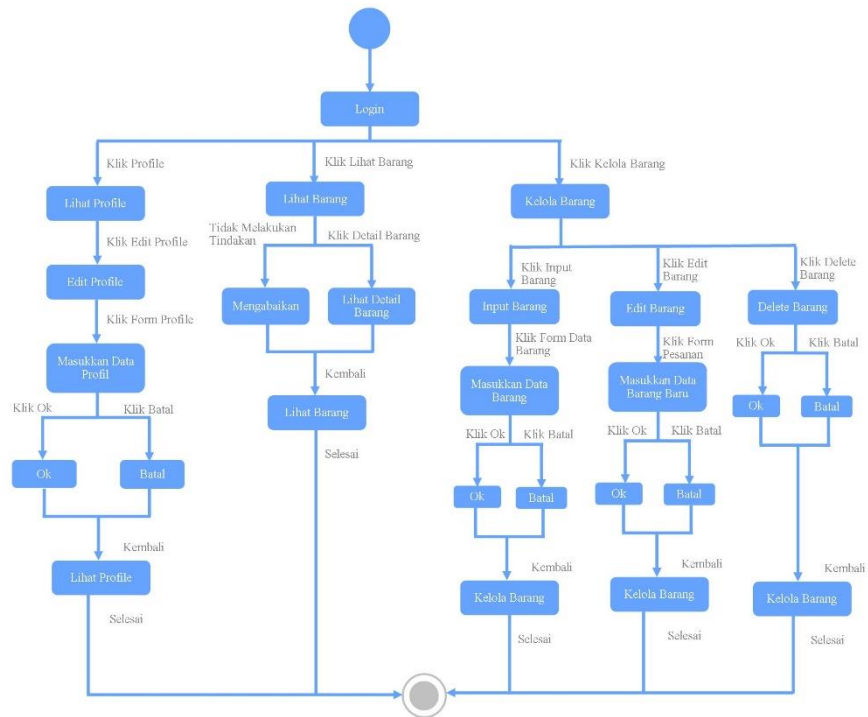
Gambar 4 Statechart Pelanggan

State Chart Konsultan



Gambar 5 Statechart Konsultan

State Chart Admin



Gambar 6 Statechart Admin

State Chart Manager



Gambar 7 Statechart Manager

### 3.6 Perancangan Antarmuka

LOGO Halaman Edit Profile

LOGO SELAMAT DATANG DI RAKIT-KOMPUTER Akun

DATA DIRI

FORM DATA DIRI

BACK SUBMIT

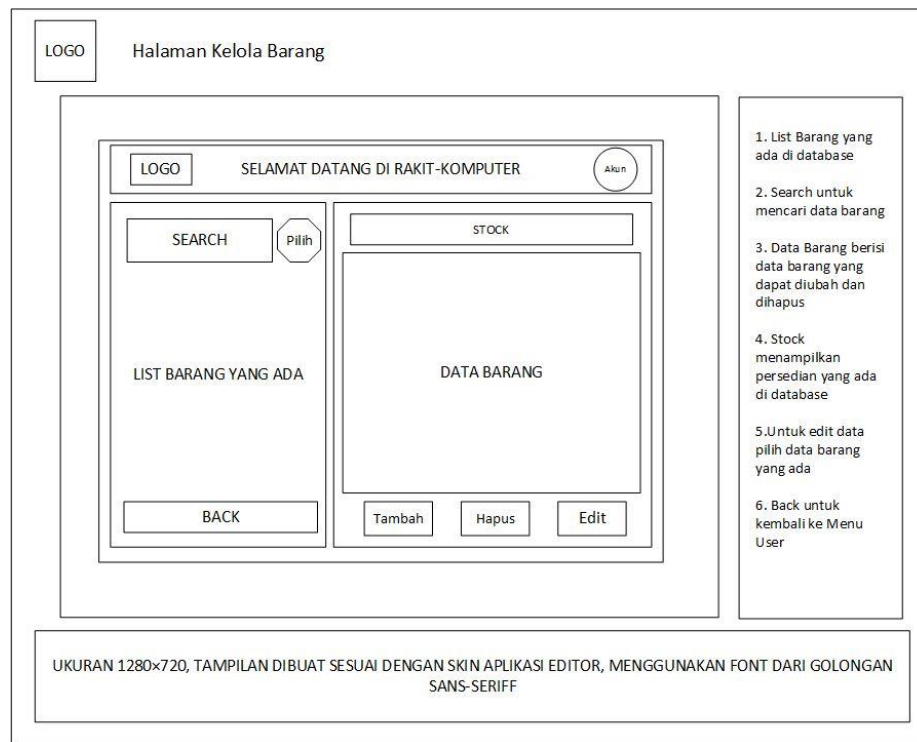
1. Form Data Diri berisi data diri yang ingin di ubah
2. Back Untuk kembali ke Menu masing-masing.
3. Submit untuk melakukan perubahan data-diri

UKURAN 1280x720, TAMPILAN DIBUAT SESUAI DENGAN SKIN APLIKASI EDITOR, MENGGUNAKAN FONT DARI GOLONGAN SANS-SERIFF

*Perancangan Muka 1 Edit Profile*

Antarmuka : Perancangan Muka 1 Edit Profile

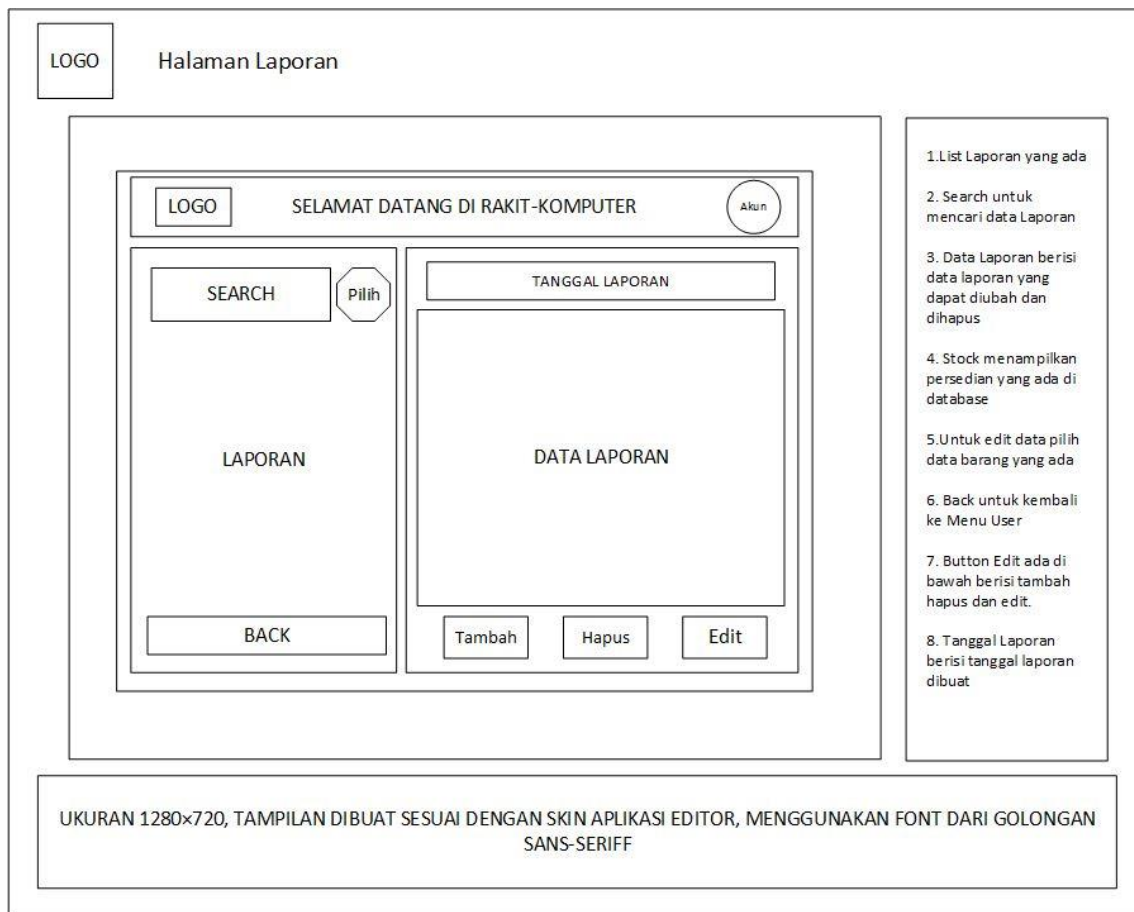
<b>Id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
Button 1	Button	Submit	Button diklik maka akan melakukan proses submit perubahan yang dilakukan didasarkan isian dari Form, dan kemudian akan di proses oleh Controller dan kemudian akan di kembalikan ke halaman User.
Button 2	Button	Back	Button Di Klik akan Kembali ke menu masing-masing user dan tidak melakukan perubahan apapun
Form 1	RTF Box	Form Data Diri	Merupakan bentuk form yang berisi isian guna mengubah profile/data diri dari seorang user (admin, manajer, konsultan, dan pelanggan)



*Perancangan Muka 2 Halaman Kelola Barang*

Antarmuka : Perancangan Muka 2 Halaman Kelola Barang

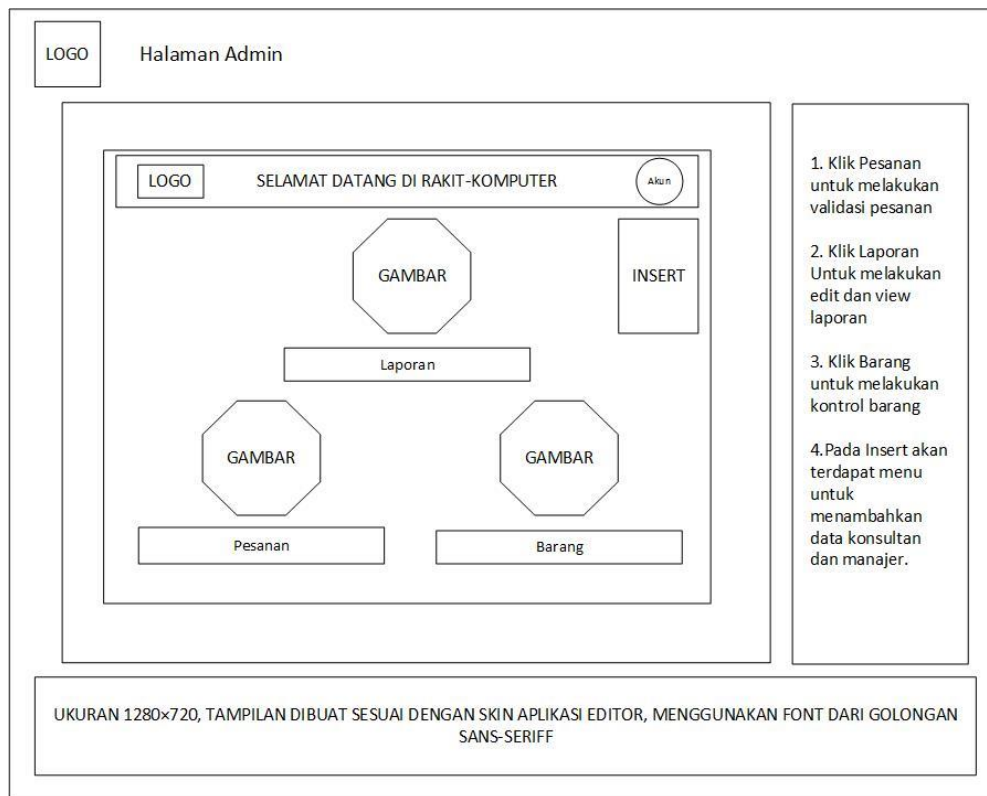
<b>Id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
Button 2	Button	Back	Jika Diklik maka akan menuju Kembali ke halaman utama dari seorang Admin.
Button 3	Button	Tambah	Button ini digunakan untuk menambahkan data barang dengan data yang berasal dari Form yang ada diatasnya.
Button 4	Button	Hapus	User harus memilih 1 barang yang ada dan kemudian form akan menampilkan datanya, lalu jika diklik maka data barang tersebut akan dihapus dari database
Button 5	Button	Edit	User harus memilih barang kemudian nanti akan mengisi id pada form dan user dapat menggantinya sesuai kebutuhan, lalu jika diklik maka data akan di update di database
Button 6	Button	Pilih	Jika Diklik maka akan memilih 1 barang dan menampilkan datanya di Form Data Barang
Form 2	Form	Data Barang	Sekumpulan dari beberapa media input yang digunakan untuk keperluan CRUD.
TextField 1	TextField	Stock	Text field ini menampilkan jumlah stock yang terbaca di database.
TexField 2	TextField	Search	Digunakan untuk mencari Barang yang ingin di lakukan kontrol.
Dropdown 1	Dropdown	Akun	Digunakan untuk mengakses Profile dari User dan terdapat dropdown menuju editProfile().
Table 1	Table	List Barang	Table yang berisi data batang belanjaan yang sudah dibeli.



Perancangan Muka 3 Halaman Laporan

Antarmuka : Perancangan Muka 3 Halaman Laporan

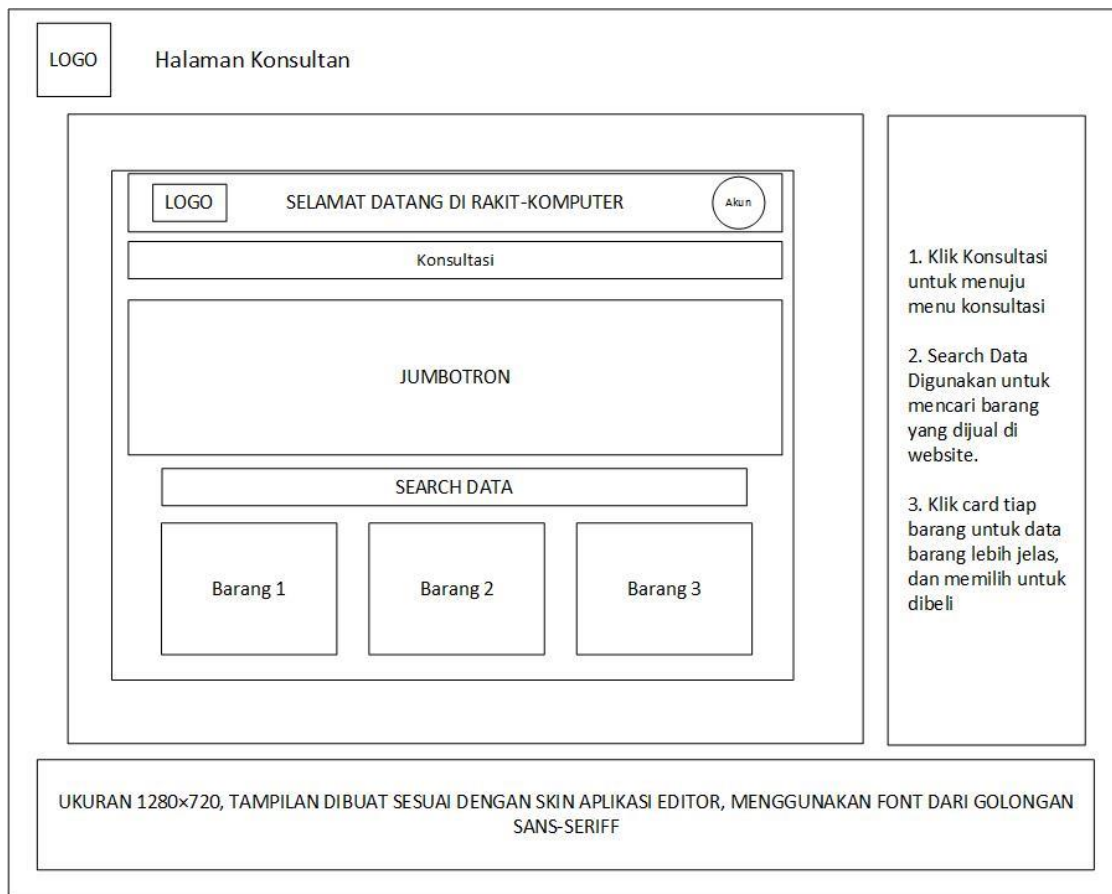
<b>Id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
Button 7	Button	Back	Jika Diklik maka akan menuju Kembali ke halaman utama dari seorang Admin/Manajer.
Button 8	Button	Tambah	Button ini digunakan untuk menambahkan data Laporan dengan data yang berasal dari Form yang ada diatasnya.
Button 9	Button	Hapus	User harus memilih 1 Laporan yang ada dan kemudian form akan menampilkan datanya, lalu jika diklik maka data Laporan tersebut akan dihapus dari database
Button 10	Button	Edit	User harus memilih laporan kemudian nanti akan mengisi id pada form dan user dapat menggantinya sesuai kebutuhan, lalu jika diklik maka data akan di update di database
Button 11	Button	Pilih	Jika Diklik maka akan memilih 1 Laporan dan menampilkan datanya di Form Data Barang
Form 3	Form	Data Laporan	Sekumpulan dari beberapa media input yang digunakan untuk keperluan CRUD.
TextField 3	TextField	Tanggal Laporan	Text Field ini menampilkan kapan terakhir kali laporan ini mendapatkan perubahan edit/akses.
TexField 4	TextField	Search	Digunakan untuk mencari Laporan yang ingin di lakukan kontrol.
Dropdown 2	Dropdown	Akun	Digunakan untuk mengakses Profile dari User dan terdapat dropdown menuju editProfile().
Table 2	Table	Laporan	Tabel yang berisi data Laporan yang ada.



*Perancangan Muka 4 Halaman Admin*

Antarmuka : Perancangan Muka 4 Halaman Admin

Id_Objek	Jenis	Nama	Keterangan
Button 12	Button	Laporan	Jika diklik akan menuju ke halaman Laporan dan dapat dilakukan control laporan oleh Admin.
Button 13	Button	Pesanan	Jika diklik, akan Menuju ke halaman pesanan dan dapat dilakukan control pesanan oleh Admin.
Button 14	Button	Barang	Jika diklik akan menuju ke halaman Barang dan dapat melakukan control barang oleh admin
Dropdown 3	Dropdown	Akun	Digunakan untuk mengakses Profile dari User dan terdapat dropdown menuju editProfile().
Gambar 1	Gambar	Gambar	Berisi Gambar Interaktif yang menandakan bahwa itu adalah menu yang berkaitan dengan laporan.
Gambar 2	Gambar	Gambar	Berisi Gambar Interaktif yang menandakan bahwa itu adalah menu yang berkaitan dengan Pesanan.
Gambar 3	Gambar	Gambar	Berisi Gambar Interaktif yang menandakan bahwa itu adalah menu yang berkaitan dengan Barang.
Dropdown 4	Dropdown	Akun	Digunakan untuk mengakses Profile dari User dan terdapat dropdown menuju editProfile().
Dropdown 5	Dropdown	Insert	Digunakan untuk mengakses halaman insert konsultan dan menghapusnya dan juga dapat menghapus manajer karena nanti akan ada 2 pilihan



*Perancangan Muka 5 Halaman Konsultan*

Antarmuka : Perancangan Muka 5 Halaman Konsultan

<b>ID_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
Card 1	Card	Barang 1	Card ini menampilkan salah satu barang yang di jual atau barang yang ada didatabase 1 baris dari card ini terdiri dari 3 card.
Card 2	Card	Barang 2	Card ini menampilkan salah satu barang yang di jual atau barang yang ada didatabase 1 baris dari card ini terdiri dari 3 card.
Card 3	Card	Barang 3	Card ini menampilkan salah satu barang yang di jual atau barang yang ada didatabase 1 baris dari card ini terdiri dari 3 card. Akan ada 2 atau 3 baris lainnya.
Textfield 5	Textfield	Search Data	Search Data ini digunakan untuk mencari data barang yang diinginkan yang berada d Website.
Button 15	Button	Konsultasi	Jika diklik akan menuju ke halaman Konsultasi
Dropdown 6	Dropdown	Akun	Digunakan untuk mengakses Profile dari User dan terdapat dropdown menuju editProfile().
Jumbotron 1	Jumbotron	Jumbotron	File berupa gambar yang bergerak menunjukkan pesan atau barang yang sedang memiliki penjualan yang baik.

LOGO

Halaman Login

LOGO

SELAMAT DATANG  
DI  
RAKIT-KOMPUTER

USERNAME

PASSWORD

Login

JUMBOTRON

1. Isi Username  
2. Isi Password  
3. Klik Login Untuk  
Masuk ke Menu  
Masing-Masing

UKURAN 1280x720, TAMPILAN DIBUAT SESUAI DENGAN SKIN APLIKASI EDITOR, MENGGUNAKAN FONT DARI GOLONGAN  
SANS-SERIFF

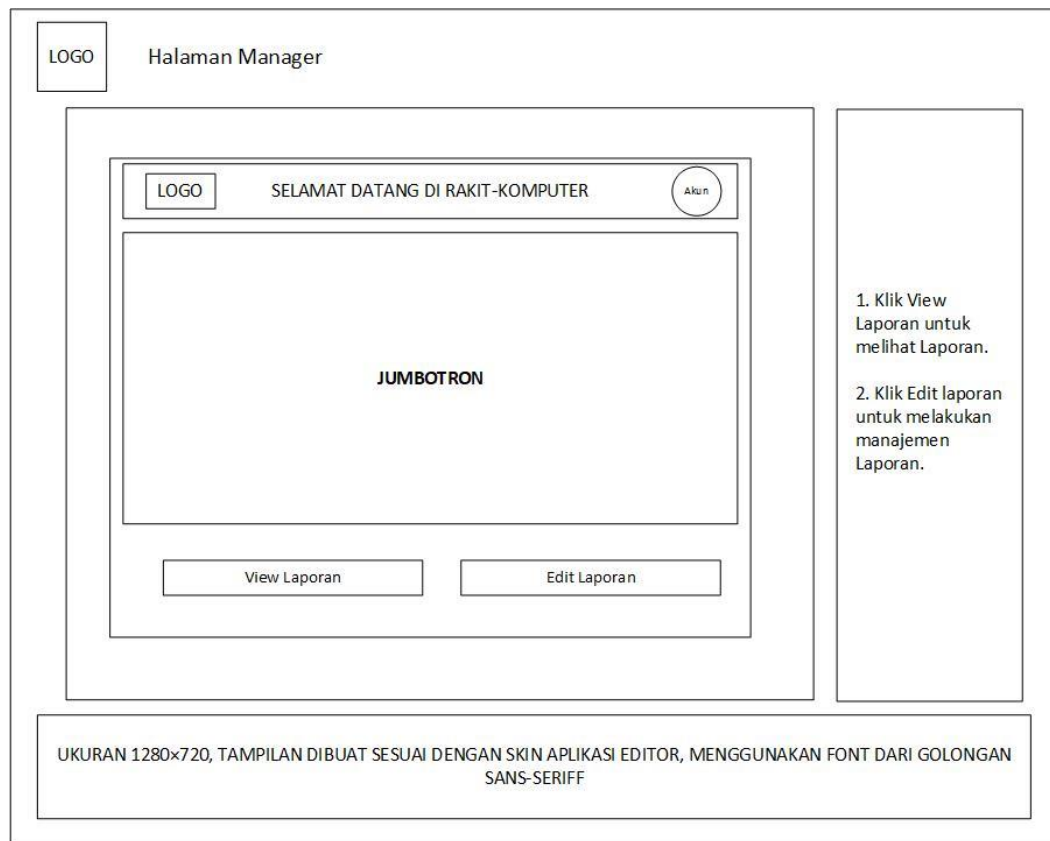
Perancangan Muka 6 Halaman Login

Antarmuka : Perancangan Muka 6 Halaman Login

<b>Id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
Jumbotron 2	Jumbotron	Jumbotron	File berupa gambar yang bergerak menunjukkan pesan atau barang yang sedang memiliki penjualan yang baik.
Button 16	Button	Login	Jika Diklik nantinya akan menuju ke masing-masing menu yang dimiliki oleh akun
Input 1	Input Text	Username	Digunakan untuk mengisi username untuk keperluan Login
Input 2	Input Text	Password	Digunakan untuk mengisi username untuk keperluan login

<b>Prodi S1 Informatika Tel-U</b>	<b>DPPL-04</b>	<b>Halaman 72 dari 81</b>
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		

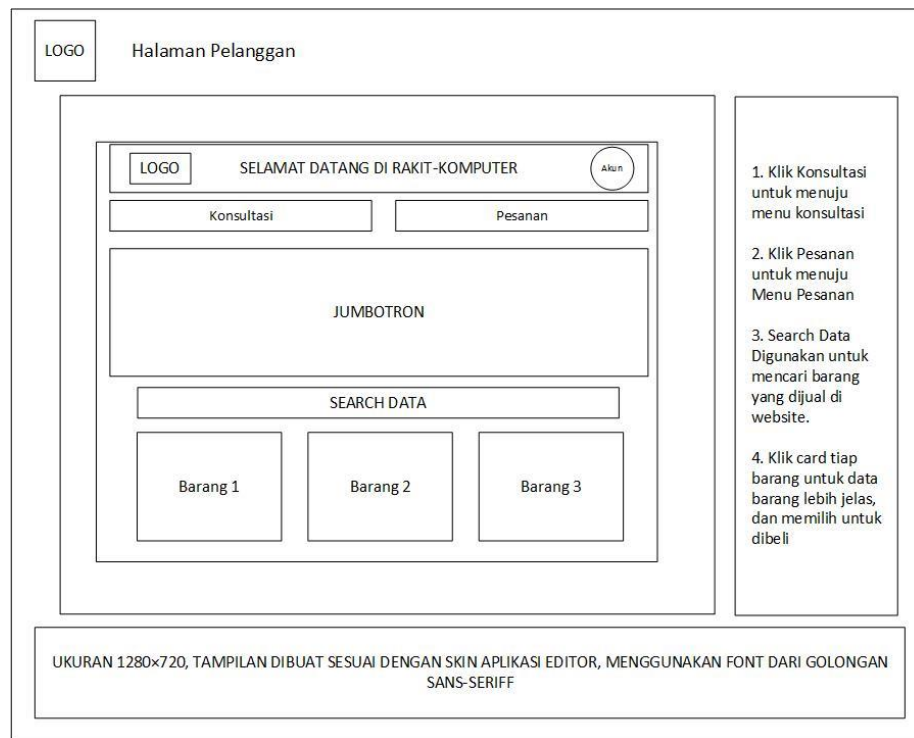




*Perancangan Muka 7 Halaman Manager*

*Antarmuka : Perancangan Muka 7 Halaman Manager*

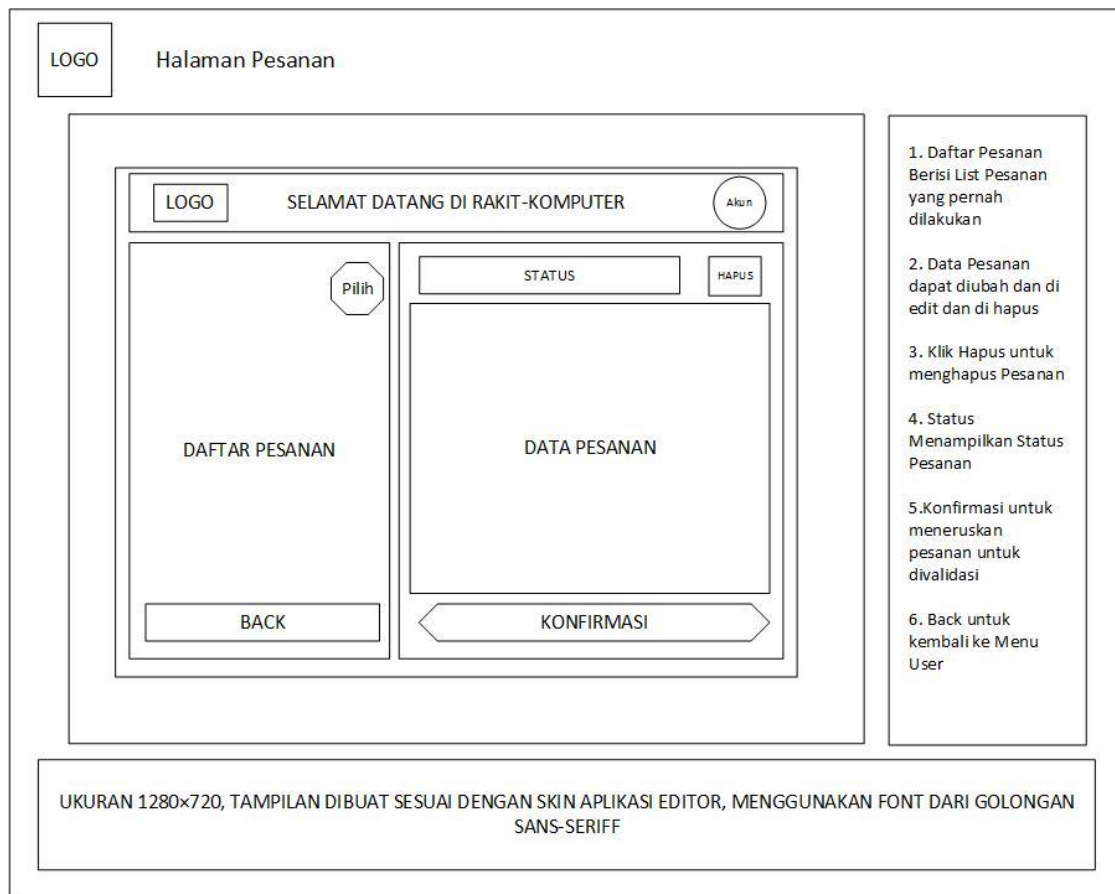
<b>Id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
Jumbotron 3	Jumbot ron	Jumbotron	File berupa gambar yang bergerak menunjukkan pesan atau barang yang sedang memiliki penjualan yang baik.
Button 17	Button	View Laporan	Jika diklik, akan menuju ke halaman laporan.
Button 18	Button	Edit Laporan	Jika diklik akan menuju halaman laporan
Dropdown 7	Dropd own	Akun	Digunakan untuk mengakses Profile dari User dan terdapat dropdown menuju editProfile().



*Perancangan Muka 8 Halaman Pelanggan*

Antarmuka : Perancangan Muka 8 Halaman Pelanggan

<b>Id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
Card 4	Card	Barang 1	Card ini menampilkan salah satu barang yang di jual atau barang yang ada didatabase 1 baris dari card ini terdiri dari 3 card.
Card 5	Card	Barang 2	Card ini menampilkan salah satu barang yang di jual atau barang yang ada didatabase 1 baris dari card ini terdiri dari 3 card.
Card 6	Card	Barang 3	Card ini menampilkan salah satu barang yang di jual atau barang yang ada didatabase 1 baris dari card ini terdiri dari 3 card. Akan ada 2 atau 3 baris lainnya.
Textfield 6	Textfield	Search Data	Search Data ini digunakan untuk mencari data barang yang diinginkan yang berada d Website.
Button 19	Button	Konsultasi	Jika diklik akan menuju ke halaman Konsultasi
Button 20	Button	Pesanan	Jika diklik akan menuju ke halaman Pesanan
Dropdown 8	Dropdown	Akun	Digunakan untuk mengakses Profile dari User dan terdapat dropdown menuju editProfile().
Jumbotron 8	Jumbotron	Jumbotron	File berupa gambar yang bergerak menunjukkan pesan atau barang yang sedang memiliki penjualan yang baik.



*Perancangan Muka 9 Halaman Pesanan*

Antarmuka : Perancangan Muka 9 Halaman Pesanan

Id Objek	Jenis	Nama	Keterangan
Table 3	Table	Daftar Pesanan	Diisi dengan penjelasan reaksi sistem, misalnya membuka layar apa, link kemana. Jika menyangkut suatu kode yang cukup rumit, acu algoritma yang telah diuraikan di atas.
Form 4	Form	Data Pesanan	Berisi Form yang dapat diisi untuk melihat pesanan apa saja yang ingin diatur..
Button 21	Button	Back	Jika Diklik akan menuju ke halaman user.
Button 22	Button	Konfirmasi	Jika Diklik akan menuju ke halaman user dan menginputkan data pesanan ke database
Button 23	Button	Hapus	Jika di klik akan menghapus data pesanan dan mengirimkan ke database
Button 24	Button	Pilih	Jika Diklik akan mengisi data pesanan dengan id pesanan yang dipilih.
Textfield 7	Textfield	Status	Menampilkan status pesanan saat diakses, sudah berakhir, sedang transaksi, belum validasi, atau sudah tervalidasi.
Dropdown 9	Dropdown	Akun	Digunakan untuk mengakses Profile dari User dan terdapat dropdown menuju editProfile().

LOGO

Halaman Register

LOGO

SILAHKAN ISI DATA BERIKUT INI

FORM DATA PENGGUNA

USERNAME

PASSWORD

Register

JUMBOTRON

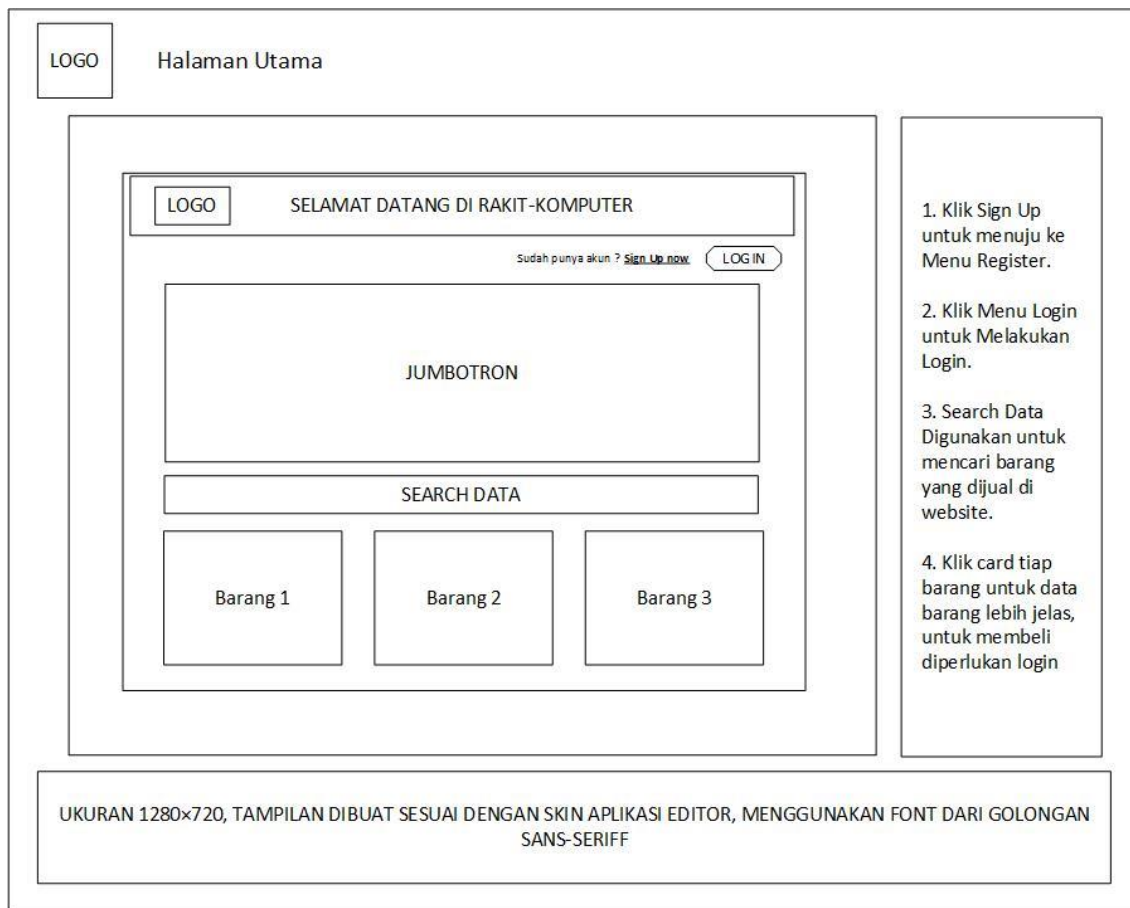
1. Form Data Pengguna Diisi dengan Pertanyaan mengenai beberapa data pribadi yang dibutuhkan seperti tanggal lahir, email, dan nama diri, serta alamat
2. Username berisi username yang akan digunakan
3. Password berisi password yang akan digunakan
4. Klik Register untuk mendaftarkan diri lalu kemudian masuk melalui login

UKURAN 1280x720, TAMPILAN DIBUAT SESUAI DENGAN SKIN APLIKASI EDITOR, MENGGUNAKAN FONT DARI GOLONGAN SANS-SERIFF

Perancangan Muka 10 Halaman Register

Antarmuka : Perancangan Muka 10 Halaman Register

<b>Id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
FormRegist	Form	Form Data Pengguna	Berisi Sekumpulan isian input yang digunakan nantinya untuk Algoritma Register.
Input 3	Input Text	Username	Isi Text Username yang tersedia dan unique
Input 4	Input Pass	Password	Isi Text Dengan Password yang dibutuhkan
Button 25	Button	Register	Jika Diklik akan mengaktifkan Algoritma Register dan melanjutkan ke halaman user.
Jumbotron 9	Jumbotron	Jumbotron	File berupa gambar yang bergerak menunjukkan pesan atau barang yang sedang memiliki penjualan yang baik.



*Perancangan Muka 11 Halaman Utama*

Antarmuka : Perancangan Muka 11 Halaman Utama

<b>Id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
Card 7	Card	Barang 1	Card ini menampilkan salah satu barang yang di jual atau barang yang ada didatabase 1 baris dari card ini terdiri dari 3 card.
Card 8	Card	Barang 2	Card ini menampilkan salah satu barang yang di jual atau barang yang ada didatabase 1 baris dari card ini terdiri dari 3 card.
Card 9	Card	Barang 3	Card ini menampilkan salah satu barang yang di jual atau barang yang ada didatabase 1 baris dari card ini terdiri dari 3 card. Akan ada 2 atau 3 baris lainnya.
Textfield 7	Textfie ld	Search Data	Search Data ini digunakan untuk mencari data barang yang diinginkan yang berada d Website.
Button 26	Button	Login	Jika diklik akan menuju ke halaman Login
Text 1	Text	Sign-up	Jika Diklik akan menuju halaman register.
Jumbotron 10	Jumbot ron	Jumbotron	File berupa gambar yang bergerak menunjukkan pesan atau barang yang sedang memiliki penjualan yang baik.

LOGO

Halaman Insert Konsultan

LOGO

HOME

SILAHKAN ISI DATA BERIKUT INI

FORM DATA KONSULTAN

USERNAME

PASSWORD

Register

List Konsultan

Hapus

1. Form Data Konsultan D diisi dengan Pertanyaan mengenai beberapa data pribadi yang dibutuhkan seperti tanggal lahir, email, dan nama diri, serta alamat
2. Username berisi username yang akan digunakan
- 3.Password berisi password yang akan digunakan
4. Klik Register untuk mendaftarkan diri lalu kemudian masuk melalui login
- 5.Klik Hapus untuk menghapus data Konsultan yang dipilih
6. Klik Home untuk kembali ke Halaman Admin

UKURAN 1280x720, TAMPILAN DIBUAT SESUAI DENGAN SKIN APLIKASI EDITOR, MENGGUNAKAN FONT DARI GOLONGAN SANS-SERIFF

Perancangan Muka 12 Insert Konsultan

Antarmuka : Perancangan Muka 12 Insert Konsultan

<b>Id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
FormRegist 2	Form	Form Data Konsultan	Berisi Sekumpulan isian input yang digunakan nantinya untuk Algoritma Insert.
Input 5	Input Text	Username	Isi Text Username yang tersedia dan unique
Input 6	Input Pass	Password	Isi Text Dengan Password yang dibutuhkan
Button 27	Button	Register	Jika Diklik akan mengaktifkan Algoritma Register dan melanjutkan ke halaman user.
Table 4	Table	List Konsultan	Berisi list konsultan yang berada di system dan dapat dipilih untuk dilakukan penghapusan
Button 27	Button	Hapus	Digunakan untuk menghapus Data Konsultan yang sudah dipilih

LOGO

Halaman Insert Manajer

LOGO

HOME

SILAHKAN ISI DATA BERIKUT INI

FORM DATA MANAJER

USERNAME

PASSWORD

Register

List Konsultan

Hapus

1. Form Data Manajer Diisi dengan Pertanyaan mengenai beberapa data pribadi yang dibutuhkan seperti tanggal lahir, email, dan nama diri, serta alamat  
2. Username berisi username yang akan digunakan  
3. Password berisi password yang akan digunakan  
4. Klik Register untuk mendaftarkan diri lalu kemudian masuk melalui login  
5. Klik Hapus untuk menghapus data Manajer yang dipilih  
6. Klik Home untuk kembali ke halaman admin

UKURAN 1280x720, TAMPILAN DIBUAT SESUAI DENGAN SKIN APLIKASI EDITOR, MENGGUNAKAN FONT DARI GOLONGAN SANS-SERIFF

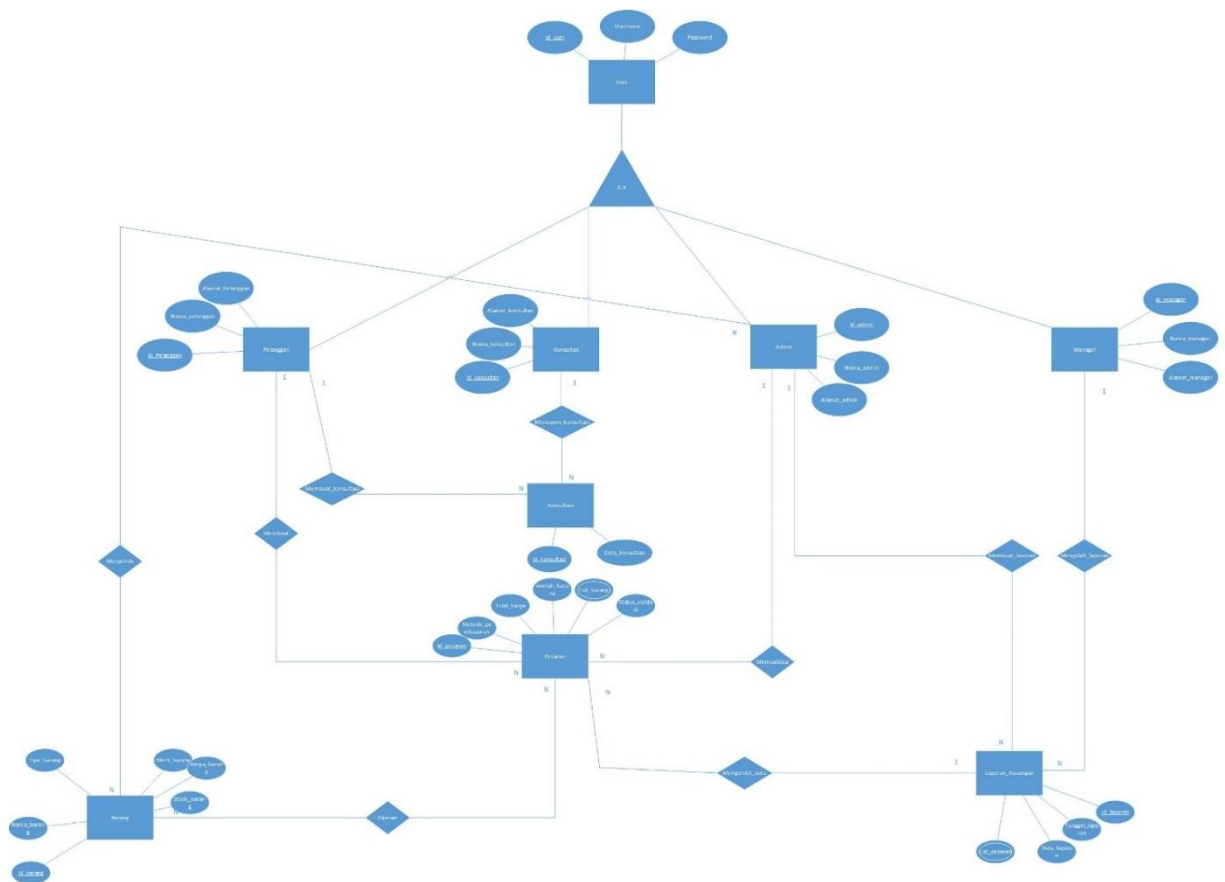
Perancangan Muka 13 Insert Manajer

Antarmuka : Perancangan Muka 13 Insert Manajer

<b>Id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
FormRegist 3	Form	Form Data Manajer	Berisi Sekumpulan isian input yang digunakan nantinya untuk Algoritma Insert.
Input 7	Input Text	Username	Isi Text Username yang tersedia dan unique
Input 8	Input Pass	Password	Isi Text Dengan Password yang dibutuhkan
Button 28	Button	Register	Jika Diklik akan mengaktifkan Algoritma Register dan melanjutkan ke halaman user.
Table 5	Table	List Manajer	Berisi list konsultan yang berada di system dan dapat dipilih untuk dilakukan penghapusan
Button 29	Button	Hapus	Digunakan untuk menghapus Data Konsultan yang sudah dipilih

<b>Prodi S1 Informatika Tel-U</b>	<b>DPPL-04</b>	<b>Halaman 79 dari 81</b>
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang mereproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		

### 3.7 Perancangan Representasi Persistensi Kelas



Gambar 8 ERD Rakit-Komputer



## 4 Matriks Kerunutan

Kelas	Use Case Terkait
Pelanggan	editPesanan, inputPesanan, deletePesanan, viewPesanan, inputKonsultasi, Pembayaran
Admin	inputBarang, deleteBarang, editBarang
Konsultan	deleteKonsultasi
Pelanggan, Konsultan	viewKonsultasi
Admin, Manajer	viewLaporanKeuangan, editLaporanKeuangan
Pelanggan, Admin, Konsultan, Manajer	Register, viewBarang, editProfile, Login