**DPPL**-1998

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

Sistem Apotek

untuk:

Apotek

Dipersiapkan oleh:

Kelompok 4

IF-40-04

Arbie Nabillasalsa Burkon/1301164082

Badrus Sholehk/1301164131

Hovely Wahyu Simatupang/1301164167

Septyan Indra Bayu/1301164264

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-1998* | | 16 |
| Revisi | *<nomor revisi>* | *Tgl: <isi tanggal>* |

DAFTAR PERUBAHAN

|  |  |
| --- | --- |
| Revisi | Deskripsi |
| A | Bab 1  Bab 2  Flowmap |
| B | Context Diagram + DFD (3.2.1)  ERD (3.3)  Perbaikan.....  ..... |
| C | Deskripsi proses (3.2.2)  Deskripsi Data (3.3)  Kebutuhan antarmuka eksternal (3.1)  Kebutuhan Non Fungsional (3.4)  Perbaikan........... |
| D | Atribut Kualitas Perangkat Lunak (3.5)  Batasan Perancangan (3.6)  Matriks keterunutan (4)  Melengkapi Lampiran ()  Perbaikan............. |
| 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](#_Toc473623651)

[1. Pendahuluan 5](#_Toc473623652)

[1.1 Tujuan Penulisan Dokumen 5](#_Toc473623653)

[1.2 Lingkup Masalah 5](#_Toc473623654)

[1.3 Definisi dan Istilah 5](#_Toc473623655)

[1.4 Aturan Penamaan dan Penomoran 5](#_Toc473623656)

[1.5 Referensi 5](#_Toc473623657)

[1.6 Ikhtisar Dokumen 5](#_Toc473623658)

[2 Deskripsi Perancangan Global 6](#_Toc473623659)

[2.1 Rancangan Lingkungan Implementasi 6](#_Toc473623660)

[2.2 Deskripsi Arsitektural 6](#_Toc473623661)

[2.3 Deskripsi Komponen 6](#_Toc473623662)

[3 Perancangan Rinci 7](#_Toc473623663)

[3.1 Realisasi Use Case 7](#_Toc473623664)

[3.1.1 Use Case <nama use case 1> 7](#_Toc473623665)

[3.2 Perancangan Detil Kelas 7](#_Toc473623666)

[3.2.1 Kelas <nama kelas> 7](#_Toc473623667)

[3.2.2 Kelas <nama kelas> 8](#_Toc473623668)

[3.3 Diagram Kelas Keseluruhan 8](#_Toc473623669)

[3.4 Algoritma/Query 8](#_Toc473623670)

[3.5 Diagram Statechart 8](#_Toc473623671)

[3.6 Perancangan Antarmuka 8](#_Toc473623672)

[3.7 Perancangan Representasi Persistensi Kelas 9](#_Toc473623673)

[4 Matriks Kerunutan 9](#_Toc473623674)

Setelah Daftar Isi Boleh ada Daftar Tabel dan Daftar Gambar

# 1. Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen ini berisi penjelasan mengenai Deskripsi Perancangan Perangakat Lunak (DPPL) untuk Sistem Apotek. 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

Pengelolaan obat yang selama ini masih berlangsung secara manual menimbulkan kesulitan bagi admin dan apoteker untuk mengetahui stock obat dalam waktu yang singkat.Hal tersebut akan membuat pembeli menunggu lama saat akan membeli obat. Untuk itu diperlukan 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), sublime

## 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 | Tampilan dashboard menu utama |
| 4 | Menu obat pil | Tampilan dashboard menu obat pil |
| 5 | Menu obat sirup | Tampilan dashboard menu obat sirup |
| 6 | Menu obat tablet | Tampilan dashboard menu obat tablet |
| 7 | Menu obat serbuk | Tampilan dashboard menu obat serbuk |
| 8 | Menu obat suspensi | Tampilan dashboard menu obat suspensi |
| 9 | Setting | Tampilan menu setting |
| 10 | Return obat | Tampilan menu mereturn obat |
| 11 | Menu transaksi | Tampilan tentang transaksi apotek |
| 12 | Menu kelola akun | Tampilan untuk admin kelola akun |
| 13 | Menu kelola obat | Tampilan untuk admin kelola obat |

# Perancangan Rinci

## Realisasi Use Case

### Nama Use Case: Login

Use case : Login

Actor : admin, supplier, pembeli, apoteker

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

Post-condition : actor telah melakukan login

Description : actor melakukan login agar dapat menggunakan program

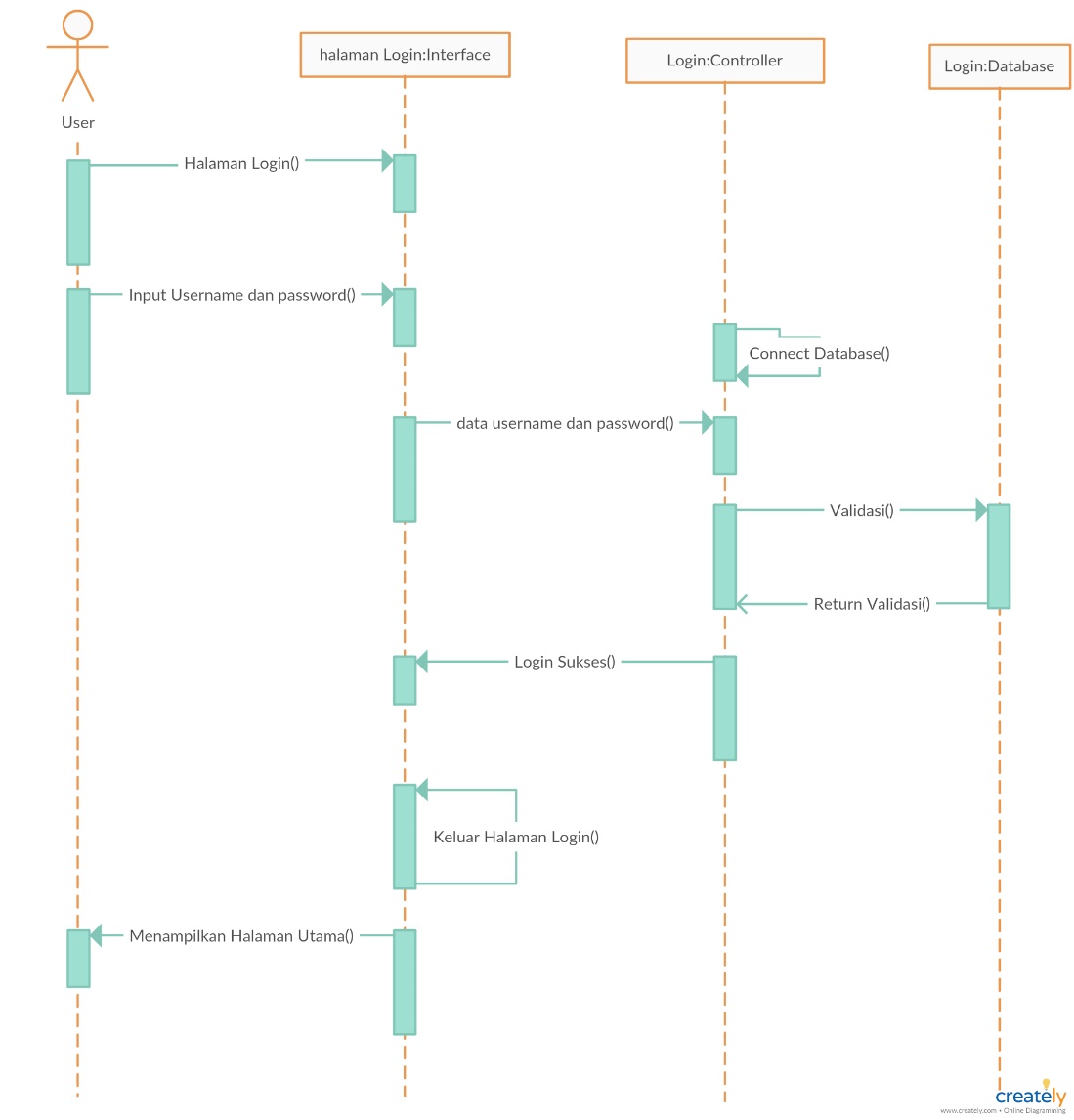
|  |  |
| --- | --- |
| Admin, Supplier, Pembeli,Apoteker | System |
| 1. Actor menginputkan username dan password |  |
|  | 1. System akan melakukan validasi username dan password |
|  | 1. Jika tidak valid, maka login gagal dan akan kembali ke langkah 1 |
|  | 1. Jika valid, maka login berhasil dan system akan menampilkan halaman utama |
| 1. Actor 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:*

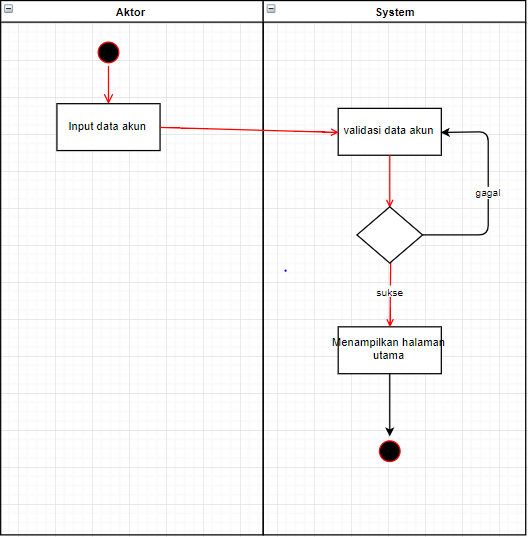
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| *1* | *Obat\_doc* | *Obat\_form* |
| *2* | *Obat\_form* | *Obat\_form* |
| *3* | *Obat\_list* | *Obat\_list* |

#### 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 |

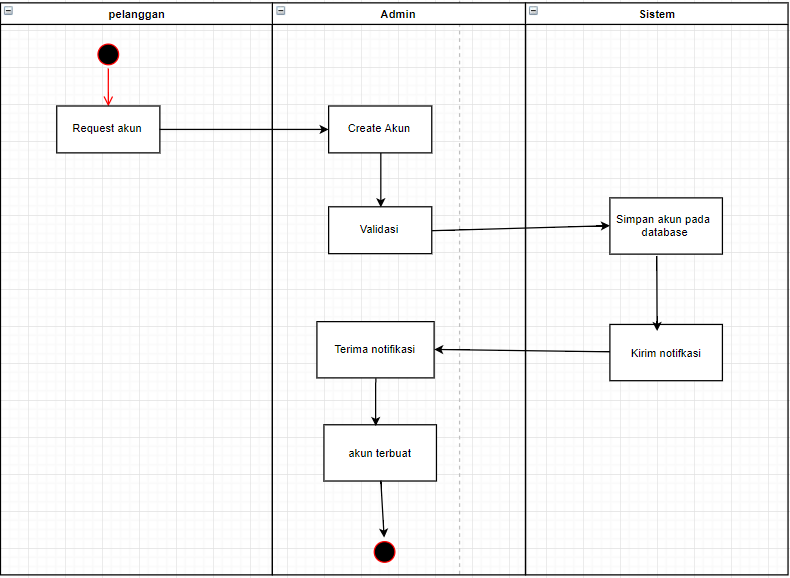
#### Identifikasi Kelas

*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

#### Activity Diagram



Gambar Activity Diagram Use Case Create Akun

### Nama Use Case: Update Akun

Use case : update akun

Actor : admin

Pre-condition : admin ingin melakukan edit pada data akun

Post-condition : admin telah berhasil melakukan edit pada data akun

Description : admin melakukan update akun agar dapat melakukan edit pada data akun

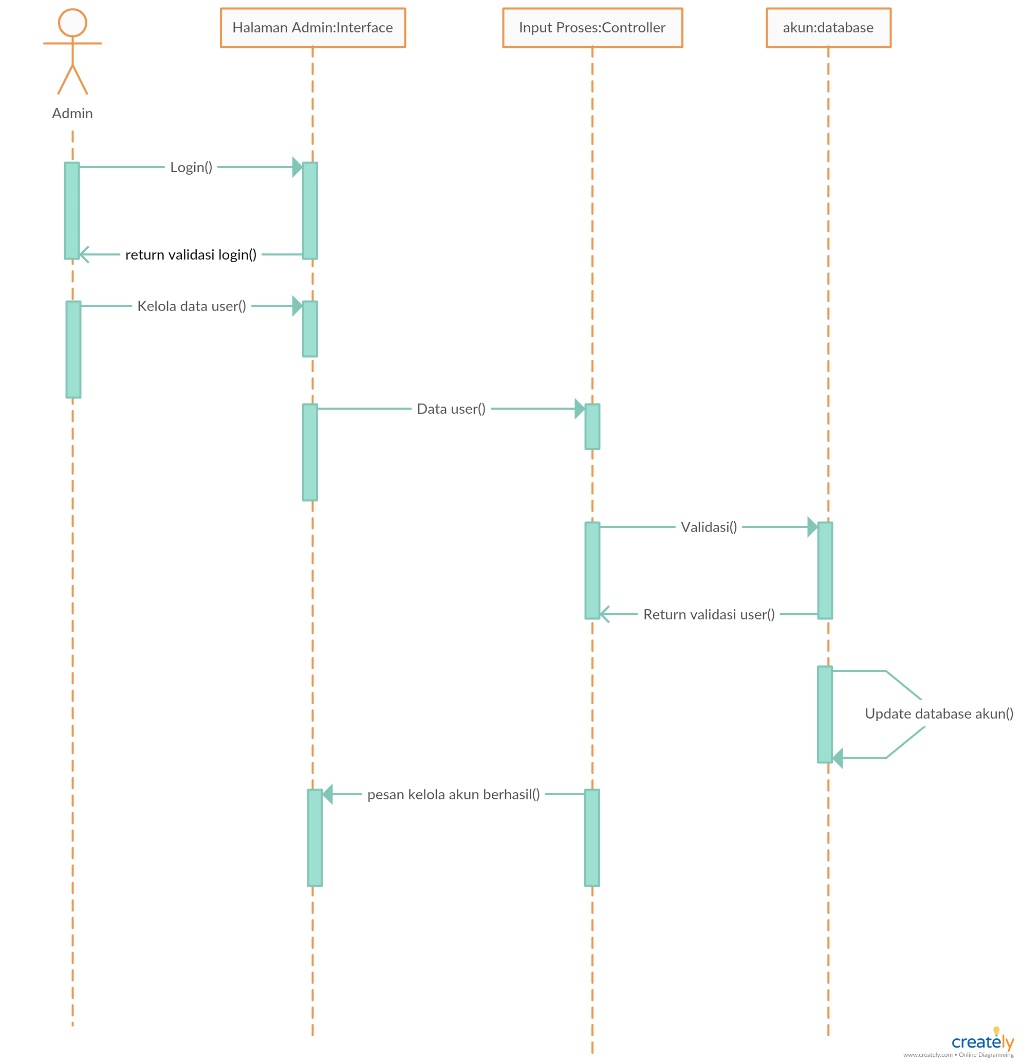
|  |  |
| --- | --- |
| Admin | System |
| 1. Admin memilih fitur kelola akun |  |
| 1. Admin memilih fitur update akun |  |
| 1. Admin akan memasukkan username akun yang akan di edit |  |
|  | 1. System akan menampilkan data akun username yang diinginkan oleh admin |
| 1. Admin akan menginputkan data yang akan diubah pada data acoount |  |
|  | 1. System akan menyimpan perubahan data yang dilakukan oleh admin ke dalam database |

#### 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:*

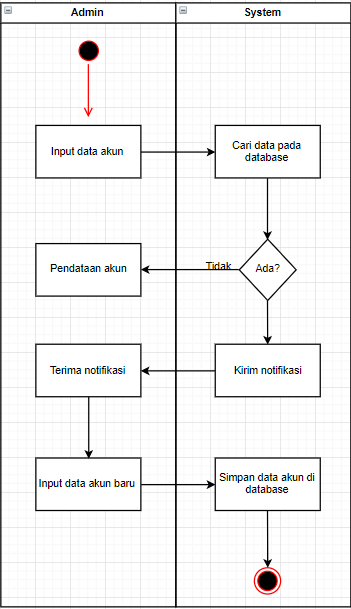
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



Gambar Sequence Diagram Use Case Update Akun

#### Activity Diagram



Gambar Activity Diagram Use Case Update 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 fitur kelola akun |  |
| 1. Admin memilih fitur view akun |  |
| 1. Admin akan memasukkan username akun yang akan dilihat |  |
|  | 1. System akan menampilkan data akun username yang diinginkan oleh admin |
| 1. Admin dapat melihat data akun yang diinginkan |  |

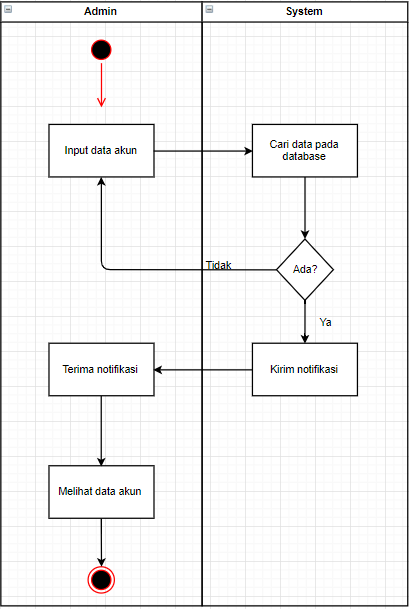
#### Identifikasi Kelas

*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. Admin memilih fitur delete akun |  |
|  | 1. System akan menampilkan halaman delete akun |
| 1. Admin akan memasukkan username akun yang akan dihapus |  |
|  | 1. System akan menampilkan data akun username yang diinginkan oleh admin |
| 1. Admin akan menghapus akun |  |
|  | 1. System akan melakukan validasi |
|  | 1. Jika admin memilih cancel, maka kembali ke langkah 3 |
|  | 1. Jika admin memilih delete, maka system akan menghapus data akun dari dalam database |

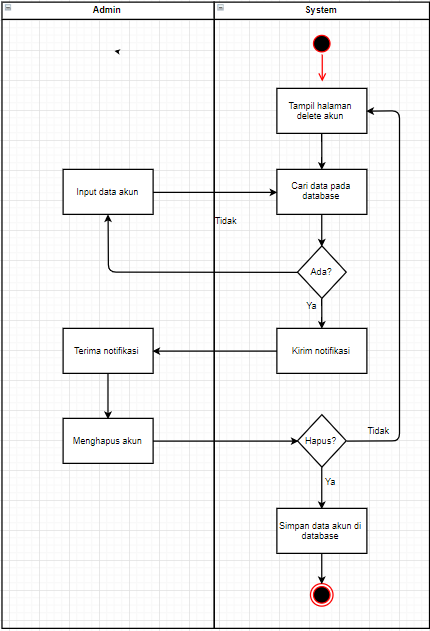
#### Identifikasi Kelas

*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: Pendataan Resep

Use case : pendataan resep

Actor : pembeli, apoteker

Pre-condition : actor ingin melakukan pendataan resep

Post-condition : actor telah melakukan pendataan resep

Description : actor melakukan pendataan resep agar dapat menyimpan data resep

|  |  |  |
| --- | --- | --- |
| Pembeli | Apoteker | System |
| 1. Pembeli akan memberikan resep pada apoteker |  |  |
|  | 1. Apoteker akan menginputkan data resep |  |
|  |  | 1. System akan menyimpan data resep pada database |

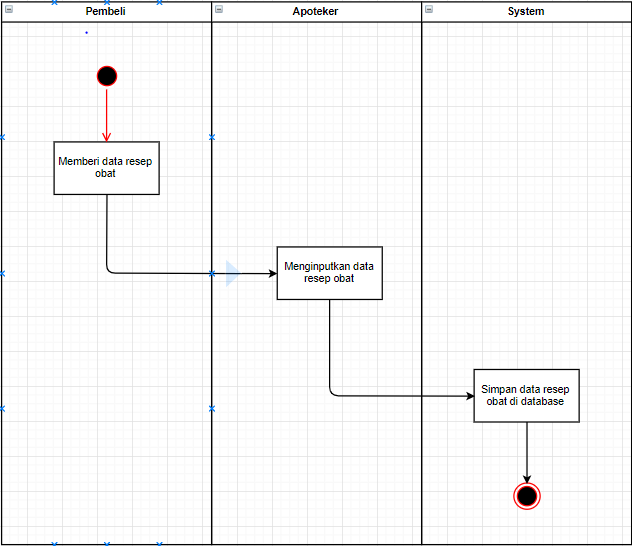
#### Identifikasi Kelas

*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 Pendataan Resep

### Nama Use Case: Pendataan Racikan

Use case : pendataan racikan

Actor : pembeli, apoteker

Pre-condition : actor ingin melakukan pendataan racikan

Post-condition : actor telah melakukan pendataan racikan

Description : actor melakukan pendataan racikan agar dapat menyimpan data obat racikan

|  |  |  |
| --- | --- | --- |
| Pembeli | Apoteker | System |
| 1. Pembeli akan memberikan resep obat racikan pada apoteker |  |  |
|  | 1. Apoteker akan menginputkan data obat racikan |  |
|  |  | 1. System akan menyimpan data obat racikan pada database |

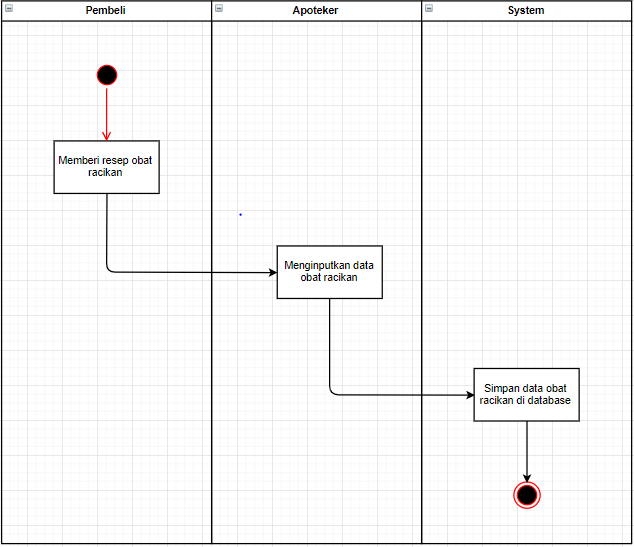
#### Identifikasi Kelas

*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 Pendataan Racikan

### Nama Use Case: Pembuatan Obat

Use case : pembuatan obat

Actor : apoteker

Pre-condition : apoteker ingin melakukan pembuatan obat

Post-condition : apoteker telah melakukan pembuatan obat

Description : apoteker melakukan pembuatan obat agar dapat membuat obat yang dibutuhkan oleh pembeli

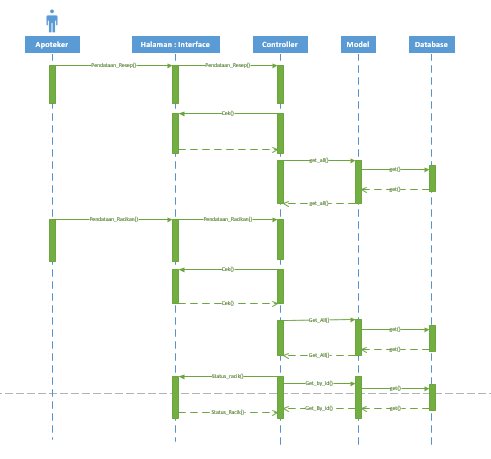
|  |  |
| --- | --- |
| Apoteker | System |
| 1. Apoteker telah melakukan pendataan resep |  |
| 1. Apoteker akan melakukan pencarian data resep |  |
|  | 1. System akan menampilkan data obat yang diinginkan oleh apoteker |
| 1. Apoteker akan membuat obat sesuai dengan resep yang ditampilkan oleh system |  |

#### 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:*

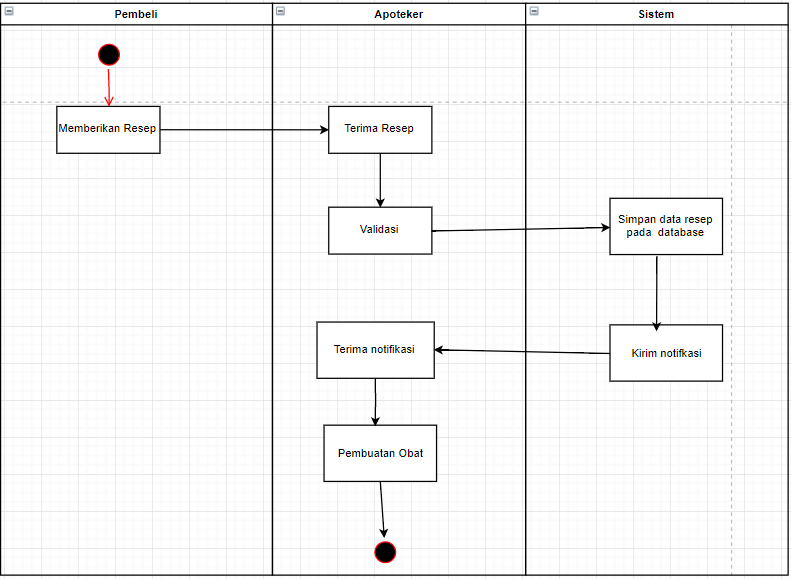
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



Gambar Sequence Diagram untuk Use Case Pembuatan Obat

#### Activity Diagram



Gambar Activity Diagram Use Case Pembuatan Obat

### Nama Use Case: Cek Jumlah Obat

Use case : cek jumlah obat

Actor : admin, apoteker

Pre-condition : actor ingin melakukan cek jumlah obat

Post-condition : actor telah melakukan cek jumlah obat

Description : actor melakukan cek jumlah obat agar dapat mengetahui stock obat yang tersedia

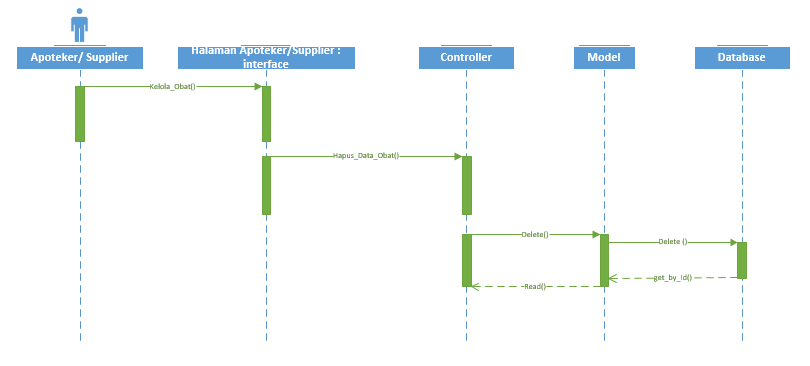
|  |  |
| --- | --- |
| Admin, Apoteker | System |
| 1. Actor memilih fitur pengadaan obat |  |
| 1. Actor memilih fitur cek jumlah obat |  |
|  | 1. System akan menampilkan data obat yang terdapat pada database |
| 1. Actor dapat melihat data obat dan dapat melakukan cek jumlah obat |  |

#### 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:*

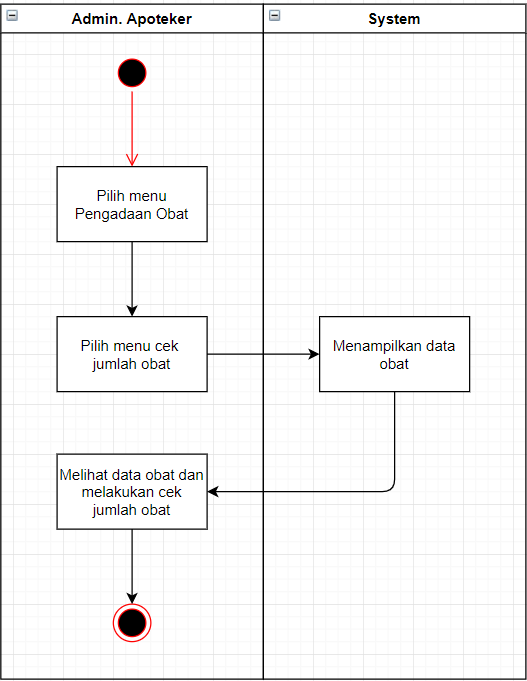
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



Gambar Sequence Diagram untuk Pengadaan Obat

#### Activity Diagram



Gambar Activity Diagram Use Case Cek Jumlah Obat

### Nama Use Case: Update Obat

Use case : update obat

Actor : admin

Pre-condition : admin ingin melakukan update obat

Post-condition : admin telah melakukan update obat

Description : admin melakukan update obat agar dapat mengubah data obat

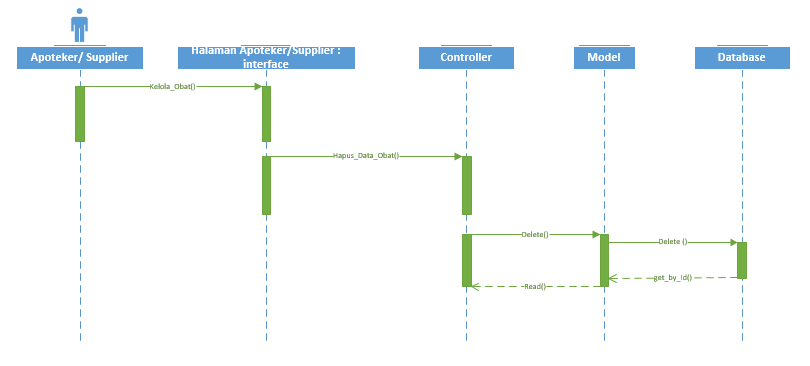
|  |  |
| --- | --- |
| Admin | System |
| 1. Admin memilih fitur pengadaan obat |  |
| 1. Admin memilih fitur update obat |  |
|  | 1. System akan menampilkan data obat |
| 1. Admin akan menginputkan data yang akan diubah pada data obat |  |
|  | 1. System akan menyimpan perubahan data yang dilakukan oleh admin ke dalam database |

#### 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:*

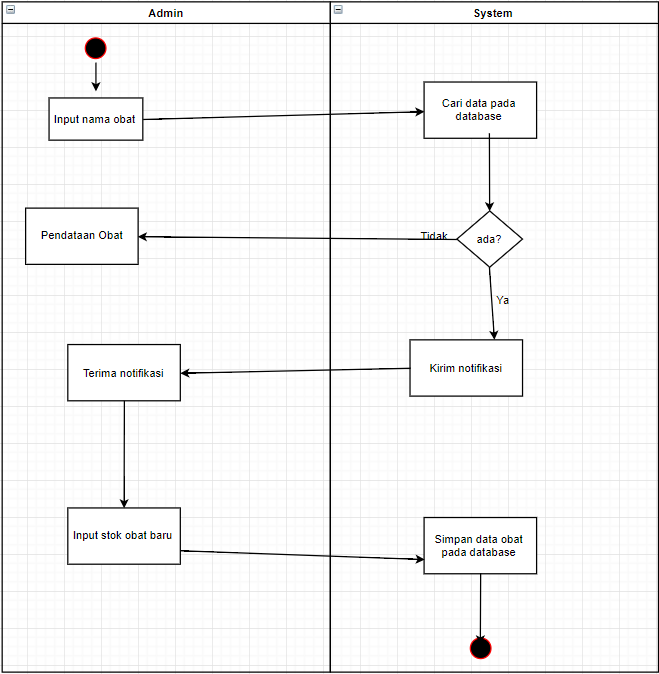
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



Gambar Sequence Diagram untuk Pengadaan Obat

#### Activity Diagram



Gambar Activity Diagram Use Case Update Obat

### Nama Use Case: Add Obat

Use case : add obat

Actor : admin, supplier

Pre-condition : actor ingin menambahkan data obat baru

Post-condition : actor telah berhasil menambahkan data obat baru

Description : actor melakukan add obat agar dapat menambahkan data obat baru pada database

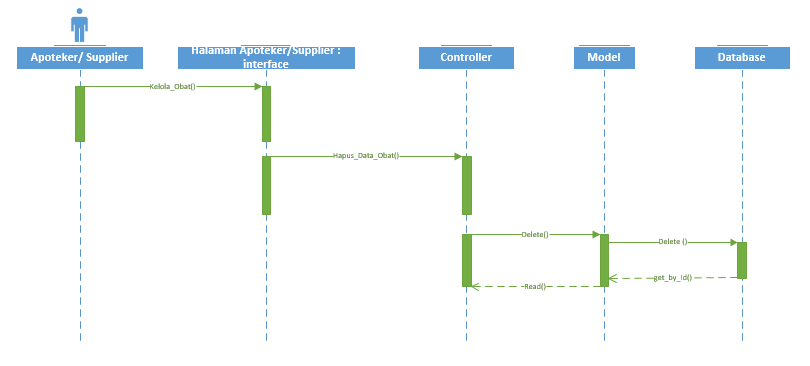
|  |  |
| --- | --- |
| Admin, Supplier | System |
| 1. Actor memilih fitur pengadaan obat |  |
| 1. Actor memilih fitur add obat |  |
|  | 1. System akan menampilkan form add obat |
| 1. Actor akan menginputkan data yang dibutuhkan di dalam form dan melakukan submit data |  |
|  | 1. System akan menyimpan data obat baru ke dalam database |

#### 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:*

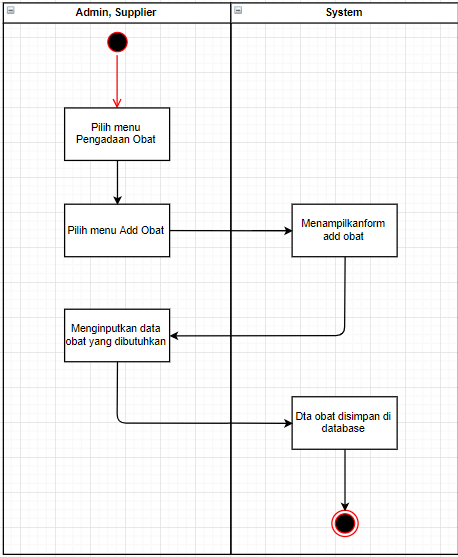
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



Gambar Sequence Diagram untuk Pengadaan Obat

#### Activity Diagram



Gambar Activity Diagram Use Case Add Obat

### Nama Use Case: Delete Obat

Use case : delete obat

Actor : admin

Pre-condition : admin ingin mengahapus data obat

Post-condition : admin telah menghapus data obat

Description : admin melakukan delete obat agar dapat menghapus data obat dari dalam database

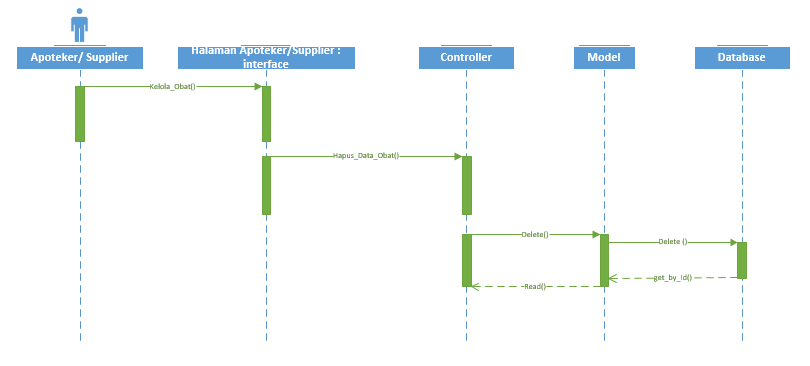
|  |  |
| --- | --- |
| Admin | System |
| 1. Admin memilih fitur pengadaan obat |  |
| 1. dmin memilih fitur delete obat |  |
|  | 1. System akan menampilkan halaman delete obat |
| 1. Admin akan memasukkan nama obat yang akan dihapus |  |
|  | 1. System akan menampilkan data obat yang diinginkan oleh admin |
| 1. Admin akan menghapus data obat |  |
|  | 1. System akan melakukan validasi |
|  | 1. Jika admin memilih cancel, maka kembali ke langkah 3 |
|  | 1. Jika admin memilih delete, maka system akan menghapus data obat dari dalam database |

#### 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:*

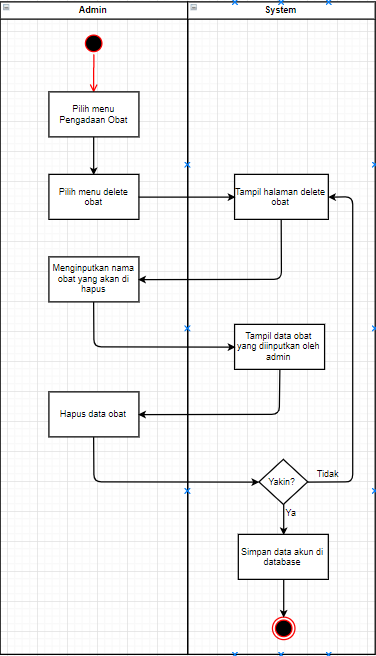
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



Gambar Sequence Diagram untuk Pengadaan Obat

#### Activity Diagram



Gambar Activity Diagram Use Case Delete Obat

### Nama Use Case: Penerimaan Obat

Use case : penerimaan obat

Actor : pembeli, apoteker

Pre-condition : actor ingin melakukan penerimaan obat

Post-condition : actor telah melakukan penerimaan obat

Description : actor melakukan pemerimaan obat agar dapat menukar obat

|  |  |  |
| --- | --- | --- |
| Pembeli | Apoteker | System |
| 1. Pembeli telah melakukan pembelian obat |  |  |
| 1. Pembeli akan menyerahkan obat kepada apoteker |  |  |
|  | 1. Apoteker akan mengecek data pembelian obat |  |
|  |  | 1. System akan menampilkan data pembelian obat |
|  | 1. Jika data sesuai dengan obat yang diberikan oleh pembeli, maka apoteker akan melakukan penerimaan obat |  |

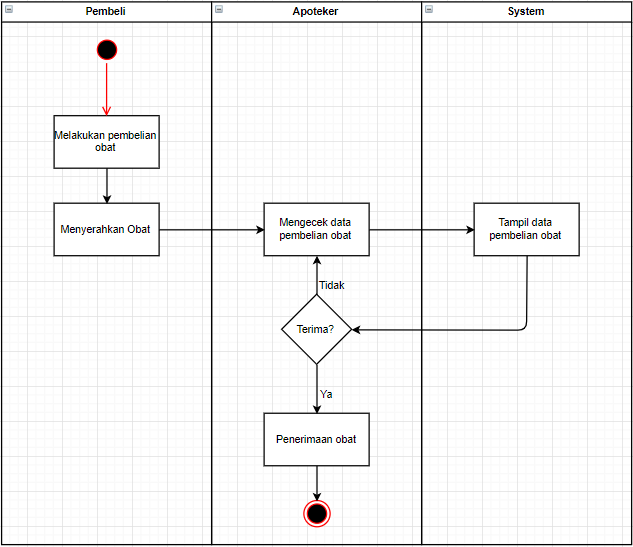
#### Identifikasi Kelas

*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 Penerimaan Obat

### Nama Use Case: Pemberian Obat Baru

Use case : pemberian obat baru

Actor : pembeli, apoteker

Pre-condition : actor ingin melakukan pemberian obat baru

Post-condition : actor telah melakukan pemberian obat baru

Description : actor melakukan pemberian obat baru agar dapat menukar obat

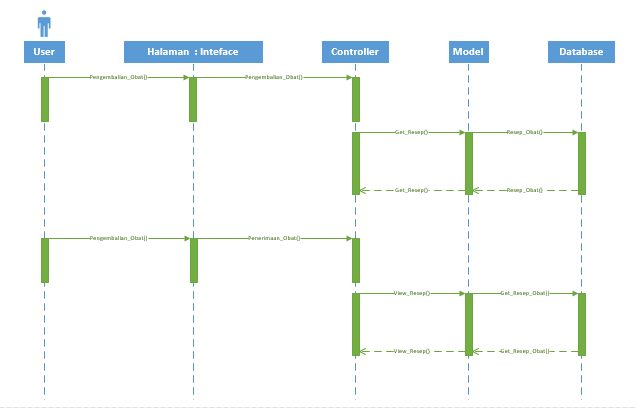
|  |  |  |
| --- | --- | --- |
| Pembeli | Apoteker | System |
| 1. Pembeli telah melakukan pembelian obat |  |  |
|  | 1. Apoteker telah melakukan penerimaan obat |  |
|  | 1. Apoteker akan mengecek data pembelian obat |  |
|  |  | 1. System akan menampilkan data pembelian obat |
|  | 1. Apoteker akan memberikan obat yang baru kepada pembeli sesuai dengan data obat |  |

#### 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:*

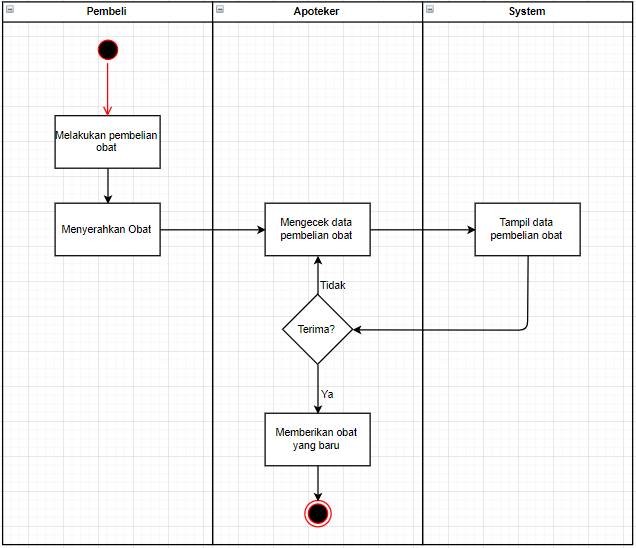
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



Gambar Sequence Diagram Pengembalian Obat

#### Activity Diagram



Gambar Activity Diagram Use Case Pemberian Obat Baru

### Nama Use Case: Transaksi Pembelian Obat

Use case : transaksi pembelian obat

Actor : apoteker

Pre-condition : apoteker ingin melakukan transaksi pembelian obat

Post-condition : apoteker telah melakukan transaksi pembelian obat

Description : apoteker melakukan transaksi pembelian obat agar dapat menyimpan data transaksi pembelian obat

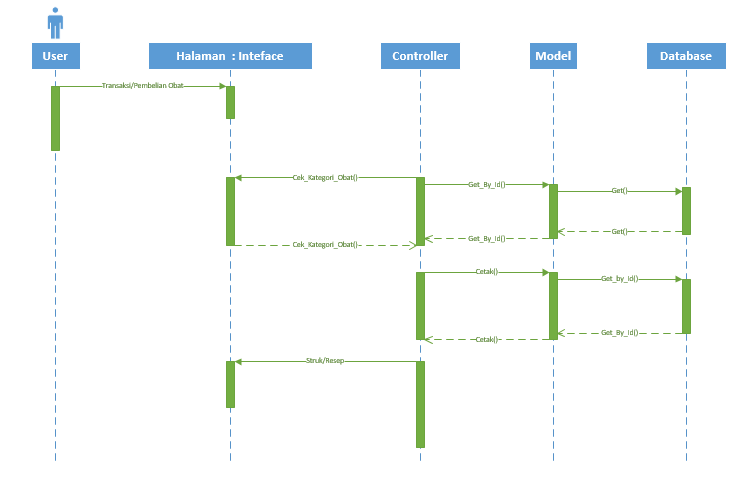
|  |  |
| --- | --- |
| Apoteker | System |
| 1. Apoteker memilih fitur transaksi obat |  |
| 1. Apoteker memilih fitur transaksi pembelian obat |  |
|  | 1. System akan menampilkan form transaksi pembelian obat |
| 1. Apoteker akan menginputkan data yang diminta oleh form sesuai dengan data transaksi pembelian obat pembeli |  |
|  | 1. System akan menyimpan data transaksi pembelian obat ke dalam database |

#### 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:*

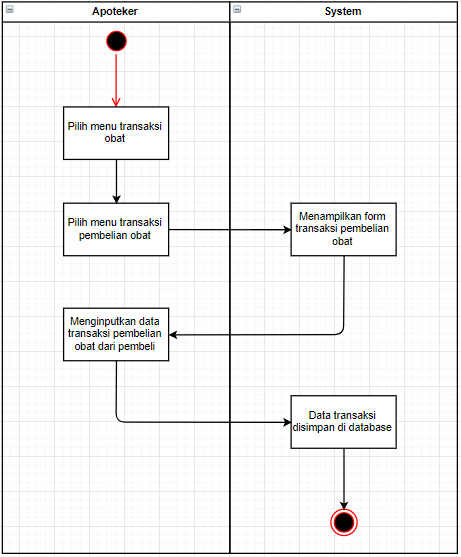
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



Gambar Sequence Diagram Use Case Transaksi Pembelian Obat

#### Activity Diagram



Gambar Activity Diagram Use Case Transaksi Pembelian Obat

### Nama Use Case: Transaksi Pembelian Bahan

Use case : transaksi pembelian bahan

Actor : apoteker

Pre-condition : apoteker ingin melakukan transaksi pembelian obat

Post-condition : apoteker telah melakukan transaksi pembelian obat

Description : apoteker melakukan transaksi pembelian obat agar dapat menyimpan data transaksi pembelian obat

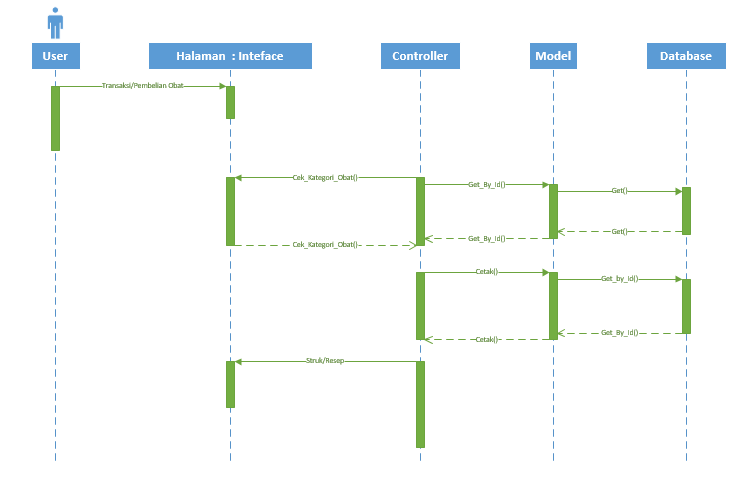
|  |  |
| --- | --- |
| Apoteker | System |
| 1. Apoteker memilih fitur transaksi obat |  |
| 1. Apoteker memilih fitur transaksi pembelian bahan |  |
|  | 1. System akan menampilkan form transaksi pembelian bahan |
| 1. Apoteker akan menginputkan data yang diminta oleh form sesuai dengan data transaksi pembelian bahan dari supplier |  |
|  | 1. System akan menyimpan data transaksi pembelian bahan ke dalam database |

#### 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:*

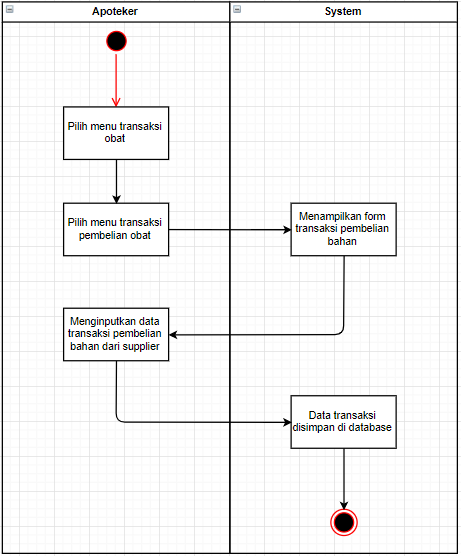
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



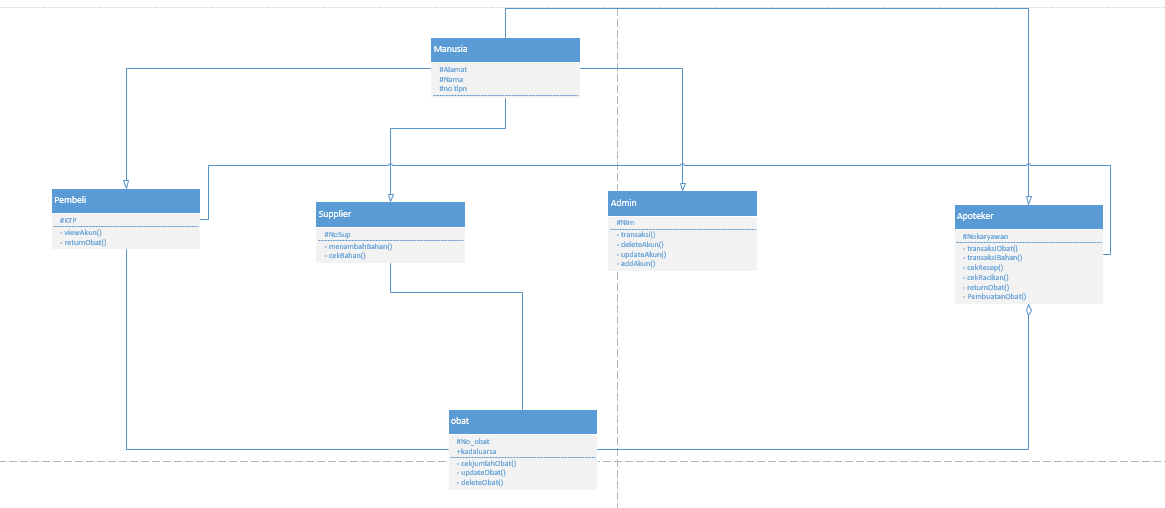
Gambar Sequence Diagram Transaksi Pembelian Bahan

#### Activity Diagram



Gambar Activity Diagram Use Case Transaksi Pembelian Bahan

#### 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.* | *Obat* |  |
|  |  |  |
|  |  |  |

*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*

Username = input

Password = input

Select \* from databaseakun where username=Username and password=Password;

If(jenis\_user==admin) then

begin

Session = login\_admin

Show HalamanAdmin

end

Else if(jenis\_user==apoteker) then

begin

Session = login\_apoteker

Show HalamanAdminApoteker

end

Else if(jenis\_user==supplier) then

begin

Session = login\_supplier

Show HalamanSupplier

End

begin

Session = login\_Pembeli

Show HalamanPembeli

end

Else

Output(“username atau password salah”)

End if

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

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-001* | Select \* from databaseakun where username=Username and password=Password; | *Mencari Username dan Password yang sesuai dengan inputan user (Pengguna)* |
|  |  |  |
|  |  |  |

*Nama Kelas :Halaman input Obat*

*Nama Operasi : Kelola Obat*

*Algoritma :Algo-002*

**Tambah Obat**

Id\_Obat, Nama\_Obat, Jenis\_Obat, Tanggal\_Kadaluarsa,Keterangan = Input\_Data = Id\_Obat, Nama\_Obat, Jenis\_Obat, Tanggal\_Kadaluarsa,Keterangan

if (select \* from Obat where Id\_Obat=Id\_Obat == null) then

insert into tanah values(Id\_Obat, Nama\_Obat, Jenis\_Obat, Tanggal\_Kadaluarsa,Keterangan)

else

Update Obat = data where Id\_Obat = Id\_Obat

**Ubah Data Obat**

Data = Id\_Obat, Nama\_Obat, Jenis\_Obat, Tanggal\_Kadaluarsa,Keterangan

update Obat set data=data where id\_Obat=id\_Obat

**Lihat Obat**

cari = input

select id\_Obat,nama\_Obat from Obat where idObat=cari or namaObat=cari

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

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-002* | select \* from Obat where Id\_Obat=Id\_Obat == null | *Mengecek di database obat apakah ada id obat yang sama, jika tidak ada, maka obat dapat ditambahkan* |
| *Q-003* | select id\_Obat,nama\_Obat from Obat where idObat=cari or namaObat=cari | *Menampilkan id\_Obat, nama\_Obat dari database jika inputan (CARI) sama dengan Id\_Obat* ***atau*** *Nama\_Obat* |
|  |  |  |

*Nama Kelas :Halaman Pembuatan Obat*

*Nama Operasi : Pembuatan Obat*

*Algoritma :Algo-003*

Input data Obat

If (data\_Obat== Resep) then

If(data\_Obat ==Racikan) then

Buat obat racikan

Else

Buat obat non\_Racikan

Else

Buat Obat non resep

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

*Query :*

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

*Nama Kelas :Halaman Kelola Akun*

*Nama Operasi : Kelola Akun*

*Algoritma :Algo-004*

**Tambah Akun**

Id\_Akun, Nama\_Akun, Jenis\_Akun, Username, Password = Input\_Data = Id\_Akun, Nama\_Akun, Jenis\_Akun, Username, Password

if (select \* from akun where Id\_akun=Id\_akun == null) then

insert into tanah values(Id\_Akun, Nama\_Akun, Jenis\_Akun, Username, Password)

else

Update Akun = data where Id\_Akun = Id\_akun

**Ubah Akun**

Data = Id\_Akun, Nama\_Akun, Jenis\_Akun, Username, Password

update Akun set data=data where id\_Akun=id\_Akun

**Lihat Akun**

cari = input

select id\_Akun,nama\_Akun from Akun where id\_Akun=cari or nama\_Akun=cari

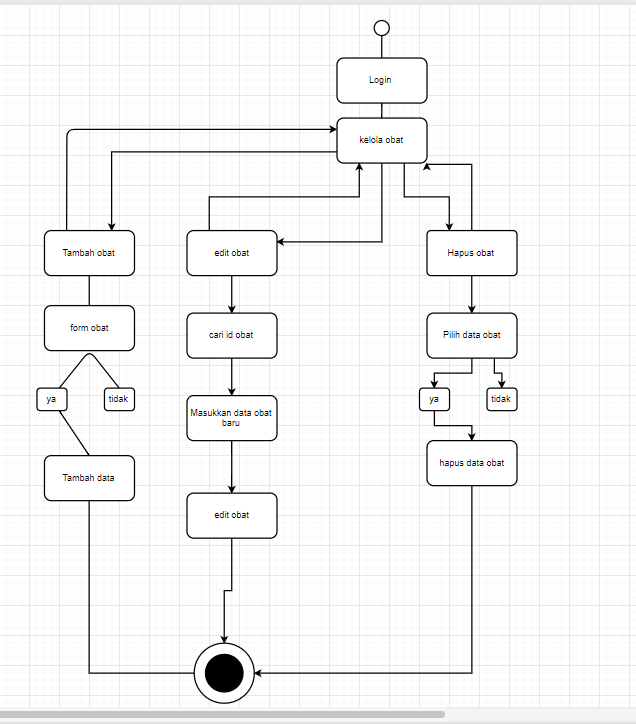
*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

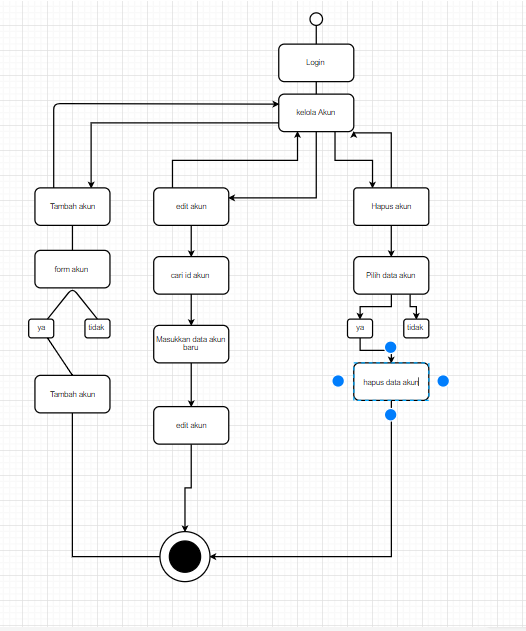
|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-004* | select \* from akun where Id\_akun=Id\_akun == null | *Mengecek database akun berdasarkan IdAkun, jika tidak ada yang sama, maka tambah akun dapat dilakukan.* |
| *Q-005* | select id\_Akun,nama\_Akun from Akun where id\_Akun=cari or nama\_Akun=cari | *Menampilkan id\_akun, nama\_akun dari database jika inputan (CARI) sama dengan Id\_akun* ***atau*** *Nama\_Akun* |
|  |  |  |

## 

## Diagram Statechart



Gambar Statechart Diagram Kelola Akun

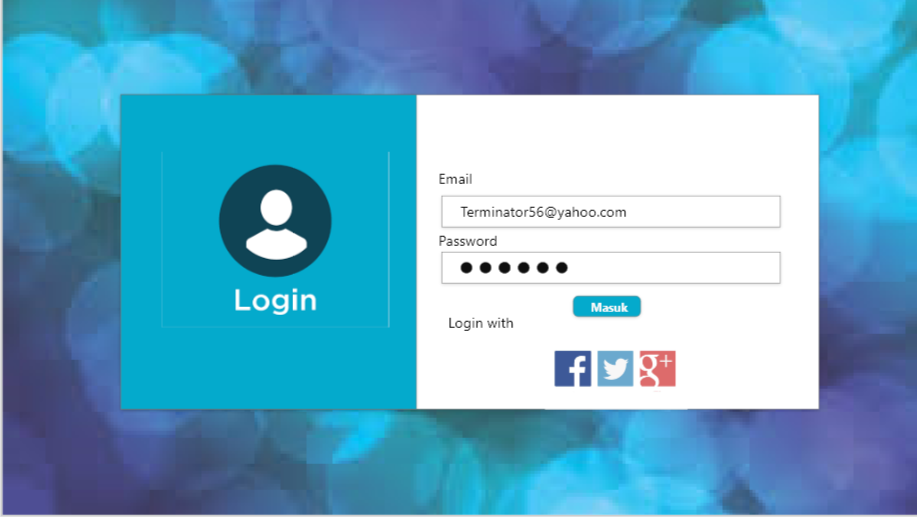


Gambar Statechart Diagram Kelola Obat

## Perancangan Antarmuka

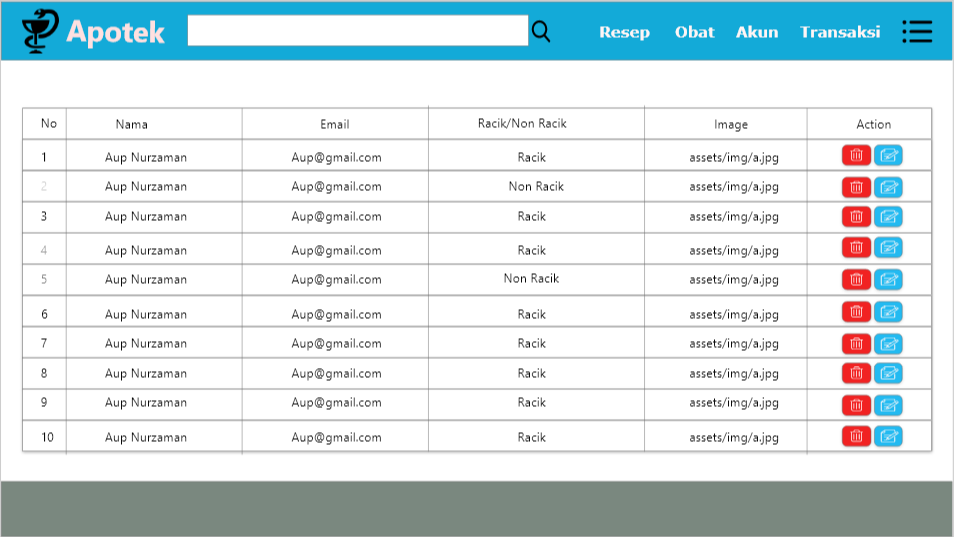
*Bagian ini diisi dengan versi awal prototipe antarmuka.*

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



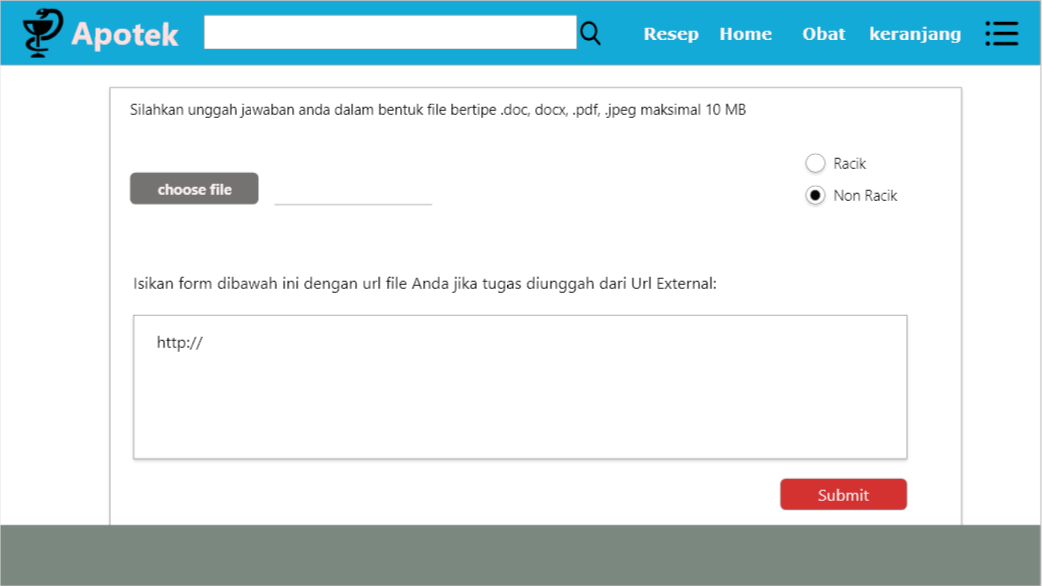
*Gambar ... Login*

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
|  |  | *Diisi dengan string yg tampil pd layar* | 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. |
| *Button1* | Button | OK | Jika diklik, akan mengaktifkan Proses AlgoXXX. |
| *RTF1* | *RTF Box* |  | Isi Teks yang disimpan pada File xxx |



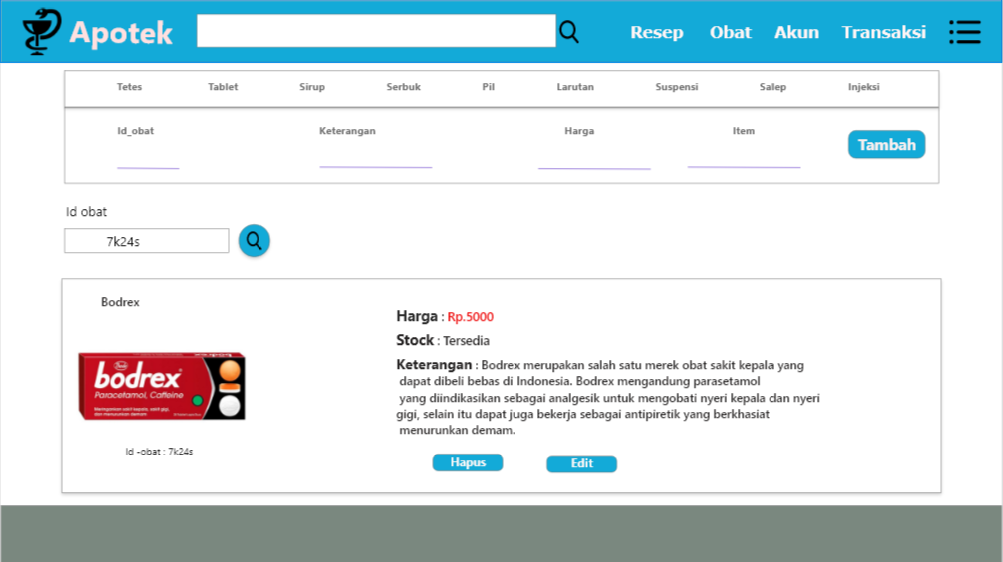
Gambar ... Kelola Akun

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
|  |  | *Diisi dengan string yg tampil pd layar* | 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. |
| *Button1* | Button | OK | Jika diklik, akan mengaktifkan Proses AlgoXXX. |
| *RTF1* | *RTF Box* |  | Isi Teks yang disimpan pada File xxx |



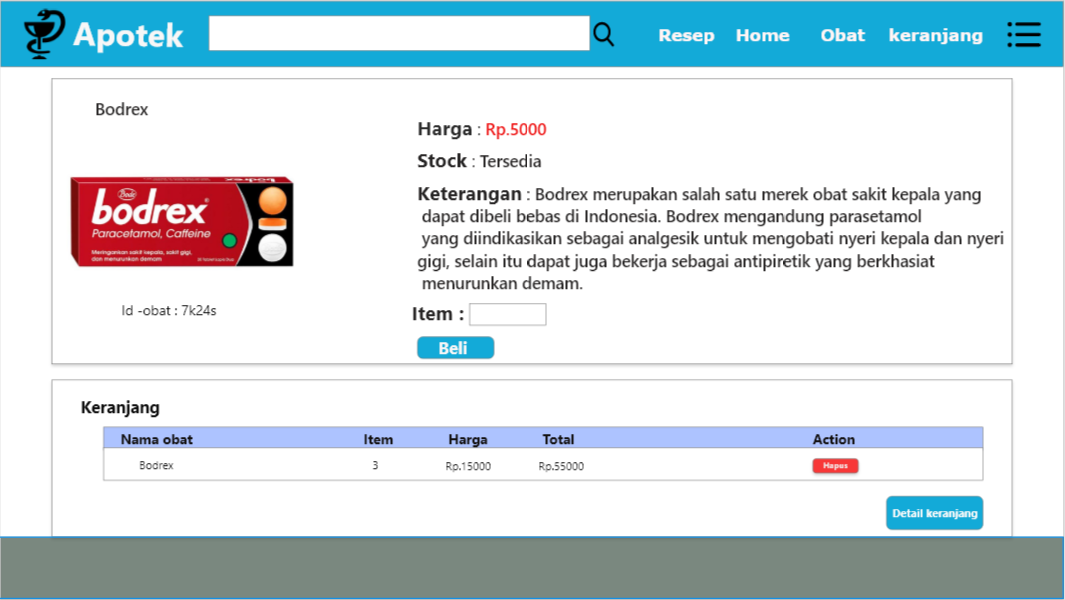
Gambar ... Pembuatan Obat

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
|  |  | *Diisi dengan string yg tampil pd layar* | 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. |
| *Button1* | Button | OK | Jika diklik, akan mengaktifkan Proses AlgoXXX. |
| *RTF1* | *RTF Box* |  | Isi Teks yang disimpan pada File xxx |



Gambar ... Kelola Obat

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
|  |  | *Diisi dengan string yg tampil pd layar* | 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. |
| *Button1* | Button | OK | Jika diklik, akan mengaktifkan Proses AlgoXXX. |
| *RTF1* | *RTF Box* |  | Isi Teks yang disimpan pada File xxx |



Gambar ... Transaksi

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
|  |  | *Diisi dengan string yg tampil pd layar* | 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. |
| *Button1* | Button | OK | Jika diklik, akan mengaktifkan Proses AlgoXXX. |
| *RTF1* | *RTF Box* |  | Isi Teks yang disimpan pada File xxx |

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 | Mengupdate data akun |
| SKPL- Apotek -004 | Menampilkan data akun |
| SKPL- Apotek -005 | Menghapus data akun |
| SKPL- Apotek -006 | Pembuatan obat |
| SKPL- Apotek -007 | Pendataan resep |
| SKPL- Apotek -008 | Pengecekan jumlah obat |
| SKPL- Apotek -009 | Mengupdate data obat |
| SKPL- Apotek -010 | Menambahkan data obat |
| SKPL- Apotek -011 | Menghapus data obat |
| SKPL- Apotek -012 | Pengembalian obat |
| SKPL- Apotek -013 | Penerimaan obat |
| SKPL- Apotek -014 | Pemberian obat baru |
| SKPL- Apotek -015 | Transaksi pembelian obat |
| SKPL- Apotek -016 | Transaksi pembelian bahan |
| SKPL- Apotek -017 | Keluar Sistem |