**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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Prodi S1- Informatika**  **Universitas Telkom** | Nomor Dokumen | | Halaman |
| *DPPL-04* | | *1/<jml #* |
| Revisi | *<nomor revisi>* | *Tgl: 21/04/2020* |

DAFTAR PERUBAHAN

|  |  |
| --- | --- |
| Revisi | Deskripsi |
| A |  |
| B |  |
| C |  |
| D |  |
| E |  |
| F |  |
| G |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX  TGL | - | A | B | C | D | E | F | G |
| Ditulis oleh |  |  |  |  |  |  |  |  |
| Diperiksa oleh |  |  |  |  |  |  |  |  |
| Disetujui oleh |  |  |  |  |  |  |  |  |

Daftar Halaman Perubahan

|  |  |  |  |
| --- | --- | --- | --- |
| Halaman | Revisi | Halaman | Revisi |
|  |  |  |  |

Daftar Isi

[1. Pendahuluan 8](#_Toc38802635)

[1.1 Tujuan Penulisan Dokumen 8](#_Toc38802636)

[1.2 Lingkup Masalah 8](#_Toc38802637)

[1.3 Definisi dan Istilah 8](#_Toc38802638)

[1.4 Referensi 9](#_Toc38802639)

[1.5 Sistematika Pembahasan 9](#_Toc38802640)

[2 Deskripsi Perancangan Global 11](#_Toc38802641)

[2.1 Rancangan Lingkungan Implementasi 11](#_Toc38802642)

[2.2 Deskripsi Arsitektural 11](#_Toc38802643)

[2.3 Deskripsi Komponen 12](#_Toc38802644)

[3 Perancangan Rinci 13](#_Toc38802645)

[3.1 Realisasi Use Case 13](#_Toc38802646)

[3.1.1 Use Case Login 13](#_Toc38802647)

[3.1.1.1 Identifikasi Kelas Login 13](#_Toc38802648)

[3.1.1.2 Sequence Diagram Login 13](#_Toc38802649)

[3.1.1.3 Diagram Kelas Login 14](#_Toc38802650)

[3.1.2 Use Case Input Pesanan 15](#_Toc38802651)

[3.1.2.1 Identifikasi Kelas Input Pesanan 15](#_Toc38802652)

[3.1.2.2 Sequence Diagram Input Pesanan 15](#_Toc38802653)

[3.1.2.3 Diagram Kelas Input Pesanan 16](#_Toc38802654)

[3.1.3 Use Case Edit Pesanan 16](#_Toc38802655)

[3.1.3.1 Identifikasi Kelas Edit Pesanan 16](#_Toc38802656)

[3.1.3.2 Sequence Diagram Edit Pesanan 17](#_Toc38802657)

[3.1.3.3 Diagram Kelas Edit Pesanan 17](#_Toc38802658)

[3.1.4 Use Case Hapus Pesanan 18](#_Toc38802659)

[3.1.4.1 Identifikasi Kelas Hapus Pesanan 18](#_Toc38802660)

[3.1.4.2 Sequence Diagram Hapus Pesanan 18](#_Toc38802661)

[3.1.4.3 Diagram Kelas Hapus Pesanan 19](#_Toc38802662)

[3.1.5 Use Case View Pesanan 19](#_Toc38802663)

[3.1.5.1 Identifikasi Kelas View Pesanan 19](#_Toc38802664)

[3.1.5.2 Sequence Diagram View Pesanan 20](#_Toc38802665)

[3.1.5.3 Diagram Kelas View Pesanan 21](#_Toc38802666)

[3.1.6 Use Case View Konsultasi 21](#_Toc38802667)

[3.1.6.1 Identifikasi Kelas View Konsultasi 21](#_Toc38802668)

[3.1.6.2 Sequence Diagram View Konsultasi 22](#_Toc38802669)

[3.1.6.3 Diagram Kelas View Konsultasi 22](#_Toc38802670)

[3.1.7 Use Case Input Konsultasi 23](#_Toc38802671)

[3.1.7.1 Identifikasi Kelas Input Konsultasi 23](#_Toc38802672)

[3.1.7.2 Sequence Diagram Input Konsultasi 23](#_Toc38802673)

[3.1.7.3 Diagram Kelas Input Konsultasi 24](#_Toc38802674)

[3.1.8 Use Case Hapus Konsultasi 24](#_Toc38802675)

[3.1.8.1 Identifikasi Kelas Hapus Konsultasi 24](#_Toc38802676)

[3.1.8.2 Sequence Diagram Hapus Konsultasi 25](#_Toc38802677)

[3.1.8.3 Diagram Kelas Hapus Konsultasi 25](#_Toc38802678)

[3.1.9 Use Case View Laporan Keuangan 26](#_Toc38802679)

[3.1.9.1 Identifikasi Kelas View Laporan Keuangan 26](#_Toc38802680)

[3.1.9.2 Sequence Diagram View Laporan Keuangan 26](#_Toc38802681)

[3.1.9.3 Diagram Kelas View Laporan Keuangan 27](#_Toc38802682)

[3.1.10 Use Case Edit Laporan Keuangan 28](#_Toc38802683)

[3.1.10.1 Identifikasi Kelas Edit Laporan Keuangan 28](#_Toc38802684)

[3.1.10.2 Sequence Diagram Edit Laporan Keuangan 28](#_Toc38802685)

[3.1.10.3 Diagram Kelas Edit Laporan Keuangan 28](#_Toc38802686)

[3.1.11 Use Case Edit Registrasi 29](#_Toc38802687)

[3.1.11.1 Identifikasi Kelas Registrasi 29](#_Toc38802688)

[3.1.11.2 Sequence Diagram Registrasi 29](#_Toc38802689)

[3.1.11.3 Diagram Kelas Registrasi 29](#_Toc38802690)

[3.1.12 Use Case Pembayaran 30](#_Toc38802691)

[3.1.12.1 Identifikasi Kelas Pembayaran 30](#_Toc38802692)

[3.1.12.2 Sequence Diagram Pembayaran 30](#_Toc38802693)

[3.1.12.3 Diagram Kelas Pembayaran 30](#_Toc38802694)

[3.1.13 Use Case Input Barang 31](#_Toc38802695)

[3.1.13.1 Identifikasi Kelas Input Barang 31](#_Toc38802696)

[3.1.13.2 Sequence Diagram Input Barang 31](#_Toc38802697)

[3.1.13.3 Diagram Kelas Input Barang 31](#_Toc38802698)

[3.1.14 Use Case Edit Barang 32](#_Toc38802699)

[3.1.14.1 Identifikasi Kelas Edit Barang 32](#_Toc38802700)

[3.1.14.2 Sequence Diagram Edit Barang 32](#_Toc38802701)

[3.1.14.3 Diagram Kelas Edit Barang 32](#_Toc38802702)

[3.1.15 Use Case Hapus Barang 33](#_Toc38802703)

[3.1.15.1 Identifikasi Kelas Hapus Barang 33](#_Toc38802704)

[3.1.15.2 Sequence Diagram Hapus Barang 33](#_Toc38802705)

[3.1.15.3 Diagram Kelas Hapus Barang 33](#_Toc38802706)

[3.1.16 Use Case View Barang 34](#_Toc38802707)

[3.1.16.1 Identifikasi Kelas View Barang 34](#_Toc38802708)

[3.1.16.2 Sequence Diagram View Barang 34](#_Toc38802709)

[3.1.16.3 Diagram Kelas Hapus Barang 34](#_Toc38802710)

[3.1.17 Use Case Edit Edit Profile 35](#_Toc38802711)

[3.1.17.1 Identifikasi Kelas Edit Profile 35](#_Toc38802712)

[3.1.17.2 Sequence Diagram Edit Profile 35](#_Toc38802713)

[3.1.17.3 Diagram Kelas Edit Profile 35](#_Toc38802714)

[3.2 Perancangan Detil Kelas 36](#_Toc38802715)

[3.2.1 Kelas Barang 36](#_Toc38802716)

[3.2.2 Kelas Pelanggan 36](#_Toc38802717)

[3.2.3 Kelas Konsultasi 37](#_Toc38802718)

[3.2.4 Kelas Konsultan 37](#_Toc38802719)

[3.2.5 Kelas User 38](#_Toc38802720)

[3.2.6 Kelas Admin 38](#_Toc38802721)

[3.2.7 Kelas Laporan Keuangan 39](#_Toc38802722)

[3.2.8 Kelas Manajer 39](#_Toc38802723)

[3.2.9 Kelas Pesanan 40](#_Toc38802724)

[3.3 Diagram Kelas Keseluruhan 41](#_Toc38802725)

[3.4 Algoritma/Query 42](#_Toc38802726)

[3.5 Statechart Diagram 56](#_Toc38802727)

[3.6 Perancangan Antarmuka 58](#_Toc38802728)

[3.7 Perancangan Representasi Persistensi Kelas 71](#_Toc38802729)

[4 Matriks Kerunutan 72](#_Toc38802730)

# 1. Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen ini berisi penjelasan mengenai Deskripsi Perancangan Perangakat 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.

## 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.

## 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

## 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

## 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 teridir 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.

# Deskripsi Perancangan Global

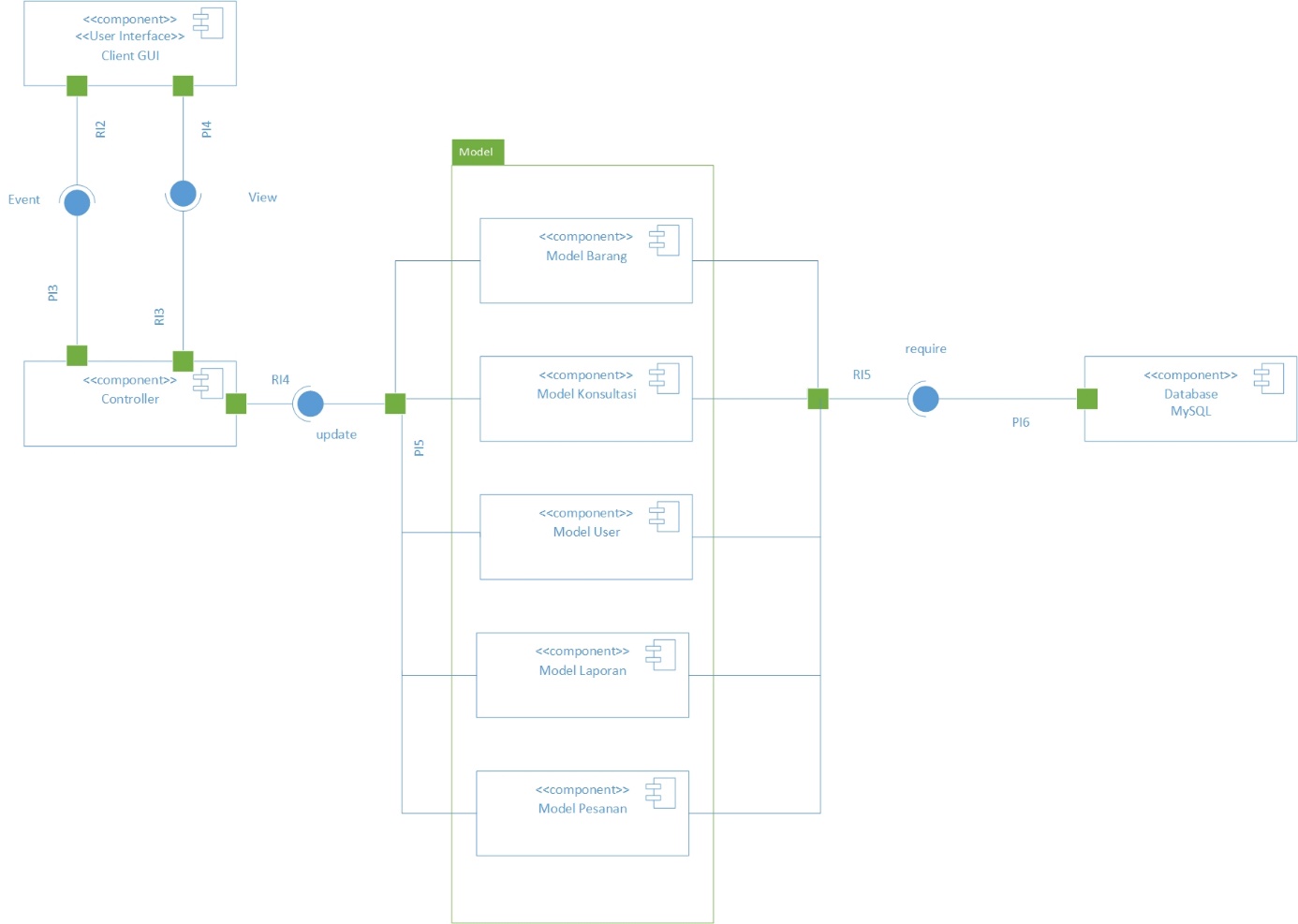
## Rancangan Lingkungan Implementasi

Sistem ini diimplementasikan dalam lingkungan sebagai berikut :

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

## 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 . Component Diagram

## 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. |

# 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

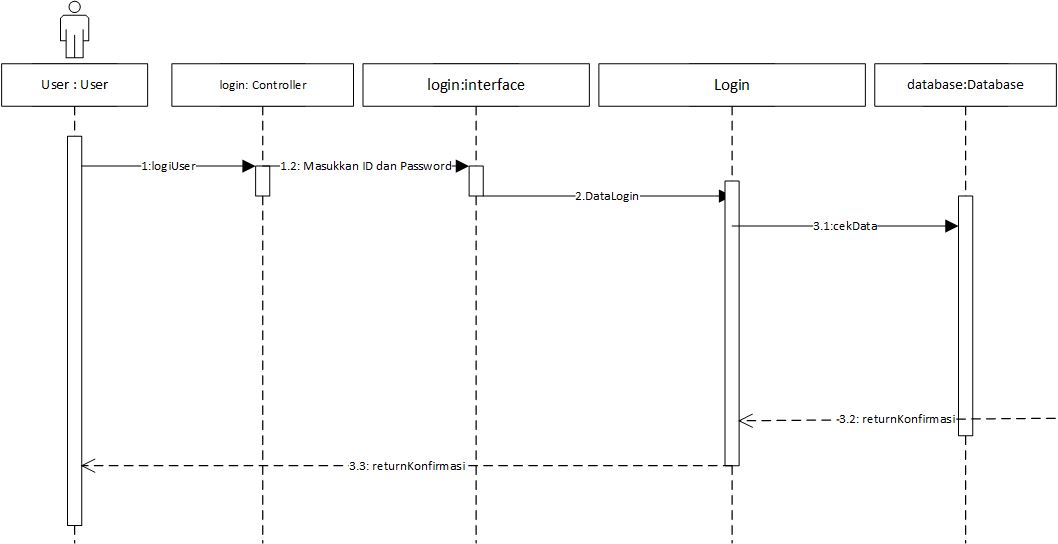
## Realisasi Use Case

### Use Case Login

#### Identifikasi Kelas Login

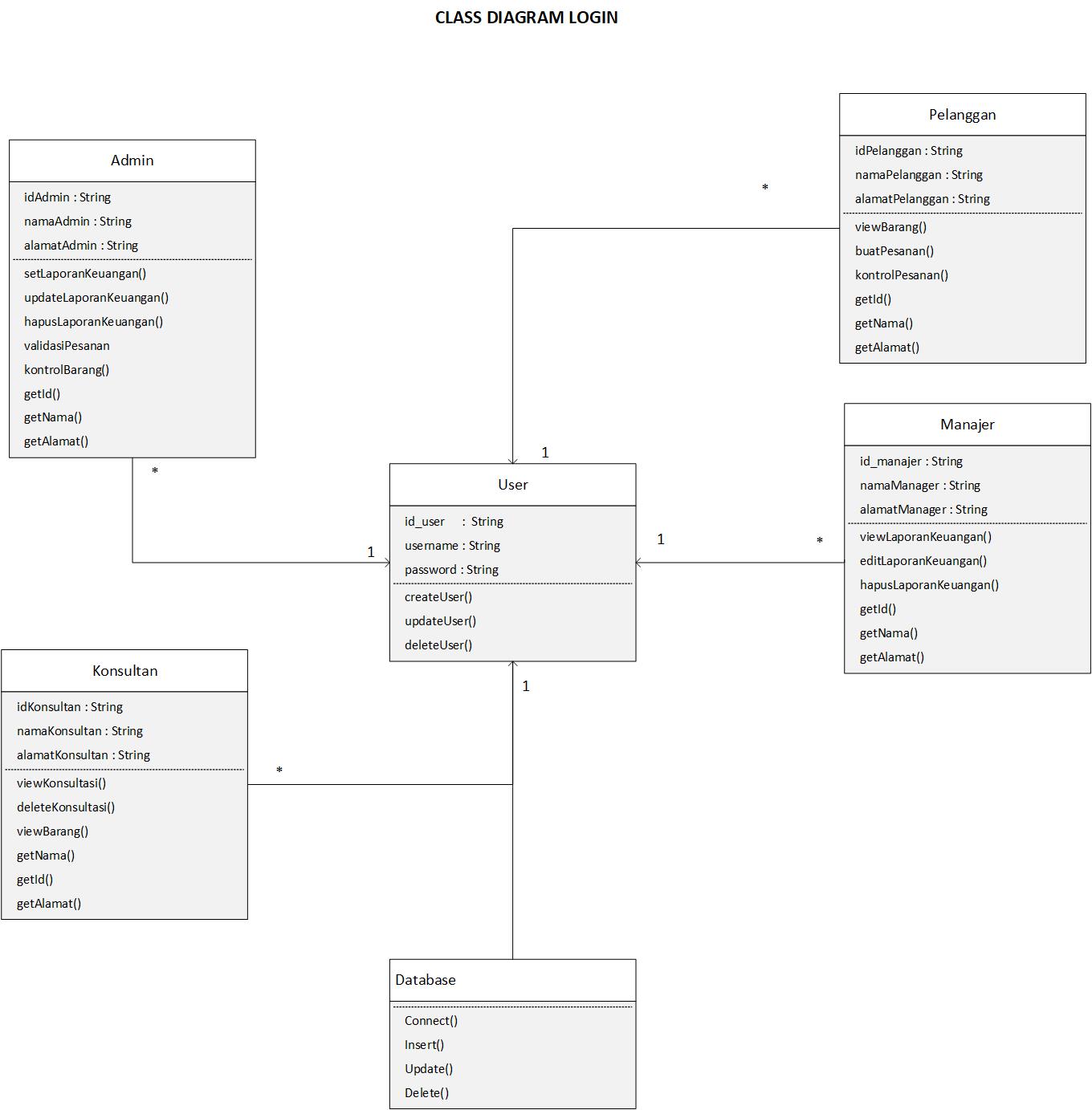
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Tipe Kelas** |
| 1 | User | Entity |
| 2 | Admin | Entity |
| 3 | Manager | Entity |
| 4 | Pelanggan | Entity |
| 5 | Konsultan | Entity |
| 6 | Database | Entity |

#### Sequence Diagram Login



Gambar .1.1 Sequence Diagram Login

#### Diagram Kelas Login



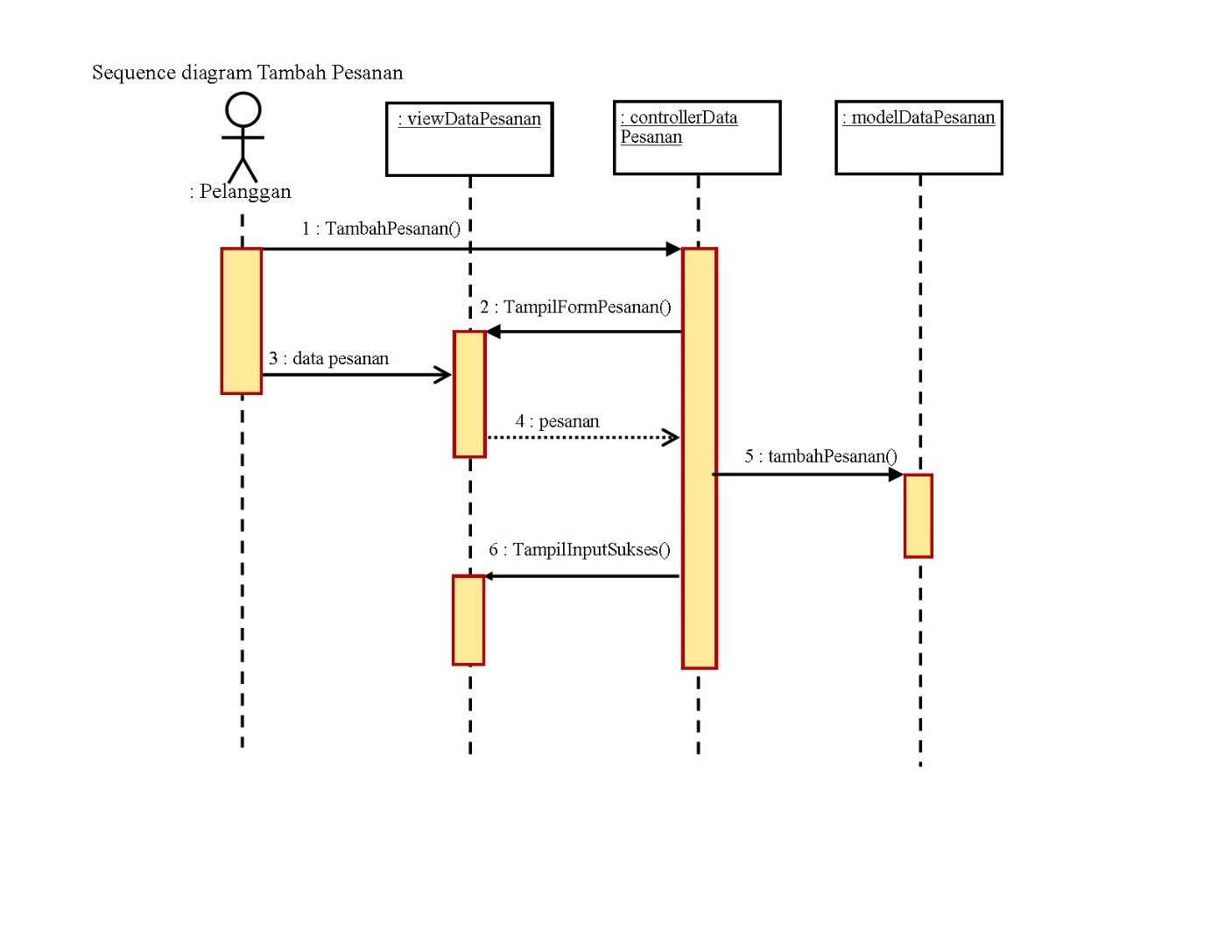
Gambar 2.1.2 Class Diagram Login

### Use Case Input Pesanan

#### Identifikasi Kelas Input Pesanan

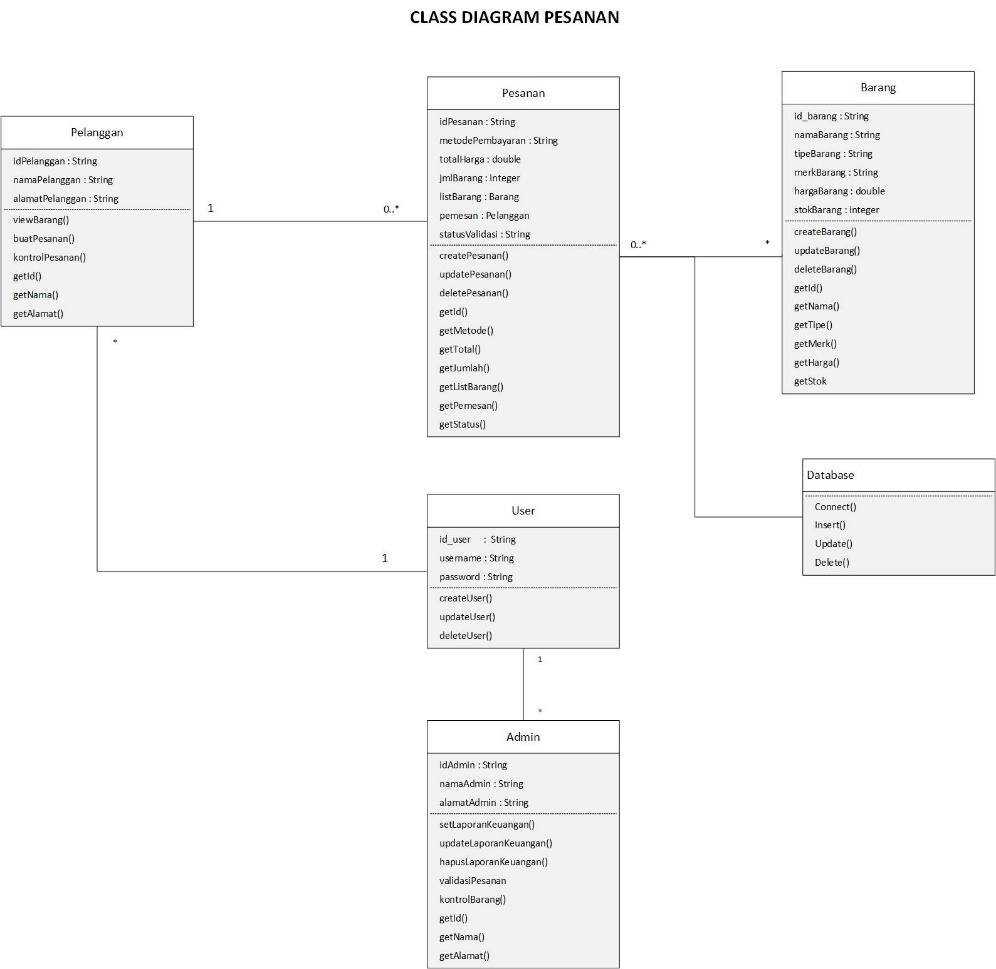
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Tipe Kelas** |
| 1 | User | Entity |
| 2 | Pelanggan | Entity |
| 3 | Pesanan | Entity |
| 4 | Barang | Entity |
| 5 | Database | Entity |

#### Sequence Diagram Input Pesanan



Gambar 2.2.1 Sequence Diagram Input Pesanan

#### Diagram Kelas Input Pesanan



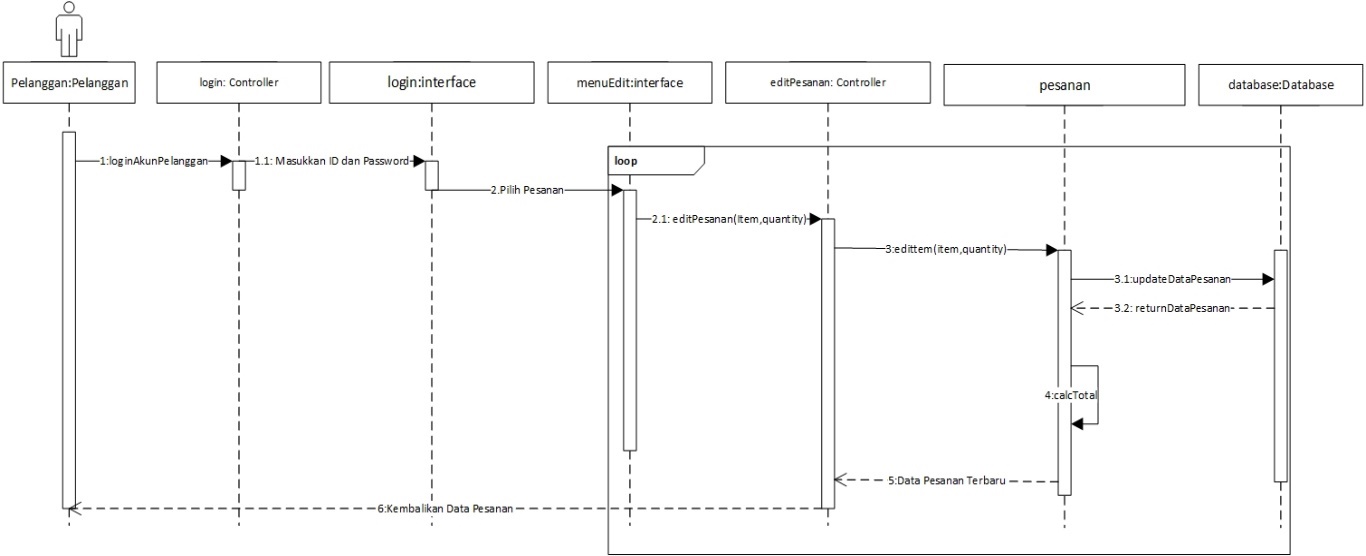
Gambar 2.2.2 Class Diagram Input Pesanan

### Use Case Edit Pesanan

#### Identifikasi Kelas Edit Pesanan

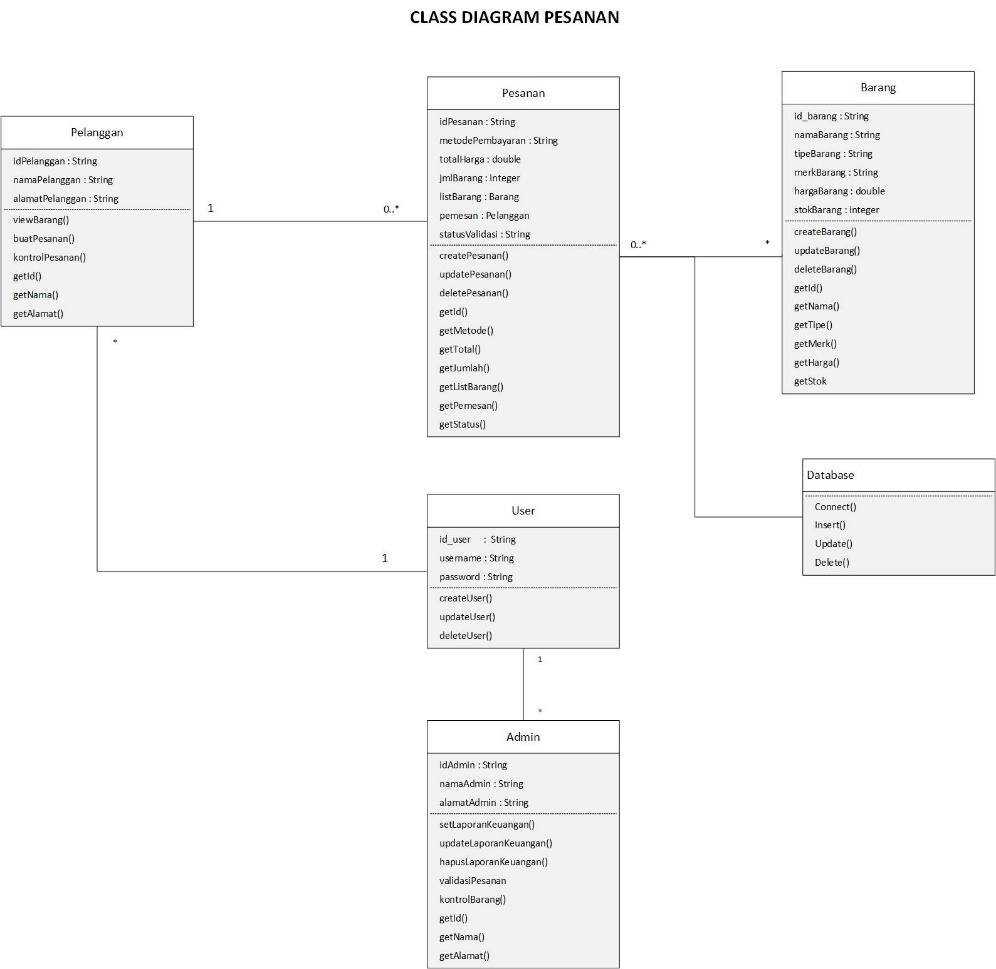
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Tipe Kelas** |
| 1 | User | Entity |
| 2 | Pelanggan | Entity |
| 3 | Pesanan | Entity |
| 4 | Barang | Entity |
| 5 | Database | Entity |

#### Sequence Diagram Edit Pesanan



Gambar 2.3.1 Sequence Diagram Edit Pesanan

#### Diagram Kelas Edit Pesanan



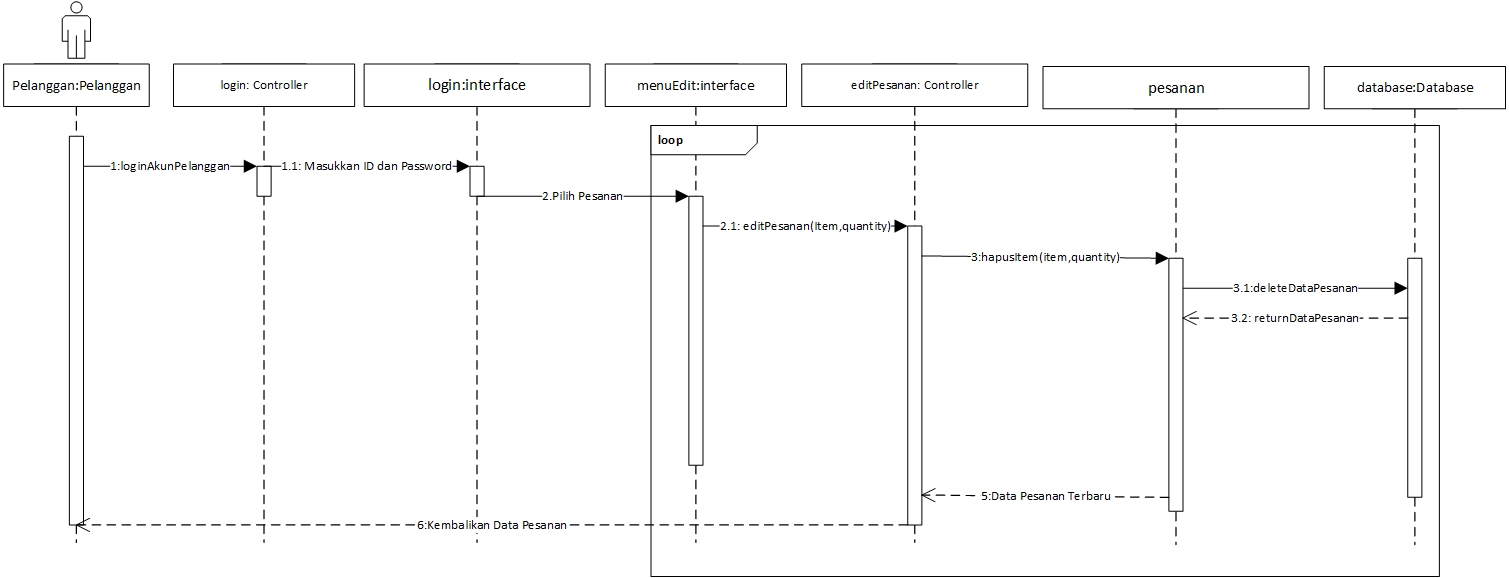
Gambar 2.3.2 Class Diagram Edit Pesanan

### Use Case Hapus Pesanan

#### Identifikasi Kelas Hapus Pesanan

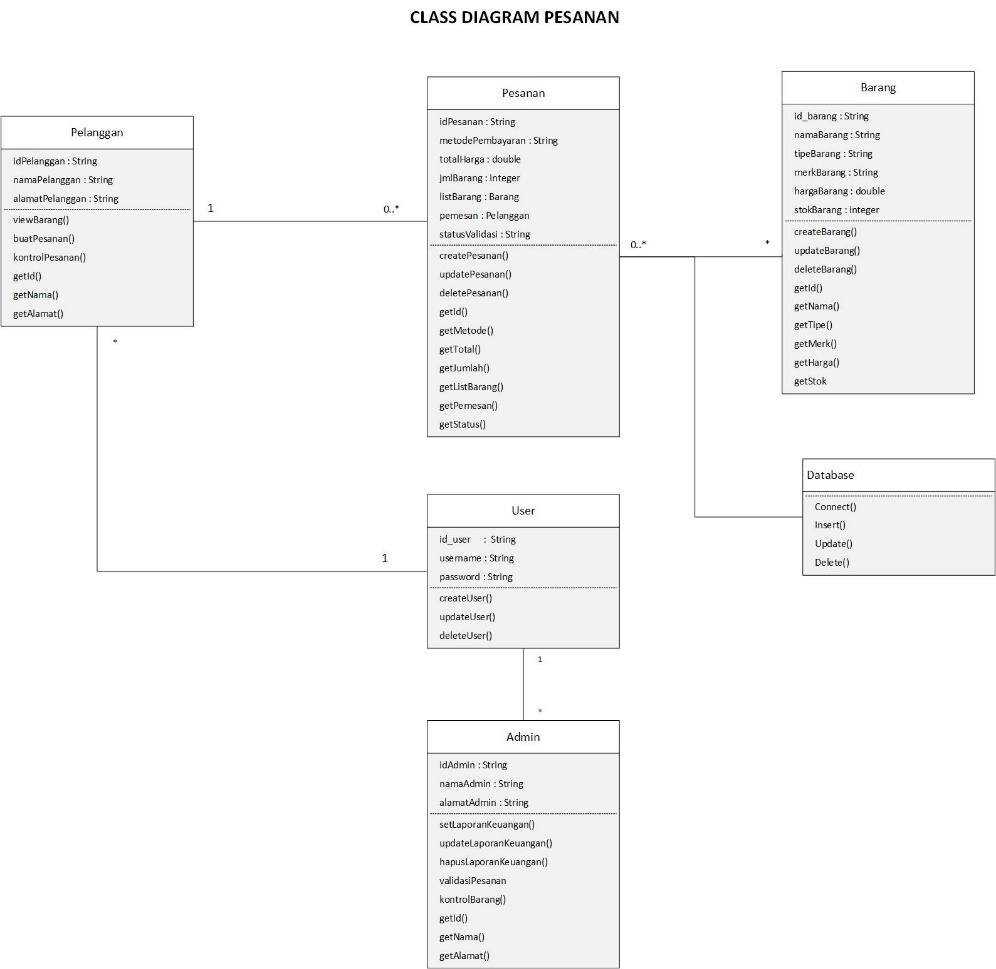
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Tipe Kelas** |
| 1 | User | Entity |
| 2 | Pelanggan | Entity |
| 3 | Pesanan | Entity |
| 4 | Barang | Entity |
| 5 | Database | Entity |

#### Sequence Diagram Hapus Pesanan



Gambar 2.4.1 Sequence Diagram Hapus Pesanan

#### Diagram Kelas Hapus Pesanan



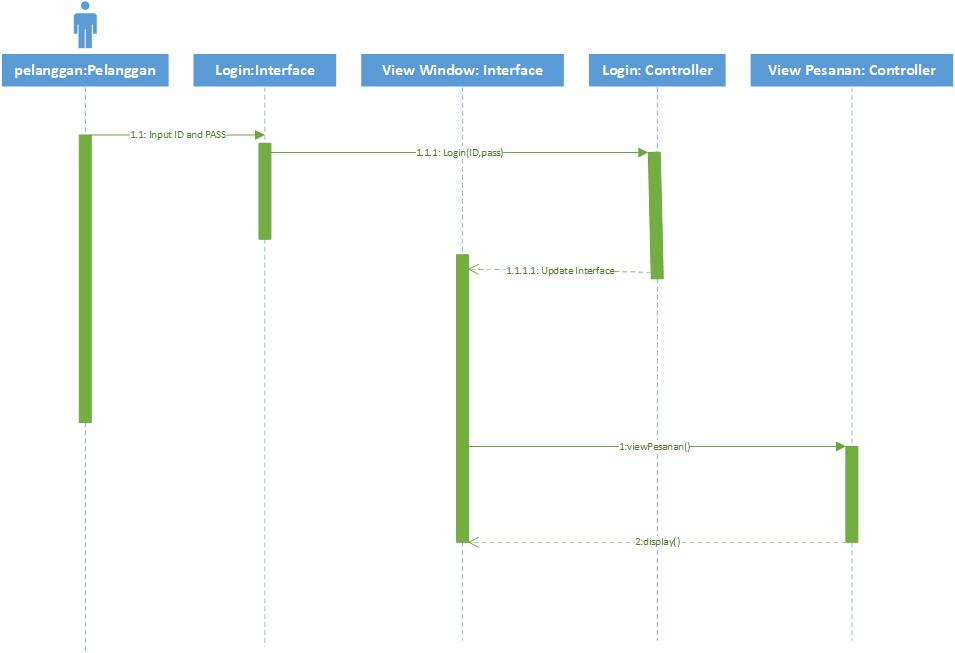
Gambar 2.4.2 Class Diagram Hapus Pesanna

### Use Case View Pesanan

#### Identifikasi Kelas View Pesanan

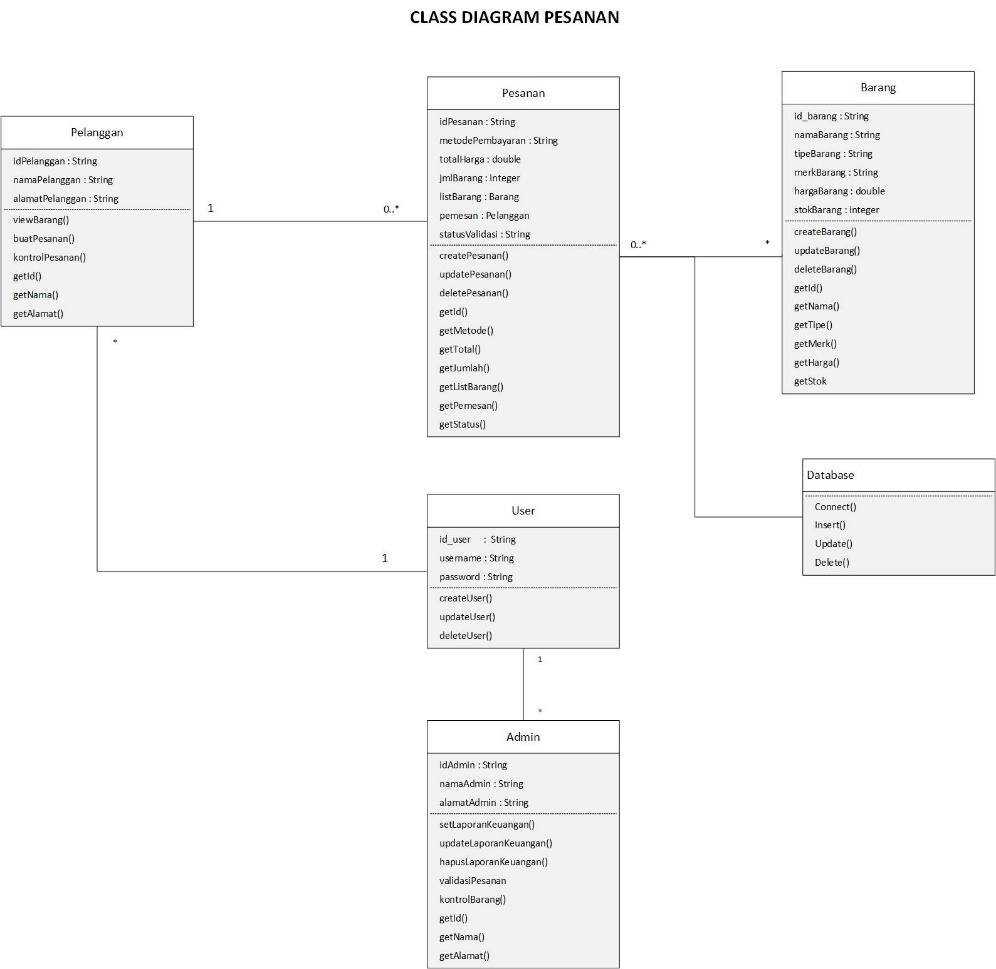
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Tipe Kelas** |
| 1 | User | Entity |
| 2 | Pelanggan | Entity |
| 3 | Pesanan | Entity |
| 4 | Barang | Entity |
| 5 | Database | Entity |

#### Sequence Diagram View Pesanan



Gambar 2.5.1 Sequence Diagram View Pesanan

#### Diagram Kelas View Pesanan



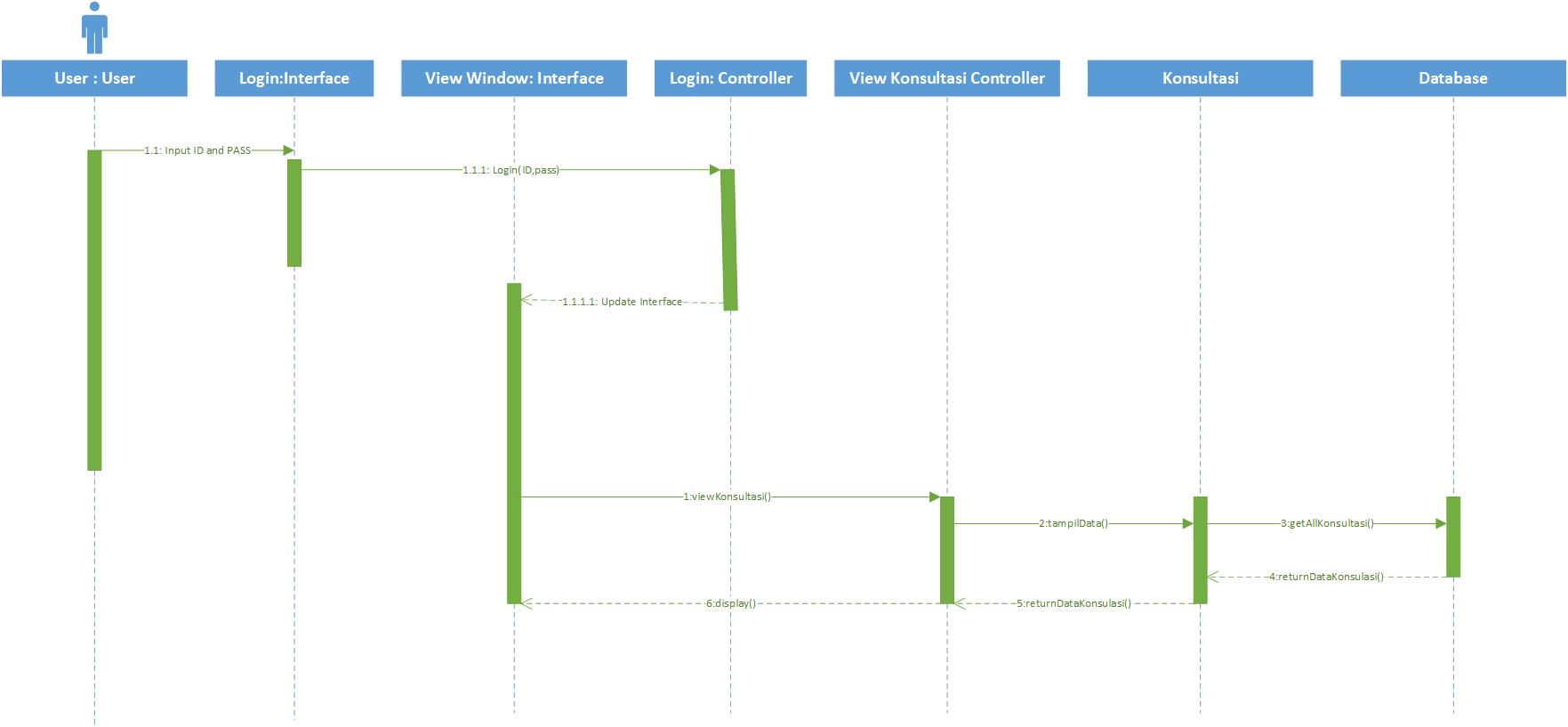
Gambar 2.5.2 Class Diagram View Pesanan

### Use Case View Konsultasi

#### Identifikasi Kelas View Konsultasi

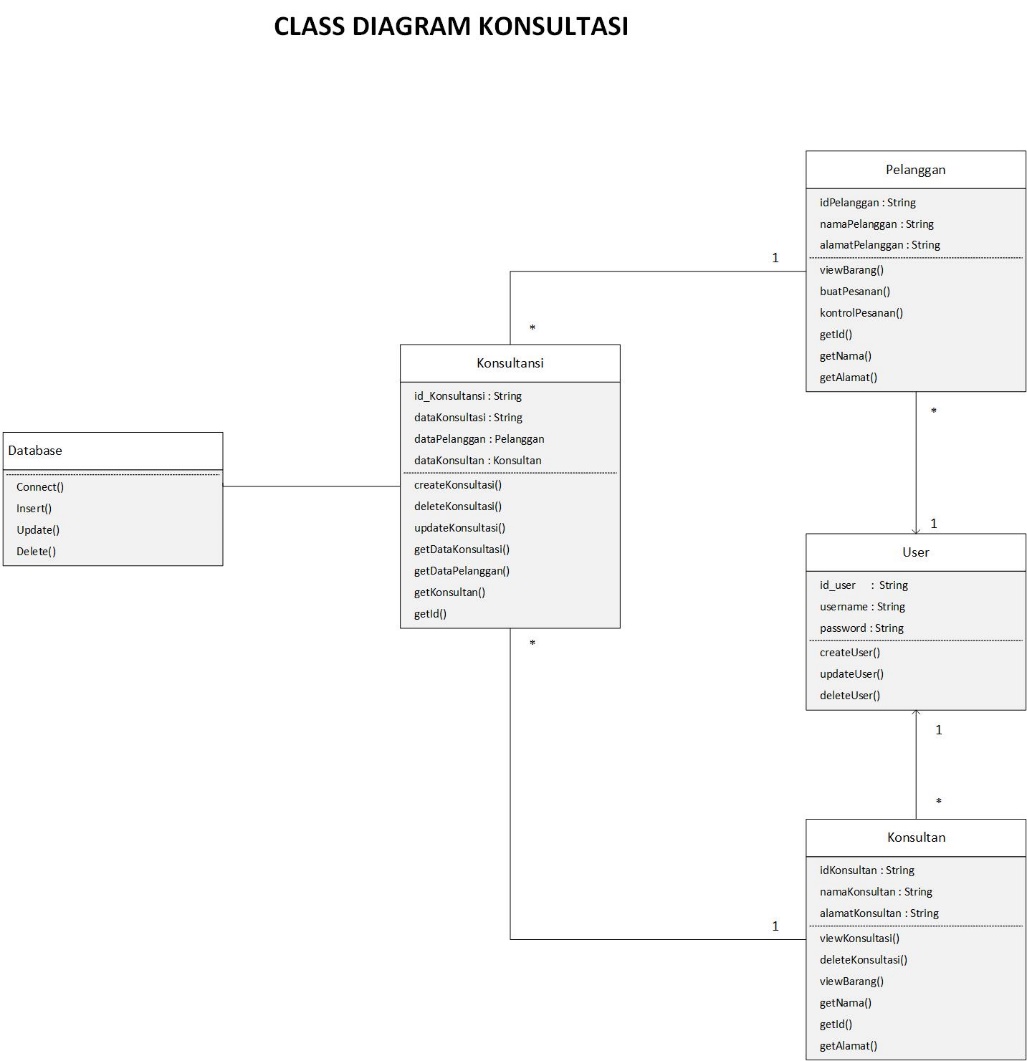
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Tipe Kelas** |
| 1 | User | Entity |
| 2 | Pelanggan | Entity |
| 3 | Konsultasi | Entity |
| 4 | Barang | Entity |
| 5 | Database | Entity |

#### Sequence Diagram View Konsultasi



Gambar 2.6.1 Sequence Diagram Input Pesanan

#### Diagram Kelas View Konsultasi



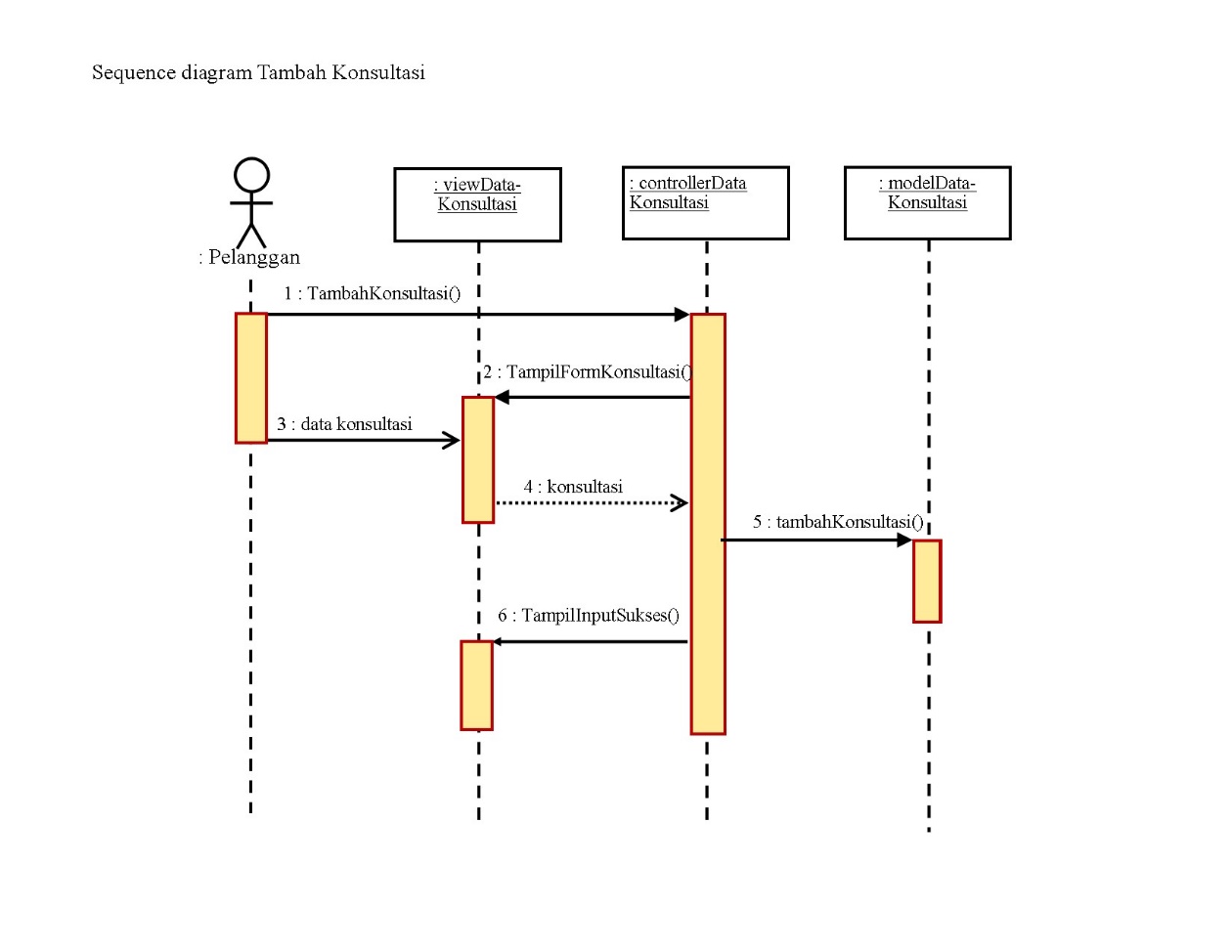
Gambar 2.6.2. Class Diagram View Konsultasi

### Use Case Input Konsultasi

#### Identifikasi Kelas Input Konsultasi

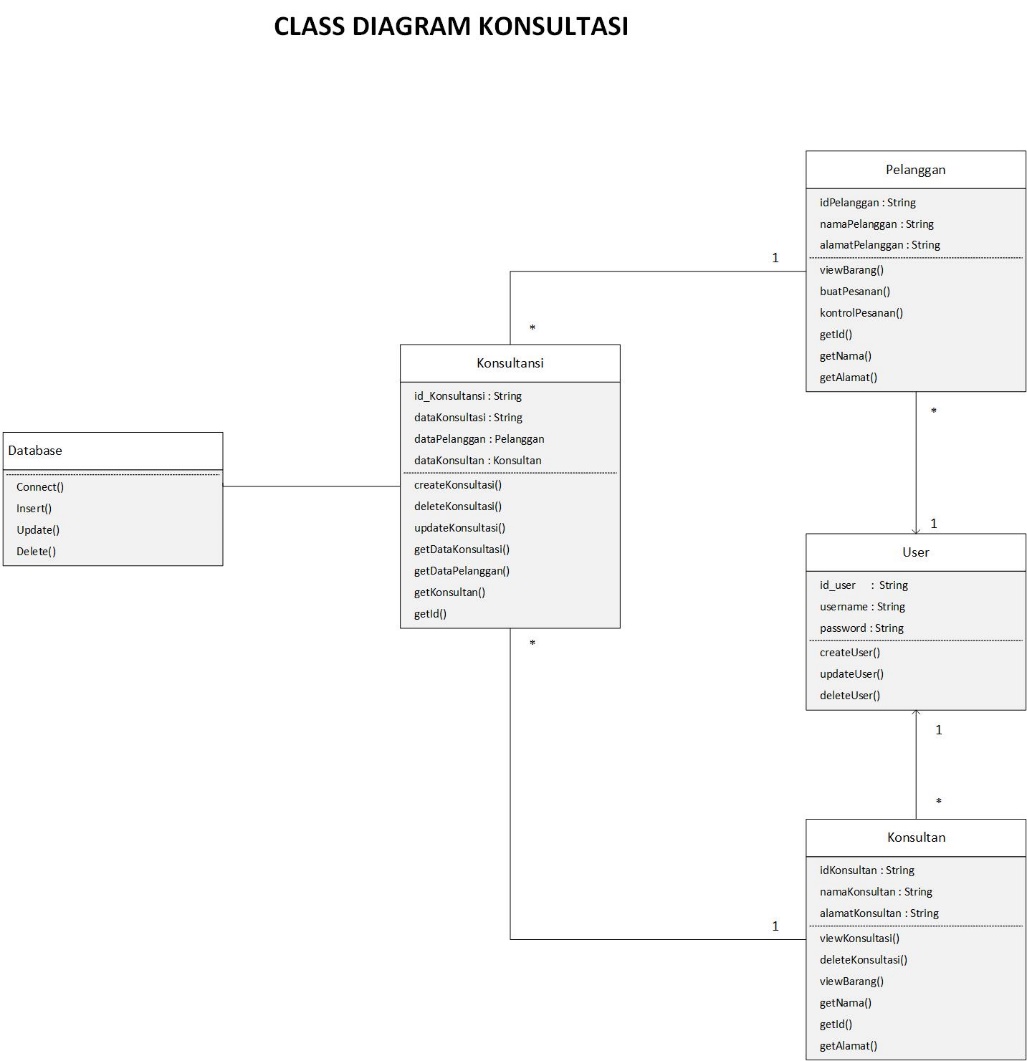
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Tipe Kelas** |
| 1 | User | Entity |
| 2 | Pelanggan | Entity |
| 3 | Konsultasi | Entity |
| 4 | Barang | Entity |
| 5 | Database | Entity |

#### Sequence Diagram Input Konsultasi



Gambar 2.7.1 Sequence Diagram Input Konsultasi

#### Diagram Kelas Input Konsultasi



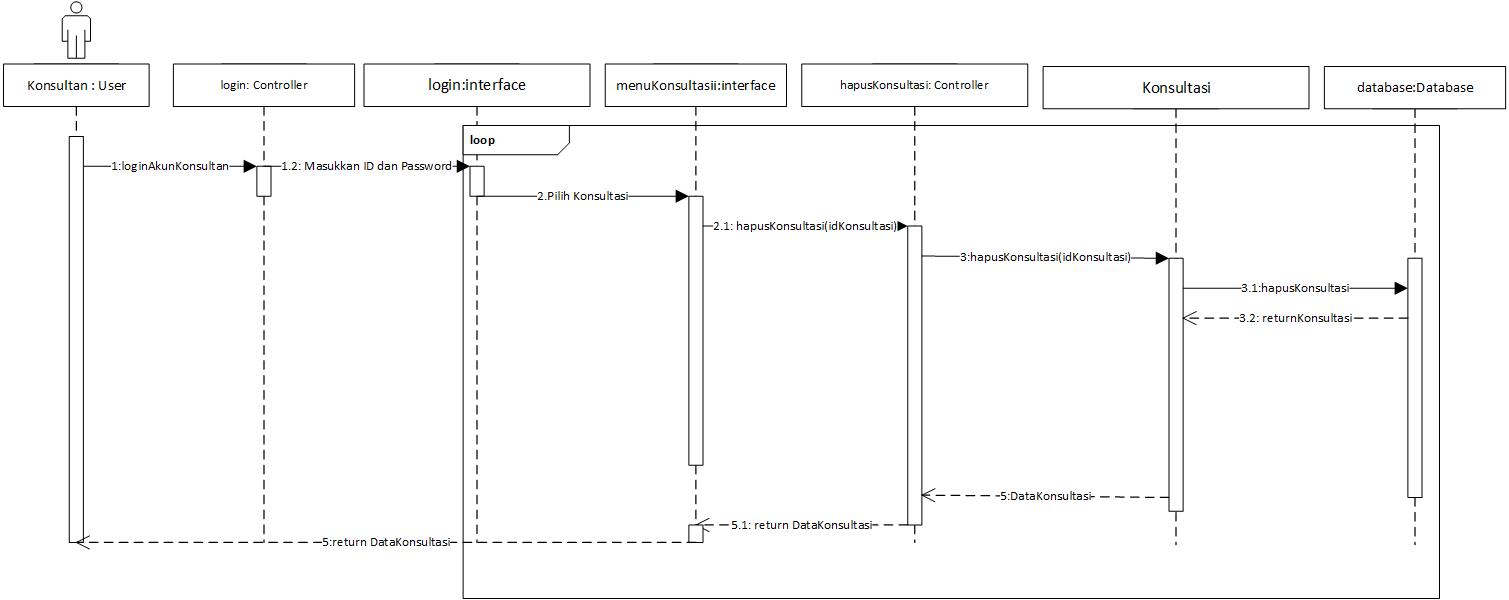
Gambar 2.7.2 Class Diagram Input Konsultasi

### Use Case Hapus Konsultasi

#### Identifikasi Kelas Hapus Konsultasi

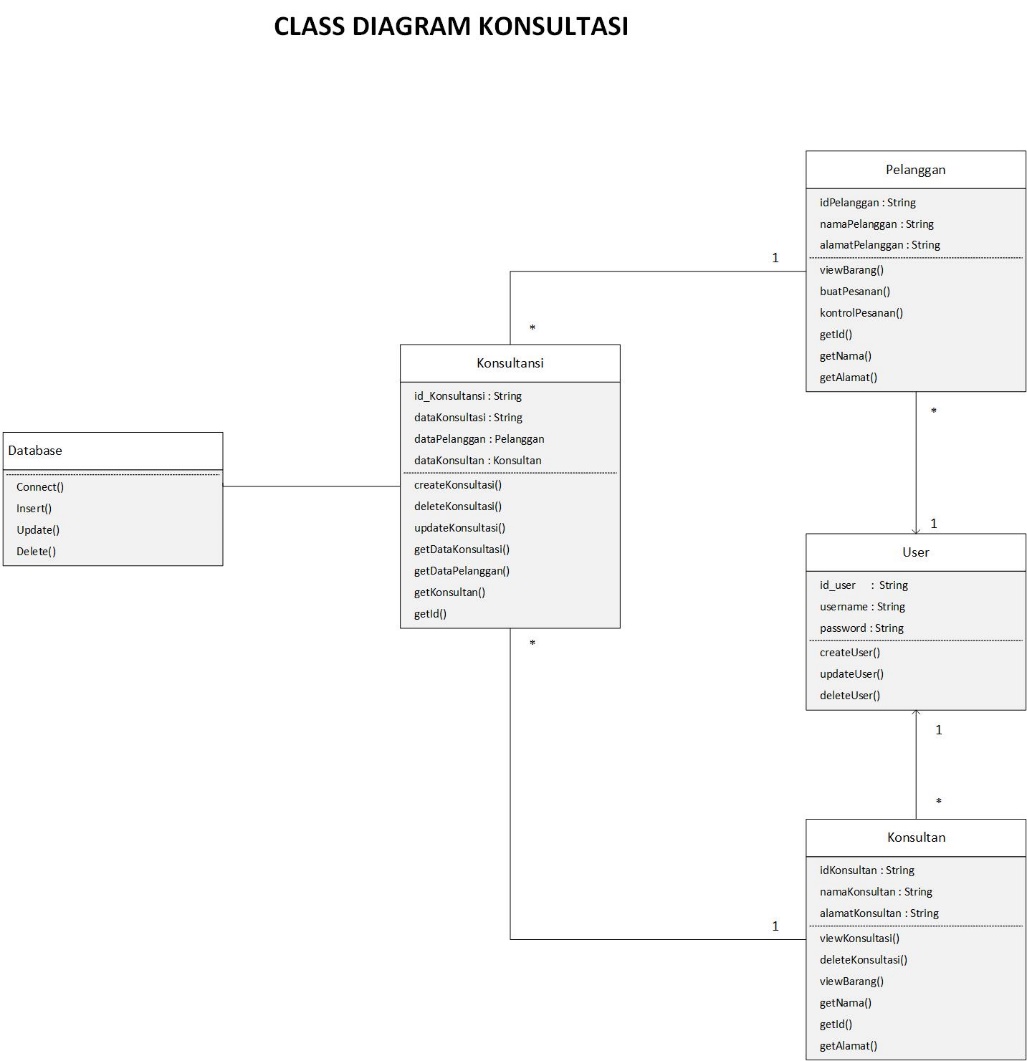
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Tipe Kelas** |
| 1 | User | Entity |
| 2 | Pelanggan | Entity |
| 3 | Konsultasi | Entity |
| 4 | Barang | Entity |
| 5 | Database | Entity |

#### Sequence Diagram Hapus Konsultasi



Gambar 2.8.1 Sequence Diagram Hapus Konsultasi

#### Diagram Kelas Hapus Konsultasi



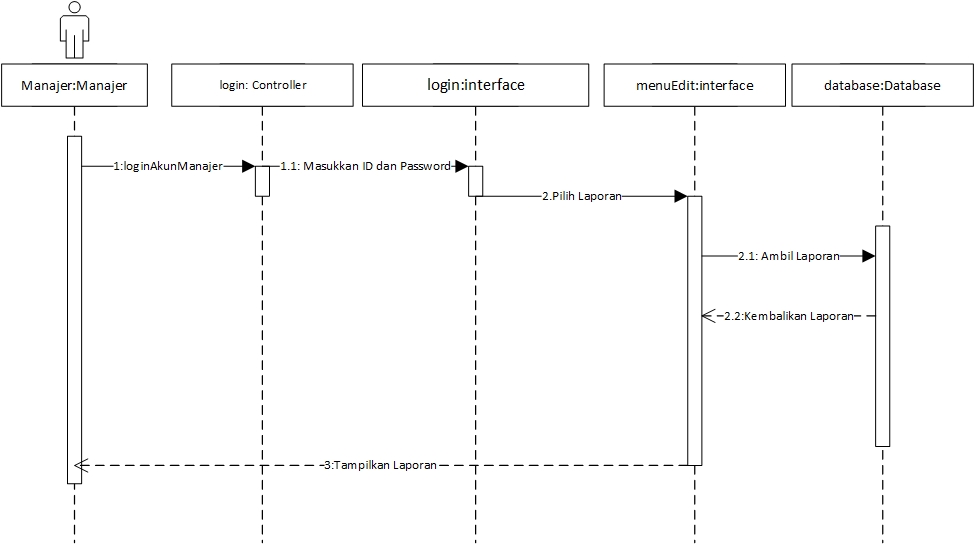
Gambar 2.8.2 Class Diagram Hapus

### Use Case View Laporan Keuangan

#### Identifikasi Kelas View Laporan Keuangan

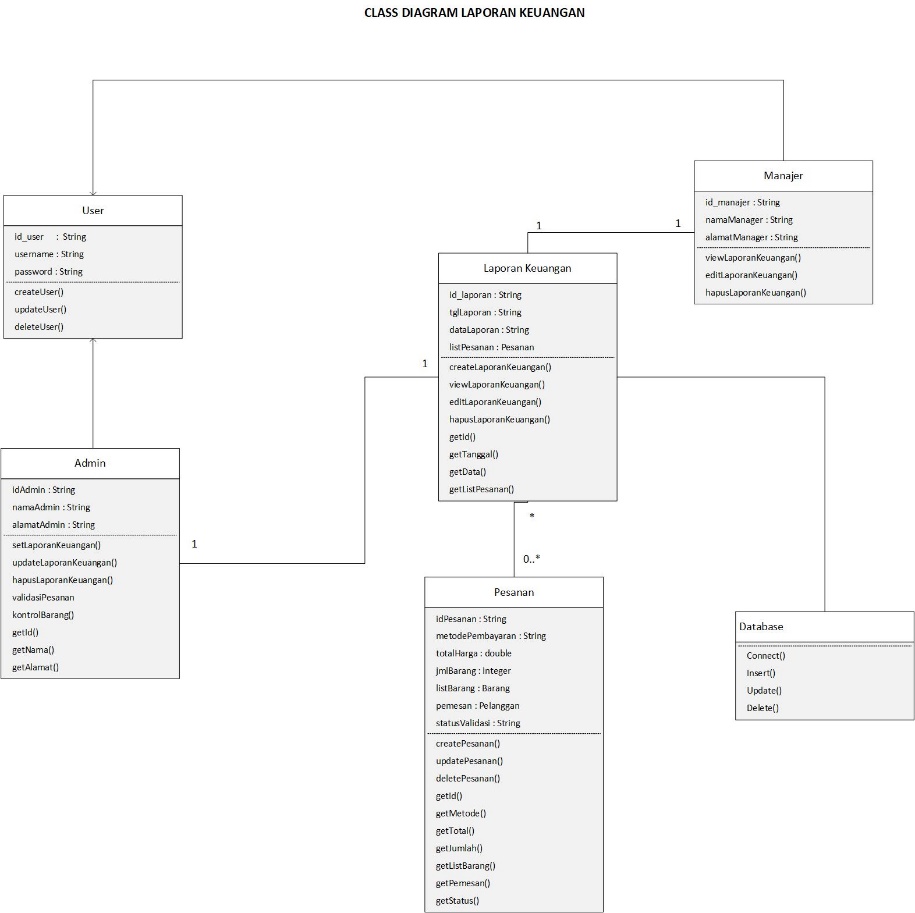
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Tipe Kelas** |
| 1 | User | Entity |
| 2 | Admin | Entity |
| 3 | Manager | Entity |
| 4 | Laporan Keuangan | Entity |
| 5 | Pesanan | Entity |
| 6 | Database | Entity |

#### Sequence Diagram View Laporan Keuangan



Gambar 2.9.1 Sequence Diagram View Laporan Keuangan

#### Diagram Kelas View Laporan Keuangan



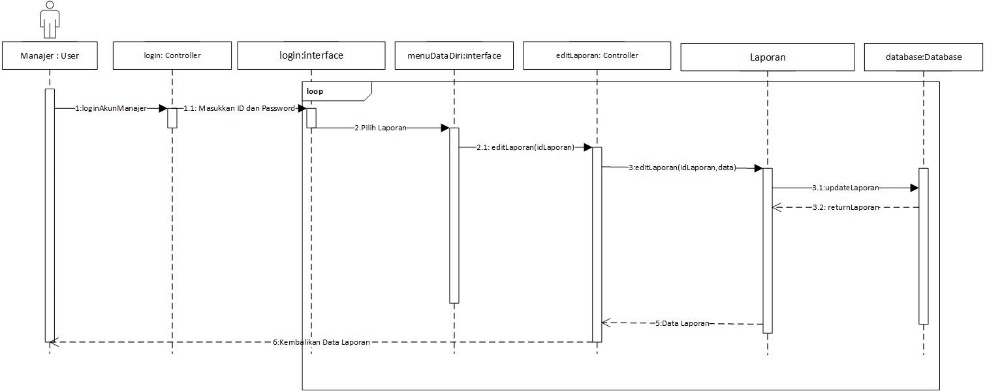
Gambar 2.9.2. Class Diagram Laporan Keuangan

### Use Case Edit Laporan Keuangan

#### Identifikasi Kelas Edit Laporan Keuangan

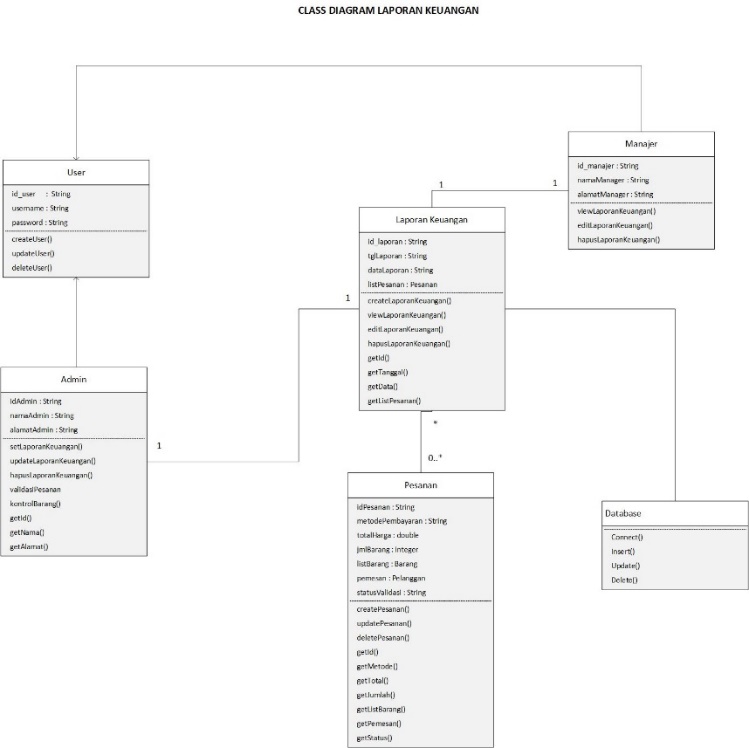
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Tipe Kelas** |
| 1 | User | Entity |
| 2 | Admin | Entity |
| 3 | Manager | Entity |
| 4 | Laporan Keuangan | Entity |
| 5 | Pesanan | Entity |
| 6 | Database | Entity |

#### Sequence Diagram Edit Laporan Keuangan



Gambar 2.10.1 Sequence Diagram Edit Laporan Keuangan

#### Diagram Kelas Edit Laporan Keuangan



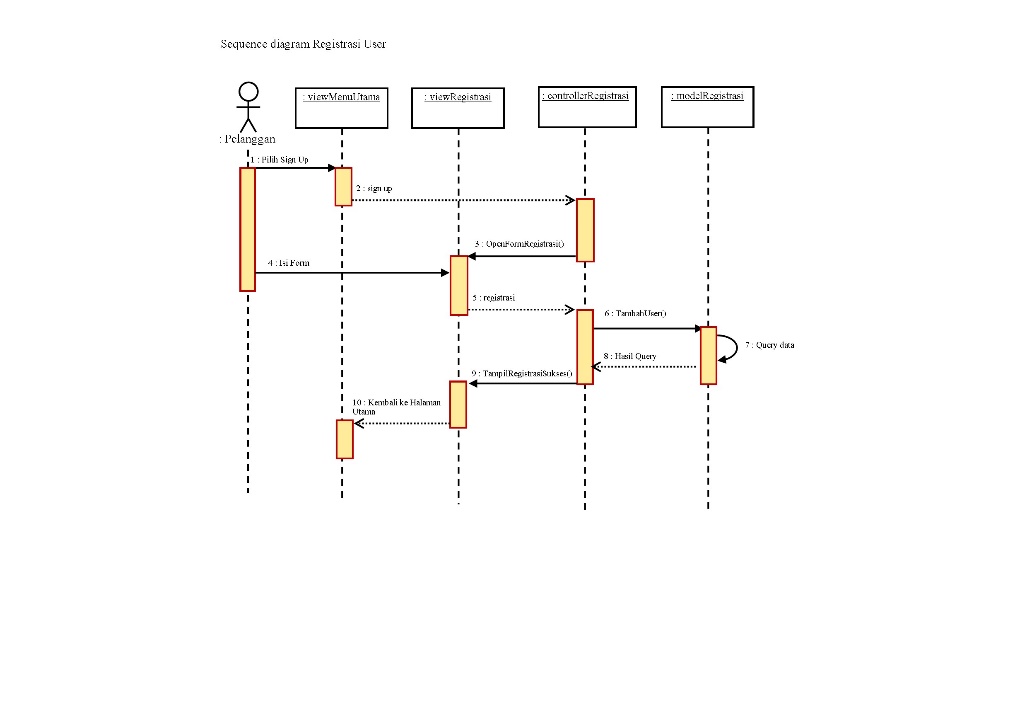
Gambar 2.10.2. Class Diagram Laporan Keuangan

### Use Case Edit Registrasi

#### Identifikasi Kelas Registrasi

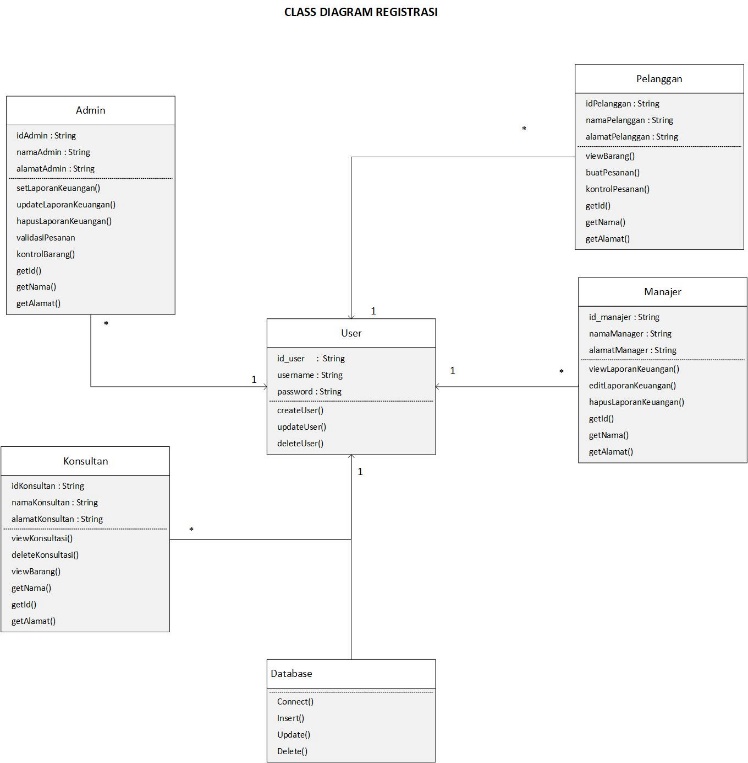
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Tipe Kelas** |
| 1 | User | Entity |
| 2 | Admin | Entity |
| 3 | Manager | Entity |
| 4 | Pelanggan | Entity |
| 5 | Konsultan | Entity |
| 6 | Database | Entity |

#### Sequence Diagram Registrasi



Gambar 2.11.1 Sequence Diagram Registrasi

#### Diagram Kelas Registrasi



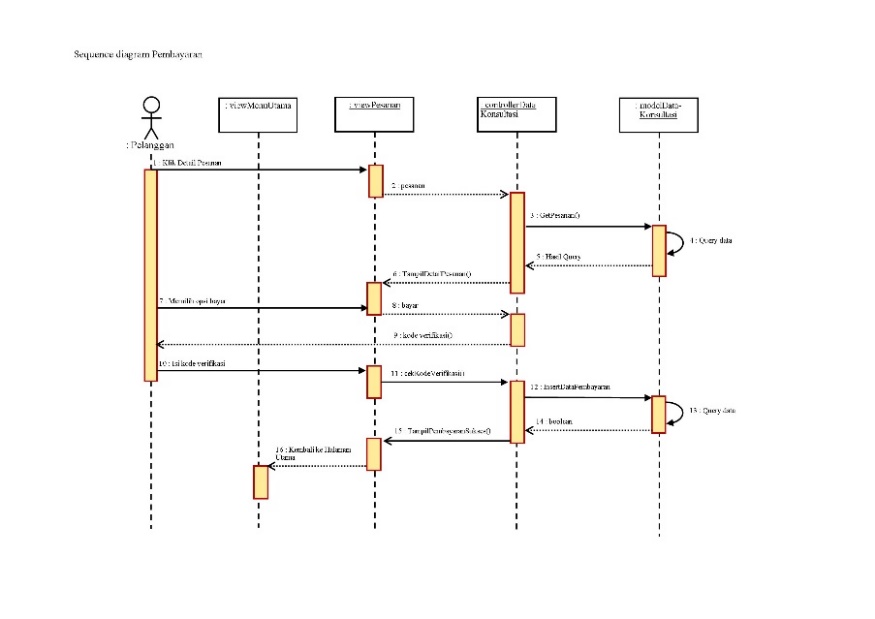
Gambar 2.11.2. Class Diagram Registrasi

### Use Case Pembayaran

#### Identifikasi Kelas Pembayaran

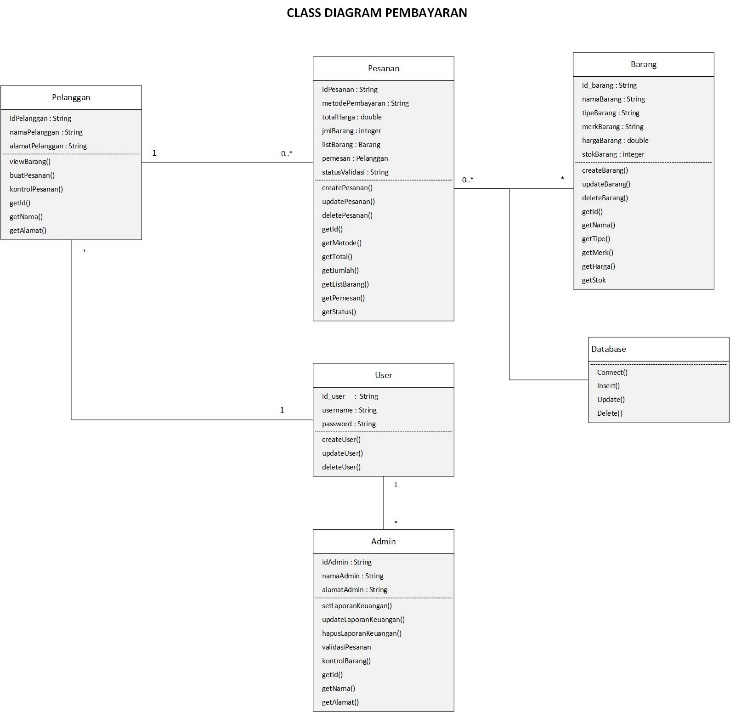
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Tipe Kelas** |
| 1 | User | Entity |
| 2 | Pelanggan | Entity |
| 3 | Pesanan | Entity |
| 4 | Barang | Entity |
| 5 | Database | Entity |

#### Sequence Diagram Pembayaran



Gambar 2.12.1 Sequence Diagram Pembayaran

#### Diagram Kelas Pembayaran



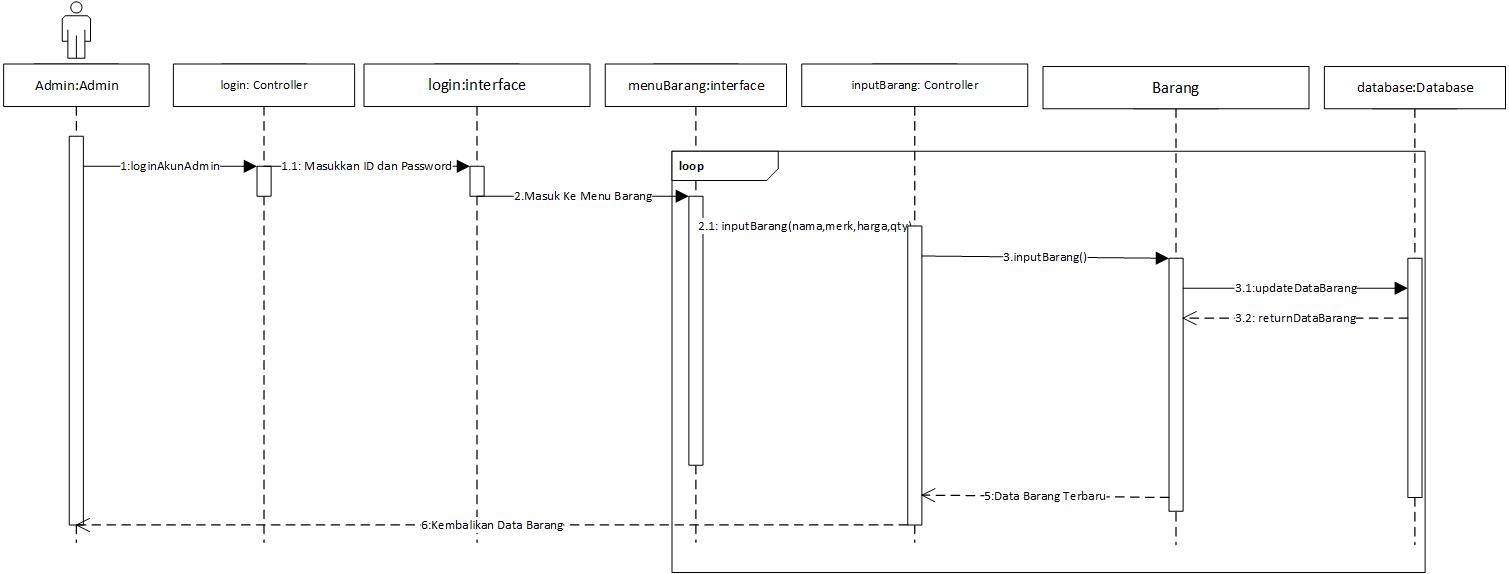
Gambar 2.12.2. Class Diagram Pembayaran

### Use Case Input Barang

#### Identifikasi Kelas Input Barang

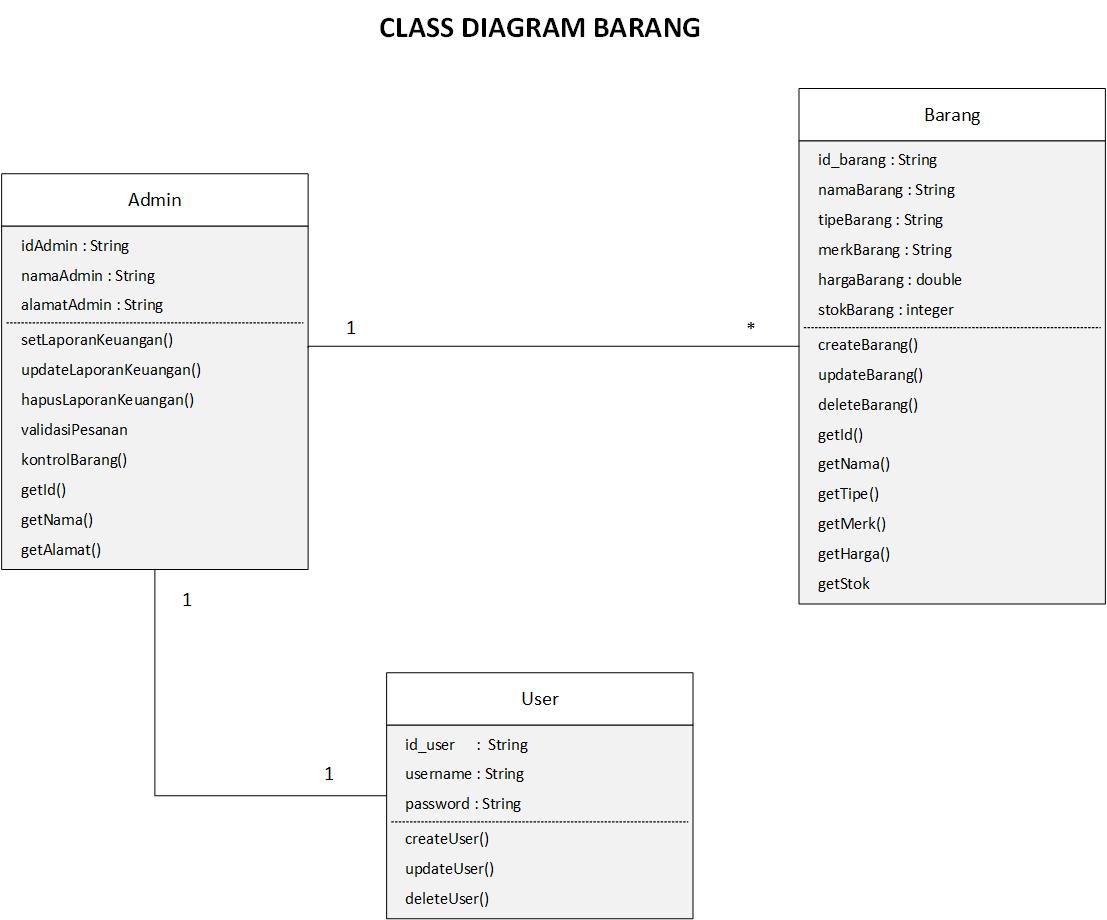
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Tipe Kelas** |
| 1 | User | Entity |
| 2 | Admin | Entity |
| 3 | Barang | Entity |
| 4 | Database | Entity |

#### Sequence Diagram Input Barang



Gambar 2.13.1 Sequence Diagram Input Barang

#### Diagram Kelas Input Barang



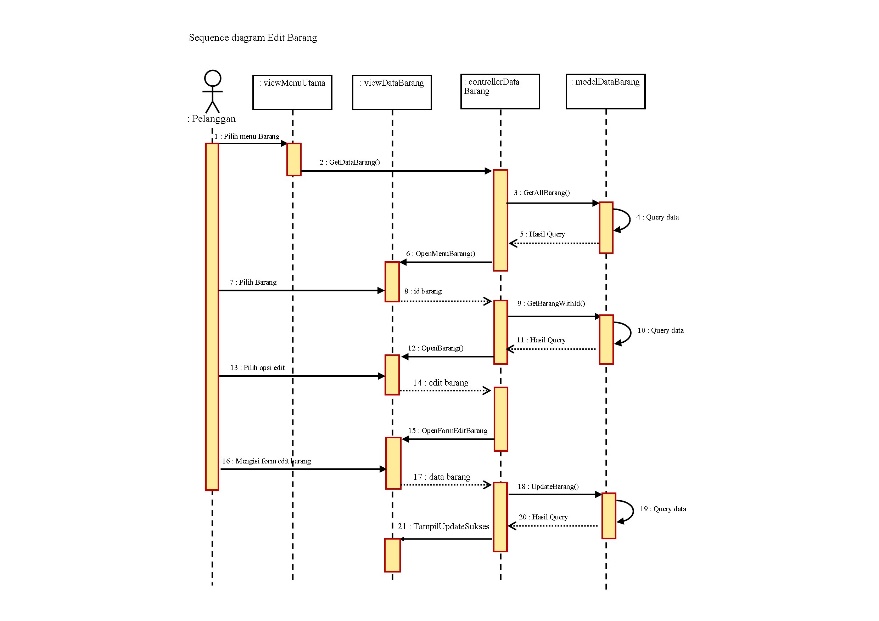
Gambar 2.13.2. Class Diagram Input Barang

### Use Case Edit Barang

#### Identifikasi Kelas Edit Barang

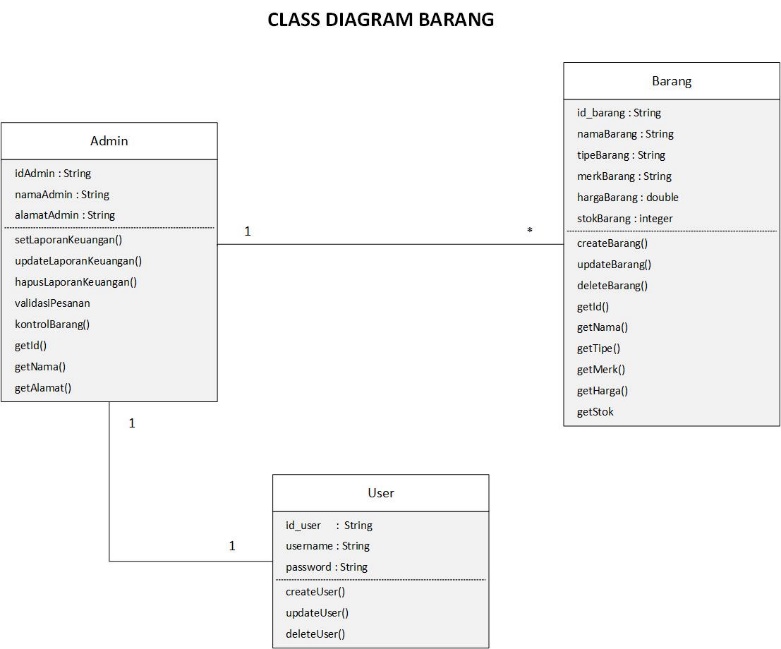
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Tipe Kelas** |
| 1 | User | Entity |
| 2 | Admin | Entity |
| 3 | Barang | Entity |
| 4 | Database | Entity |

#### Sequence Diagram Edit Barang



Gambar 2.14.1 Sequence Diagram Edit Barang

#### Diagram Kelas Edit Barang



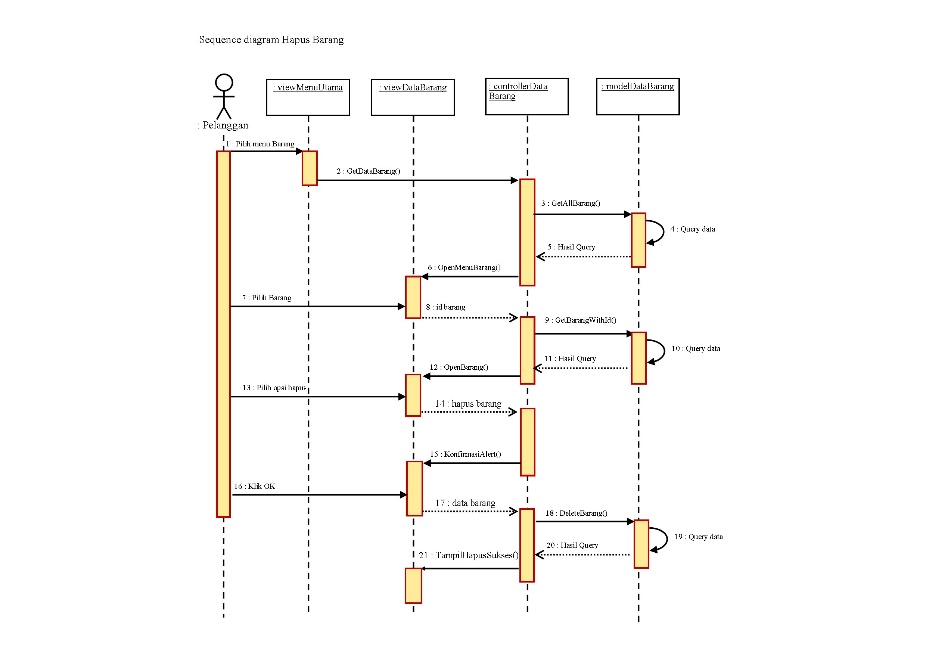
Gambar 2.14.2. Class Diagram Edit Barang

### Use Case Hapus Barang

#### Identifikasi Kelas Hapus Barang

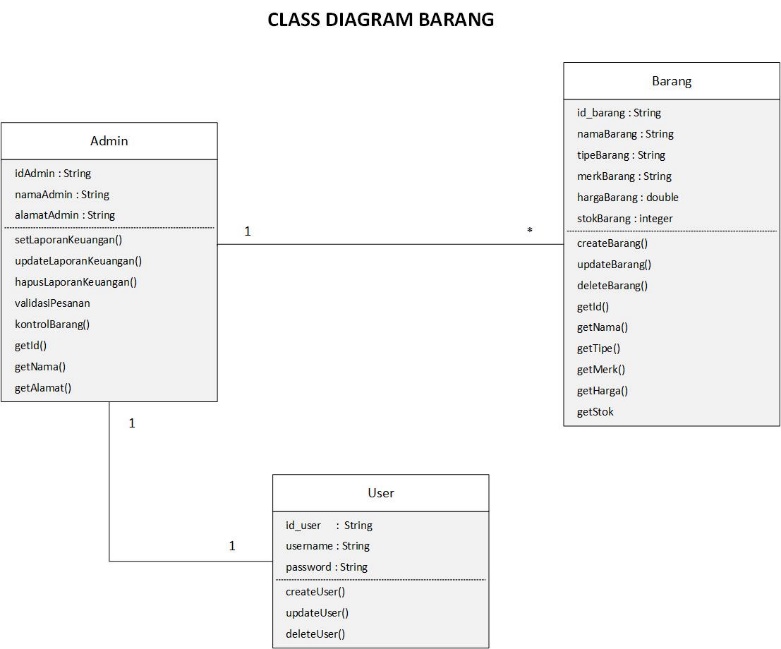
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Tipe Kelas** |
| 1 | User | Entity |
| 2 | Admin | Entity |
| 3 | Barang | Entity |
| 4 | Database | Entity |

#### Sequence Diagram Hapus Barang



Gambar 2.15.1 Sequence Diagram Hapus Barang

#### Diagram Kelas Hapus Barang



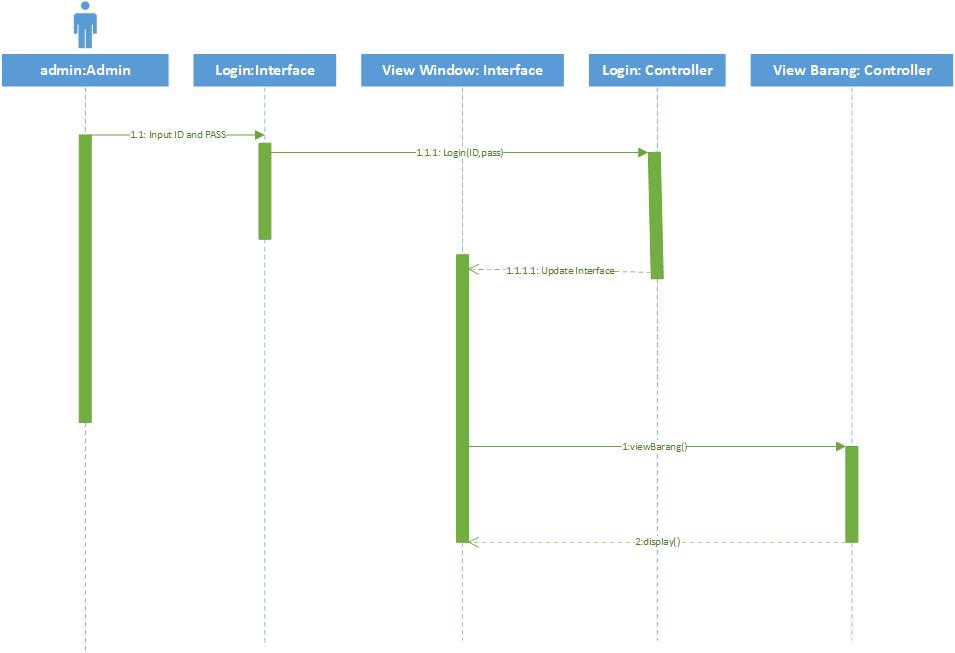
Gambar 2.15.2. Class Diagram Hapus Barang

### Use Case View Barang

#### Identifikasi Kelas View Barang

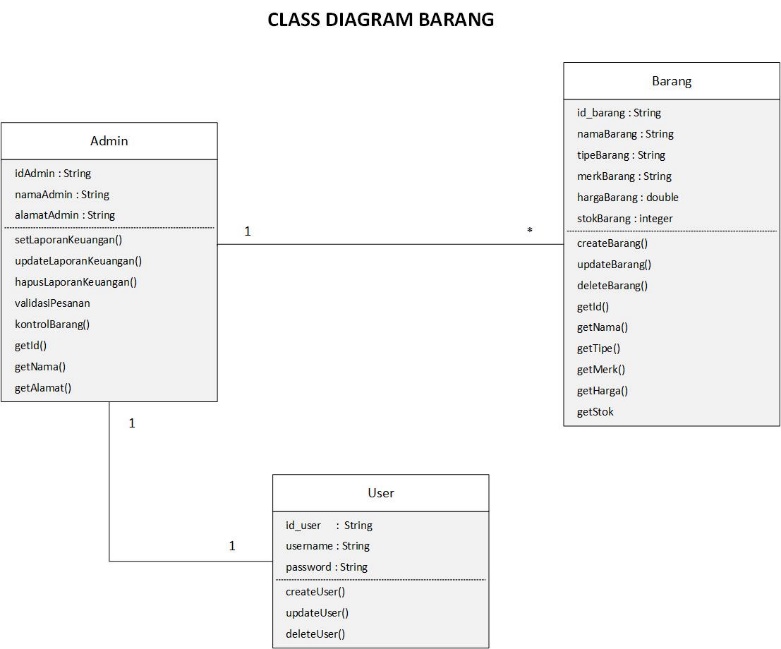
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Tipe Kelas** |
| 1 | User | Entity |
| 2 | Admin | Entity |
| 3 | Barang | Entity |
| 4 | Database | Entity |

#### Sequence Diagram View Barang



Gambar 2.16.1 Sequence Diagram View Barang

#### Diagram Kelas Hapus Barang



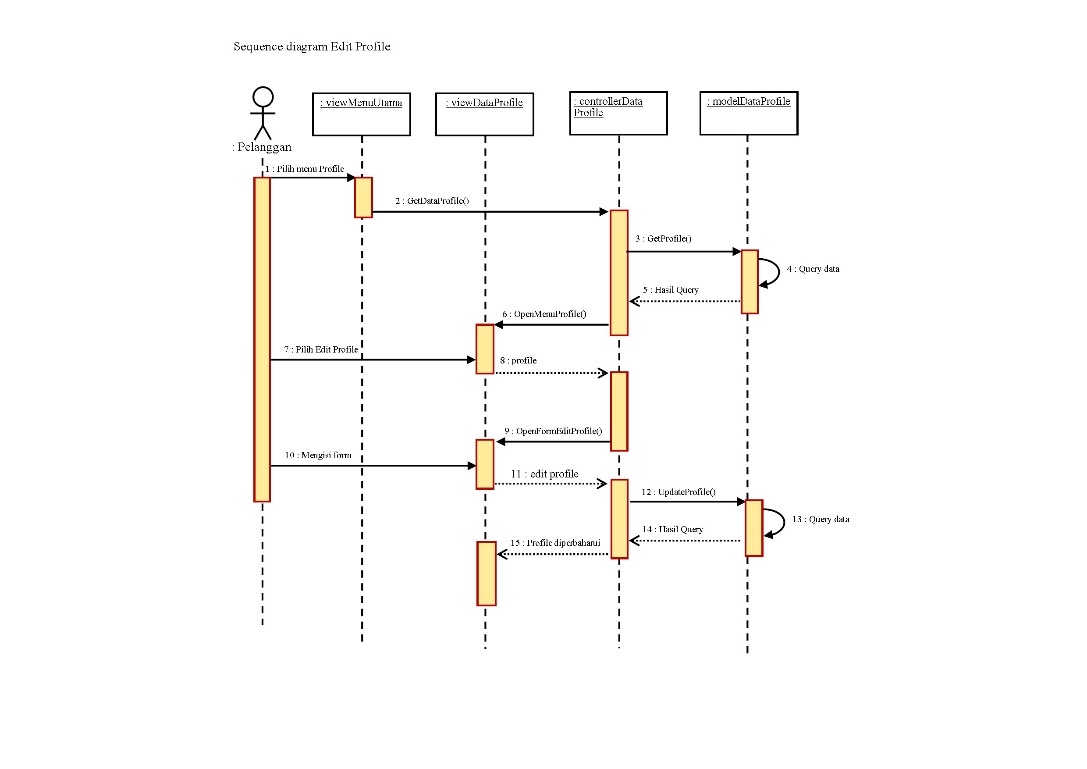
Gambar 2.16.2. Class Diagram View Barang

### Use Case Edit Edit Profile

#### Identifikasi Kelas Edit Profile

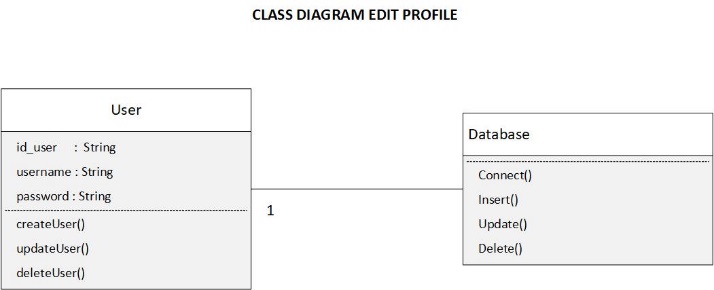
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Tipe Kelas** |
| 1 | User | Entity |
| 2 | Database | Entity |

#### Sequence Diagram Edit Profile



Gambar 2.17.1 Sequence Diagram Edit Profile

#### Diagram Kelas Edit Profile



Gambar 2.10.2. Class Diagram Edit Profile

## 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 |

### Kelas Barang

Nama Kelas : Barang

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| createBarang() | public | Operasi ini adalah operasi membuat objek Barang dan memasukkan semua keperluannya. |
| 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 |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| Id\_barang | private | String |
| namaBarang | private | String |
| tipeBarang | private | String |
| merkBarang | private | String |
| hargaBarang | private | Double |
| stokBarang | private | Integer |

### Kelas Pelanggan

Nama Kelas : Pelanggan

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| 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 |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| Id\_pelanggan | private | String |
| namaPelanggan | private | String |
| alamatPelanggan | private | String |

### Kelas Konsultasi

Nama Kelas : Konsultasi

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| 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 | Digunakna untuk mendapatkan Pelanggan siapa yang sedang melakukan konsultasi |
| getDataKonsultan | Public | Untuk mendapatkan Data dari Konsultan yang sedang menanggani Konsultasi. |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| Id\_konsultasi | private | String |
| dataKonsultasi | private | Konsultasi |
| dataPelanggan | private | Pelanggan |
| dataKonsultan | private | Konsultan |

### Kelas Konsultan

Nama Kelas : Konsultan

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| 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 | Digunakna untuk mendapatkan Alamat dari seorang Konsultan |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| Id\_konsultan | private | String |
| namaKonsultan | private | String |
| alamatKonsultan | private | String |

### Kelas User

Nama Kelas : User

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| createUser() | Public | Digunakan untuk membuat user |
| updateUser() | public | Digunakan untuk mengupdate atribut user |
| deleteUser() | public | Digunakan untuk menghapus user dari sistem |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| Id\_user | private | String |
| Username | private | String |
| Password | private | String |

### Kelas Admin

Nama Kelas : Admin

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| 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 |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| Id\_admin | private | String |
| Nama\_admin | private | String |
| alamatAdmin | private | String |

### Kelas Laporan Keuangan

Nama Kelas : Laporan Keuangan

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| 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 |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| Id\_laporan | private | String |
| tglLaporan | private | String |
| dataLaporan | private | String |
| listPesanan | private | Pesanan |

### Kelas Manajer

Nama Kelas : Manajer

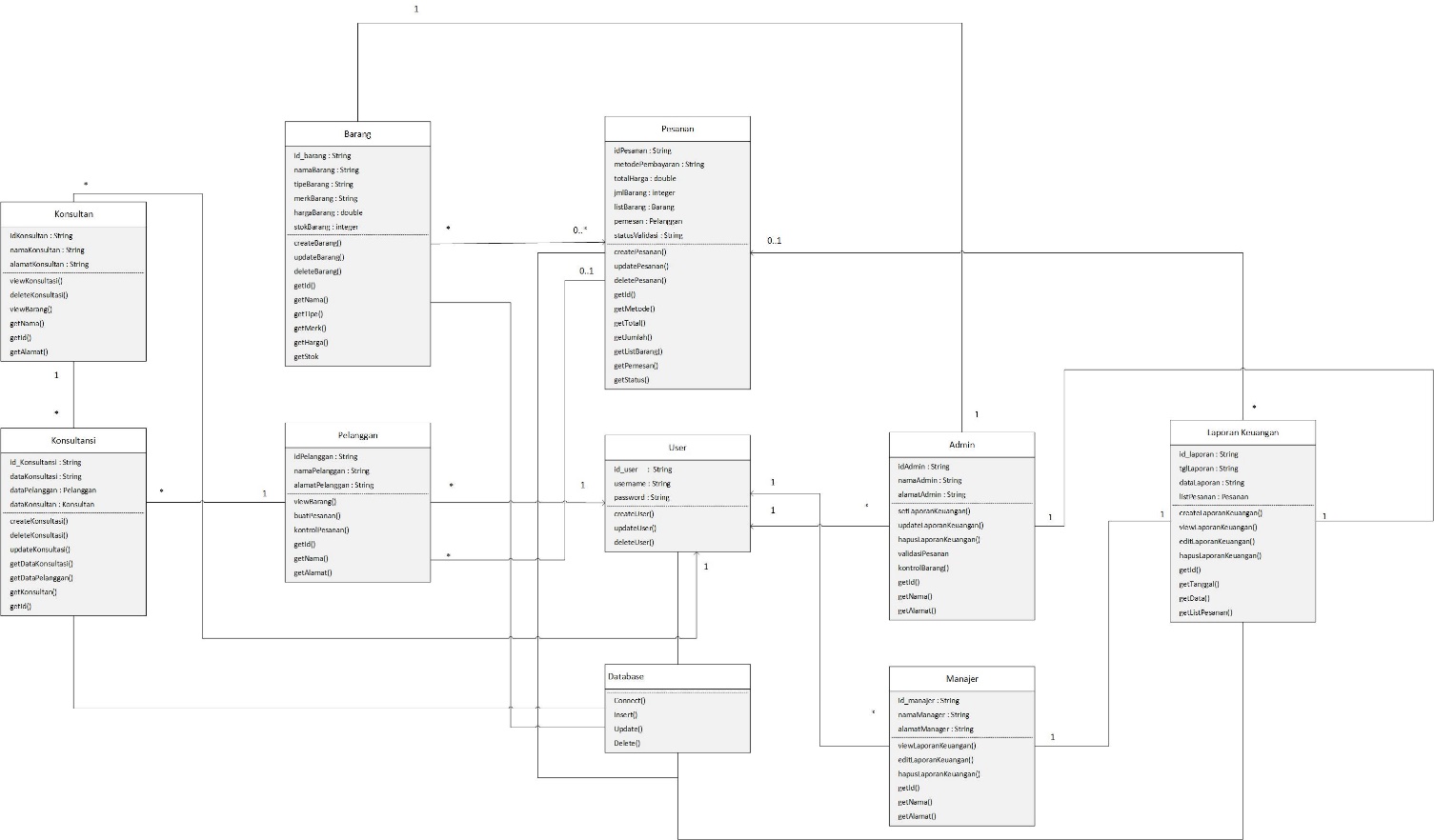
|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| viewLaporanKeuangan() | public | Digunakan melihat laporan keuangan |
| editLaporanKeuangan() | public | Digunakan untuk mengubah isi dari laporan keuangan |
| hapusLaporanKeuangan() | 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 |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| Id\_manager | private | String |
| namaManager | private | String |
| alamatManager | private | String |

### Kelas Pesanan

Nama Kelas : Pesanan

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| 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 |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| Id\_pesanan | private | String |
| metodePembayaran | private | String |
| totalHarga | private | double |
| jmlBarang | private | Integer |
| listBarang | private | Barang |
| pemesan | private | Pelanggan |
| statusValidasi | private | String |

## Diagram Kelas Keseluruhan



Gambar 4 Class Diagram

## 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, ….  FROM barang where Field n = ‘Kondisi’; | Digunakan untuk mengambil data tertentu dengan penkondisian tertentu pada tabel barang |

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

alamatPelangan = 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

alamatPelangan = 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 Dikosongkan’)

}

}

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’)

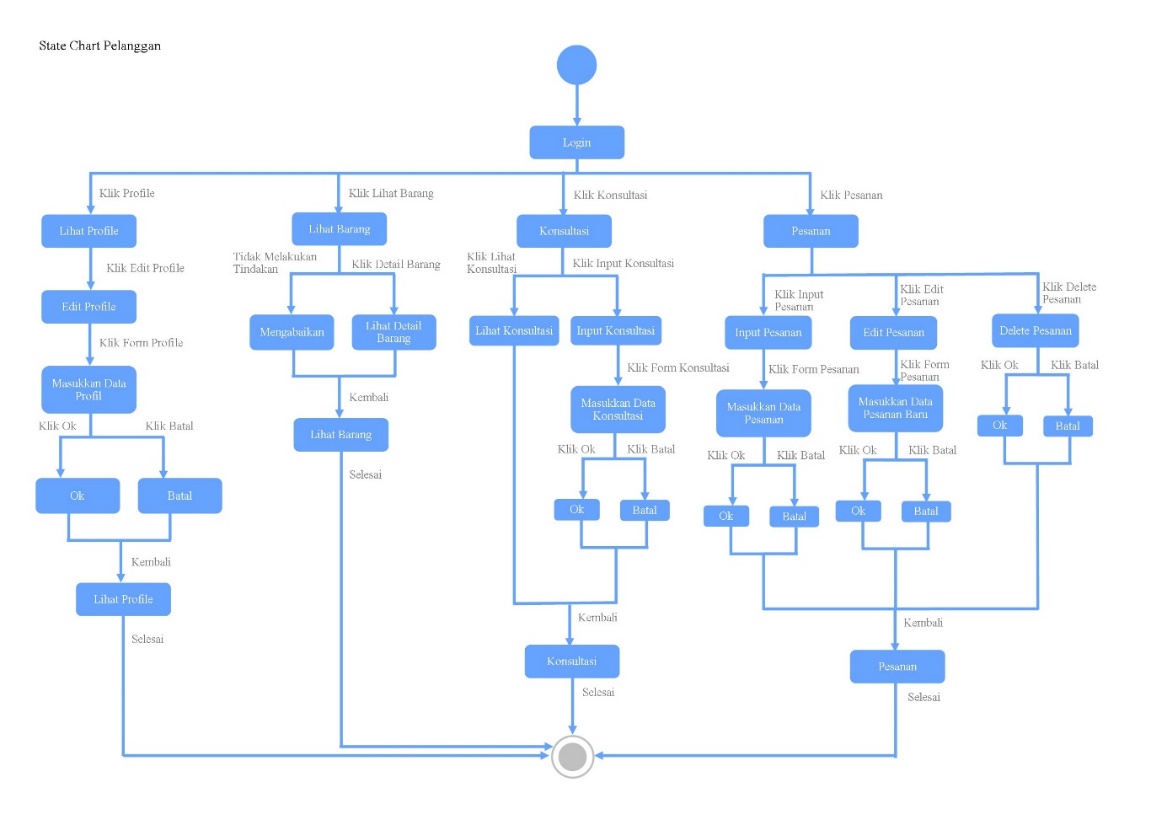
output(Laporan);

}

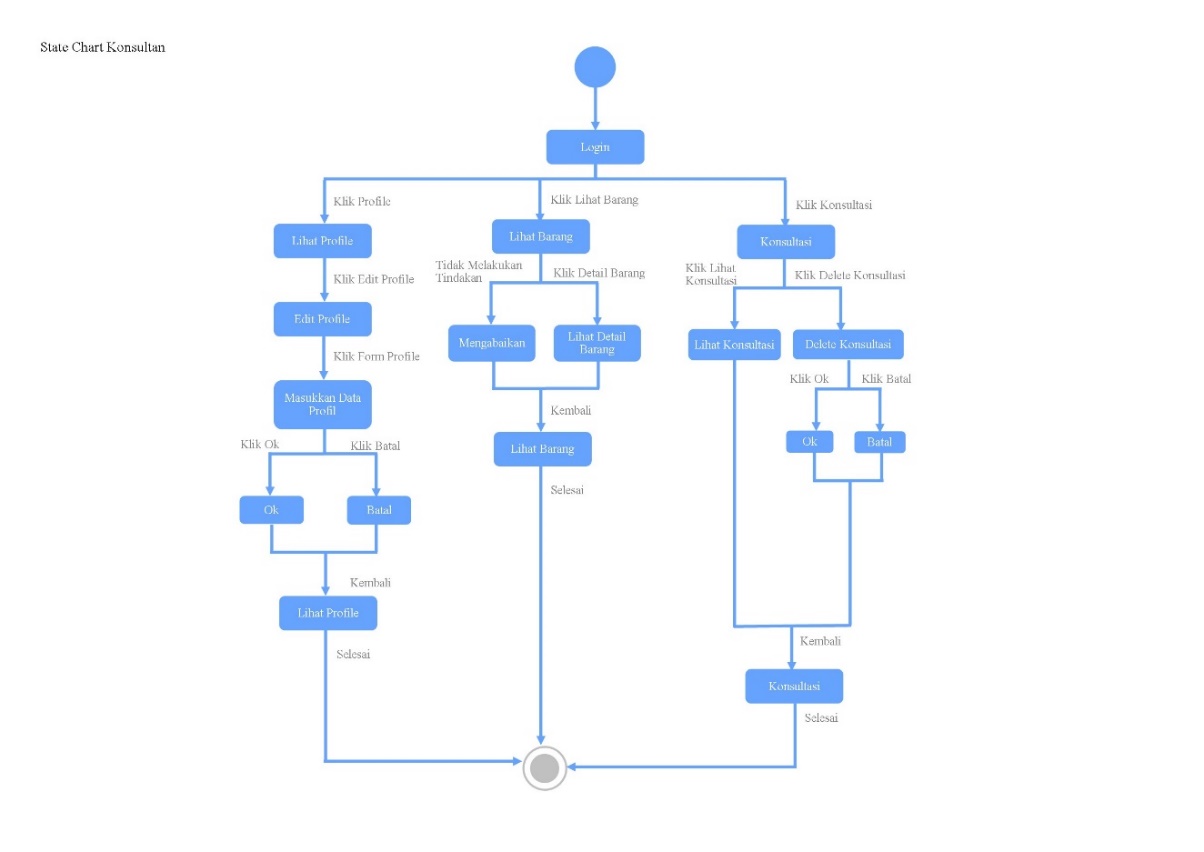
*Query :* Digunakan untuk methods pada Class Barang, yaitu CRUD data ke database

|  |  |  |
| --- | --- | --- |
| **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 |

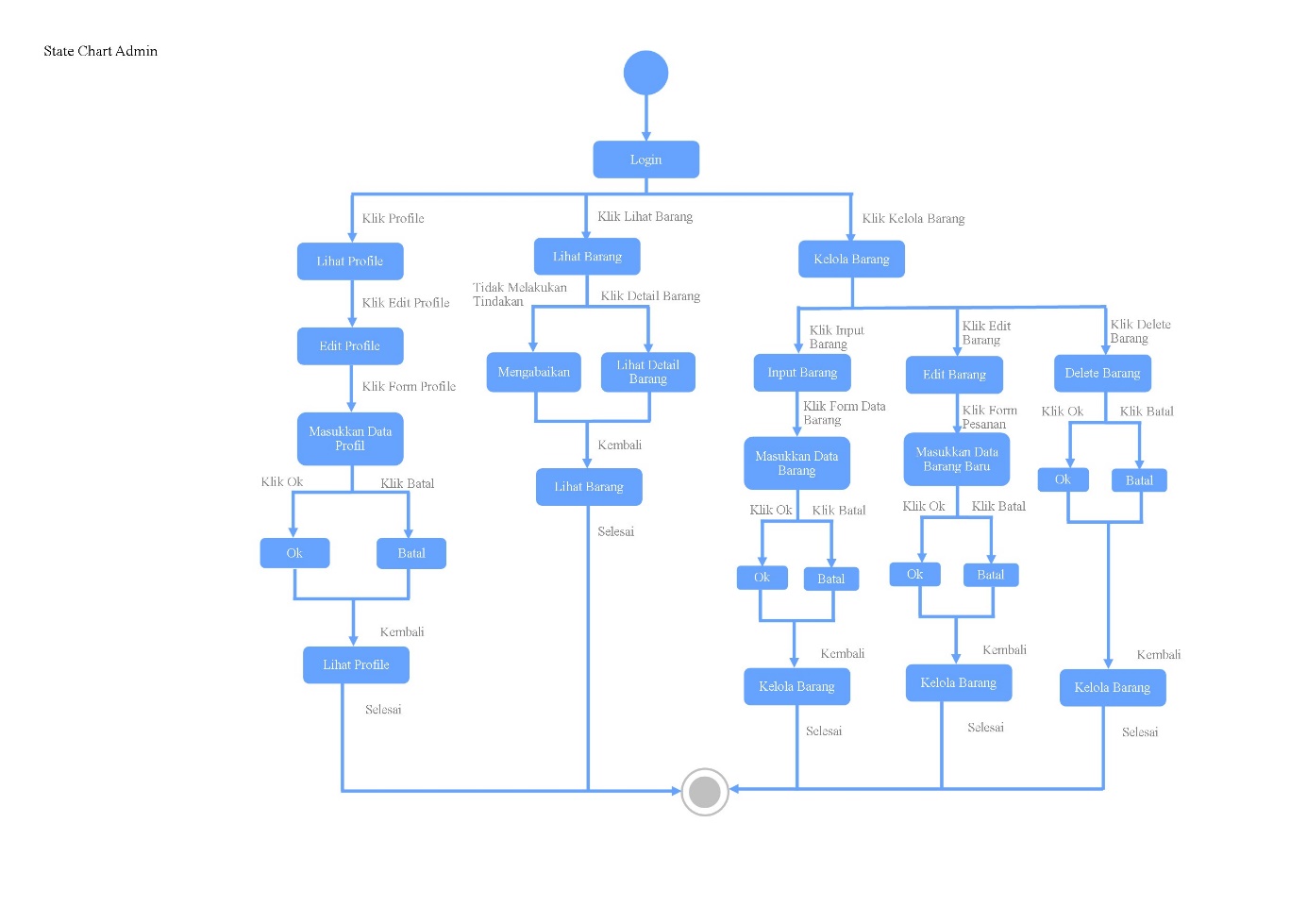
## Statechart Diagram



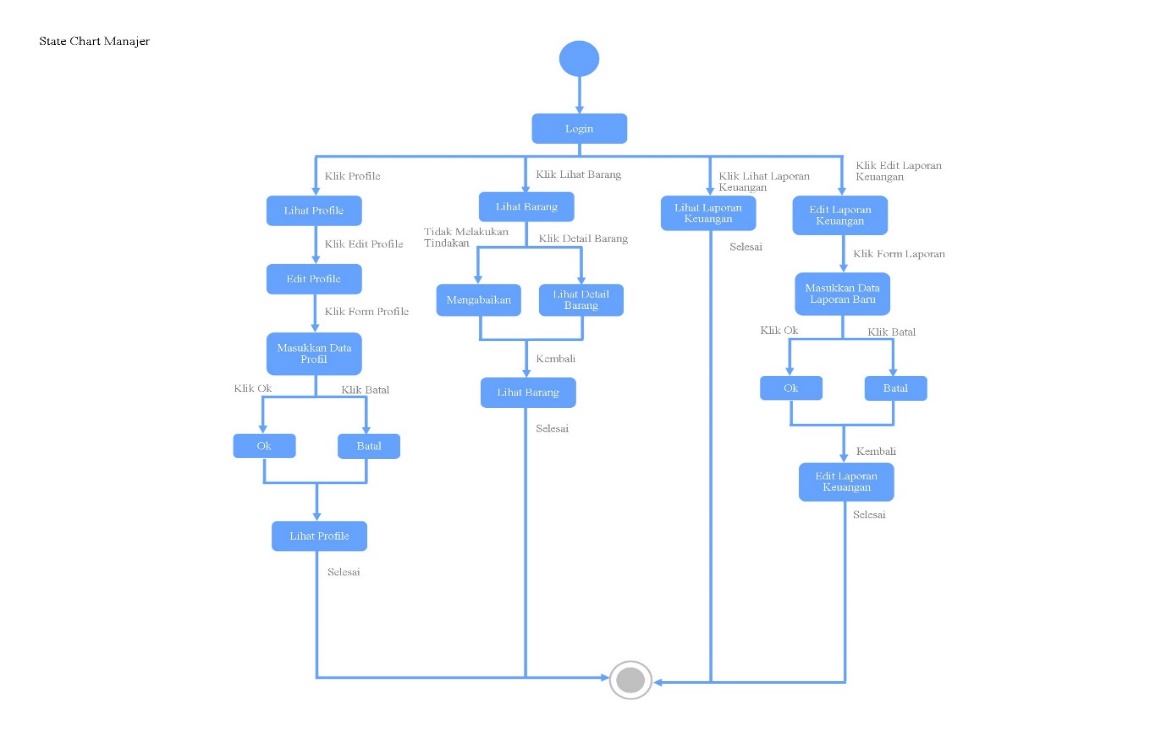
Gambar Statechart Pelanggan



Gambar Statechart Konsultan

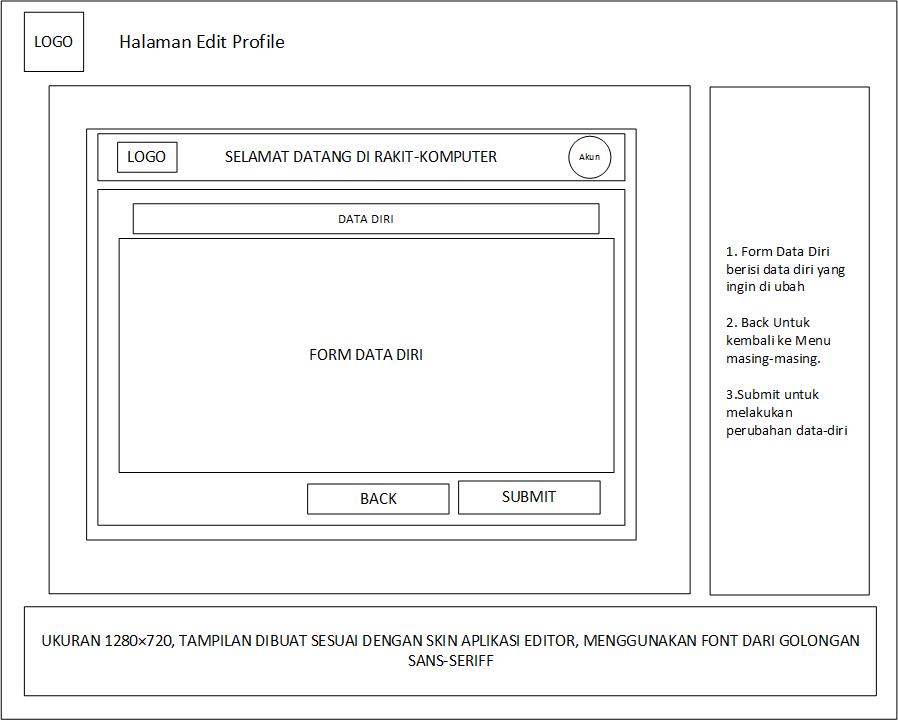


Gambar Statechart Admin



Gambar Statechart Manajer

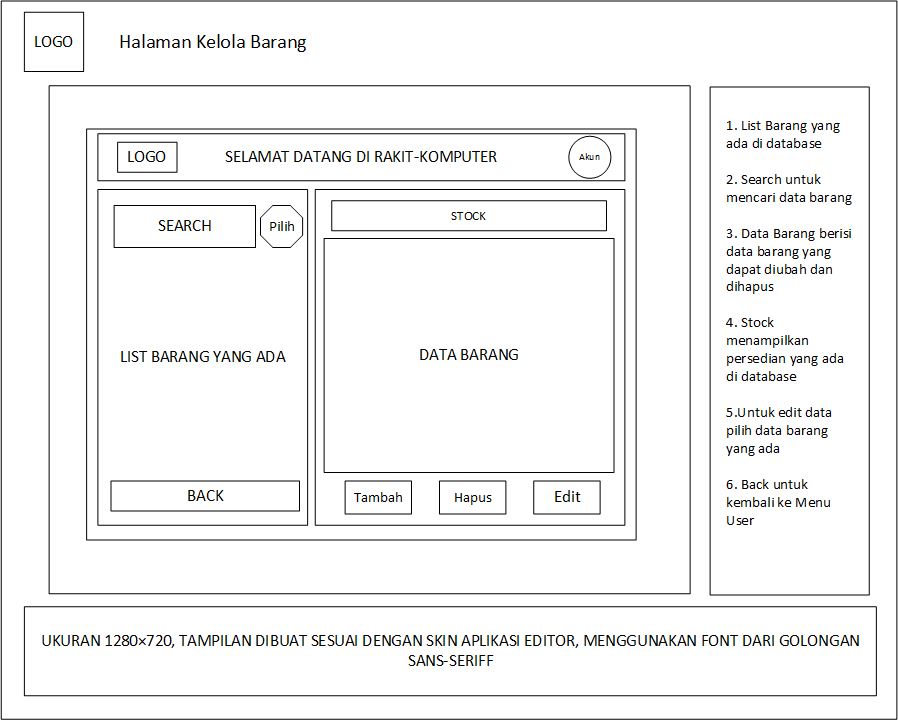
## Perancangan Antarmuka

**

Perancangan Muka 1 Edit Profile

*Antarmuka : Perancangan Muka 1 Edit Profile*

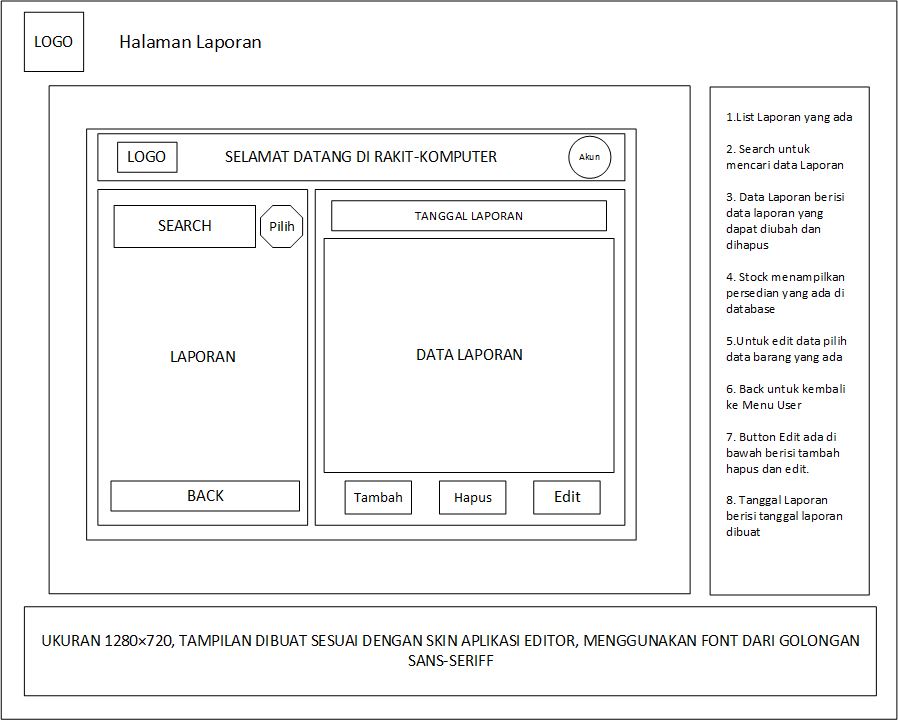
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| Button 1 | Button | Submit | Button diklik maka akan melakukan proses submit perubahan yang dilakukan didasarkan isian dari Form, dan kemudian akan di proses oleh Controller dan kemudian akan di kembalikan 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*

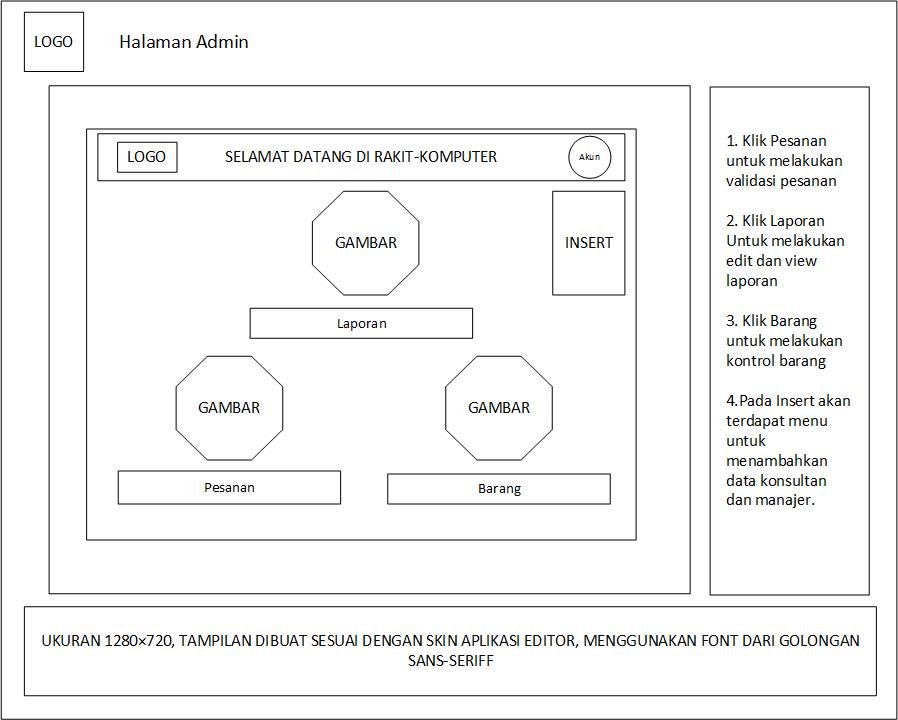
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| Button 2 | Button | Back | Jika Diklik maka akan menuju Kembali ke halaman utama dari seorang Admin. |
| Button 3 | Button | Tambah | Button ini digunakan untuk menambahkan data barang dengan data yang 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*

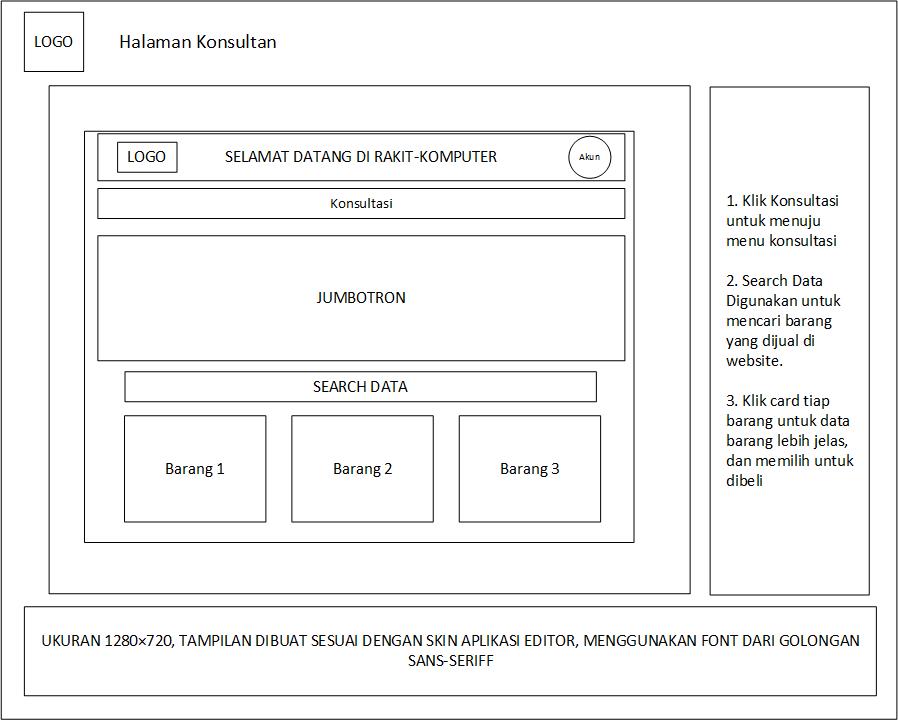
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| Button 7 | Button | Back | Jika Diklik maka akan menuju Kembali ke halaman utama dari seorang Admin/Manajer. |
| Button 8 | Button | Tambah | Button ini digunakan untuk menambahkan data Laporan dengan 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*

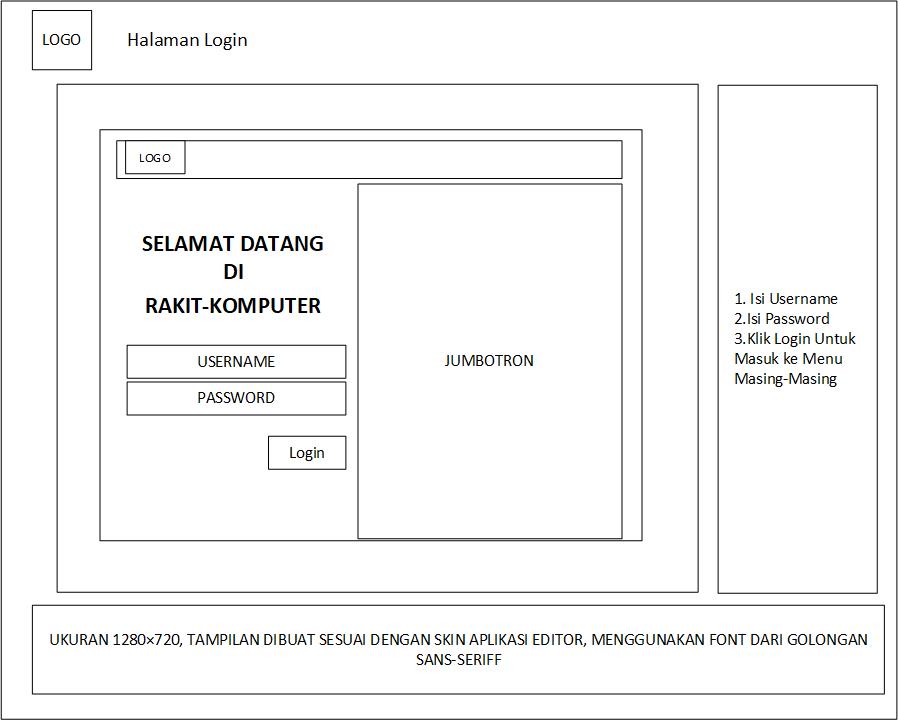
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| Button 12 | Button | Laporan | Jika diklik akan menuju ke halaman Laporan dan dapat dilakukan control laporan oleh Admin. |
| Button 13 | Button | Pesanan | Jika diklik,akan Menuju ke halaman pesanan dan 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 Interakti 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 konsutan dan menghapusnya dan juga dapat menghapus manajer karena nanti akan ada 2 pilihan |

**

Perancangan Muka 5 Halaman Konsultan

*Antarmuka : Perancangan Muka 5 Halaman Konsultan*

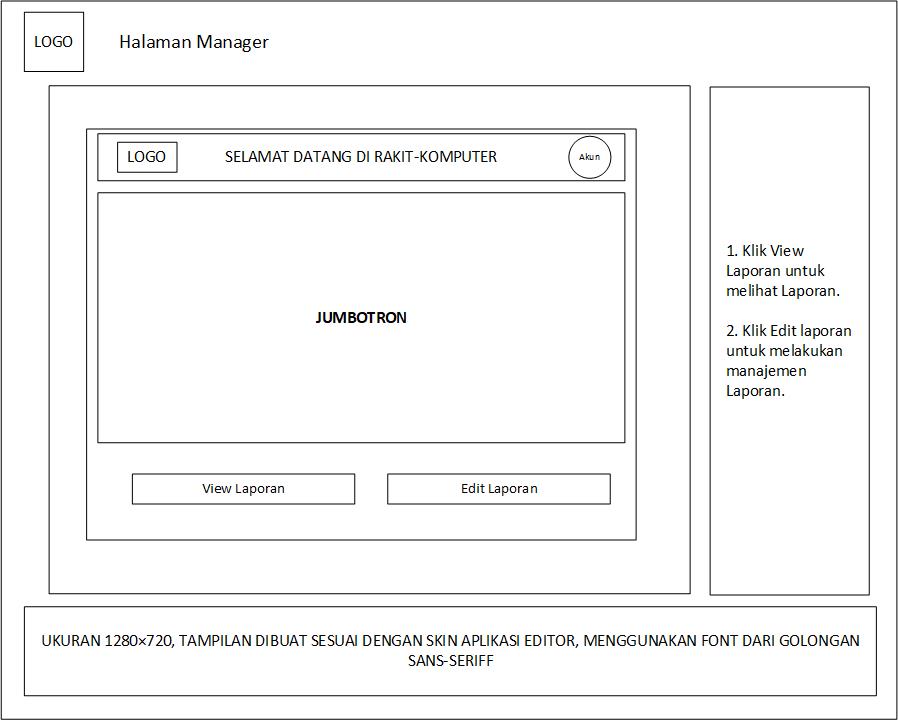
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| Card 1 | Card | Barang 1 | Card ini menampilkan salah satu barang yang di jual atau barang yang ada didatabase 1 baris dari card ini terdiri dari 3 card. |
| Card 2 | Card | Barang 2 | Card ini menampilkan 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. |

**

Perancangan Muka 6 Halaman Login

*Antarmuka : Perancangan Muka 6 Halaman Login*

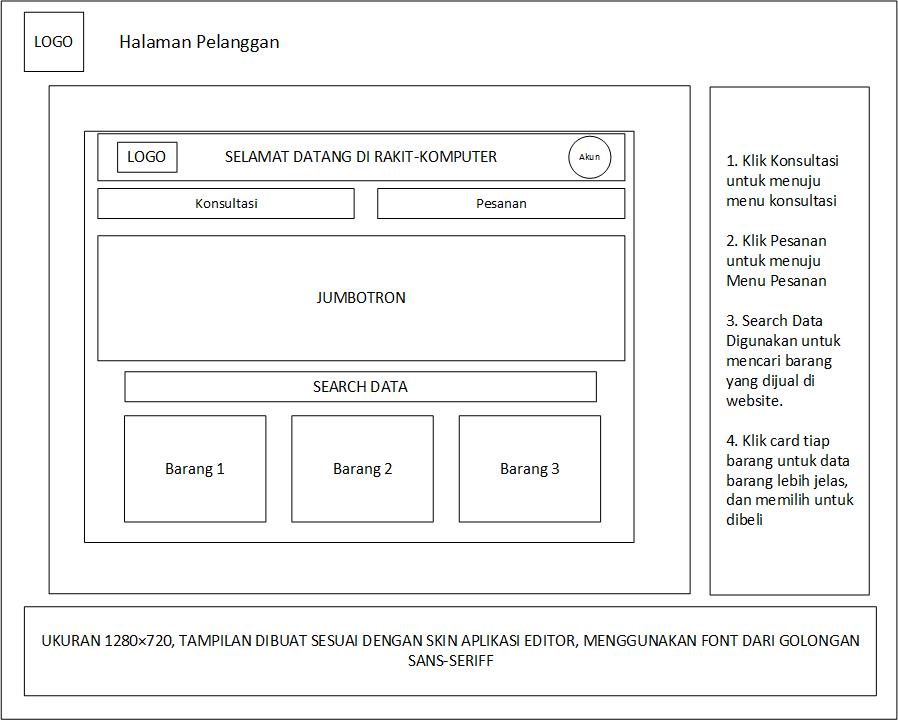
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| Jumbotron 2 | Jumbotron | Jumbotron | File berupa gambar yang bergerak menunjukkan pesan atau barang yang sedang memiliki penjualan yang baik. |
| Button 16 | Button | Login | Jika Diklik nantinya akan 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 |

**

Perancangan Muka 7 Halaman Manager

*Antarmuka : Perancangan Muka 7 Halaman Manager*

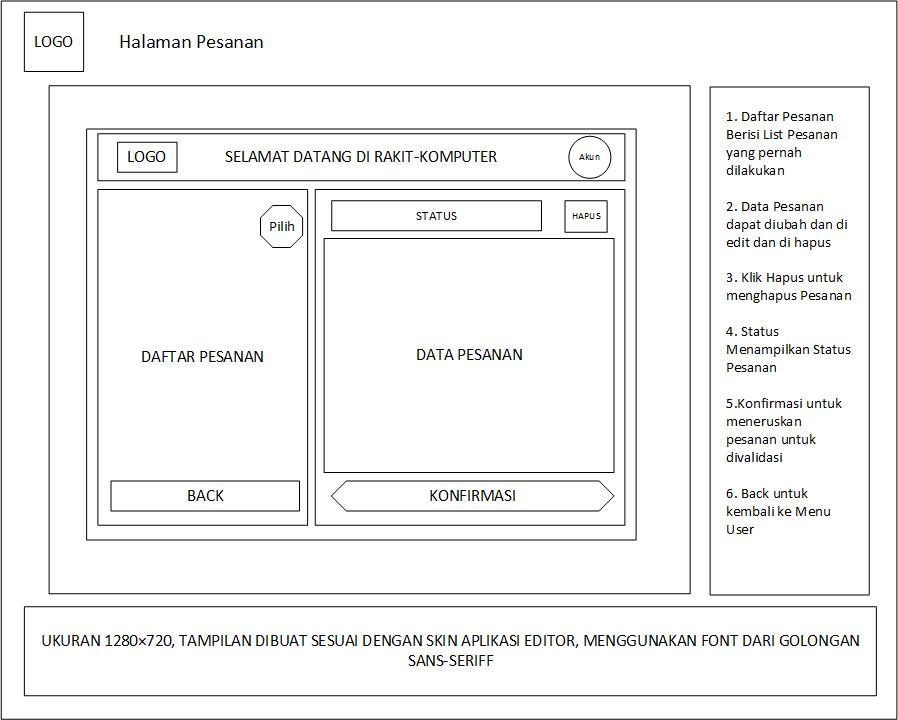
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| Jumbotron 3 | Jumbotron | Jumbotron | File berupa gambar yang bergerak menunjukkan pesan atau barang yang sedang memiliki penjualan yang baik. |
| Button 17 | Button | View Laporan | Jika diklik, akan menuju ke halaman laporan. |
| Button 18 | Button | Edit Laporan | Jika diklik akan menuju halaman laporan |
| Dropdown 7 | Dropdown | Akun | Digunakan untuk mengakses Profile dari User dan terdapat dropdown menuju editProfile(). |

**

Perancangan Muka 8 Halaman Pelanggan

*Antarmuka : Perancangan Muka 8 Halaman Pelanggan*

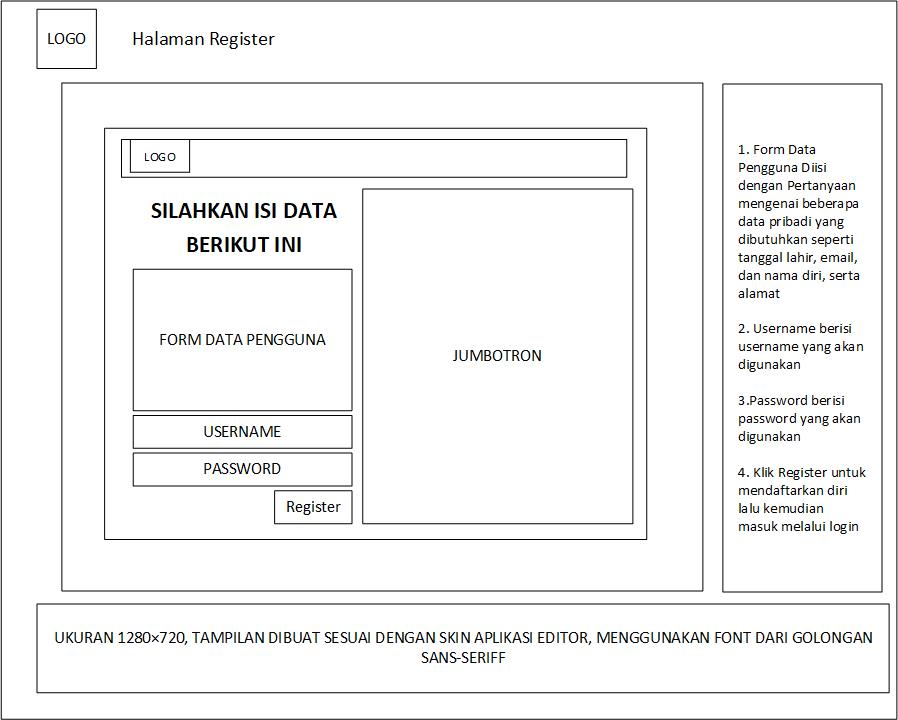
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| Card 4 | Card | Barang 1 | Card ini menampilkan salah satu barang yang di jual atau barang yang ada didatabase 1 baris dari card ini terdiri dari 3 card. |
| Card 5 | Card | Barang 2 | Card ini menampilkan 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*

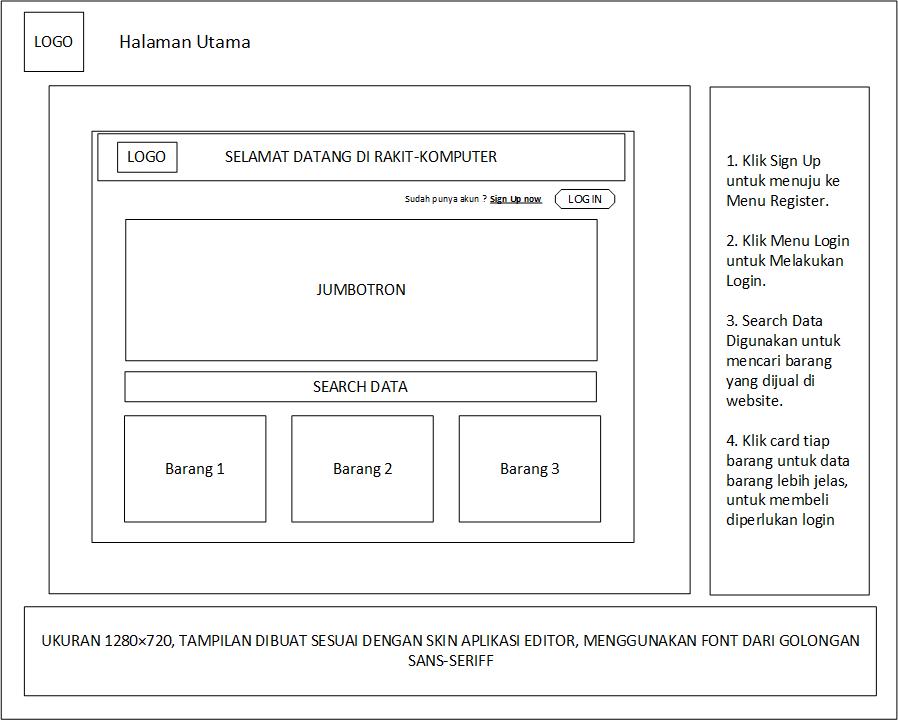
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| Table 3 | Table | Daftar Pesanan | Diisi dengan penjelasan reaksi sistem, misalnya membuka layar apa, link kemana. Jika menyangkut suatu kode yang cukup rumit, acu algoritma yang telah diuraikan di atas. |
| 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 | Konfimasi | 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(). |

**

Perancangan Muka 10 Halaman Register

*Antarmuka : Perancangan Muka 10 Halaman Register*

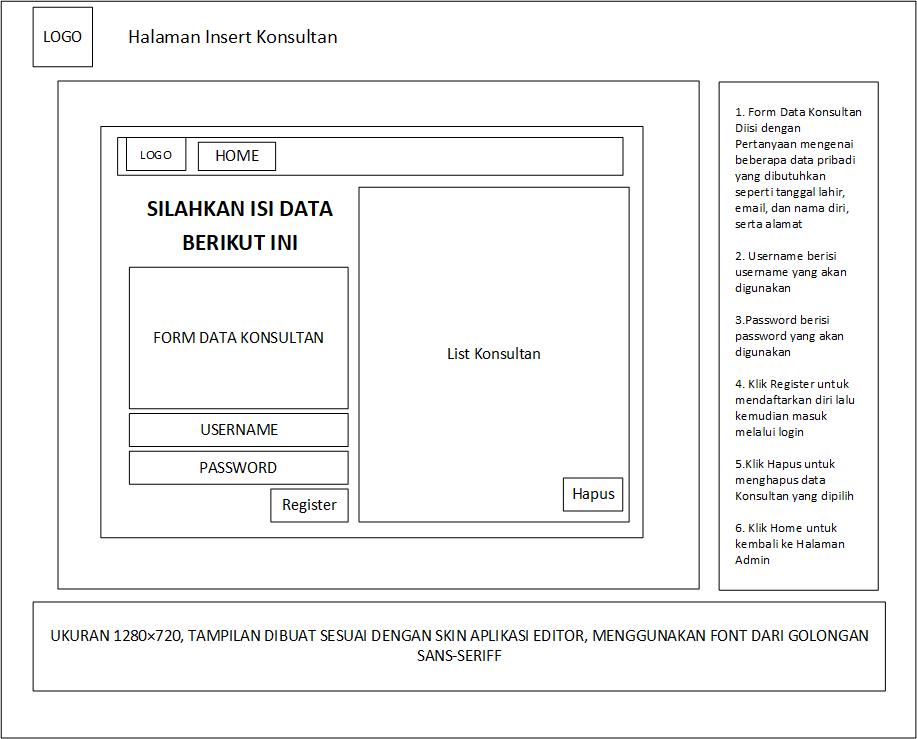
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| FormRegist | Form | Form Data Pengguna | Berisi Sekumpulan isian input yang digunakan nantinya untuk Algoritma Register. |
| Input 3 | Input Text | Username | Isi Text Username yang tersedia dan unique |
| 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*

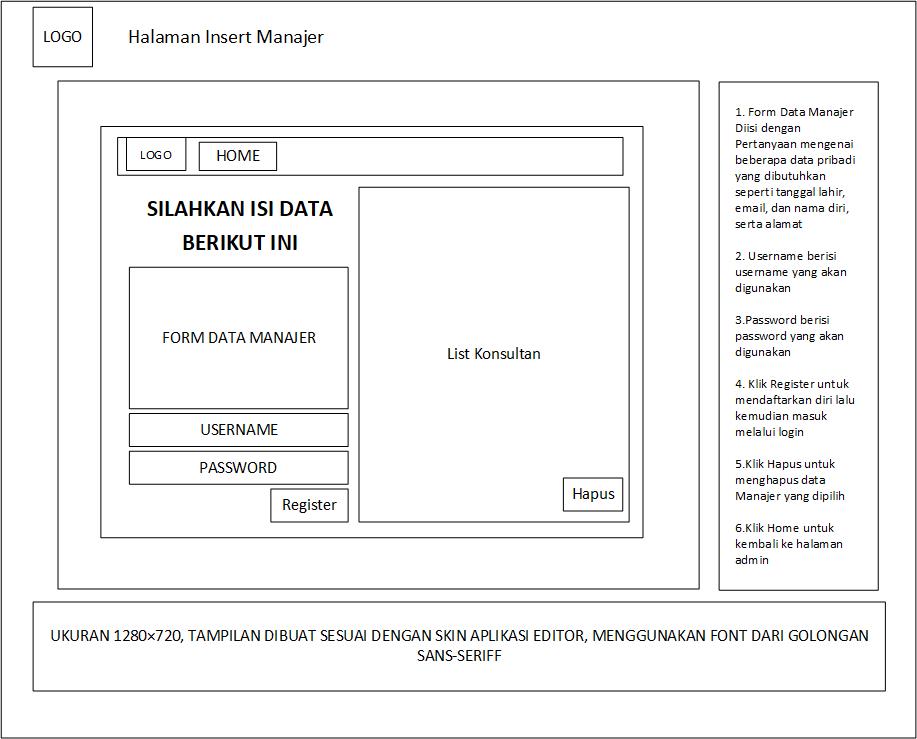
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| Card 7 | Card | Barang 1 | Card ini menampilkan salah satu barang yang di jual atau barang yang ada didatabase 1 baris dari card ini terdiri dari 3 card. |
| Card 8 | Card | Barang 2 | Card ini menampilkan 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 | Textfield | 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 | Jumbotron | Jumbotron | File berupa gambar yang bergerak menunjukkan pesan atau barang yang sedang memiliki penjualan yang baik. |

**

Perancangan Muka 12 Insert Konsultan

*Antarmuka : Perancangan Muka 12 Insert Konsultan*

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| 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 |

**

Perancangan Muka 13 Insert Manajer

*Antarmuka : Perancangan Muka 13 Insert Manajer*

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| 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 |

## Perancangan Representasi Persistensi Kelas

*A close up of a map

Description automatically generated*

Gambar ERD Rakit-Komputer

# 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 |

# 