**DPPL**-2019

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

Sistem Apotek

untuk:

Apotek

Dipersiapkan oleh:

Kelompok Apotek

IFK-41-02

Hatami Ra'is Bukhari 1301154303

Raynaldo Listianov 1301154359

William Adi A.K 1301150786

Dimas Maulana

Program Studi S1 Teknik Informatika – Fakultas Informatika

Universitas Telkom

Jalan Telekomunikasi Terusan Buah Batu, Bandung

Indonesia

|  |  |  |  |
| --- | --- | --- | --- |
| Program Studi S1 Teknik Informatika  -  Fakultas Informatika | Nomor Dokumen | | Halaman |
| *DPPL-2019* | | 35 |
| Revisi | *0* | *Tgl: 10-12-2019* |

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

[Daftar Isi 4](#_Toc27102232)

[1. Pendahuluan 5](#_Toc27102233)

[1.1 Tujuan Penulisan Dokumen 5](#_Toc27102234)

[1.2 Lingkup Masalah 5](#_Toc27102235)

[1.3 Definisi dan Istilah 5](#_Toc27102236)

[1.3.1 DPPL 5](#_Toc27102237)

[1.3.2 SKPL 5](#_Toc27102238)

[1.4 Aturan Penamaan dan Penomoran 5](#_Toc27102239)

[1.5 Referensi 6](#_Toc27102240)

[1.6 Ikhtisar Dokumen 6](#_Toc27102241)

[1.6.1 Pendahuluan 6](#_Toc27102242)

[1.6.2 Deskripsi Perancangan Global 6](#_Toc27102243)

[1.6.3 Perancangan Rinci 6](#_Toc27102244)

[1.6.4 Matriks Keterunutan 7](#_Toc27102245)

[1.7 Deskripsi Perancangan Global 8](#_Toc27102246)

[1.8 Rancangan Lingkungan Implementasi 8](#_Toc27102247)

[1.9 Deskripsi Arsitektural 8](#_Toc27102248)

[1.10 Deskripsi Komponen 8](#_Toc27102249)

[2 Perancangan Rinci 9](#_Toc27102250)

[2.1 Realisasi Use Case 9](#_Toc27102251)

[2.1.1 Nama Use Case: Login 9](#_Toc27102252)

[2.1.2 Nama Use Case: Create Akun 11](#_Toc27102253)

[2.1.3 Nama Use Case: Edit Akun 13](#_Toc27102254)

[2.1.4 Nama Use Case: View Akun 15](#_Toc27102255)

[2.1.5 Nama Use Case: Delete Akun 16](#_Toc27102256)

[2.1.6 Nama Use Case: Tambah Obat 18](#_Toc27102257)

[2.1.7 Nama Use Case: View Obat 19](#_Toc27102258)

[2.1.8 Nama Use Case: Pembuatan Obat 19](#_Toc27102259)

[2.1.9 Nama Use Case: Cek Jumlah Obat 22](#_Toc27102260)

[2.1.10 Nama Use Case: Update Obat 23](#_Toc27102261)

[2.1.11 Nama Use Case: Add Obat 25](#_Toc27102262)

[2.1.12 Nama Use Case: Delete Obat 27](#_Toc27102263)

[2.1.13 Nama Use Case: Penerimaan Obat 28](#_Toc27102264)

[2.1.14 Nama Use Case: Pemberian Obat Baru 29](#_Toc27102265)

[2.1.15 Nama Use Case: Transaksi Pembelian Obat 31](#_Toc27102266)

[2.1.16 Nama Use Case: Transaksi Pembelian Bahan 33](#_Toc27102267)

[2.2 Perancangan Detil Kelas 36](#_Toc27102268)

[2.2.1 Kelas <nama kelas> 36](#_Toc27102269)

[2.2.2 Kelas <nama kelas> 37](#_Toc27102270)

[2.3 Diagram Kelas Keseluruhan 37](#_Toc27102271)

[2.4 Algoritma/Query 37](#_Toc27102272)

[2.5 Diagram Statechart 40](#_Toc27102273)

[2.6 Perancangan Antarmuka 41](#_Toc27102274)

[2.7 Perancangan Representasi Persistensi Kelas 44](#_Toc27102275)

[3 Matriks Kerunutan 44](#_Toc27102276)

Setelah Daftar Isi Boleh ada Daftar Tabel dan Daftar Gambar

# 1. Pendahuluan

## Tujuan Penulisan Dokumen

Pada dokumen ini berisi penjelasan mengenai Deskripsi Perancangan Perangakat Lunak (DPPL) untuk Sistem Apotek yang dimana tujuan penulisan dokumen ini adalah untuk mendeskripsikan secara rinci mengenai perangkat lunak yang akan dibangun oleh penulis dan tim pengembangnya. Dokumen ini digunakan untuk jadi 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

Pengelolaan obat yang selama ini masih berlangsung secara manual sering kali menimbulkan kesulitan bagi admin dan apoteker untuk mengetahui berbagai data penting tentang apoteker seperti nama obat, jumlah stock obat dll.Hal tersebut akan membuat seorang pembeli menunggu lama di saat akan membeli suatu obat di apotik. Untuk itu diperlukan sebuah sistem apotek untuk dapat menyimpan data obat secara terurut dan terperinci agar mudah dicari saat dibutuhkan.

## Definisi dan Istilah

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

### DPPL

DPPL adalah singkatan dari Deskripsi Perancangan Perangkat Lunak yaitu dokumen yang mendeskripsikan dan menjabarkan secara tereperinci mengenai perancangan perangkat lunak yang akan dibangun.

### SKPL

SKPL adalah singkatan dari Spesifikasi Kebutuhan Perangkat Lunak atau sering juga disebut sebagai *Software Requirements Spesification* (SRS) merupakan spesifikasi dari perangkat lunak yang akan dikembangkan.

## Aturan Penamaan dan Penomoran

Penulisan dokumen DPPL ini menggunakan berbagai macam aturan penamaan dan penomoran yang berbeda-beda untuk beberapa bagian tertentu. Aturan penamaan dan penomoran yang digunakan berdasarkan hal/bagian tersebut adalah seperti yang tercantum pada Tabel 1 berikut ini.

Tabel 1 Aturan Penamaan dan Penomoran

| **Hal/Bagian** | **Aturan Penomoran/Penamaan** |
| --- | --- |
| Algoritma | Algo-xxx, dengan xxx sebagai digit/nomor algoritma |
| Query | Q-xxx, dengan xxx sebagai digit/nomor query |
| Perancangan Antarmuka | Axx, dengan xx sebagai nomor antarmuka  Mxx, dengan xx sebagai nomor pesan |

## Referensi

Referensi yang digunakan dalam pembuatan dokumen DPPL ini adalah sebagai berikut:

1. Dokumen Panduan Pengisian Deskripsi Perancangan Perangkat Lunak (DPPL) Berorientasi Objek
2. Dokumen SKPL Sistem Informasi Inventaris
3. Modul Implementasi dan Pengujian Perangkat Lunak Fakultas Informatika Telkom University.

## Ikhtisar Dokumen

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 teridiri dari empat bab dengan perincian sebagai berikut:

### 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.

### Deskripsi Perancangan Global

Deskripsi perancangan global berisi tentang rancangan dari perangkat lunak yang akan dibangun meliputi, rancangan lingkungan implementasi, deskripsi arsitektural, dan deskripsi komponen.

### Perancangan Rinci

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

### Matriks Keterunutan

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

## Deskripsi Perancangan Global

Deskripsi perancangan global meliputi penjelasan tentang racangan lingkungan implementasi, deskripsi arsitektural, dan deskripsi komponen yang membangun perangkat lunak.

## Rancangan Lingkungan Implementasi

Sistem ini diimplementasikan dalam lingkungan sebagai berikut :

1. Sistem informasi : Windows
2. Bahasa Pemrograman : PHP, HTML
3. DBMS : MySQL
4. Development Tools : PHP myAdmin(xampp), Visual studio code

## Deskripsi Arsitektural

Merupakan gambaran arsitektur atau komponen yang akan diterapkan pada perangkat lunak “Sistem Apotek” untuk mempermudah pengembang dalam mengembangkan atau mengimplementasikan perangkat lunak ini.

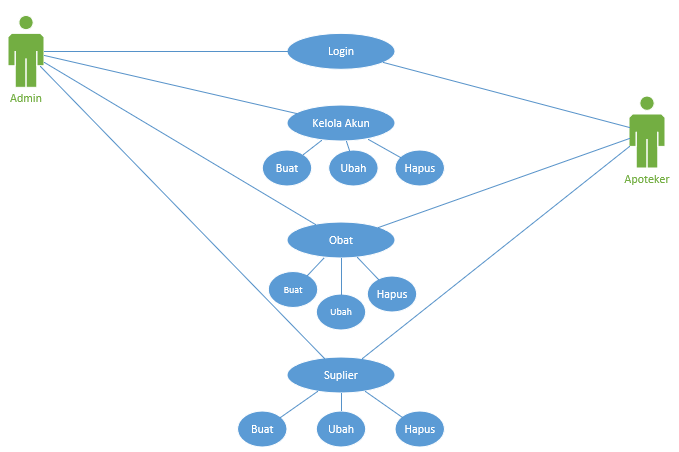
## Deskripsi Komponen

Diisi dengan daftar modul. Daftar modul bisa dalam bentuk tabel berikut:

| **No** | **Nama Komponen** | **Keterangan** |
| --- | --- | --- |
| 1 | Login\_user | Tampilan dari halaman login users |
| 2 | Login Admin | Tampilan dari halaman login Admin |
| 3 | Home/Dashboard | Tampilan dashboard menu utama |
| 4 | Kelola akun | Tampilan dari dashboard untuk kelola akun |
| 5 | View akun | Tampilan dari kelola akun menampilkan akun yang ada |
| 6 | Tambah Akun | Tampilan dari kelola akun menu untuk tambah akun |
| 7 | Delete Akun | Tampilan dari kelola akun menu untuk hapus akun |
| 8 | Kelola obat | Tampilan dari dashboard untuk kelola obat |
| 9 | Tambah Obat | Tampilan dari kelola obat menu untuk tambah obat |
| 10 | View Obat | Tampilan dari kelola obat menampilkan data obat yang tersimpan |
| 11 | Delete Obat | Tampilan dari kelola obat menu untuk hapus obat |
| 12 | Kelola Suplier | Tampilan dari dashboard untuk kelola suplier |
| 13 | Tambah Suplier | Tampilan dari kelola suplier menu untuk tambah suplier |
| 14 | Delete Suplier | Tampilan dari kelola suplier menu untuk hapus suplier |
| 15 | View Suplier | Tampilan dari kelola suplier menampilkan daftar suplier |
| 16 | Log Out | Log out dari aplikasi |

# Perancangan Rinci

## Realisasi Use Case



### Nama Use Case: Login

Use case : Login

Actor : admin, apoteker

Pre-condition : user ingin menggunakan program dan belum melakukan login

Post-condition : user telah melakukan login

Description : user melakukan login agar dapat menggunakan program

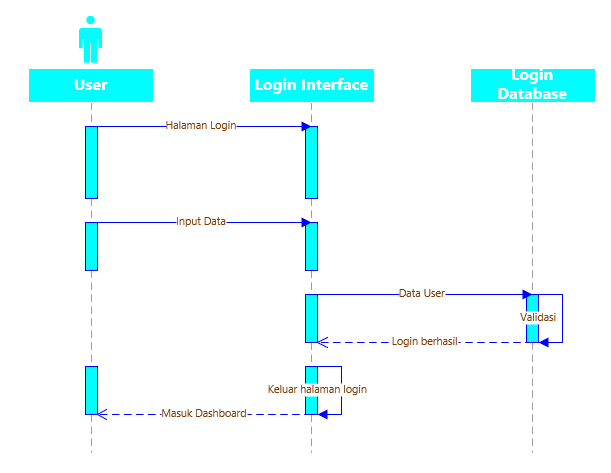
|  |  |
| --- | --- |
| Admin, Supplier, Pembeli,Apoteker | System |
| 1. User menginputkan username dan password |  |
|  | 1. System akan melakukan validasi username dan password |
|  | 1. Jika hasil didapat tidak valid, maka login gagal dan akan kembali ke langkah 1 |
|  | 1. Jika hasil valid didapat, maka login berhasil dan system akan menampilkan halaman utama |
| 1. User dapat menggunakan berbagai fitur dalam program |  |

#### Identifikasi Kelas

*Identifikasi kelas yang terkait dengan use case tersebut.Kelas di tahap perancangan dapat berbeda dengan dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

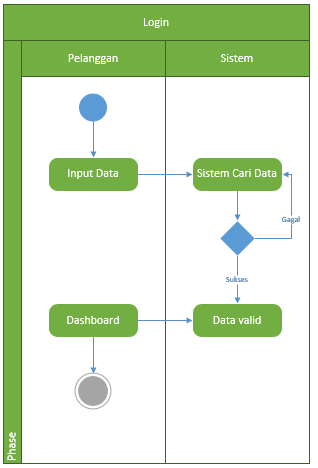
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| *1* |  |  |
| *2* |  |  |
| *3* |  |  |

#### Sequence Diagram



Gambar Sequence Diagram Use Case Login

#### Activity Diagram



Gambar Activity Diagram Use Case Login

### Nama Use Case: Create Akun

Use case : Create akun

Actor : admin

Pre-condition : admin ingin menambahkan akun baru

Post-condition : admin telah berhasil menambahkan akun baru

Description : admin melakukan create akun agar dapat menambahkan user baru pada program

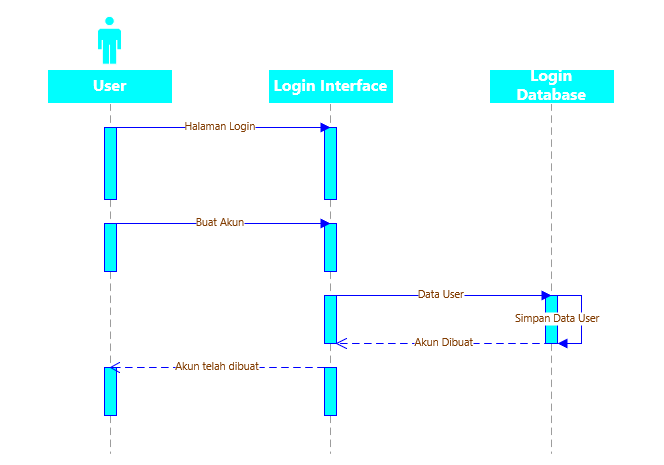
|  |  |
| --- | --- |
| Admin | System |
| 1. Admin memilih fitur kelola akun |  |
| 1. Admin memilih fitur create akun |  |
|  | 1. System akan menampilkan form create akun |
| 1. Admin akan menginputkan data yang dibutuhkan di dalam form dan melakukan submit data |  |
|  | 1. System akan menyimpan data akun baru ke dalam database |
|  | 1. Akun baru sudah berhasil dibuat |

#### Identifikasi Kelas

*Identifikasi kelas yang terkait dengan use case tersebut.Kelas di tahap perancangan dapat berbeda dengan dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

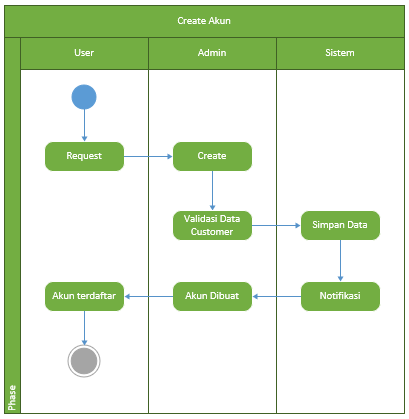
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



Gambar Sequence Diagram Use Case Create Akun

#### Activity Diagram



Gambar Activity Diagram Use Case Create Akun

### Nama Use Case: View Akun

Use case : view akun

Actor : admin

Pre-condition : admin ingin melihat data akun

Post-condition : admin telah melihat data akun

Description : admin melakukan view akun agar dapat melihat data akun

|  |  |
| --- | --- |
| Admin | System |
| 1. Admin memilih menu kelola akun |  |
|  | 1. Sistem cari data akun yang ada didatabase |
|  | 1. Sistem akan menampilkan data akun yang ada |
| 1. Admin dapat melihat data akun yang ada |  |

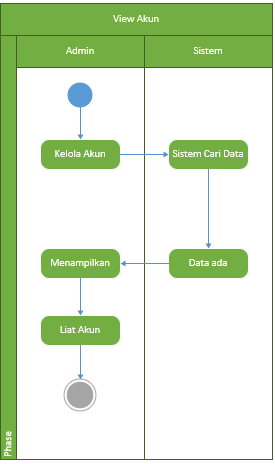
#### Identifikasi Kelas

*Identifikasi kelas yang terkait dengan use case tersebut.Kelas di tahap perancangan dapat berbeda dengan dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram

#### Activity Diagram



Gambar Activity Diagram Use Case View Akun

### Nama Use Case: Delete Akun

Use case : delete akun

Actor : admin

Pre-condition : admin ingin mengahapus akun

Post-condition : admin telah menghapus akun

Description : admin melakukan delete akun agar dapat menghapus akun

|  |  |
| --- | --- |
| Admin | System |
| 1. Admin memilih fitur kelola akun |  |
|  | 1. Sistem memeriksa data akun yang ada |
|  | 1. Sistem menampilkan data akun |
| 1. Admin melihat data akun |  |
| 1. Admin memilih hapus akun |  |
|  | 1. Data akun terhapus disimpan |

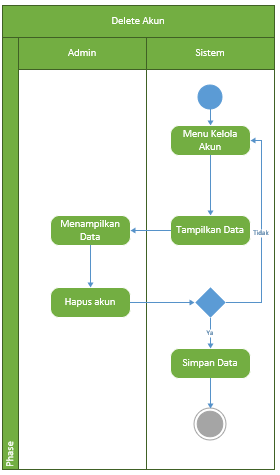
#### Identifikasi Kelas

*Identifikasi kelas yang terkait dengan use case tersebut.Kelas di tahap perancangan dapat berbeda dengan dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram

#### Activity Diagram



Gambar Activity Diagram Use Case Delete Akun

### Nama Use Case: Tambah Obat

Use case : Tambah Obat

Actor : apoteker

Pre-condition : apoteker ingin menambah obat

Post-condition : apoteker telah menambahkan obat

Description : apoteker melakukan penambahan obat

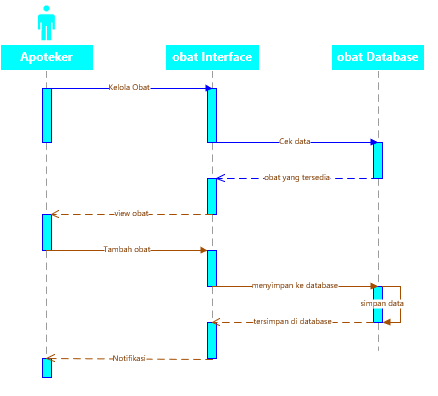
|  |  |
| --- | --- |
| Apoteker | System |
| 1. Apoteker ingin input obat baru |  |
|  | 1. Sistem mencari data obat |
| 1. Apoteker melihat data obat yang ada |  |
| 1. Apoteker Input Obat baru |  |
|  | 1. Sistem menyimpan data Obat Batu |
|  | 1. Data obat disimpan |
| 1. Apoteker menerima notifikasi data obat baru telah disimpan |  |

#### Identifikasi Kelas

*Identifikasi kelas yang terkait dengan use case tersebut.Kelas di tahap perancangan dapat berbeda dengan dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

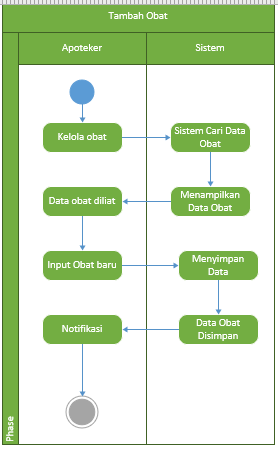
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



Gambar Sequence Diagram Use Case Tambah Obat

#### Activity Diagram



Gambar Activity Diagram Use Case Tambah Obat

### Nama Use Case: View Obat

Use case : View Obat

Actor : apoteker

Pre-condition : apoteker melihat data obat

Post-condition : apoteker telah melihat data obat

Description : apoteker mencoba melihat data obat yang tersimpan di database

|  |  |
| --- | --- |
| Apoteker | System |
| 1. Apoteker masuk ke menu kelola obat |  |
|  | 1. Sistem Mencoba mencari data obat yang tersimpan di database |
|  | 1. Sistem menampilkan data obat |
| 1. Apoteker melihat data obat yang ada |  |

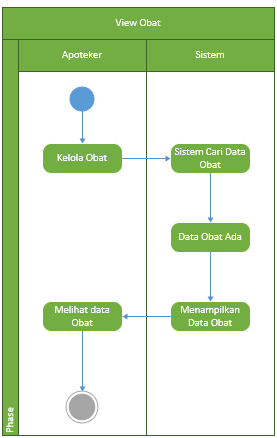
#### Identifikasi Kelas

*Identifikasi kelas yang terkait dengan use case tersebut.Kelas di tahap perancangan dapat berbeda dengan dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram

#### Activity Diagram



Gambar Activity Diagram Use Case View Obat

### Nama Use Case: Delete Obat

Use case : Delete obat

Actor : apoteker

Pre-condition : apoteker ingin menghapus obat

Post-condition : apoteker telah menghapus obat

Description : apoteker menghapus obat data obat yang ada

|  |  |
| --- | --- |
| Admin | System |
| 1. Apoteker memilih fitur kelola obat |  |
|  | 1. Sistem memeriksa data obat yang ada |
|  | 1. Sistem menampilkan data obat |
| 1. Apoteker melihat data obat |  |
| 1. Apoteker memilih hapus obat |  |
|  | 1. Data obat terhapus disimpan |

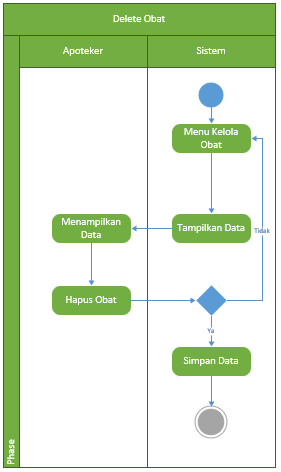
#### Identifikasi Kelas

*Identifikasi kelas yang terkait dengan use case tersebut.Kelas di tahap perancangan dapat berbeda dengan dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram

#### Activity Diagram



Gambar Activity Diagram Use Delete Obat

### Nama Use Case: Tambah Suplier

Use case : Tambah Suplier

Actor : admin, apoteker

Pre-condition : actor ingin menambah data suplier

Post-condition : actor telah menambah data suplier

Description : actor melakukan penambahan supplier ke database

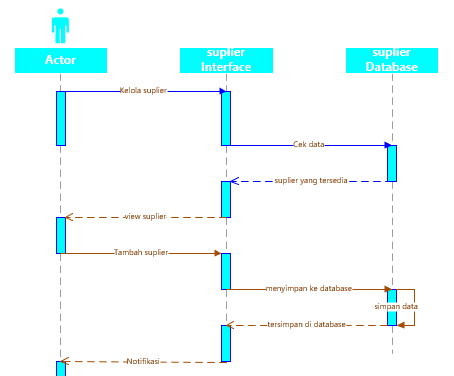
|  |  |
| --- | --- |
| Apoteker | System |
| 1. Actor ingin input suplier baru |  |
|  | 1. Sistem mencari data suplier |
| 1. Actor melihat data suplier yang ada |  |
| 1. Actor Input suplier baru |  |
|  | 1. Sistem menyimpan data suplier baru |
|  | 1. Data supplier disimpan |
| 1. Actor menerima notifikasi data suplier baru telah disimpan |  |

#### Identifikasi Kelas

*Identifikasi kelas yang terkait dengan use case tersebut.Kelas di tahap perancangan dapat berbeda dengan dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

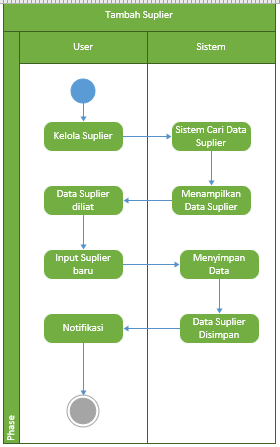
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



Gambar Sequence Diagram Use Case Tambah Suplier

#### Activity Diagram



Gambar Activity Diagram Use Case Tambah Suplier

### Nama Use Case: View Suplier

Use case : View Suplier

Actor : admin,apoteker

Pre-condition : actor ingin melihat data suplier

Post-condition : actor telah melihat data suplier

Description : actor mencoba melihat data supplier yang ada di database

|  |  |
| --- | --- |
| Apoteker | System |
| 1. Actor masuk ke menu kelola suplier |  |
|  | 1. Sistem mencoba mencari data suplier yang tersimpan di database |
|  | 1. Sistem menampilkan data supplier |
| 1. Actor melihat data suplier yang ada |  |

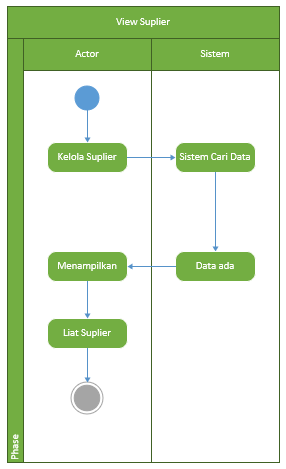
#### Identifikasi Kelas

*Identifikasi kelas yang terkait dengan use case tersebut.Kelas di tahap perancangan dapat berbeda dengan dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram

#### Activity Diagram



Gambar Activity Diagram Use Case View Suplier

### Nama Use Case: Delete Suplier

Use case : Delete Suplier

Actor : admin, apoteker

Pre-condition : actor menghapus suplier

Post-condition : actor telah berhasil menghapus suplier

Description : actor mencoba melakakukan hapus supplier dari database

|  |  |
| --- | --- |
| Admin | System |
| 1. Actor memilih fitur kelola suplier |  |
|  | 1. Sistem memeriksa data suplier yang ada |
|  | 1. Sistem menampilkan data suplier |
| 1. Actor melihat data suplier |  |
| 1. Actor memilih hapus suplier |  |
|  | 1. Data suplier terhapus disimpan |

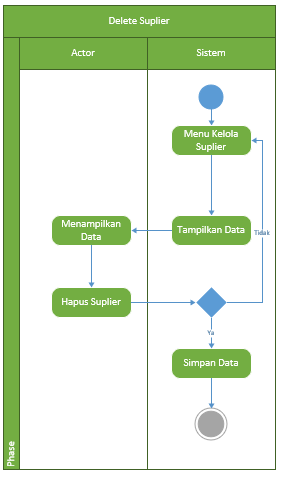
#### Identifikasi Kelas

*Identifikasi kelas yang terkait dengan use case tersebut.Kelas di tahap perancangan dapat berbeda dengan dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram

#### Activity Diagram



Gambar Activity Diagram Use Case Delete Suplier

### Nama Use Case: Ubah Akun

Use case : Ubah Akun

Actor : admin

Pre-condition : admin ingin mengumbah data akun

Post-condition : admin telah mengubah data akun

Description : admin mencoba mengubah data akun dalam database

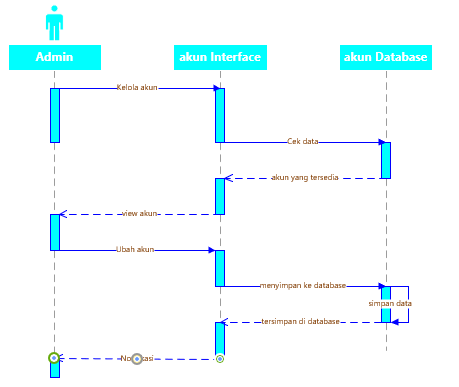
|  |  |
| --- | --- |
| Apoteker | System |
| 1. Admin ingin ubah data akun |  |
|  | 1. Sistem mencari data akun |
| 1. Admin melihat data akun yang ada |  |
| 1. Admin mengubah data akun |  |
|  | 1. Sistem menyimpan data akun yang baru |
|  | 1. Data akun disimpan |
| 1. Admin menerima notifikasi data baru telah disimpan |  |

#### Identifikasi Kelas

*Identifikasi kelas yang terkait dengan use case tersebut.Kelas di tahap perancangan dapat berbeda dengan dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

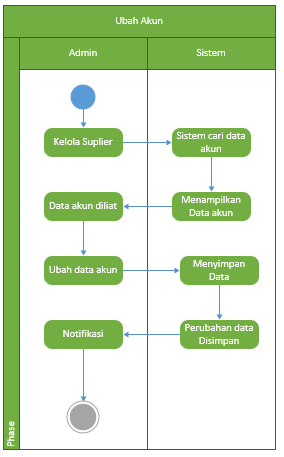
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



Gambar Sequence Diagram untuk Ubah Akun

#### Activity Diagram



Gambar Activity Diagram Use Case Ubah Akun

### Nama Use Case: Ubah Obat

Use case : Ubah obat

Actor : apoteker

Pre-condition : apoteker ingin mengumbah data obat

Post-condition : apoteker telah mengubah data obat

Description : apoteker mencoba mengubah data obat dalam database

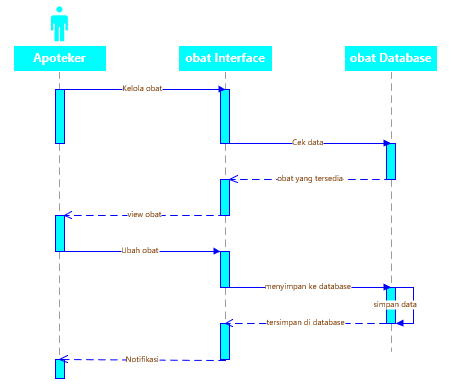
|  |  |
| --- | --- |
| Apoteker | System |
| 1. Apoteker ingin ubah data obat |  |
|  | 1. Sistem mencari data obat |
| 1. Apoteker melihat data obat yang ada |  |
| 1. Apoteker mengubah data obat |  |
|  | 1. Sistem menyimpan data obat yang baru |
|  | 1. Data obat disimpan |
| 1. Apoteker menerima notifikasi data baru telah disimpan |  |

#### Identifikasi Kelas

*Identifikasi kelas yang terkait dengan use case tersebut.Kelas di tahap perancangan dapat berbeda dengan dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

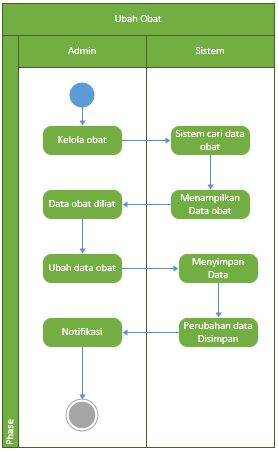
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



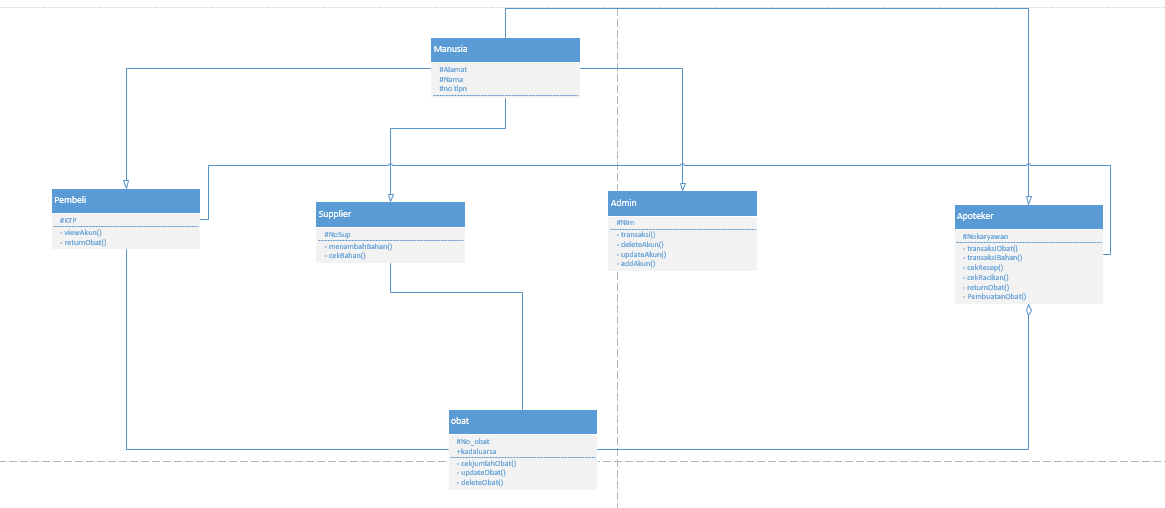
Gambar Sequence Diagram untuk Ubah Obat

#### Activity Diagram



Gambar Activity Diagram Use Case Ubah Obat

#### Diagram Kelas



Gambar Diagram Kelas

## Perancangan Detil Kelas

*Bagian ini diisi dengan daftar seluruh kelas dalam tabel berikut:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| *1.* |  |  |
|  |  |  |
|  |  |  |

*Untuk setiap kelas:*

* *identifikasi operasi (mengacu pada tanggung-jawab kelas), termasuk visibility-nya*
* *identifikasi atribut, termasuk visibility-nya*

### Kelas <nama kelas>

Bagian ini diisi dengan daftar operasi dan atribut untuk setiap kelas.

*Nama Kelas : ……..*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *Diisi dengan signature operasi* |  |  |
|  |  |  |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *Diisi dengan nama atribut* |  | *Tuliskan tipenya sesuai dengan yang dikenal pada bahasa pemrograman yang digunakan* |
|  |  |  |
|  |  |  |

### Kelas <nama kelas>

## Diagram Kelas Keseluruhan

*Bagian ini diisi dengan diagram kelas keseluruhan.*

## Algoritma/Query

*Bagian ini hanya diisi untuk kerangka algoritma untuk proses-proses yang dianggap cukup penting. Implementasi skeleton code juga sudah dapat dilakukan untuk kelas-kelas yang terdefinisi pada bahasa pemrograman tertentu. Boleh dibuat subbab per kelas.*

*Nama Kelas :Halaman Login*

*Nama Operasi : Login*

*Algoritma :Algo-001*

Nama Kelas :Halaman Login

Nama Operasi :Login

Algoritma :Algo-001

Username = input

Password = input

include function cek\_login

type username

type password

if (username && password != db)

alert'>Username dan Password tidak sesuai! Hubungi admin!

else

run cek\_login

end

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
|  |  |  |
|  |  |  |
|  |  |  |

*Nama Kelas :Periksa Tipe Login*

*Nama Operasi : Cek login*

*Algoritma :Algo-002*

Nama Sub Kelas :Periksa Tipe Login

Nama Operasi :cek login

Algoritma :Algo-002

select \* from user where username='$username' and password='$password'

if(level = admin)

begin

session = username as admin

show admin\_dashboard

end

else if (level = apoteker)

begin

session = username as apoteker

show apoteker\_dashboard

end

end

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-001* | select \* from user where username='$username' and password='$password' | *Mencari level username dan password yang sesuai dengan inputan* |

*Nama Kelas : Kelola Aku*

*Nama Operasi : Proses akun*

*Algoritma :Algo-003*

Nama Sub Kelas :Kelola akun

Nama Operasi :Proses Akun

Algoritma :Algo-003

-Tambah Akun

INSERT INTO user (nama, username, password, level) VALUES('".$nama."','".$username."','".$password."','".$level."')";

-Ubah Akun

UPDATE user SET ".$perubahan." WHERE id=$id";

-Delete

DELETE FROM user WHERE id=" . $id;

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-002* | INSERT INTO user (nama, username, password, level) VALUES('".$nama."','".$username."','".$password."','".$level."')"; | *Untuk Menambah akun dalam proses akun* |
| *Q-003* | UPDATE user SET ".$perubahan." WHERE id=$id"; | *Untuk Mengubah Akun dalam proses akun* |
| *Q-004* | DELETE FROM user WHERE id=" . $id; | *Untuk Menghapus Akun dalam proses akun* |

*Nama Kelas :Halaman Kelola obat*

*Nama Operasi : Proses Obat*

*Algoritma :Algo-004*

Nama Sub Kelas :Kelola Obat

Nama Operasi :Proses Obat

Algoritma :Algo-004

-Tambah Obat

INSERT INTO obat (nama, jenis, tanggal masuk, tanggal kadaluarsa, jumlah, harga) VALUES('".$nama."','".$jenis."','".$tgl\_masuk."','".$tgl\_kadaluarsa."','".$jumlah."')";

-Ubah Obat

UPDATE obat SET ".$perubahan." WHERE id=$id";

-Delete Obat

DELETE FROM obat WHERE id=" . $id;

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-005* | INSERT INTO obat (nama, jenis, tanggal masuk, tanggal kadaluarsa, jumlah, harga) VALUES('".$nama."','".$jenis."','".$tgl\_masuk."','".$tgl\_kadaluarsa."','".$jumlah."')"; | *Untuk menambah obat dalam database* |
| *Q-006* | UPDATE obat SET ".$perubahan." WHERE id=$id"; | *Untuk mengubah obat dalam database* |
| *Q-007* | *DELETE FROM obat WHERE id=" . $id;* | *Untuk hapus obat dalam data base* |

*Nama Kelas :Halaman Kelola Suplier*

*Nama Operasi : proses Suplier*

*Algoritma :Algo-005*

Nama Sub Kelas :Kelola Suplier

Nama Operasi :Proses suplier

Algoritma :Algo-005

-Tambah Suplier

INSERT INTO supplier (nama, kontak, alamat) VALUES('".$nama."','".$kontak."','".$alamat."')";

-Ubah Suplier

UPDATE user SET ".$perubahan." WHERE id=$id";

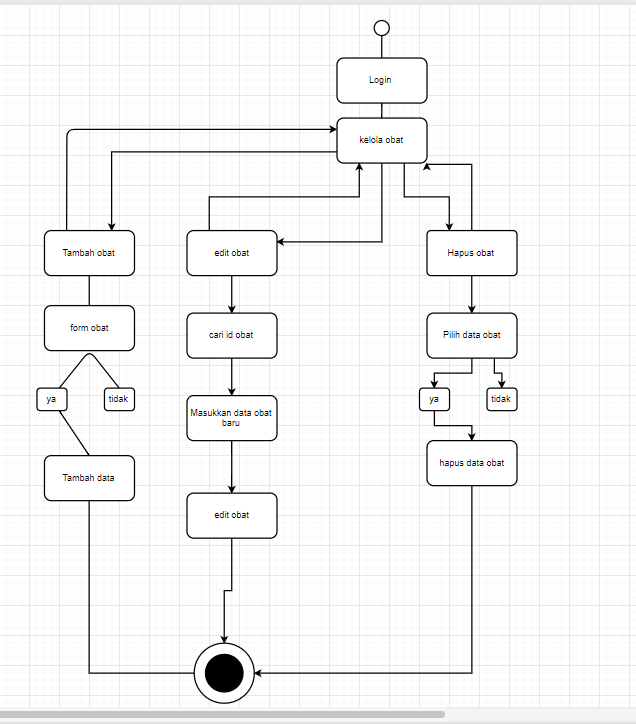
-Delete Suplier

DELETE FROM supplier WHERE id=" . $id;

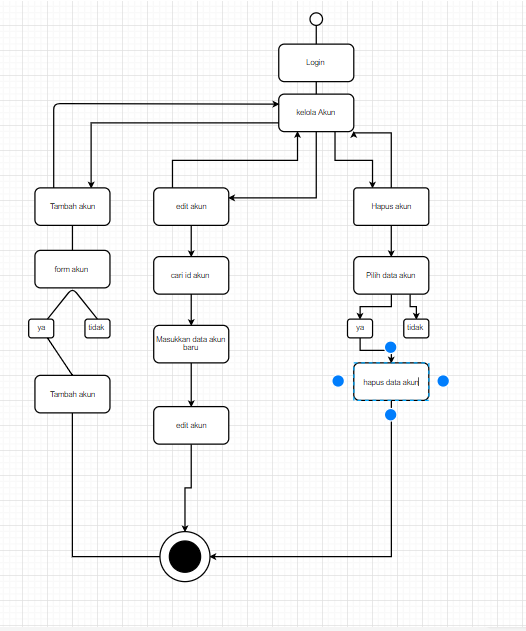
*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-008* | INSERT INTO supplier (nama, kontak, alamat) VALUES('".$nama."','".$kontak."','".$alamat."')"; | *Untuk menambah suplier dalam database* |
| *Q-009* | UPDATE user SET ".$perubahan." WHERE id=$id"; | *Untuk mengubah suplier dalam database* |
| *Q-010* | DELETE FROM supplier WHERE id=" . $id; | *Untuk hapus suplier dalam database* |

## Diagram Statechart



Gambar Statechart Diagram Kelola Obat

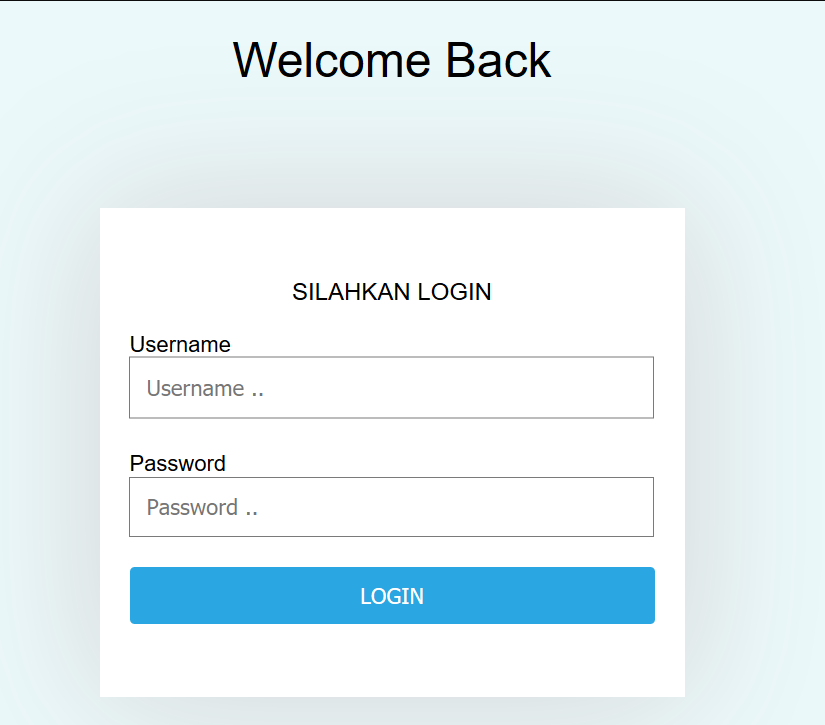


Gambar Statechart Diagram Kelola Akun

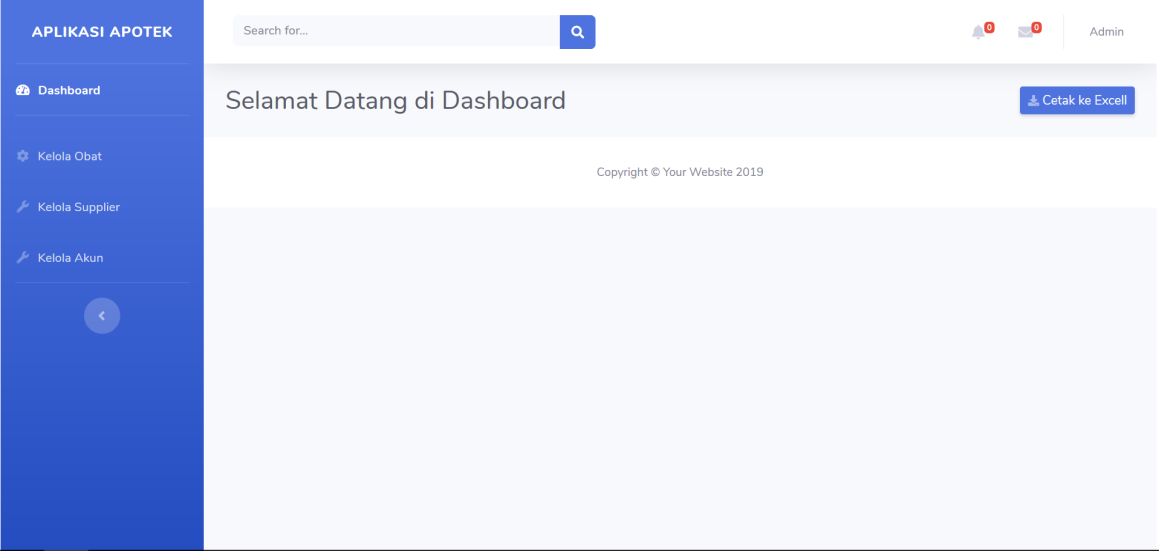
## Perancangan Antarmuka

*Bagian ini diisi dengan versi awal prototipe antarmuka.*

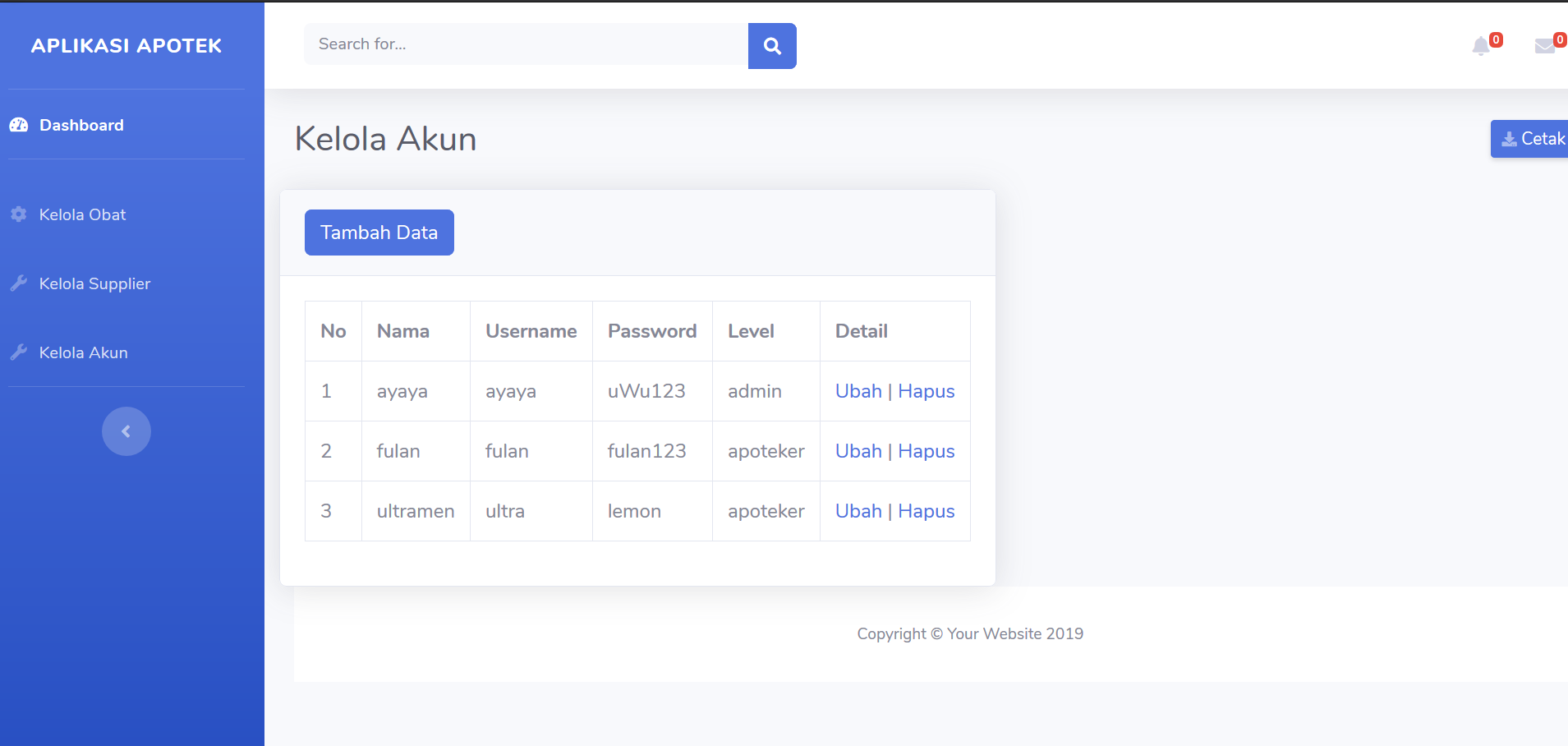
*Selanjutnya, untuk setiap antarmuka/layar, tuliskan spesifikasi detilnya, misalnya seperti di bawah ini:*



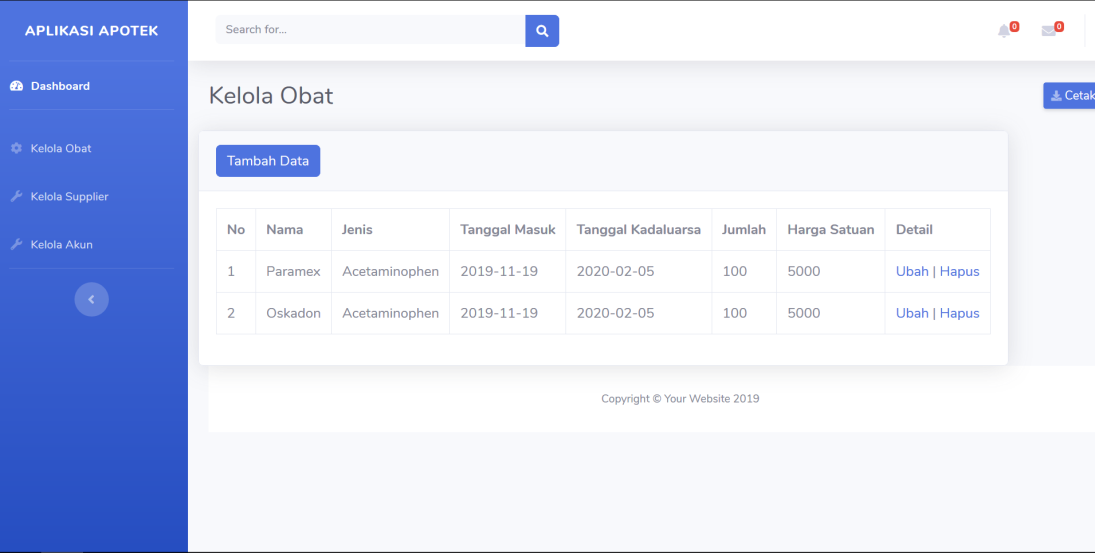
*Gambar ... Login*



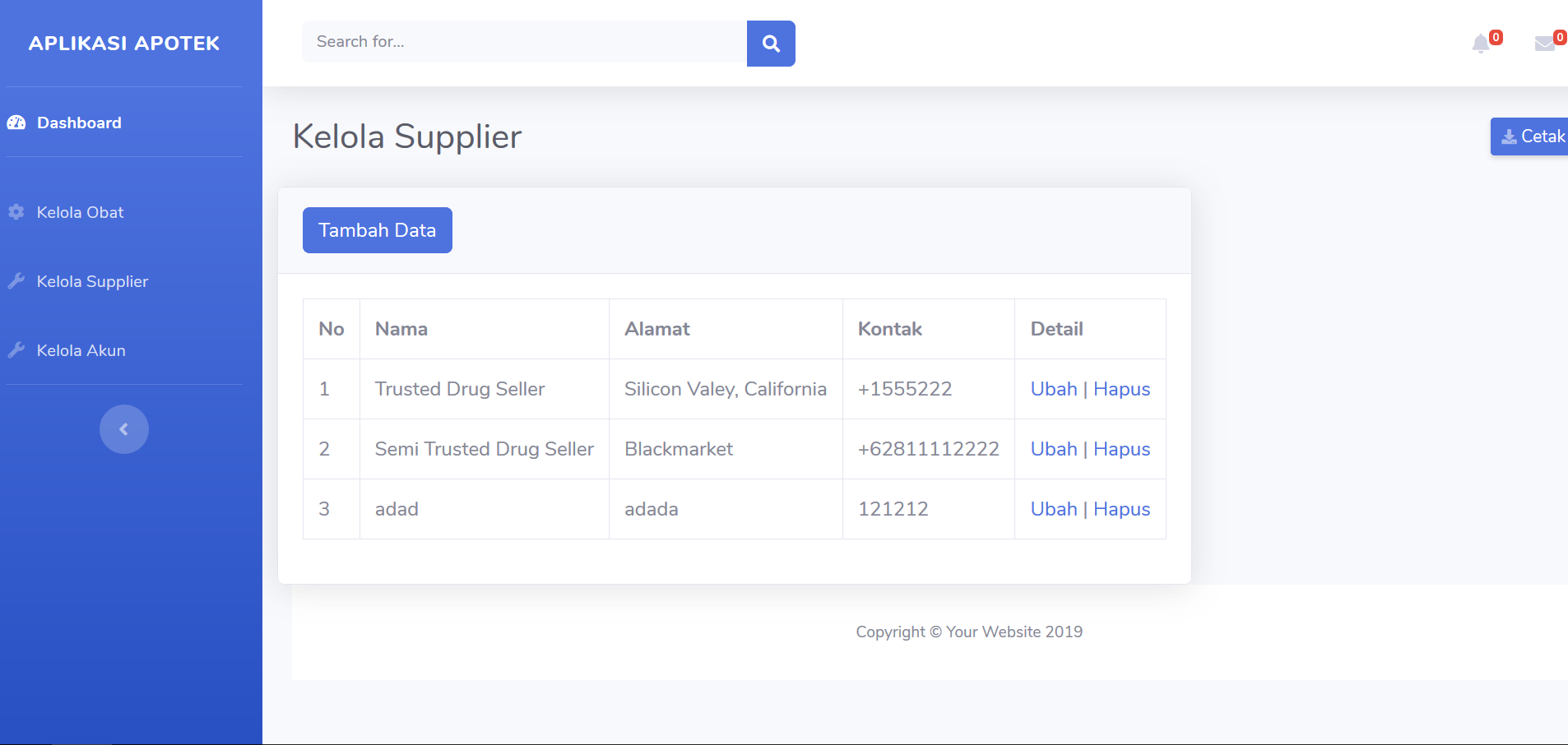
Gambar ... Dashboard



Gambar ... Kelola Akun



Gambar ... Kelola Obat



Gambar ... Kelola Suplier

Jika objek dikaitkan ke File lain (misalnya file gambar, file teks), berikan nama file terkait dan deskripsi ringkas dalam kolom keterangan

## Perancangan Representasi Persistensi Kelas

*Bagian ini diisi dengan rancangan skema basisdata dan traceability-nya terhadap kelas entity.*

# Matriks Kerunutan

Mapping use case dengan kelas-kelas terkait

|  |  |
| --- | --- |
| Kebutuhan Fungsionalitas | Keterangan |
| SKPL-Apotek-001 | Masuk Sistem |
| SKPL- Apotek -002 | Menambahkan data akun |
| SKPL- Apotek -003 | Mengubah data akun |
| SKPL- Apotek -004 | Menampilkan data akun |
| SKPL- Apotek -005 | Menghapus data akun |
| SKPL- Apotek -006 | Menambahkan data obat |
| SKPL- Apotek -007 | Mengubah data Obat |
| SKPL- Apotek -008 | Menampilkan data obat |
| SKPL- Apotek -009 | Menghapus data obat |
| SKPL- Apotek -010 | Menambahkan data supplier |
| SKPL- Apotek -011 | Menampilkan data supplier |
| SKPL- Apotek -012 | Menghapus data Suplier |
| SKPL- Apotek -013 | Logout aplikasi |