

DPPL-xx

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

Hello Madicine


Dipersiapkan oleh:

Rafly Ahmad Yanuar	1301194119
Firsta Daffa Bintang Wilaga	1301194215
Ahmad Luthfi Ridho	1301194142

Program Studi Informatika

Fakultas Informatika

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

 Telkom University	Prodi S1- Informatika Universitas Telkom	Nomor Dokumen		Halaman
		DPPL-XX <xx:no grp>		<#>/<jml #
		Revisi	<nomor revisi>	Tgl: <isi tanggal>

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	5
Tujuan Penulisan Dokumen	6
Lingkup Masalah	6
Definisi dan Istilah	6
Referensi	6
Sistematika Pembahasan	6
Pendahuluan	6
Deskripsi Perancangan Global	7
Perancangan Rinci	7
Matriks Keruntutan	7
Deskripsi Perancangan Global	8
Rancangan Lingkungan Implementasi	8
Deskripsi Arsitektural	8
Deskripsi Komponen	9
Perancangan Rinci	10
Realisasi Use Case	10
Use Case Register	10
Identifikasi Kelas	10
Sequence Diagram	11
Diagram Kelas	12
Use Case Login	12
Identifikasi Kelas	13
Sequence Diagram	13
Diagram Kelas	14
Browse Medicine	14
Identifikasi Kelas	15
Sequence Diagram	15
Diagram Kelas	16
Use Case Make Order	16
Identifikasi Kelas	17
Sequence Diagram	17
Diagram Kelas	18
Use Case Checkout	18
Identifikasi Kelas	19
Sequence Diagram	19
Diagram Kelas	20

Use Case Delete Order	21
Identifikasi Kelas	21
Sequence Diagram	21
Diagram Kelas	22
Use Case EditProfile	23
Identifikasi Kelas	23
Sequence Diagram	24
Diagram Kelas	25
Use Case Organize Product	25
Identifikasi Kelas	26
Sequence Diagram	27
Diagram Kelas	28
Use Case Order Confirmation	28
Identifikasi Kelas	29
Sequence Diagram	30
Diagram Kelas	31
Use Case LogOut	31
Identifikasi Kelas	32
Sequence Diagram	32
Diagram Kelas	32
Perancangan Detail Kelas	33
Kelas <UserController>	33
Kelas <MedicineController>	34
Kelas <LoginController>	34
Kelas <OrderController>	35
Kelas <User>	35
Kelas <Medicine>	35
Kelas <Order>	36
Diagram Kelas Keseluruhan	36
Algoritma/Query	36
Perancangan Antarmuka	37
Perancangan Representasi Persistensi Kelas	42
Matriks Kerunutan	42

1. Pendahuluan

1.1 Tujuan Penulisan Dokumen

Dokumen ini berisi penjelasan mengenai Deskripsi Perancangan Perangkat Lunak (DPPL) untuk Aplikasi web Hello Medicine. 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 di akhir pembangunan perangkat lunak.

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

1.2 Lingkup Masalah

Perangkat lunak yang akan dikembangkan dalam makalah ini adalah Online Marketplace Hello Medicine, yang merupakan perangkat lunak yang dibangun untuk mempermudah komunikasi antar entitas yang ada di dunia medis, seperti pelaku yang sedang sakit. Aplikasi ini digunakan sebagai media pembelian obat secara online. Aktivitas yang dapat dilakukan pada aplikasi ini yaitu pelaku dapat membuat akun dan mencari obat yang dibutuhkan,

1.3 Definisi dan Istilah

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

- DPPL: Dokumen yang mendeskripsikan dan menjabarkan secara terperinci mengenai perancangan perangkat lunak yang akan dibangun.
- SKPL: Singkatan dari Spesifikasi Kebutuhan Perangkat Lunak atau sering juga disebut sebagai Software Requirements Specification (SRS) merupakan spesifikasi dari perangkat lunak yang akan dikembangkan.
- DBMS: Singkatan dari “Database Management System” yaitu sistem pengorganisasian dan sistem pengolahan Database pada komputer.
- PHP: Bahasa pemrograman server side yang digunakan untuk membangun web yang dinamis

1.4 Referensi

1. IF-43-09. (2021). SKPL Hello Medicine.

1.5 Sistematika Pembahasan

Dokumen DPPL ini berisikan deskripsi perancangan perangkat lunak Hello Medicine yang akan dikembangkan berdasarkan dokumen SKPL. Pada dokumen DPPL ini akan dijelaskan rincian dari perancangan perangkat lunak sehingga dapat diimplementasikan. Dokumen ini secara garis besar terdiri dari empat bab dengan rincian sebagai berikut:

1.5.1 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.

1.5.2 Deskripsi Perancangan Global

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

1.5.3 Perancangan Rinci

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

1.5.4 Matriks Kerunutan

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

2 Deskripsi Perancangan Global

2.1 Rancangan Lingkungan Implementasi

Sistem ini diimplementasikan dalam lingkungan sebagai berikut:

Sistem operasi: Microsoft Windows 11

Bahasa pemrograman: HTML, CSS, JavaScript, PHP

DBMS: PostgreSQL

Development tools: VisualStudio Code

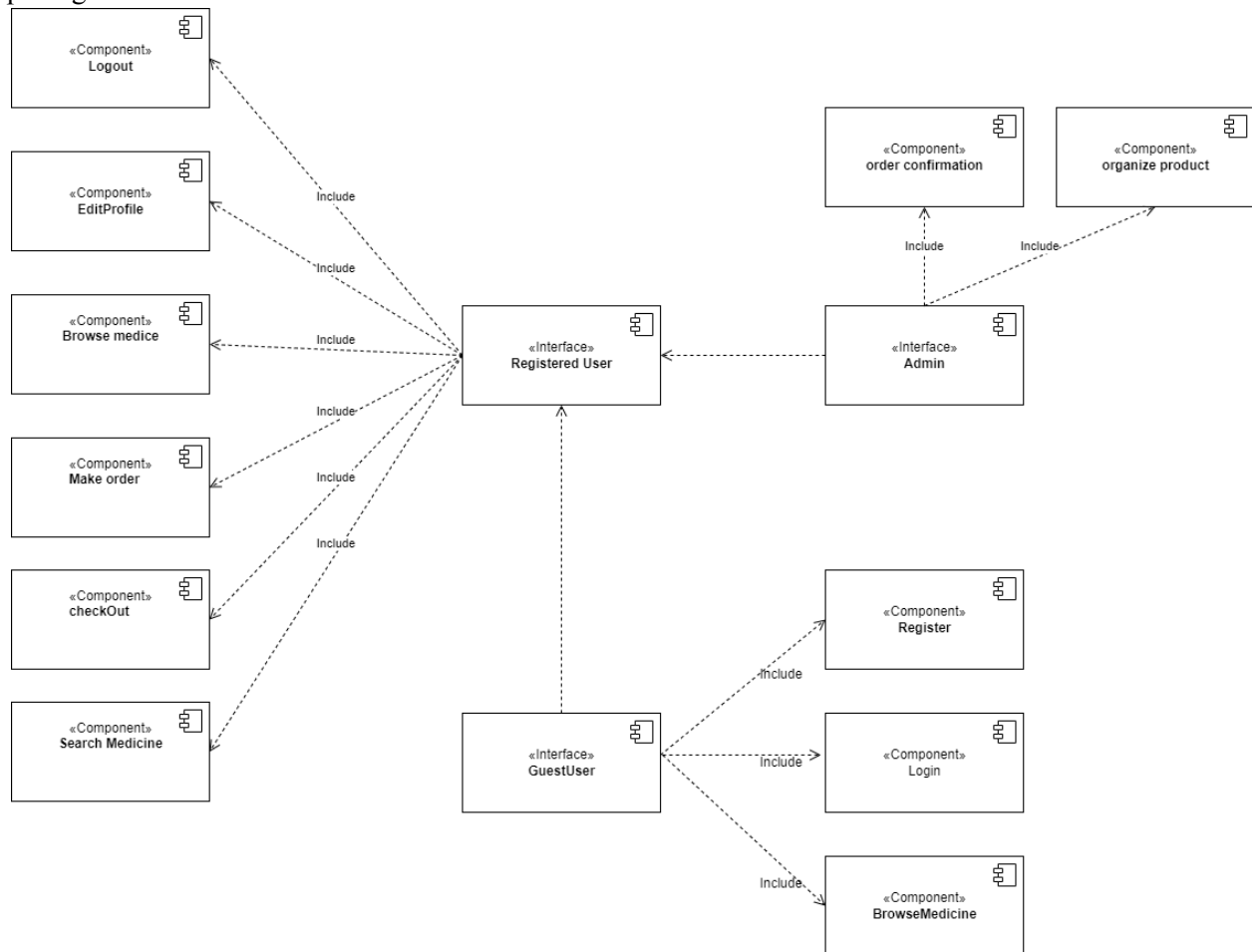
Software yang dibutuhkan pada sisi pengguna:

Browser: Google Chrome/Microsoft Edge

Koneksi Internet

2.2 Deskripsi Arsitektural

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



2.3 Deskripsi Komponen

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

No	Nama Komponen	Keterangan
1	Register	Membuat Akun
2	Login	Autentikasi Akun
3	BrowseMedicine	Mencari Obat dengan manual
4	SearchMedicine	Mencari Obat dengan menggunakan Nama atau Komponen
5	Make Order	Membuat orderan
6	Checkout	Melakukan checkout setelah membuat orderan
7	DeleteOrder	Pemesan dapat menghapus obat di dalam keranjangnya yang belum di checkout
8	EditProfile	Untuk menyunting Profile
9	OrganizeProduct	User Admin dapat mengorganisir obat seperti, menambah obat. Menghapus obat, dan mengedit obat
10	Order confirmation	User Admin mengkonfirmasi pesanan yang masuk
11	Logout	Menghilangkan Autentikasi Akun

3 Perancangan Rinci

3.1 Realisasi Use Case

3.1.1 Use Case Register

Nama Use Case	Register	
Deskripsi	Proses pembuatan akun	
Aktor	User	
Pre-Kondisi	User sudah membuka menu register	
Post-Kondisi	Akun user sudah terdaftar pada database	
Skenario Utama	Aktor	Sistem
	1. User menginput data registrasi	
	2. User meng-klik tombol regis	
		3. Sistem memvalidasi data
		4. Sistem mengarahkan user ke halaman welcome untuk melakukan login
Skenario Eksepsional (Alternative Flow)	Aktor	Sistem
		Sistem menampilkan pesan error apabila data gagal divalidasi

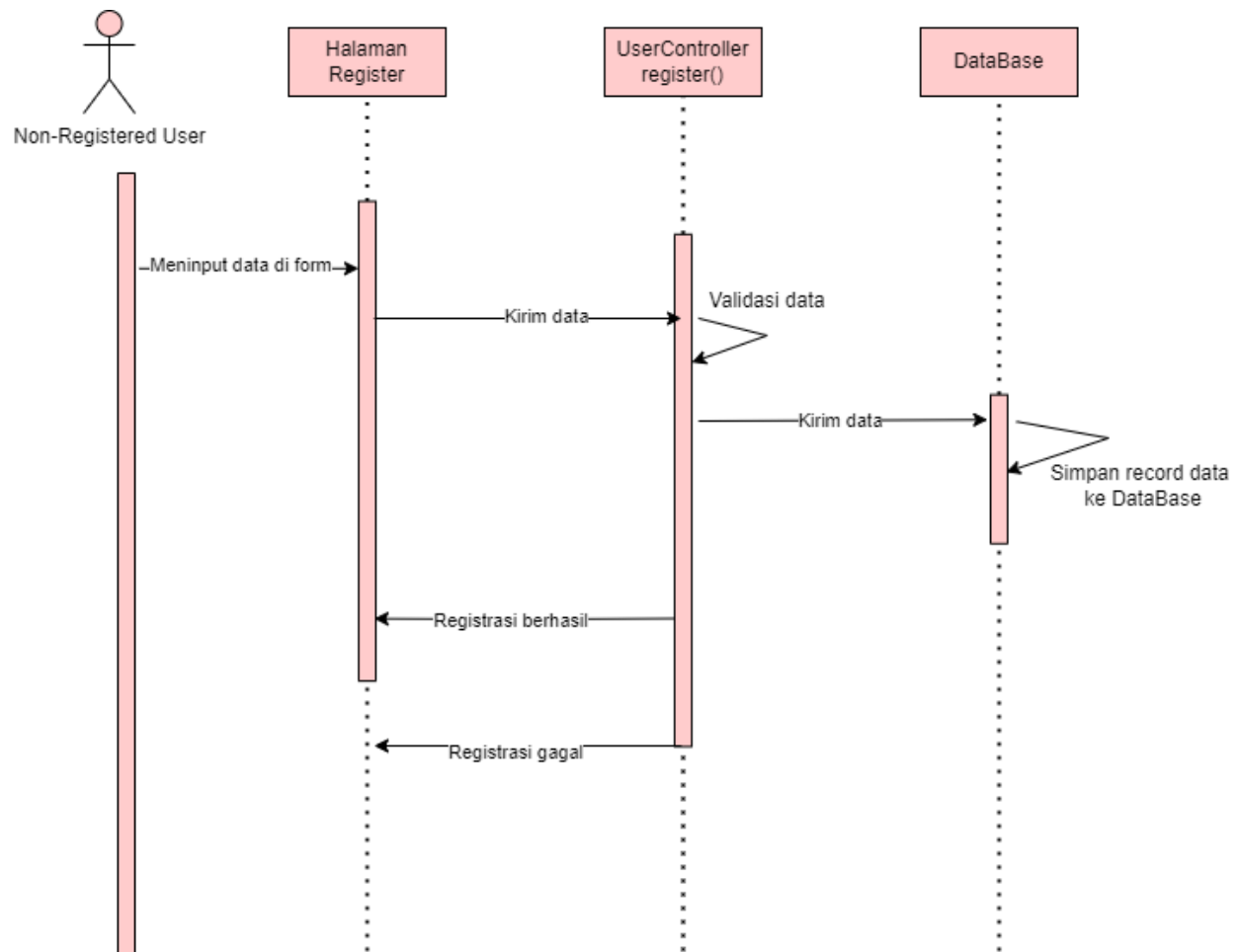
3.1.1.1 Identifikasi Kelas

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:

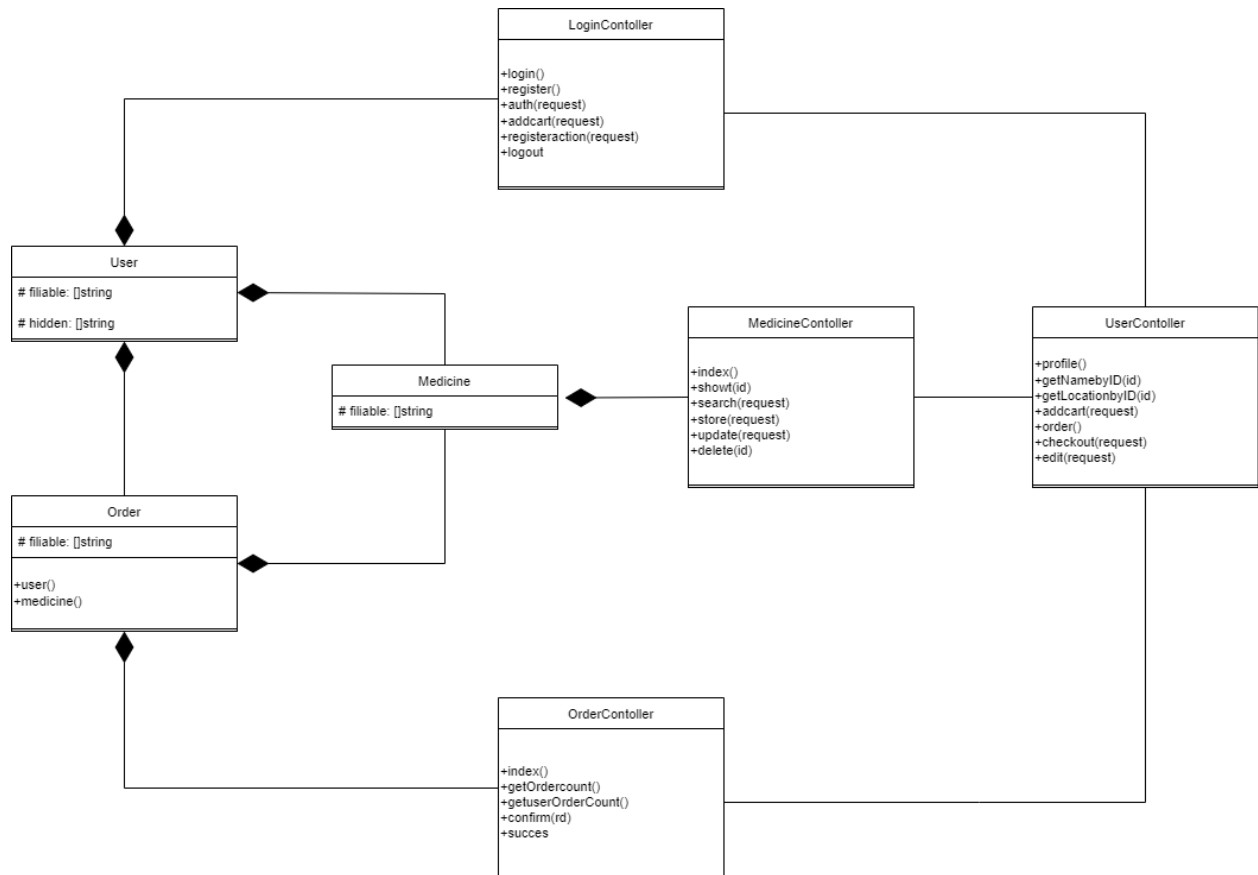
No	Nama Kelas Perancangan	Tipe Kelas
1	LoginController	Controller

*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller

3.1.1.2 Sequence Diagram



3.1.1.3 Diagram Kelas



3.1.2 Use Case Login

Nama Use Case	Login	
Deskripsi	Proses autentikasi untuk memastikan user menggunakan akun yang valid	
Aktor	User	
Pre-Kondisi	Akun sudah harus terdaftar di DataBase	
Post-Kondisi	Sistem akan memberi akses kepada user, jika login gagal user harus mengulang input email atau password yang benar	
Skenario Utama	Aktor	Sistem
	1. Membuka menu login	
	2. Mengisi data di	

	form login	
		3. Memvalidasi input
		4. Sistem mengarahkan user ke halaman home
Skenario Eksepsional (Alternative Flow)	Aktor	Sistem
		4a. Jika salah sistem menampilkan pesan error

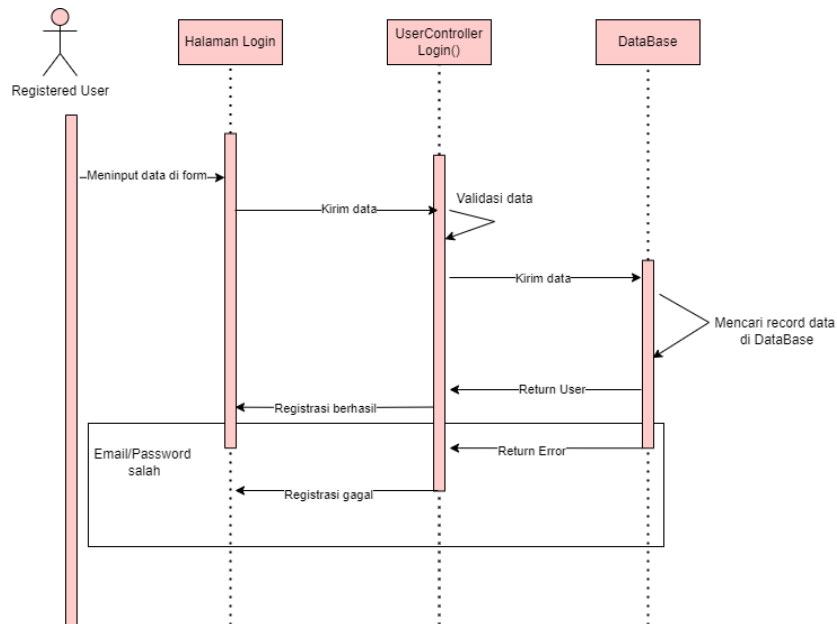
3.1.2.1 Identifikasi Kelas

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:

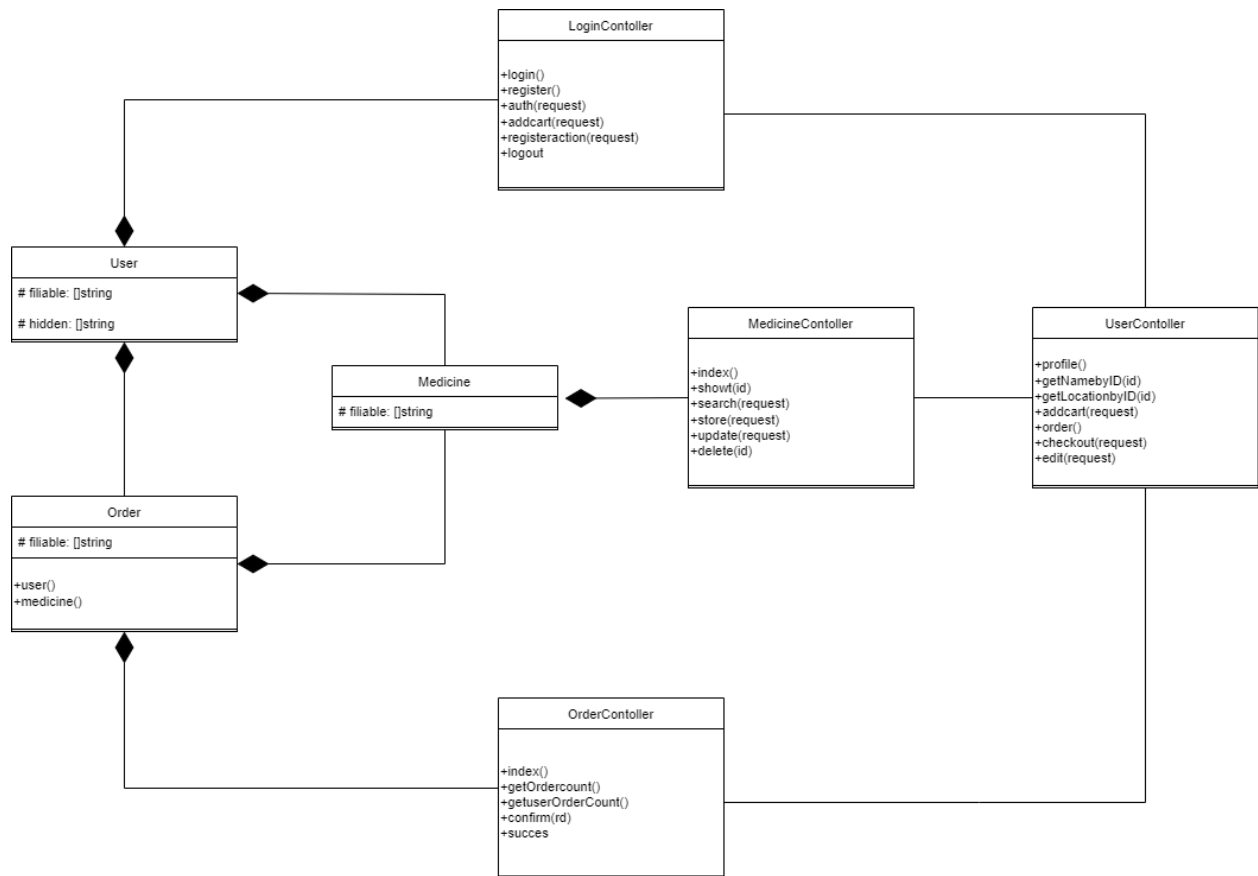
No	Nama Kelas Perancangan	Tipe Kelas
1	LoginController	Controller
2	User	Model

*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller

3.1.2.2 Sequence Diagram



3.1.2.3 Diagram Kelas



3.1.3 Browse Medicine

Nama Use Case	Browse Medicine	
Deskripsi	Proses mencari obat yang diinginkan	
Aktor	User	
Pre-Kondisi	User sudah melakukan login	
Post-Kondisi	User menemukan obat yang dicari	
Skenario Utama	Aktor	Sistem
	1. User login dan masuk ke halaman home	
	2. User mencari obat	
		3. Sistem memvalidasi akun user

		4. Sistem menampilkan obat di halaman home
Skenario Eksepsional (Alternative Flow)	Aktor	Sistem
		3a. Sistem menampilkan pesan error jika akun yang dimasukan salah

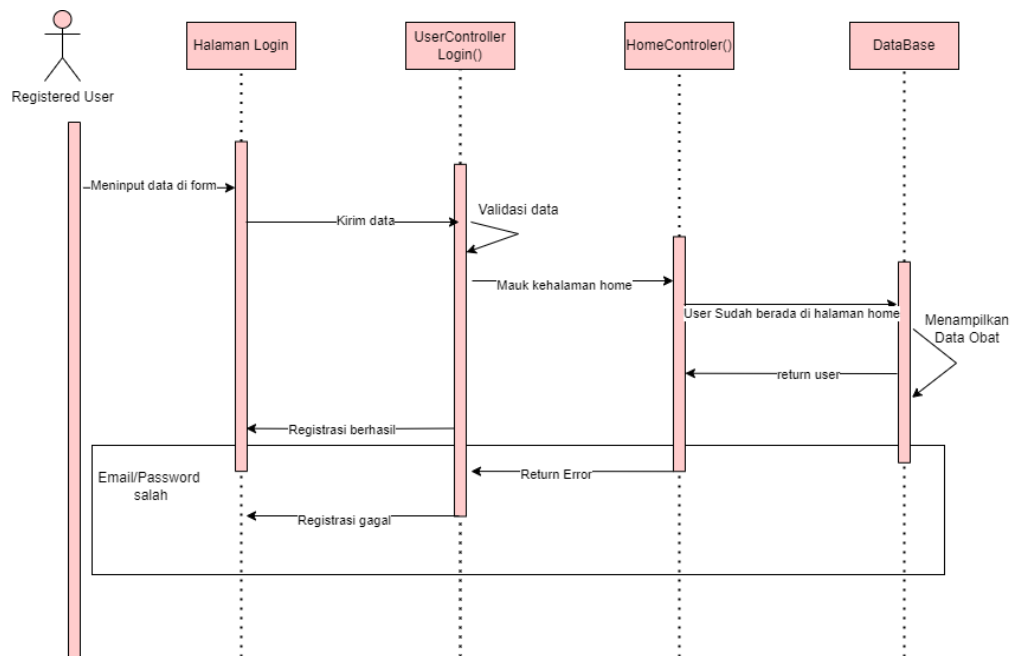
3.1.3.1 Identifikasi Kelas

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:

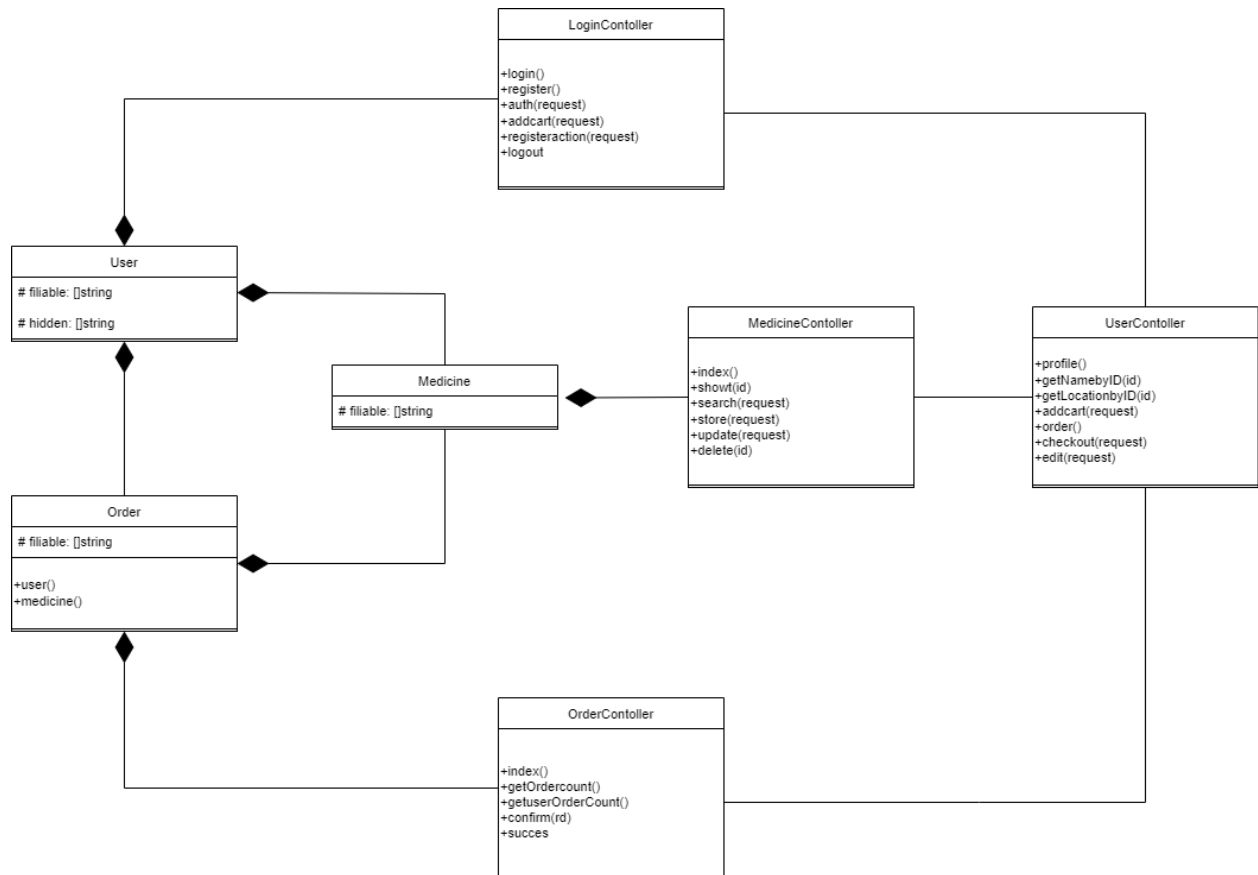
No	Nama Kelas Perancangan	Tipe Kelas
1	Medicine Controller	Controller
2	Medicine	Model

*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller

3.1.3.2 Sequence Diagram



3.1.3.3 Diagram Kelas



3.1.4 Use Case Make Order

Nama Use Case	MakeOrder	
Deskripsi	Proses melakukan pemesanan obat	
Aktor	User	
Pre-Kondisi	User sudah melakukan login	
Post-Kondisi	Obat yang dipilih user sudah masuk kedalam keranjang	
Skenario Utama	Aktor	Sistem
	1. User melakukan login	
	2. User memilih obat	
		3. Sistem memvalidasi data

		4. Sistem mengarahkan user ke halaman pesan
Skenario Eksepsional (Alternative Flow)	Aktor	Sistem
		3a. Sistem akan memberi pesan error jika login gagal

3.1.4.1 Identifikasi Kelas

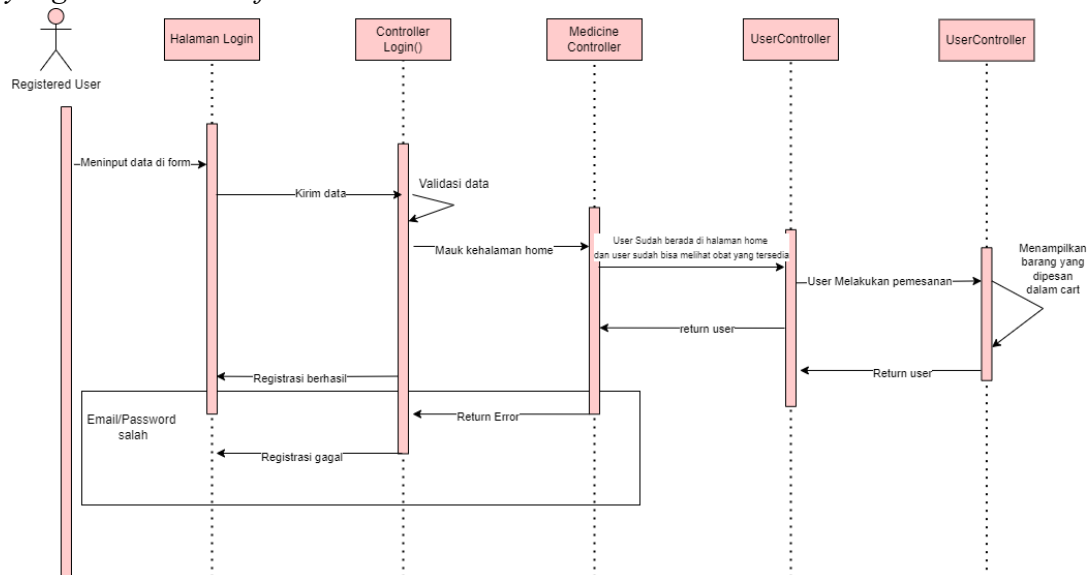
Identifikasi kelas yang terkait dengan use case tersebut. Kelas di tahap perancangan dapat berbeda dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:

No	Nama Kelas Perancangan	Tipe Kelas
1	Order Controller	Controller
2	Order	Model
3	Medicine Controller	Controller
4	User Controller	Controller

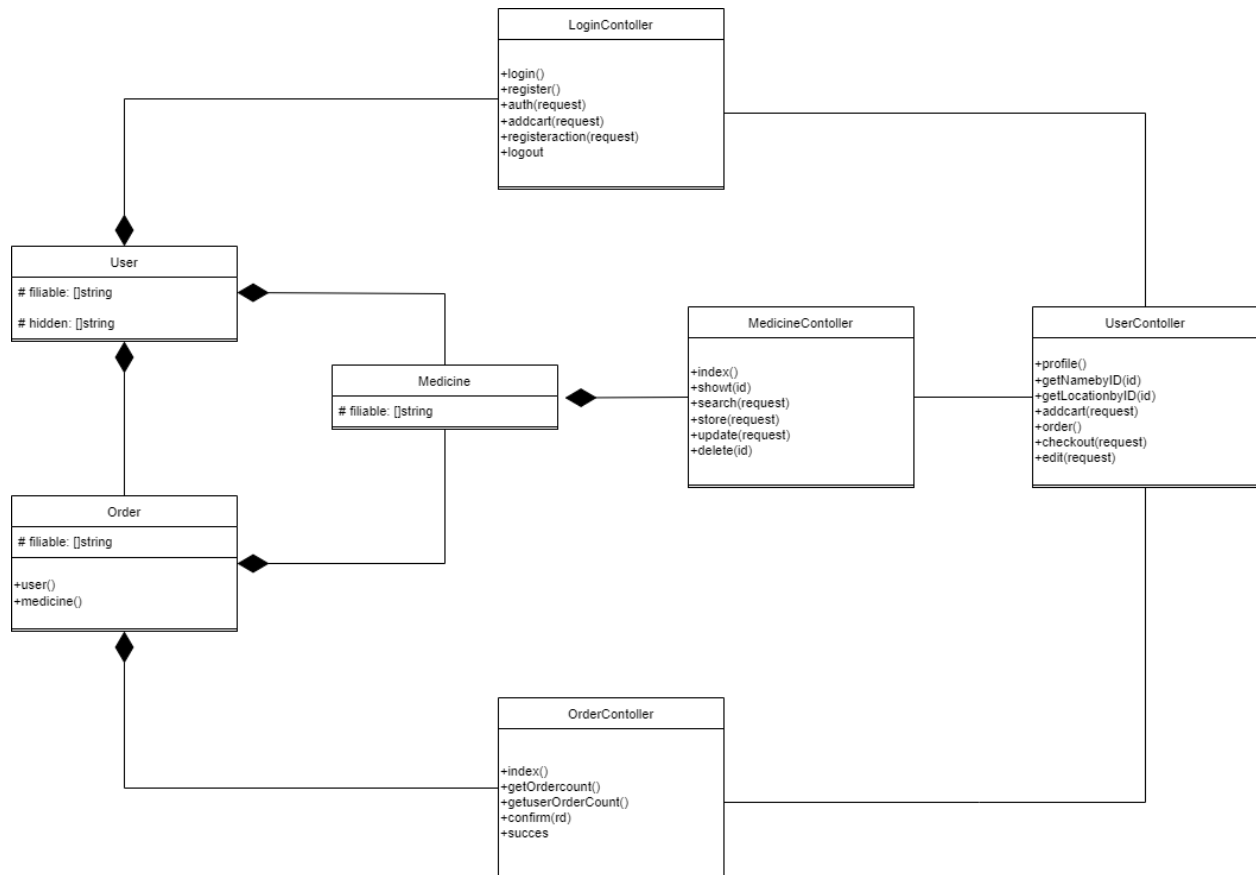
*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller

3.1.4.2 Sequence Diagram

Buatlah diagram sequence untuk setiap skenario use case. Skenario melibatkan kelas-kelas yang sudah diidentifikasi.



3.1.4.3 Diagram Kelas



3.1.5 Use Case Checkout

Nama Use Case	Checkout	
Deskripsi	Proses melakukan Checkout dan pembayaran setelah melakukan pemesanan	
Aktor	User	
Pre-Kondisi	User sudah melakukan pemesanan dan pesanan tersimpan di cart	
Post-Kondisi	Pesanan berhasil ter checkout dan terkirim ke pada admin untuk di konfirmasi	
Skenario Utama	Aktor	Sistem
	1. User melakukan pemesanan	
	2. User melakukan	

	checkout dan pembayaran	
		3. Sistem memvalidasi data
		4. Sistem menyimpan hasil pesanan di dalam cart
		5. Sistem mengeluarkan form dan note untuk no rekening pembeli dan note no rekening Admin
Skenario Eksepsional (Alternative Flow)	Aktor	Sistem

3.1.5.1 Identifikasi Kelas

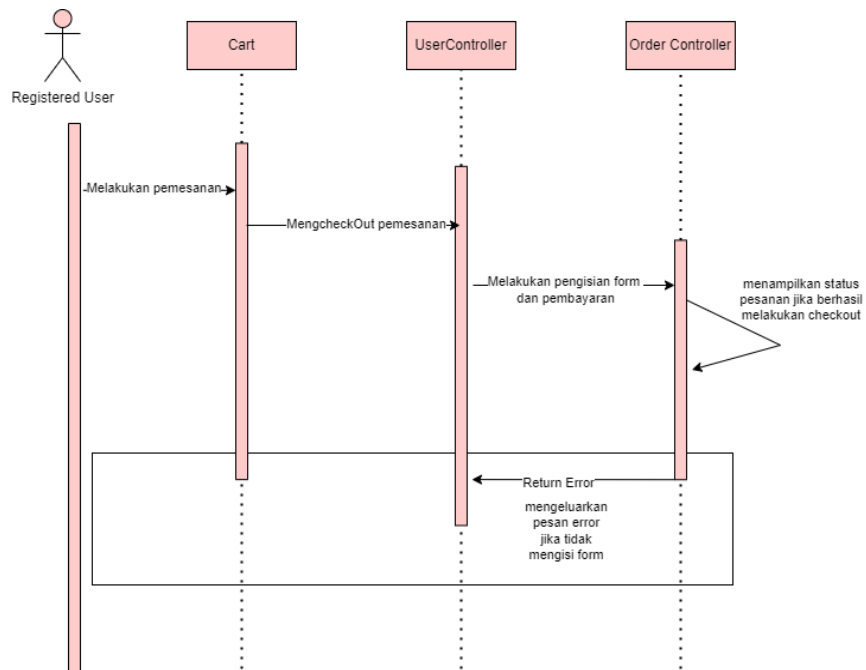
Identifikasi kelas yang terkait dengan use case tersebut. Kelas di tahap perancangan dapat berbeda dengan dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:

No	Nama Kelas Perancangan	Tipe Kelas
1	User Controller	Controller
2	Order Controller	Controller
3	Order	model

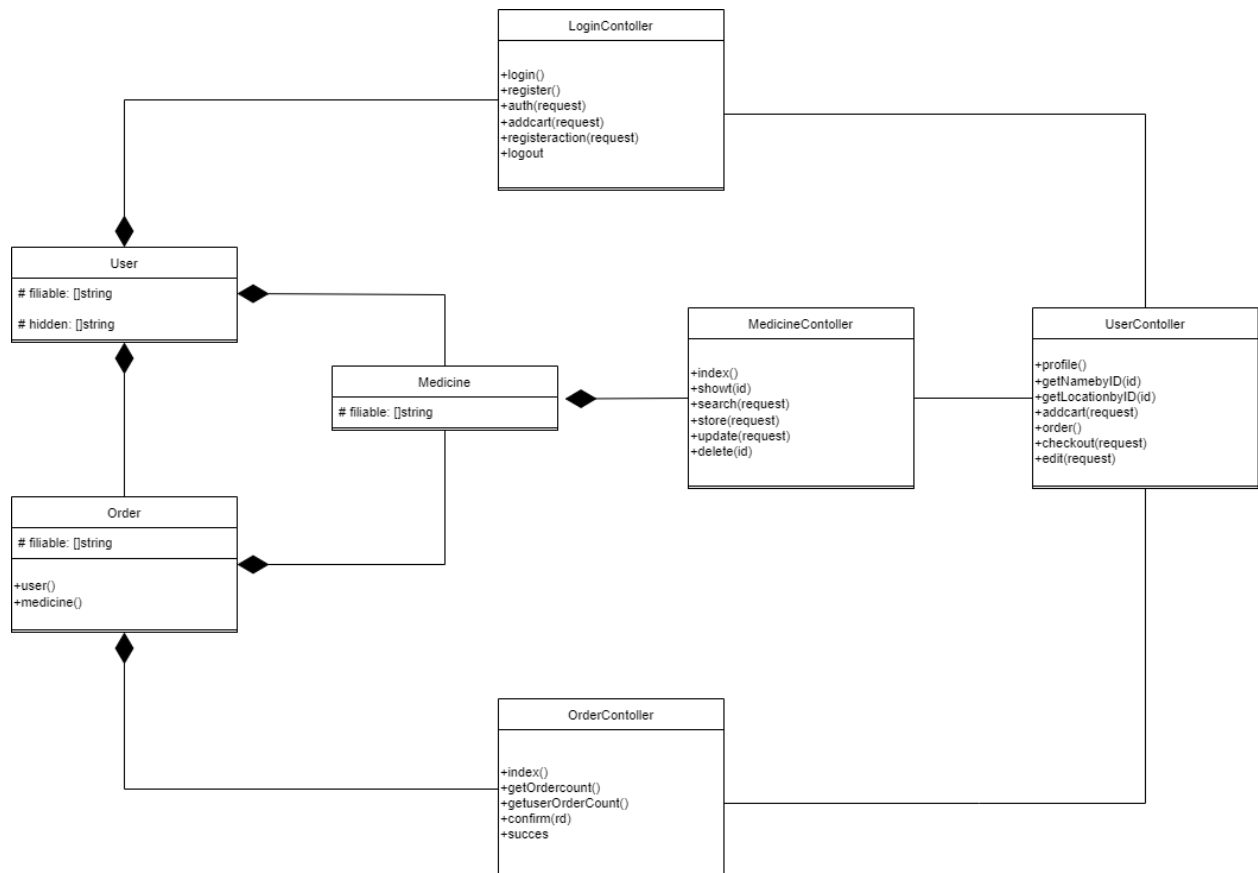
**Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

3.1.5.2 Sequence Diagram

Buatlah **diagram sequence** untuk setiap skenario use case. Skenario melibatkan kelas-kelas yang sudah diidentifikasi.



3.1.5.3 Diagram Kelas



3.1.6 Use Case Delete Order

Nama Use Case	Delete Order	
Deskripsi	Proses penghapusan obat yang sudah tersimpan di cart tetapi belum melakukan checkout	
Aktor	User	
Pre-Kondisi	User sudah melakukan pemesanan	
Post-Kondisi	Pesanan yang sudah berada di cart user akan hilang	
Skenario Utama	Aktor	Sistem
	1. User sudah melakukan pemesanan	
	2. User melakukan penghapusan obat pada cart	
		3. Sistem ngerespon dan menghapus pesanan yang berada dalam cart
Skenario Eksepsional (Alternative Flow)	Aktor	Sistem

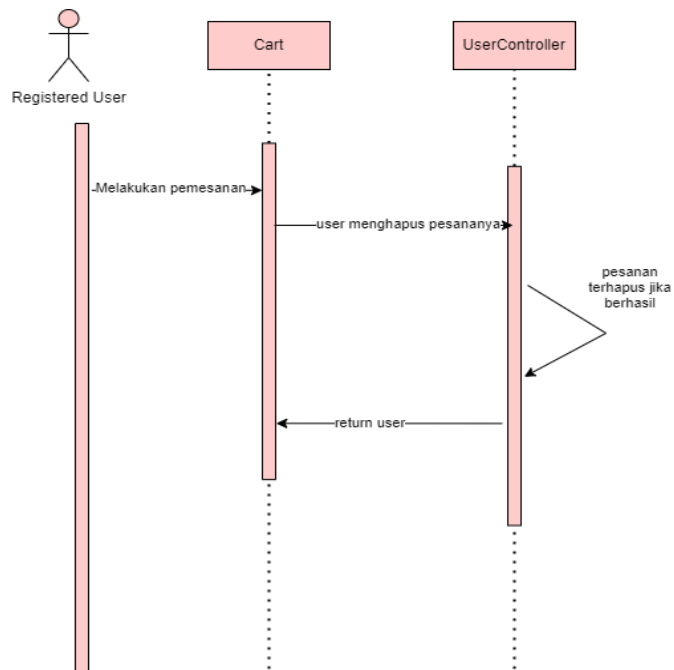
3.1.6.1 Identifikasi Kelas

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:

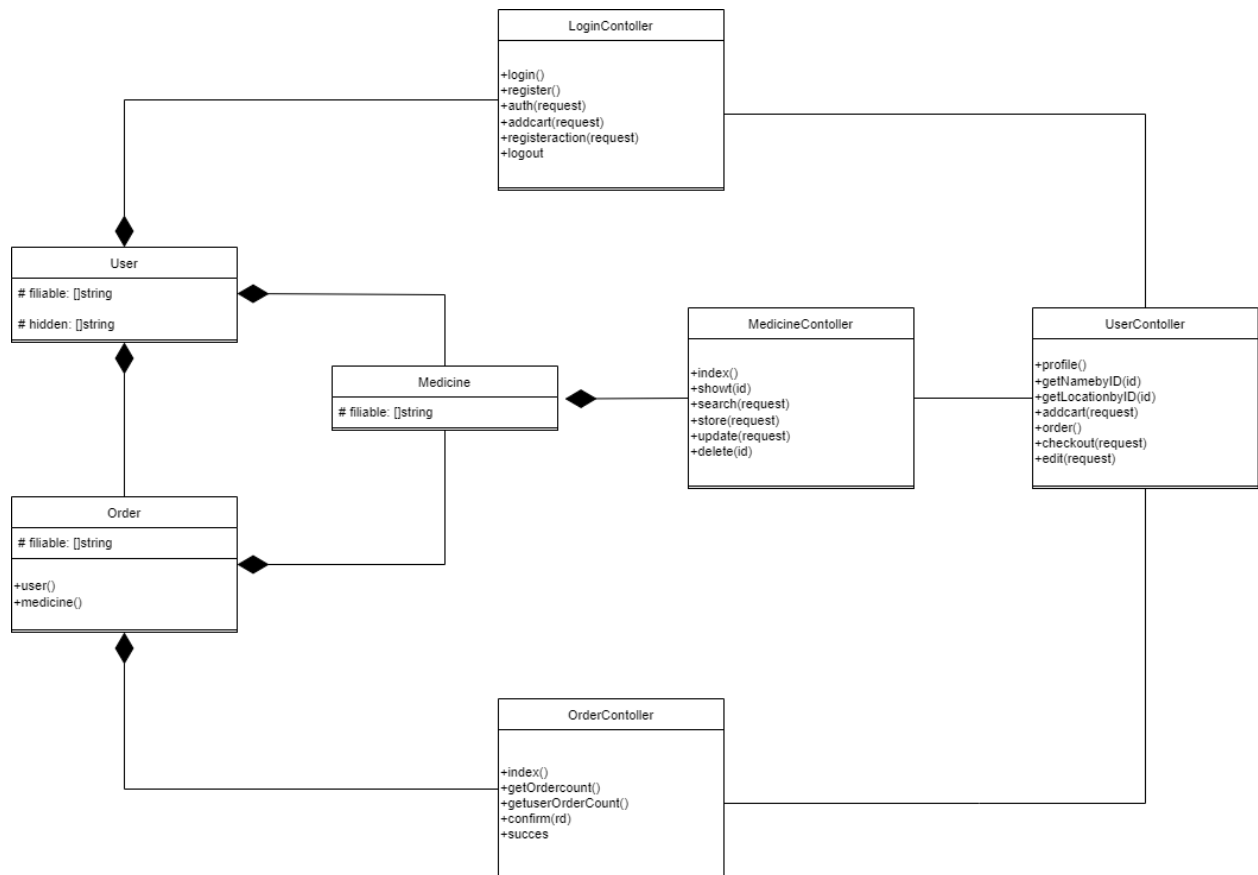
No	Nama Kelas Perancangan	Tipe Kelas
1	User Controller	Controller
2	Order	model

**Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

3.1.6.2 Sequence Diagram



3.1.6.3 Diagram Kelas



3.1.7 Use Case EditProfile

Nama Use Case	EditProfile	
Deskripsi	Proses menyunting profile	
Aktor	User	
Pre-Kondisi	User sudah memiliki data saat melakukan registrasi	
Post-Kondisi	Data user sudah berubah menjadi yang terbaru	
Skenario Utama	Aktor	Sistem
	1. User membuka menu profile	
	2. User mengedit data yang sudah ada	
		3. Sistem mengrespon dan mengubah profile menjadi terbaru
Skenario Eksepsional (Alternative Flow)	Aktor	Sistem
		3a. Mengeluarkan pesan error ketika data ada yang tidak terisi

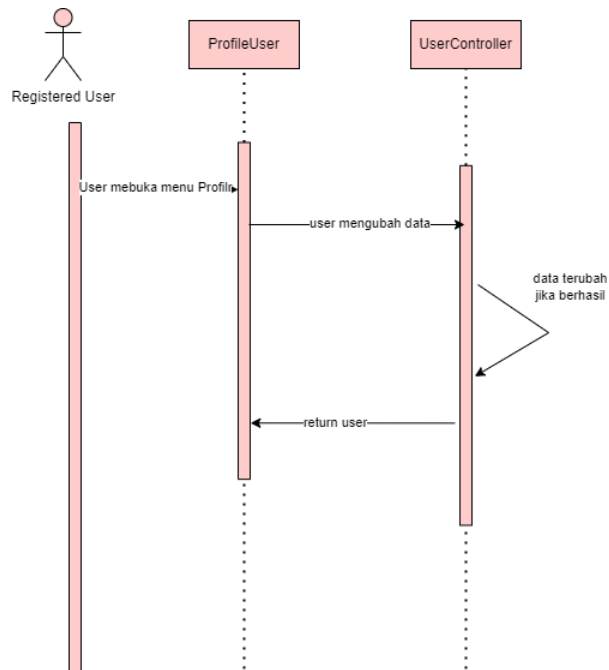
3.1.7.1 Identifikasi Kelas

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:

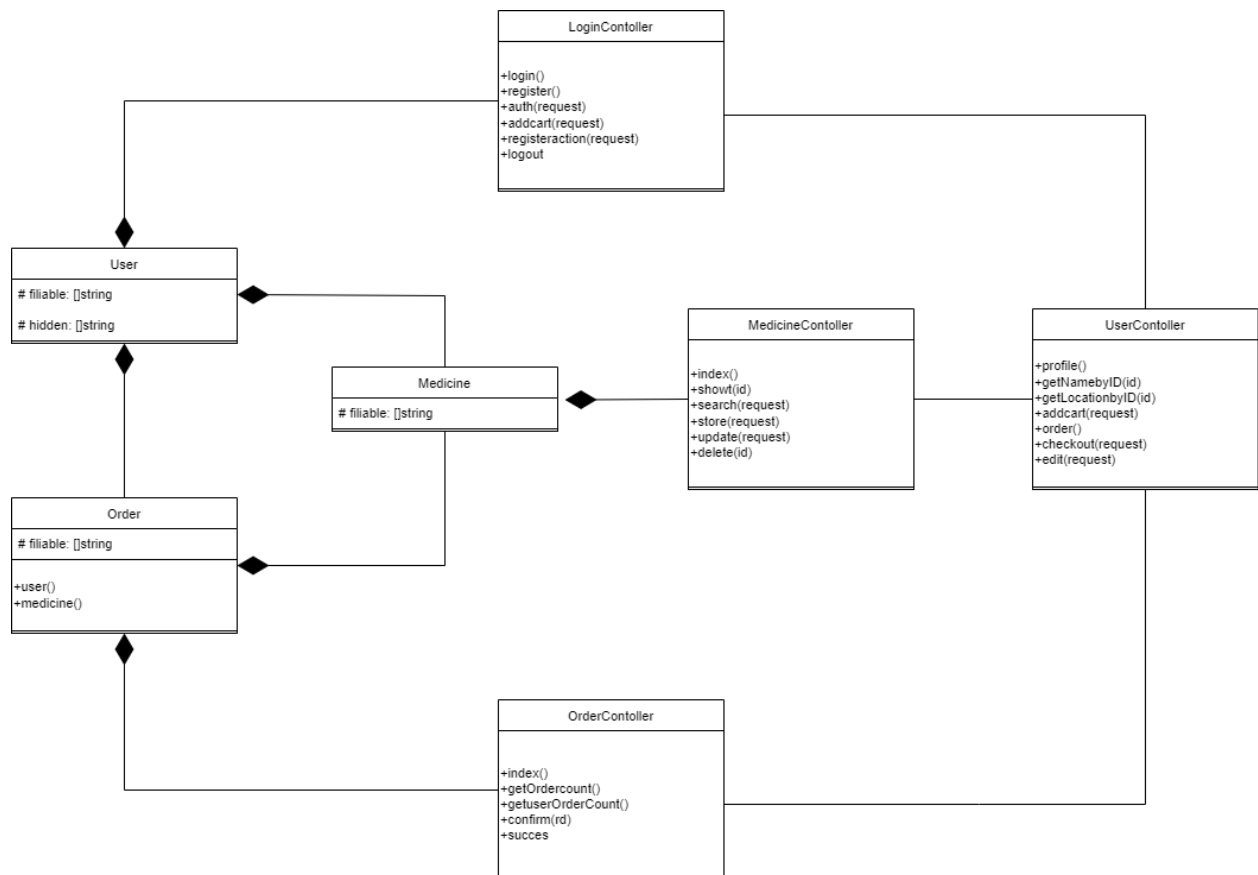
No	Nama Kelas Perancangan	Tipe Kelas
1	User Controller	Controller
2	Order	model

*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller

3.1.7.2 Sequence Diagram



3.1.7.3 Diagram Kelas



3.1.8 Use Case Organize Product

Nama Use Case	Organize Product	
Deskripsi	Proses mengatur produk	
Aktor	Admin	
Pre-Kondisi	Data obat sebelumnya sudah ada	
Post-Kondisi	Data obat, jumlah obat dan lain lain berubah	
Skenario Utama	Aktor	Sistem
	1. Admin melakukan login	
	2. Admin mengubah data obat	
	3. Admin menambah obat	
	4. Admin menghapus obat	
		5. Sistem memvalidasi data
		6. Sistem merespon perubahan data
		7. Sistem merespon menambahkan obat dan menampilkan obat yang baru ditambah
		8. Sistem merespon penghapusan obat
Skenario Eksepsional (Alternative Flow)	Aktor	Sistem
		5a. Mengeluarkan pesan error ketika validasi data ada yang salah 6a. Mengeluarkan pesan error ketika ada forum yang tidak diisi 7a. Mengeluarkan pesan error ketika ada forum yang tidak diisi

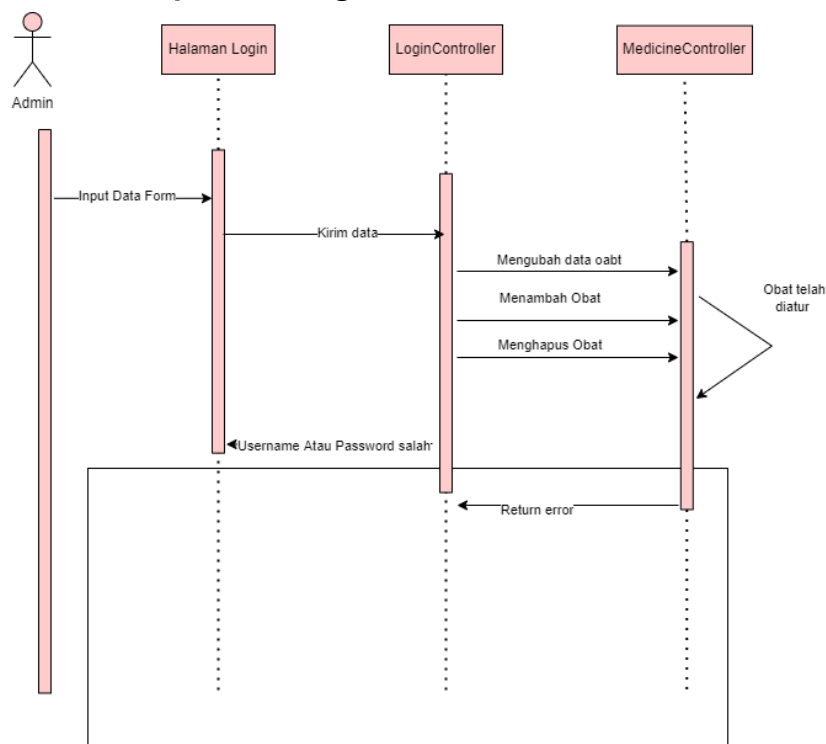
3.1.8.1 Identifikasi Kelas

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

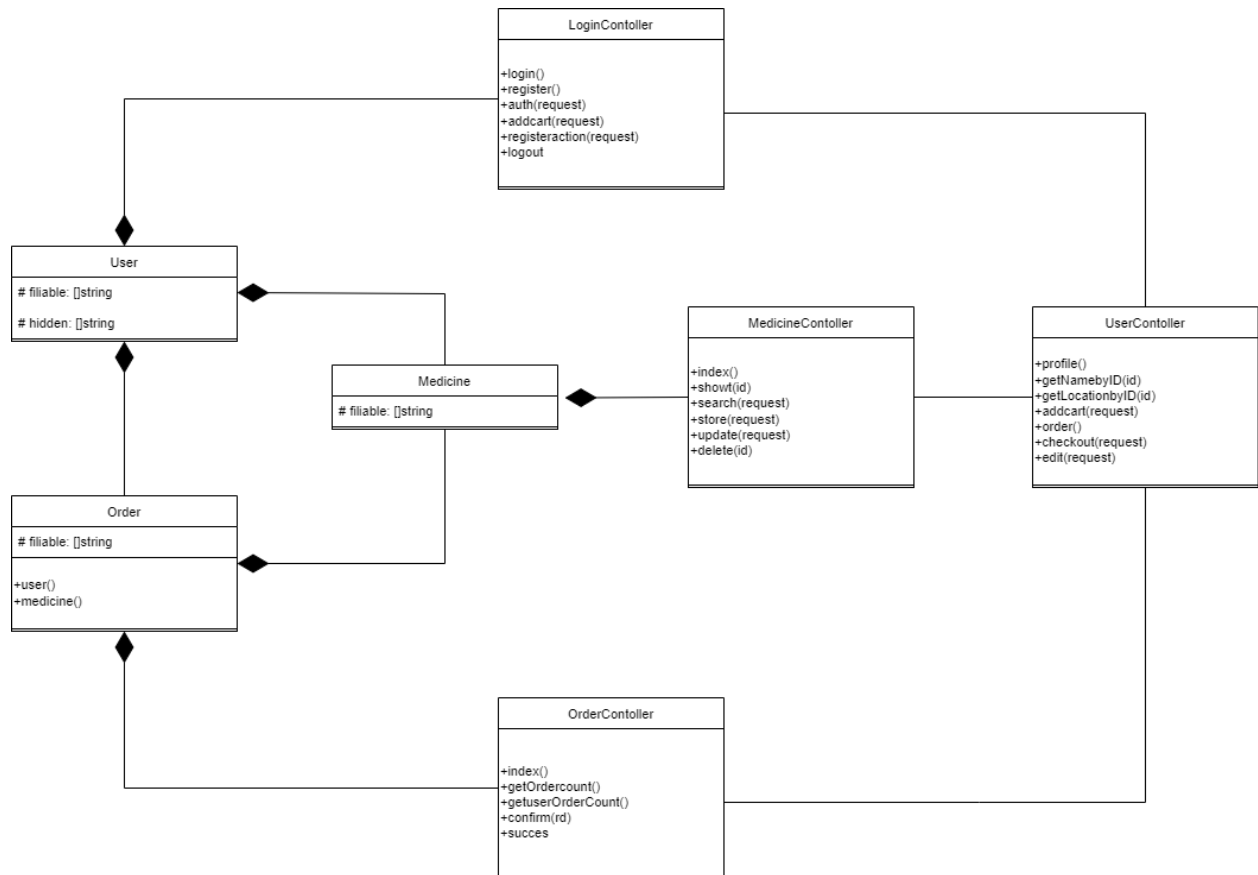
No	Nama Kelas Perancangan	Tipe Kelas
1	Medicine Controller	Controller
2	Medicine	model
3	Login Controller	Controller
4	User	model

*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller

3.1.8.2 Sequence Diagram



3.1.8.3 Diagram Kelas



3.1.9 Use Case Order Confirmation

Nama Use Case	Order Confirmation	
Deskripsi	Proses mengkonfirmasi pesanan yang masuk	
Aktor	Admin	
Pre-Kondisi	Pesanan yang masuk belum di konfirmasi	
Post-Kondisi	Pesanan telah dikonfirmasi	
Skenario Utama	Aktor	Sistem
	1. Admin membuka cart pesanan	
	2. Admin Melihat melalui aplikasi Mobile Banking untuk memastikan	

	apakah pembeli sudah membayar pesanannya	
	3. Admin mengkonfirmasi pesanan yang sudah dibayar	
		4. Sistem mengrespon konfirmasi admin dan akan mengubah status pesanan di web pembeli menjadi”sudah dikonfirmasi”
Skenario Eksepsional (Alternative Flow)	Aktor	Sistem

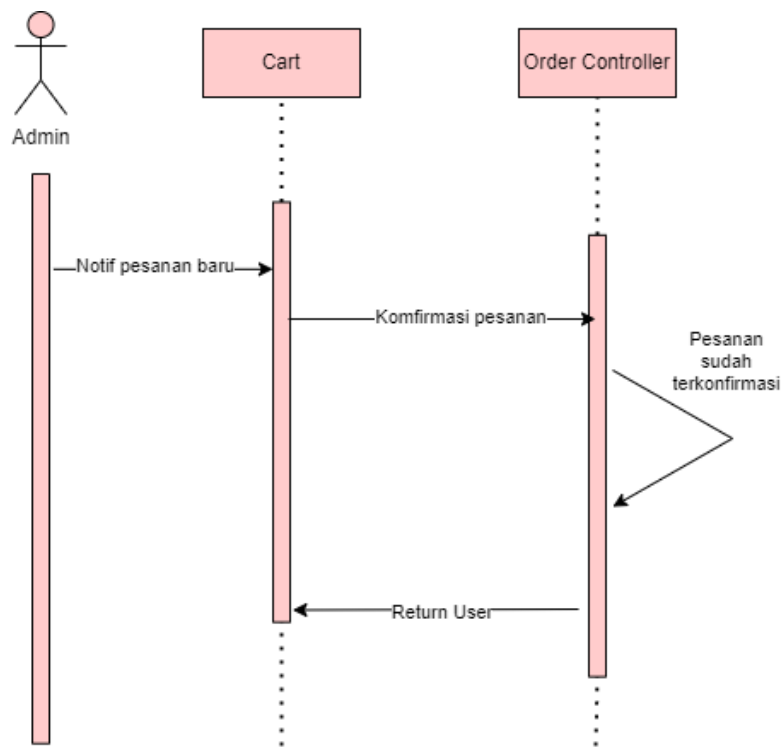
3.1.9.1 Identifikasi Kelas

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:

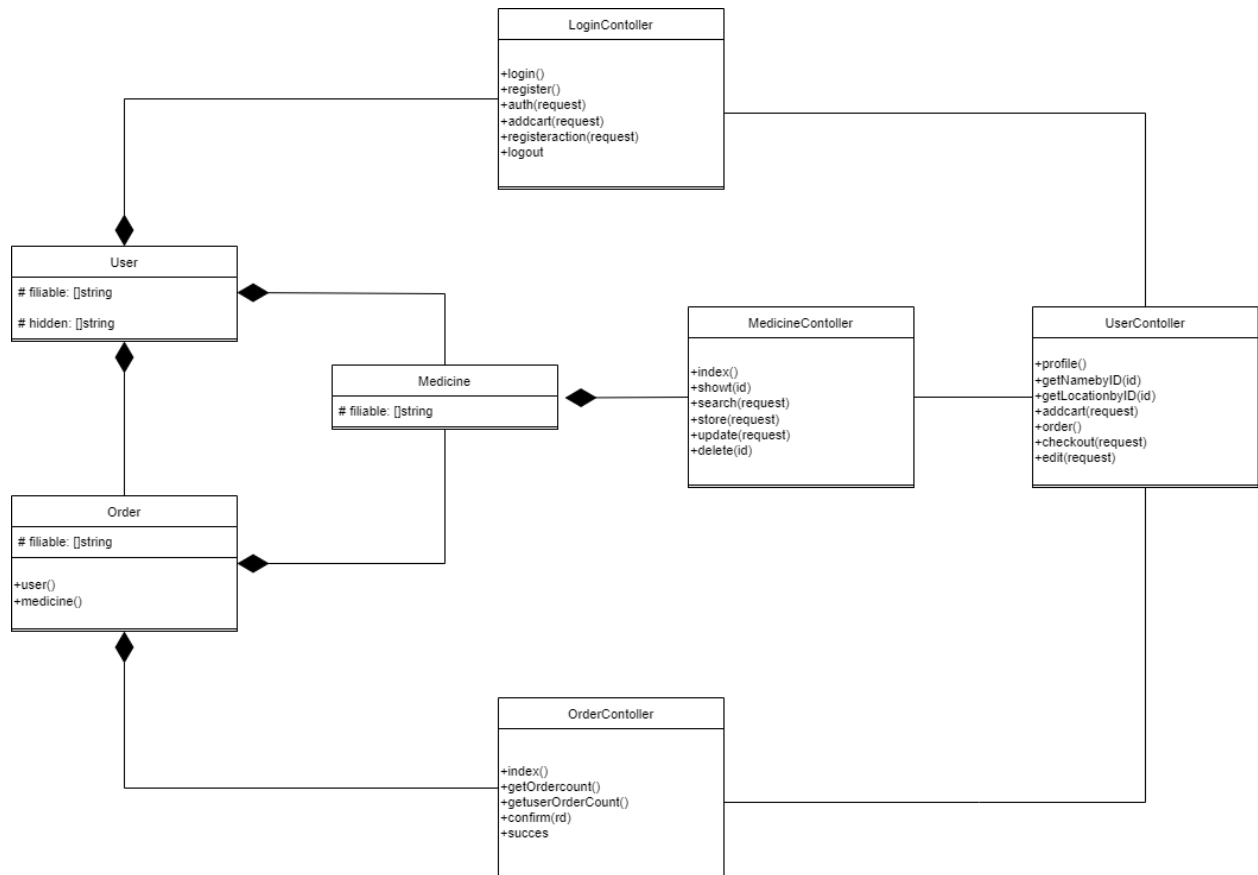
No	Nama Kelas Perancangan	Tipe Kelas
1	Order Controller	Controller
2	Order	model

**Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

3.1.9.2 Sequence Diagram



3.1.9.3 Diagram Kelas



3.1.10 Use Case LogOut

Nama Use Case	LogOut	
Deskripsi	Proses menghilangkan autentikasi akun	
Aktor	User	
Pre-Kondisi	User sedang login	
Post-Kondisi	User sudah keluar dari akunnya	
Skenario Utama	Aktor	Sistem
	1. User sedang berada di halaman home	
	2. User menekan menu logout	
		3. Sistem menerima respon dan

		mengeluarkan akun yang sedang login
Skenario Eksepsional (Alternative Flow)	Aktor	Sistem

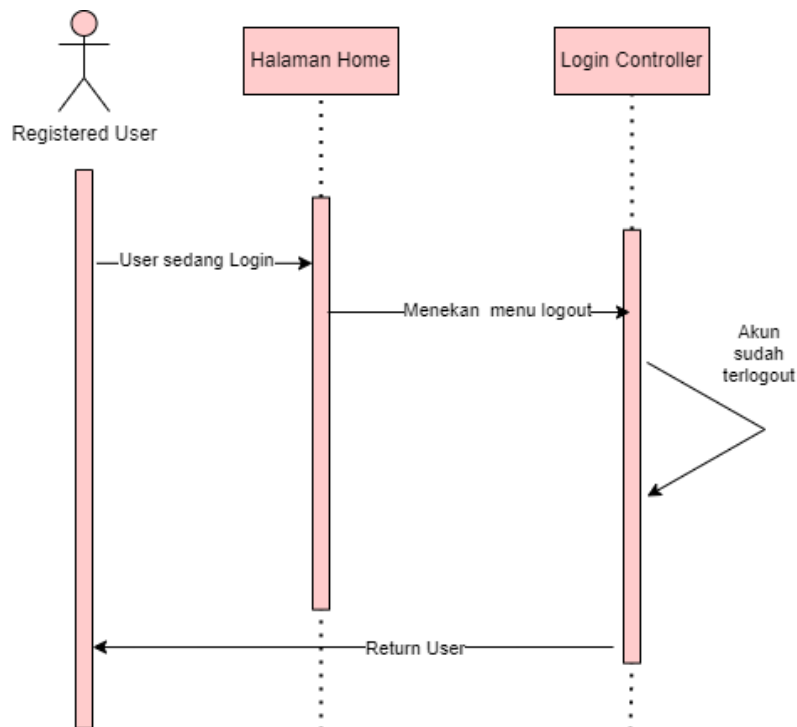
3.1.10.1 Identifikasi Kelas

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

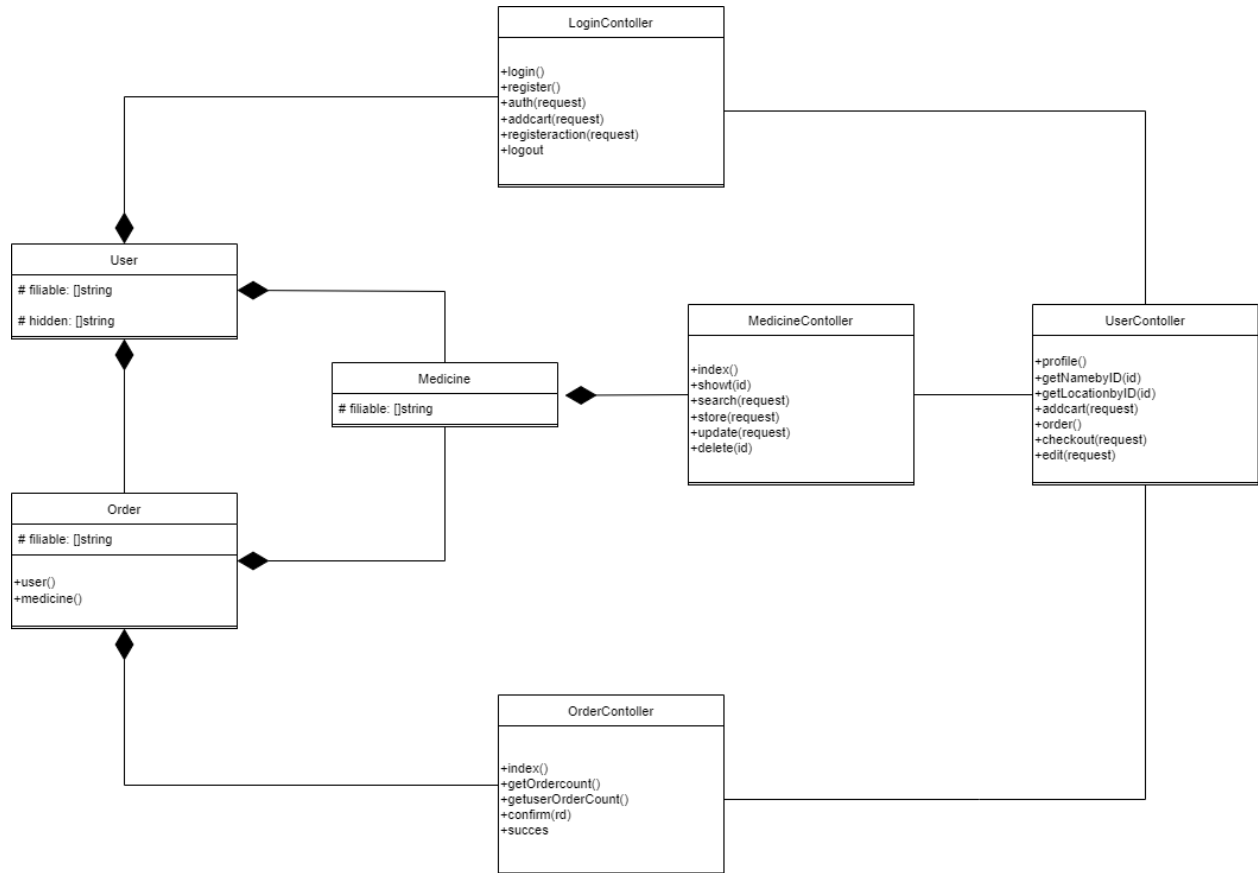
No	Nama Kelas Perancangan	Tipe Kelas
1	Login Controller	Controller
2	User	Model

*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller

3.1.10.2 Sequence Diagram



3.1.10.3 Diagram Kelas



3.2 Perancangan Detil Kelas

Bagian ini diisi dengan daftar seluruh kelas dalam tabel berikut:

No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1	User Controller	User Controller
2	Medicine Controller	Medicine Controller
3	Order Controller	Order Controller
4	LoginController	Login Controller
5	User	User
6	Medicine	Medicine
7	Order	Order

Untuk setiap kelas:

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

3.2.1 Kelas <UserController>

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

Nama Kelas : UserController

Nama Operasi	Visibility	Keterangan
--------------	------------	------------

	<i>(private, public)</i>	
<i>Profile</i>	<i>Public</i>	<i>Method ini menampilkan profile user</i>
<i>AddCart</i>	<i>Public</i>	<i>Method ini untuk memasukan pesanan ke dalam cart</i>
<i>Order</i>	<i>Public</i>	<i>Method ini untuk melakukan pemesanan</i>
<i>CheckOut</i>	<i>Public</i>	<i>Method ini untuk mengcheckout pesanan yang berada di cart</i>
<i>Edit</i>	<i>Public</i>	<i>Method ini untuk mengubah data profile</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>selected field</i>	<i>protected</i>	<i>Array of string</i>

3.2.2 Kelas <MedicineController>

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

Nama Kelas : Medicine Controller

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
<i>Search</i>	<i>Public</i>	<i>Method ini digunakan untuk mencari obat berdasarkan nama atau komposisi</i>
<i>Store</i>	<i>Public</i>	<i>Method ini digunakan untuk menambah obat</i>
<i>Update</i>	<i>Public</i>	<i>Method ini digunakan untuk mengubah data obat</i>
<i>Delete</i>	<i>Public</i>	<i>Method ini digunakan untuk menghapus obat</i>
<i>Show</i>	<i>Public</i>	<i>Method ini untuk menampilkan obat</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>selected field</i>	<i>protected</i>	<i>Array of string</i>
<i>Medicine</i>	<i>protected</i>	<i>Array of string</i>

3.2.3 Kelas <LoginController>

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

Nama Kelas : Login Controller

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
<i>Auth login</i>	<i>Public</i>	<i>Method ini digunakan untuk me-login-kan user kedalam</i>
<i>Registeraction</i>	<i>Public</i>	<i>Method ini digunakan untuk mendaftarkan user ke dalam sistem</i>

<i>Logout</i>	<i>Public</i>	<i>Method ini digunakan untuk me-logout-kan user</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>default values</i>	<i>protected</i>	<i>Associative array</i>

3.2.4 Kelas <OrderController>

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

Nama Kelas : *OrderController*

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
<i>GetOrderCount</i>	<i>Public</i>	<i>Method ini digunakan untuk memberi status</i>
<i>GetUserOrderCount</i>	<i>Public</i>	<i>Method ini digunakan untuk memberi status pesanan pada pembeli</i>
<i>Confirm</i>	<i>Public</i>	<i>Method ini digunakan untuk mengkonfirmasi pesanan</i>
<i>Succes</i>		<i>Method ini digunakan untuk memberi pesan sukses</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>selected field</i>	<i>protected</i>	<i>Array of string</i>

3.2.5 Kelas <User>

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

Nama Kelas : *User*

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>fillable</i>	<i>protected</i>	<i>Array of string</i>

3.2.6 Kelas <Medicine>

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

Nama Kelas : *Medicine*

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>fillable</i>	<i>protected</i>	<i>Array of string</i>

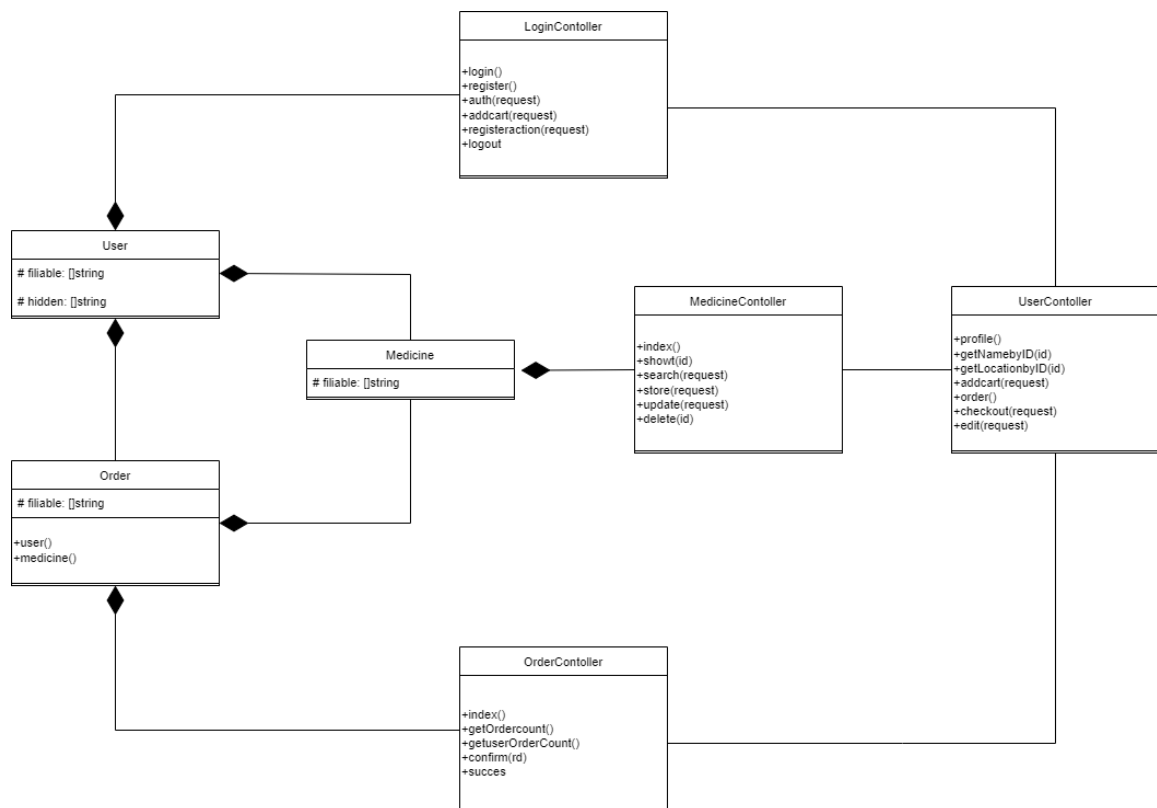
3.2.7 Kelas <Order>

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

Nama Kelas : Order

Nama Operasi	Visibility (private, public)	Keterangan
Nama Atribut	Visibility (private, public)	Tipe
fillable	protected	Array of string

3.3 Diagram Kelas Keseluruhan



3.4 Algoritma/Query

3.4.1 LoginController

Nama Operasi : *registeraction*

Algoritma : *(Algo-xxx)*

```

public function registeraction(Request $request)
{
    if (!$request->validate([
        'username' => 'required',
    ]))
    {
        return back()->withErrors($request->errors());
    }

    $user = new User([
        'username' => $request->username,
        'password' => $request->password,
    ]);

    $user->save();

    return back();
}
  
```

```

        'password' => 'required|min:6',
        'name' => 'required',
        'gender' => 'required',
        'location' => 'required',
    ])) {
        return redirect('/register')->withErrors($request->validator);
    }
    $data = [
        'username' => $request->input('username'),
        'password' => bcrypt($request->input('password')),
        'name' => $request->input('name'),
        'gender' => $request->input('gender'),
        'location' => $request->input('location'),
        'role' => 'customer',
    ];
    $user = User::create($data);
    return redirect('/login')->with('success', "Hooray, anda berhasil
register!");
}

```

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

Query :

No Query	Query	Keterangan
Q-001	INSERT INTO "users" ("username", "password", "name", "gender", "location")	Membuat data baru (user) ke tabel users
Q-002	INSERT INTO "personal_access_tokens" ("name", "token", "abilities", "tokenable_id", "tokenable_type", "updated_at", "created_at")	Membuat data baru (access token dari user) ke tabel personal access tokens

3.4.2 MedicineController

Nama Operasi : Store

Algoritma : (Algo-xxx)

```

public function store(Request $request)
{
    // get image
    $image = $request->file('image');
    // get image name
    $imageName = $image->getClientOriginalName();
    // save image
    $image->move(public_path('images/medicines'), $imageName);
    // set data from all request
}

```

```

$data = $request->all();
// set image name to data
$data['image'] = $imageName;
// create medicine
Medicine::create($data);
// redirect to medicine page
return redirect('/medicine')->with('success', 'Medicine stored
successfully');
}

```

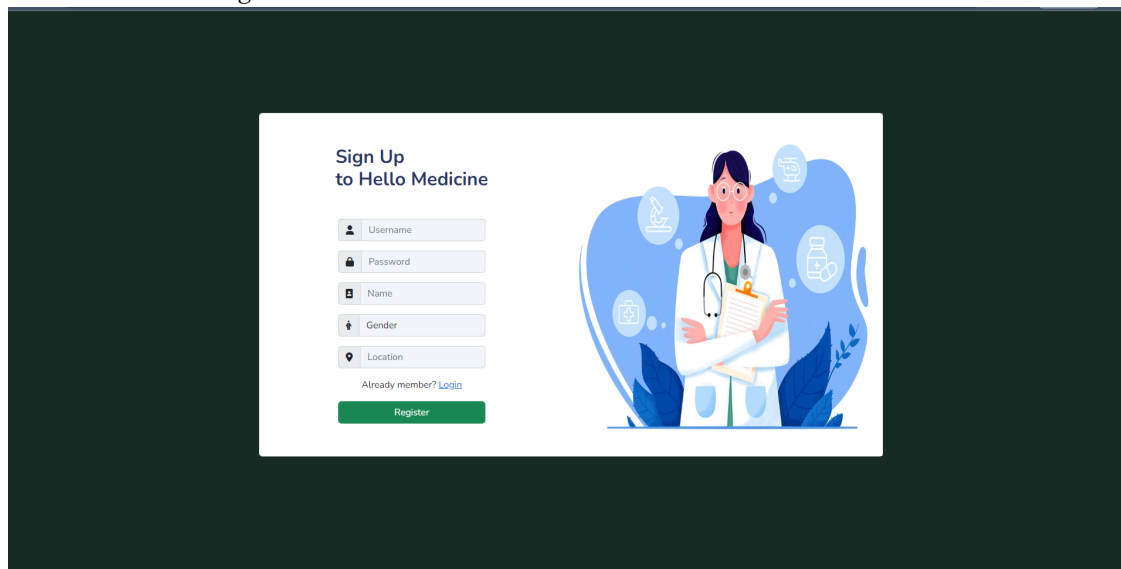
{Jika mengacu query tertentu, lengkapi tabel query di bawah}

Query

No Query	Query	Keterangan
Q-001	INSERT INTO "medicine" ("name", "composition", "quantity", "price", "image")	Membuat data baru (medicine) ke tabel medicine

3.5 Perancangan Antarmuka

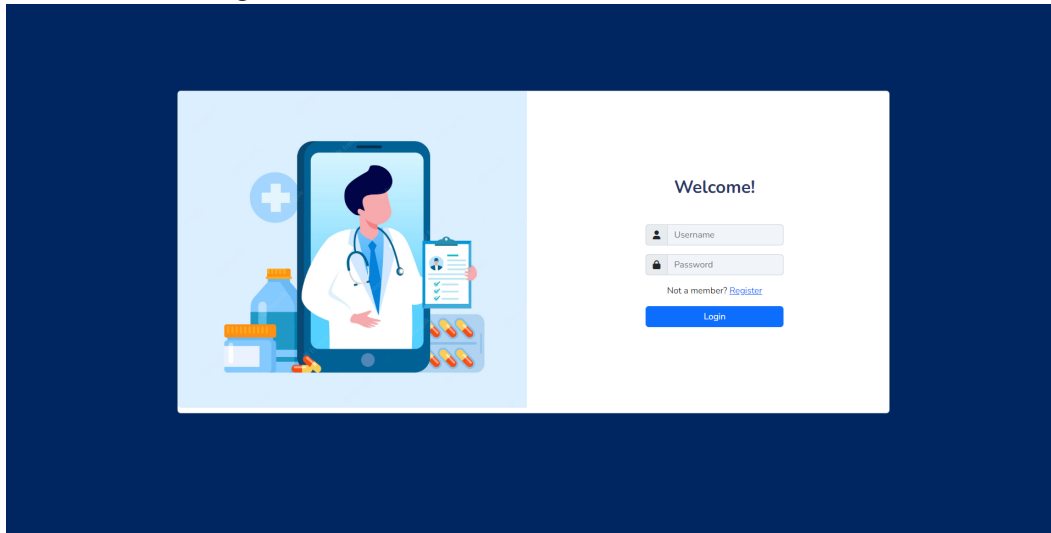
Antarmuka : Register



Id Objek	Jenis	Nama	Keterangan
Username	TextInput	username	Menerima input berupa text dari pengguna
password	TextInput	password	Menerima input berupa text dari pengguna
name	TextInput	name	Menerima input berupa text dari pengguna
gender	TextInput	gender	Menerima input berupa text dari pengguna
location	TextInput	location	Menerima input berupa text dari pengguna
register	button	register	Jika diklik akan mendaftarkan akun

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

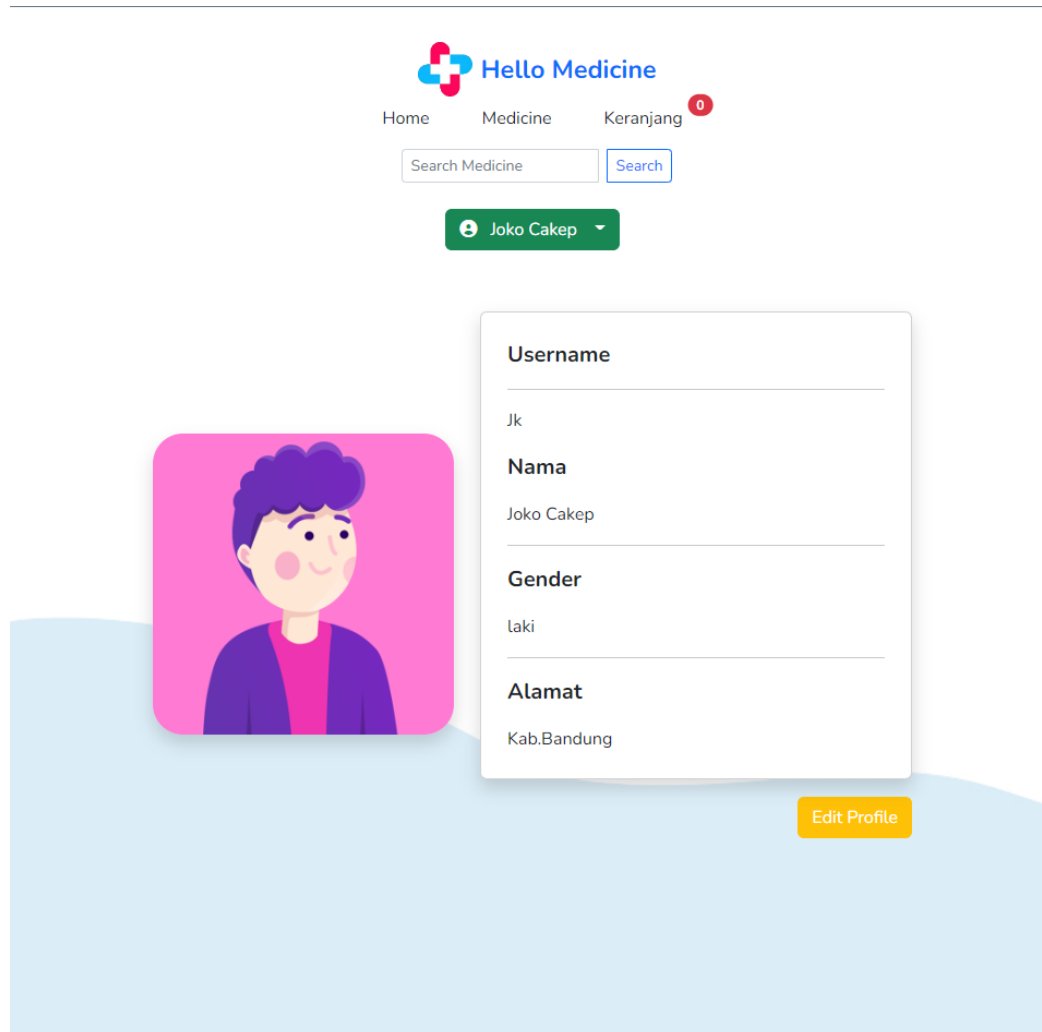
Antarmuka : Login



Id_Objek	Jenis	Nama	Keterangan
username	TextInput	username	Menerima input berupa text dari pengguna
password	TextInput	password	Menerima input berupa text dari pengguna
login	Button	login	Jika diklik akan masuk menggunakan akun

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

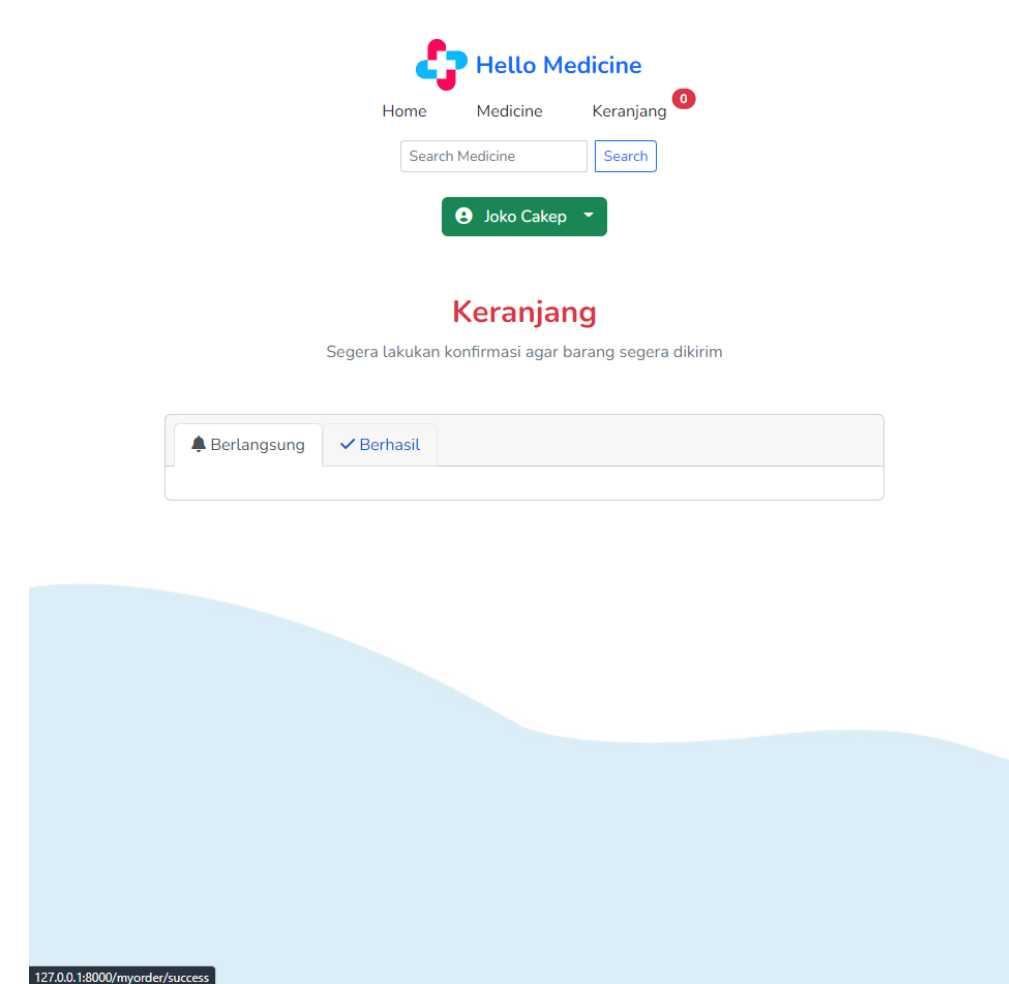
Antarmuka : Profile



Id Objek	Jenis	Nama	Keterangan
homebtn	button	Home	Jika diklik akan ke halaman home
medicinebtn	button	Medicine	Jika diklik akan ke halaman medicine
keranjangbtn	button	Keranjang	Jika diklik akan ke halaman keranjang
searchBar	Text input	SearchMedicine	Menerima inputan dari pengguna
Search btn	button	Search	Jika diklik akan mencari obat yang sudah diketik
Edit btn	button	Edit Profile	Jika diklik akan ke halaman edit profile
namebtn	button	Joko Cakep	Jika diklik akan menampilkan menu profile, keranjang, dan logout

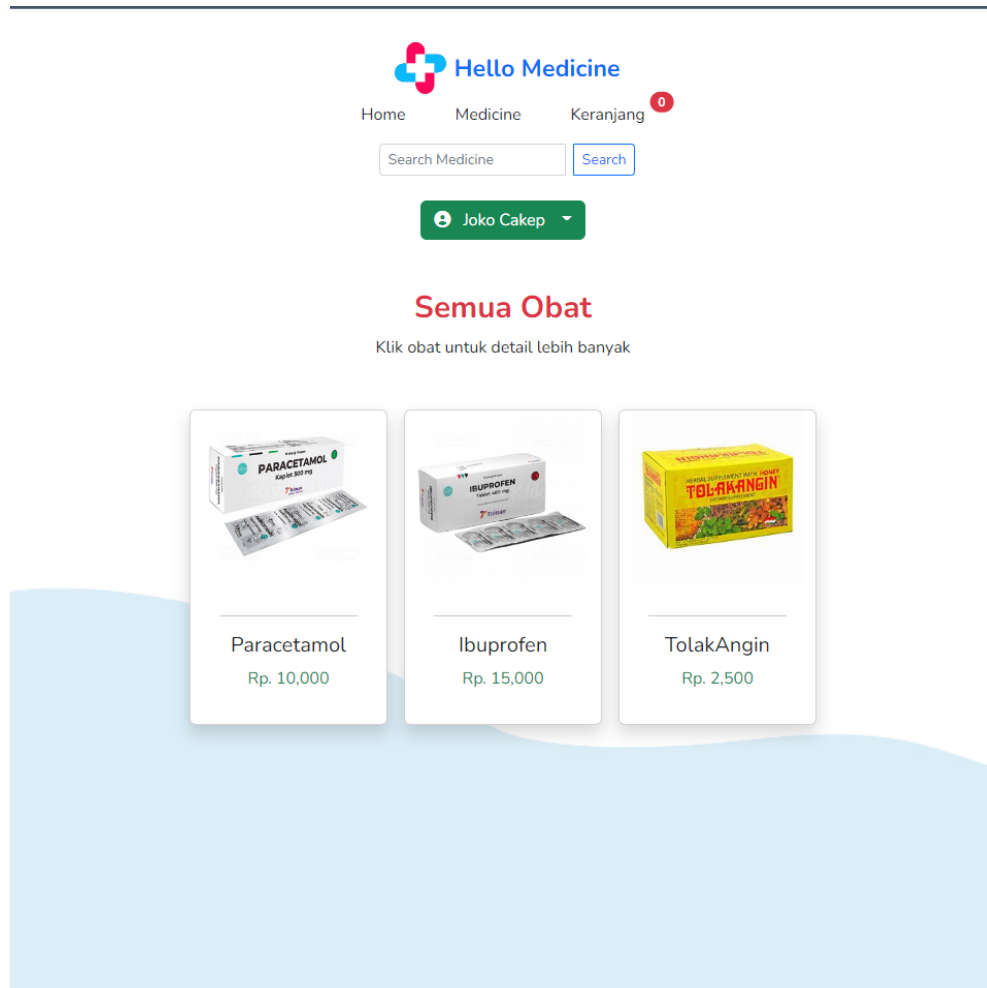
Jika objek dikaitkan ke File lain (misalnya file gambar, file teks), berikan nama file terkait dan deskripsi ringkas dalam kolom keterangan

Antarmuka : Halaman Keranjang



Id Objek	Jenis	Nama	Keterangan
homebtn	button	Home	Jika diklik akan ke halaman home
medicinebtn	button	Medicine	Jika diklik akan ke halaman medicine
keranjangbtn	button	Keranjang	Jika diklik akan ke halaman keranjang
searchBar	Text input	SearchMedicine	Menerima inputan dari pengguna
Search btn	button	Search	Jika diklik akan mencari obat yang sudah diketik
namebtn	button	Joko Cakep	Jika diklik akan menampilkan menu profile, keranjang, dan logout
berlangsungbtn	button	Berlangsung	Jika diklik akan menampilkan pesanan yang sedang berlangsung
berhasilbtn	button	Berhasil	Jika diklik akan menampilkan pesanan yang sudah berhasil

Antarmuka : Halaman medicine

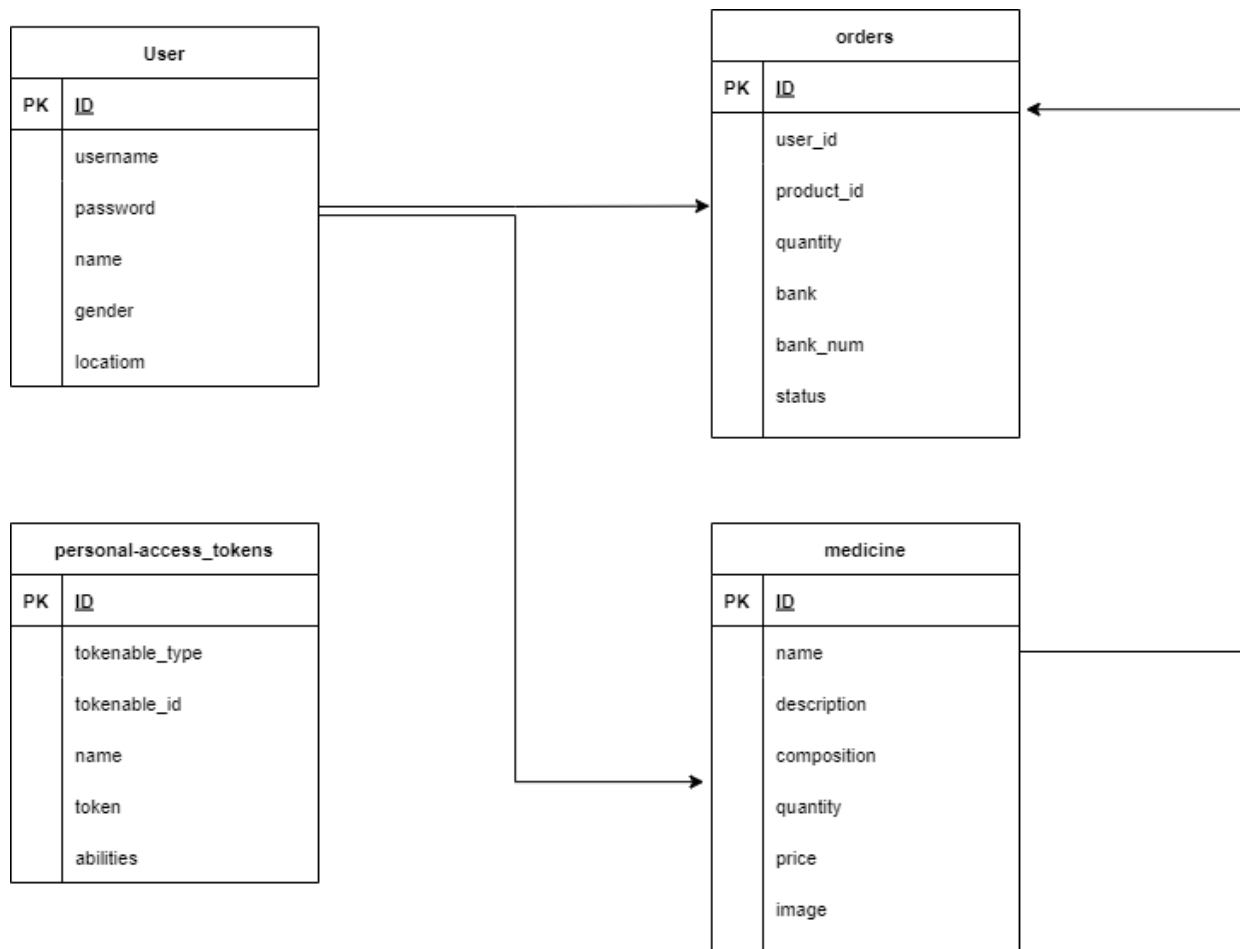


Id Objek	Jenis	Nama	Keterangan
homebtn	button	Home	Jika diklik akan ke halaman home
medicinebtn	button	Medicine	Jika diklik akan ke halaman medicine
keranjangbtn	button	Keranjang	Jika diklik akan ke halaman keranjang
searchBar	Text input	SearchMedicine	Menerima inputan dari pengguna
Search btn	button	Search	Jika diklik akan mencari obat yang sudah diketik
namebtn	button	Joko Cakep	Jika diklik akan menampilkan menu profile, keranjang, dan logout
cardObat	card	namaObat	Jika diklik akan menampilkan informasi tentang obat itu dan dapat melakukan pemesanan

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

3.6 Perancangan Representasi Persistensi Kelas

Bagian ini diisi dengan rancangan skema basis data dan traceability-nya terhadap kelas entity. (PEMBUATAN SKEMA RELASI)



4 Matriks Kerunutan

Mapping use case dengan kelas-kelas terkait

Kelas	Use Case Terkait
User, LoginController	Register
User, LoginController	Login
User, LoginController	Logout
User, UserController	Edit Profile
User(admin), MedicineController	Menambah obat baru
User(admin), MedicineController	Menghapus obat
User(admin), MedicineController	Mengupdate obat baru

User(admin), OrderController	Mengkonfirmasi pesanan
User, UserController	Melakukan checkout
User, UserController	Melakukan pemesanan