

**DPPL-003**

## **DESKRIPSI PERANCANGAN PERANGKAT LUNAK**

**CareMe!**

untuk:

Ikatan Psikologi Klinis Indonesia

Dipersiapkan oleh:

Reyhan Septri Asta	(1301190308)
Sabilla Suci Amini	(1301194347)
Azka Zainur Azifa	(1301194255)
Maulana Nur	(1301190402)
Elita Aurora Az Zahra	(1301194127)

Program Studi Informatika

Fakultas Informatika

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

Telkom University	Prodi S1- Informatika Universitas Telkom	Nomor Dokumen	Halaman
		DPPL-003	45
		Revisi	01

## 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

<b>1. Pendahuluan</b>	<b>5</b>
Tujuan Penulisan Dokumen	6
Lingkup Masalah	6
Definisi dan Istilah	7
Referensi	8
Sistematika Pembahasan	8
 <b>Deskripsi Perancangan Global</b>	 <b>8</b>
Rancangan Lingkungan Implementasi	8
Deskripsi Arsitektural	8
Deskripsi Komponen	9
 <b>Perancangan Rinci</b>	 <b>10</b>
Realisasi Use Case	10
Use Case Login Register	10
Identifikasi Kelas	10
Sequence Diagram	10
Diagram Kelas	10
Use Case Memberikan Konsultasi	11
Identifikasi Kelas	11
Sequence Diagram	12
Diagram Kelas	12
Use Case Melihat Informasi User	13
Identifikasi Kelas	13
Sequence Diagram	13
Diagram Kelas	14
Use Case Mengupload Medical Records	15
Identifikasi Kelas	15
Sequence Diagram	15
Diagram Kelas	15
Use Case Manage System	16
Identifikasi Kelas	16
Sequence Diagram	16
Diagram Kelas	17
Use Case Mengisi Biodata Lengkap User	18
Identifikasi Kelas	18
Sequence Diagram	18
Diagram Kelas	19
Use Case Berkonsultasi	19

<b>Prodi S1 Informatika Tel-U</b>	<b>DPPL-003</b>	<b>Halaman 4 dari 42</b>
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		

Identifikasi Kelas	20
Sequence Diagram	20
Diagram Kelas	20
Use Case Melakukan Komplain	21
Identifikasi Kelas	21
Sequence Diagram	21
Diagram Kelas	22
Use Case Timeline	23
Identifikasi Kelas	23
Sequence Diagram	23
Diagram Kelas	23
Use Case Melakukan Print Medical Record	24
Identifikasi Kelas	24
Sequence Diagram	24
Diagram Kelas	25
Perancangan Detil Kelas	26
Kelas User	26
Kelas Admin	26
Kelas Psikolog	27
Kelas Komplain	27
Kelas Medical Record	28
Kelas Timeline	28
Kelas Konsultasi	28
Kelas Chat	29
Diagram Kelas Keseluruhan	29
Algoritma/Query	30
Algoritma/Query Login	30
Algoritma/Query Register	31
Algoritma/Query Psikolog	31
Algoritma/Query Komplain	32
Algoritma/Query Konsultasi	33
Algoritma/Query Chat	33
Algoritma/Query Medical Record	34
Algoritma/Query Timeline	35
Perancangan Antarmuka	36
Perancangan Representasi Persistensi Kelas	41
<b>Matriks Kerunutan</b>	<b>42</b>

Setelah Daftar Isi Boleh ada Daftar Tabel dan Daftar Gambar

Prodi S1 Informatika Tel-U	DPPL-003	Halaman 5 dari 42
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		

# 1. Pendahuluan

## 1.1 Tujuan Penulisan Dokumen

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) adalah dokumen yang berisi deskripsi dari perangkat lunak yang akan dikembangkan. Dokumen ini ditulis untuk menjabarkan deskripsi kebutuhan perangkat lunak dari aplikasi *CareMe!*

Kesehatan jiwa merupakan hal yang sangat penting bagi segenap manusia untuk menjalani kehidupannya. Seiring dengan berkembangnya zaman, kebutuhan dan taraf hidup dikalangan masyarakat pun semakin meningkat. Dilihat dari masalah tersebut, ada orang yang dapat menyikapi masalah yang terjadi dalam hidupnya dengan cara yang baik adapun yang sebaliknya. Tak jarang seseorang dapat melakukan hal yang membahayakan dirinya untuk menutupi masalah tersebut.

Selain itu, di masa pandemi COVID-19 ini semakin sulit pendapatan ekonomi masyarakat dan sulitnya mencari pekerjaan, karena masyarakat diimbau untuk dirumah saja dan banyak yang terkena PHK. Hal ini membuat menambahnya beban pikiran, yang dapat menyebabkan terganggunya kesehatan mental bila tidak ditangani dengan benar.

Maka terdapat beberapa faktor yang dapat menyebabkan terganggunya kesehatan mental seperti ambisi akan taraf hidup, situasi pandemi, emosi yang tidak terkendali, tekanan dari lingkungan keluarga, dan teman. Gangguan mental umumnya dialami oleh remaja dan orang dewasa, tetapi tak jarang anak kecil pun mengalaminya. Anak - anak dapat berpotensi mengalami gangguan mental berupa trauma yang disebabkan oleh kekerasan orang tua, *bullying*, ataupun tekanan dari lingkungan keluarga. Hal ini dapat mengganggu karakter anak tersebut saat menginjak masa remaja dan dewasa.

Dikutip dari *Liputan6.com* menurut data Ikatan Psikologi Klinis (IPK) Indonesia, sejak pandemi pada Maret hingga Agustus 2020 tercatat ada sekitar 14.619 kasus masalah psikologis. Dari data tersebut dapat dikatakan bahwa tingkat gangguan kejiwaan dan depresi di Indonesia cukup tinggi. Maka, dibutuhkan beberapa solusi untuk mengurangi masalah tersebut. Oleh karena itu, dengan adanya aplikasi kesehatan mental *CareMe!* diharapkan dapat menjadi salah satu solusi tersebut

## 1.2 Lingkup Masalah

Tuliskan dengan ringkas nama aplikasi dan deskripsinya. Maksimal 1 paragraf. Sama dengan yang ditulis di SKPL.

Aplikasi *CareMe!* merupakan aplikasi yang bergerak dibidang kesehatan mental. Sesuai dengan tujuannya aplikasi ini dibuat untuk mengurangi tingkat gangguan kesehatan mental yang terjadi di Indonesia tepatnya pada saat masa pandemi dimana menurut Ikatan Psikologi(2020) terdapat 14.169 kasus masalah psikologis yang terjadi di Indonesia.

Aplikasi ini memberikan beberapa layanan yang dapat membantu masalah psikologis seperti tes kesehatan mental, seminar motivasi, emergency call, konsultasi, dan lain-lain. Selain itu, masih banyak lagi layanan yang diberikan oleh aplikasi *CareMe!* Salah satu keunggulan dari aplikasi ini adalah menyediakan layanan konsultasi yang langsung ditangani oleh psikolog bersertifikat. Hal ini dapat sangat memudahkan masyarakat untuk melakukan konsultasi.

Sebenarnya gangguan mental dapat menimpas iapa saja dari mulai anak-anak, remaja, hingga dewasa. Namun, gangguan mental lebih berpotensi menyerang remaja dan dewasa. Maka dari itu, aplikasi ini lebih diperuntukan kepada remaja SMP dan SMA juga kalangan orang dewasa. Anak - anak pun dapat menggunakan aplikasi ini dengan syarat dibimbing oleh

Prodi S1 Informatika Tel-U	DPPL-003	Halaman 6 dari 42
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		

orang tua. Selain itu, kami melakukan survei secara *questionnaire* kepada 30 responden yang dimana sebanyak 96.6% responden menyatakan bahwa dengan adanya aplikasi kesehatan mental CareMe! dapat membantu permasalahan mental yang terjadi.

### 1.3 Definisi dan Istilah

- *Bullying* : Penggunaan kekerasan, ancaman, atau paksaan untuk menyalahgunakan atau mengintimidasi orang lain.
- *Emergency Call* : Nomor telepon yang dapat dihubungi dalam keadaan darurat.
- *Questionnaire* : Teknik survei berupa beberapa pertanyaan.
- *User* : Pengguna aplikasi.
- *Login* : Menu yang digunakan untuk masuk kedalam aplikasi
- *Review* : Sebuah ringkasan, tinjauan dari beberapa sumber
- *Emergency* : Keadaan darurat
- *Rule* : Sebuah peraturan
- *Log out* : Istilah dalam hal keamanan komputer, yakni proses keluar guna memutus akses ke sistem komputer maupun layanan tertentu.
- *Admin* : Pekerjaan dalam suatu instansi yang bertugas dalam hal administratif
- *Server* : Sistem komputer yang menyediakan sebuah layanan tertentu dalam suatu jaringan komputer
- *Scripting Language* : Bahasa pemrograman penerjemah dan kompilasi kode dan dapat dijalankan secara dinamis.
- *DBMS* : Suatu perangkat lunak yang digunakan untuk mengelola basis data dan menjalankan operasi yang digunakan
- *Mobile* : Suatu aplikasi yang menggunakan telepon seluler atau handphone
- *Hotline* : Saluran siaga
- *Interface* : antarmuka
- *Library* : Perpustakaan
- *Database* : Kumpulan data yang terorganisir
- *Password* : kata sandi
- *Dashboard* : sebuah antarmuka yang menghubungkan pengguna dengan sistem
- *Top up* : isi ulang atau penambahan saldo
- *Menu bar* : daftar menu
- *Edit* : mengubah
- *Input box* : kotak yang dapat dimasukkan kalimat atau kata
- *Video call* : metode komunikasi seperti telepon dengan adanya video
- *Chat* : obrolan
- *Medical record* : rekam medis atau berkas yang berisikan catatan kesehatan seseorang.
- *Submit* : Mengajukan
- *Rating* : penilaian terhadap sesuatu
- *Event* : Acara
- *Refresh* : menyegarkan
- *Check Up* : Pemeriksaan kesehatan
- *History* : Riwayat
- *Functionality* : Fungsionalitas

## 1.4 Referensi

Dokumen spesifikasi kebutuhan perangkat lunak ini memiliki beberapa referensi dalam penyusunannya, yaitu sebagai berikut :

- Survei secara *questionnaire* kepada 30 responden melalui google form.
- Aedien, Aep. (2011). *PERSAMAAN DAN PERBEDAAN ANTARA PSIKOLOGI, SOSIOLOGI, DAN PSIKOLOGI SOSIAL* di <https://aeppsiologi.blogspot.com/2011/10/persamaan-dan-perbedaan-antara.html?m=1> .(akses 17 Desember 2020).
- Nasihudin, Ade. (2020). *IKATAN PSIKOLOG : 14.619 MASALAH PSIKOLOGIS TERCATAT SELAMA PANDEMI COVID-19* di <https://www.liputan6.com/health/read/4407289/ikatan-psikolog-14619-masalah-psikologis-tercatat-selama-pandemi-covid-19> . (akses 17 Desember 2020).

## 1.5 Sistematika Pembahasan

Bagian ini merupakan deskripsi umum dokumen. Tuliskan sistematika pembahasan dokumen DPPL ini. contoh: misalkan dalam bab1 berbicara tentang apa, bab 2 berbicara tentang apa dll

- **Bab 1** : Merupakan pendahuluan yang membahas tentang tujuan penulisan dokumen, istilah dan definisi, juga referensi yang digunakan dalam pembangunan perangkat lunak.
- **Bab 2** : Merupakan deskripsi perancangan global yang membahas tentang rancangan lingkungan implementasi, deskripsi arsitektural, dan deskripsi komponen.
- **Bab 3** : Merupakan perancangan rinci yang membahas tentang realisasi use case, class diagram, dan sequence diagram.
- **Bab 4** : Merupakan matriks keruntutan yang membahas tentang mapping use case dengan kelas terkait..

## 2 Deskripsi Perancangan Global

### 2.1 Rancangan Lingkungan Implementasi

Sebutkan Operating system, DBMS, development tools, filing system, bahasa pemrograman yang dipakai Perangkat pada sisi server yang dibutuhkan :

- Laptop (RAM min 4GB, HDD 1TB/SSD 512GB )
- Sistem Operasi : Windows 8/8.1/10
- *Scripting Language* : Dart
- DBMS : MySQL

Perangkat pada sisi client atau user yang dibutuhkan:

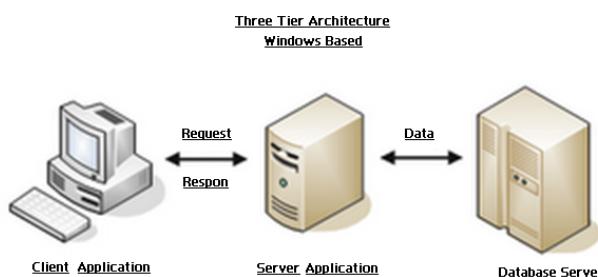
- Smartphone

### 2.2 Deskripsi Arsitektural

Berikan penjelasan singkat tentang arsitektur /L yang akan dibangun. Gambarkan dalam bentuk diagram komponen.

Prodi S1 Informatika Tel-U	DPPL-003	Halaman 8 dari 42
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		

Arsitektur perangkat lunak yang akan kami gunakan yaitu 3-tier architecture, dimana terdapat bagian tier yang dapat mendukung pembuatan aplikasi kami. Tier 1 berupa Client Tier dimana aplikasi kami memiliki 2 user yaitu pasien dan psikolog yang berperan sebagai client pada aplikasi kami. Tier 2 berupa Server Tier dimana pada tier ini terdapat admin yang dapat mengelola server dari aplikasi kami. Tier 3 berupa Database Server dimana pada tier ini terdapat penyimpanan data dari aplikasi kami. Tentunya arsitektur ini membutuhkan middleware(server tier) untuk membantu pembuatan aplikasi.



Sumber :

<http://sandyperdanabermatadewa.blogspot.com/2012/09/aplikasi-three-tier.html>

### 2.3 Deskripsi Komponen

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

No	Nama Komponen	Keterangan
1.	Login dan Registrasi	User, Psikolog, dan Admin melakukan login dan registrasi untuk masuk ke dalam aplikasi
2.	Biodata User	Aplikasi menampilkan Biodata User dan user dapat mengedit data yang ada didalamnya.
3.	Konsultasi	Fitur ini menyediakan layanan konsultasi berupa chat untuk para user yang akan langsung ditangani oleh seorang psikolog.
4.	Komplain	Fitur ini menyediakan layanan komplain yang akan menampilkan halaman komplain dari user untuk psikolog, aplikasi, atau layanan lainnya.
5.	Timeline	Fitur ini menyediakan layanan yang menampilkan informasi tentang kesehatan mental.

### 3 Perancangan Rinci

#### 3.1 Realisasi Use Case

##### 3.1.1 Use Case Login Register

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

###### 3.1.1.1 Identifikasi Kelas

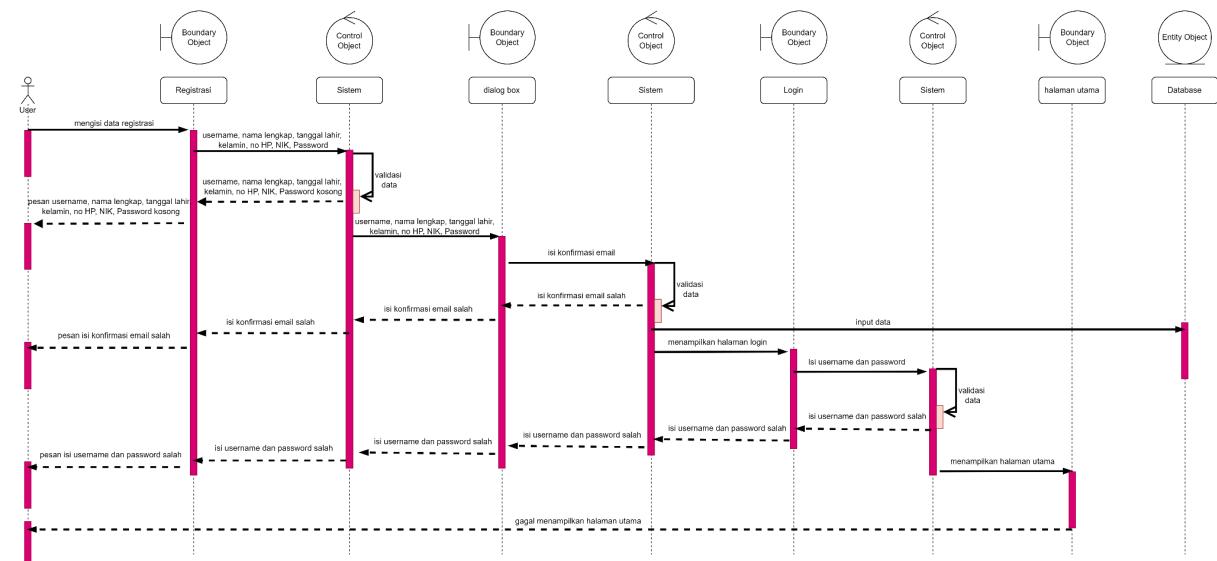
Identifikasi kelas yang terkait dengan use case tersebut. Kelas di tahap perancangan dapat berbeda dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:

No	Nama Kelas Perancangan	Tipe Kelas
1.	Login	Boundary
2.	Sistem	Controller
3.	Register	Boundary
4.	Dialog box	Boundary
5.	Halaman Utama	Boundary
6.	Database	Entity

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

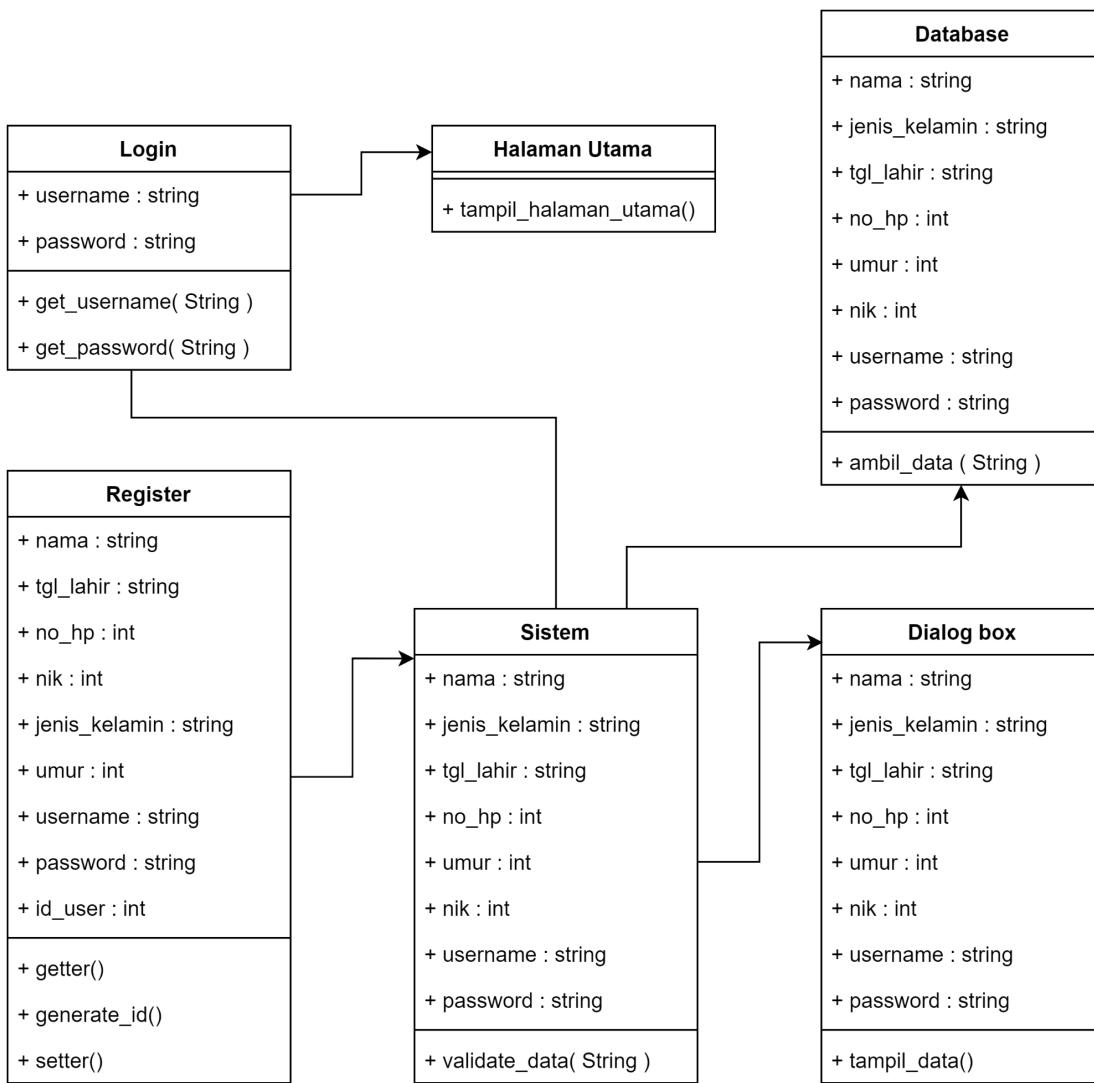
###### 3.1.1.2 Sequence Diagram

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



###### 3.1.1.3 Diagram Kelas

Buatlah diagram kelas untuk use case tersebut. buat class diagram **BUKAN KESELURUHAN, tapi PER USE CASE**



### 3.1.2 Use Case Memberikan Konsultasi

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

#### 3.1.2.1 Identifikasi Kelas

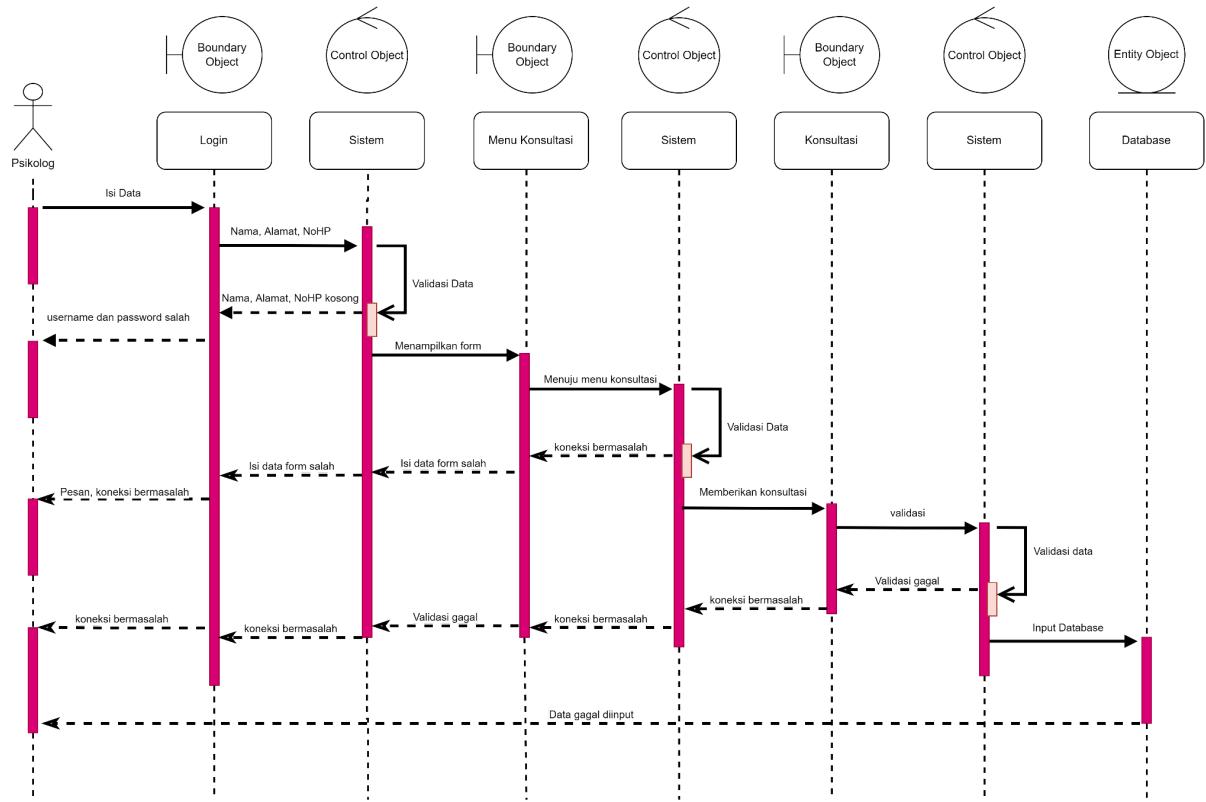
*Identifikasi kelas yang terkait dengan use case tersebut. Kelas di tahap perancangan dapat berbeda dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

No	Nama Kelas Perancangan	Tipe Kelas
1.	<i>Login</i>	
2.	<i>Sistem</i>	
3.	<i>Menu Konsultasi</i>	
4.	<i>Konsultasi</i>	
5.	<i>Database</i>	

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

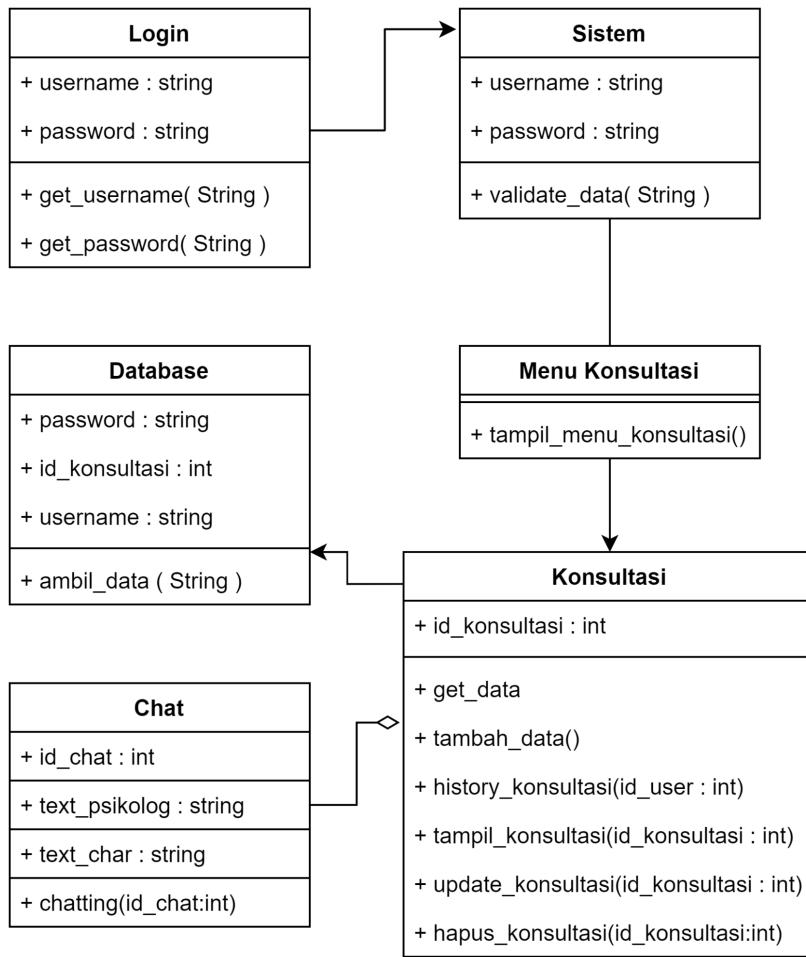
### 3.1.2.2 Sequence Diagram

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



### 3.1.2.3 Diagram Kelas

Buatlah diagram kelas untuk use case tersebut. buat class diagram **BUKAN KESELURUHAN**, tapi **PER USE CASE**



### 3.1.3 Use Case Melihat Informasi User

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

#### 3.1.3.1 Identifikasi Kelas

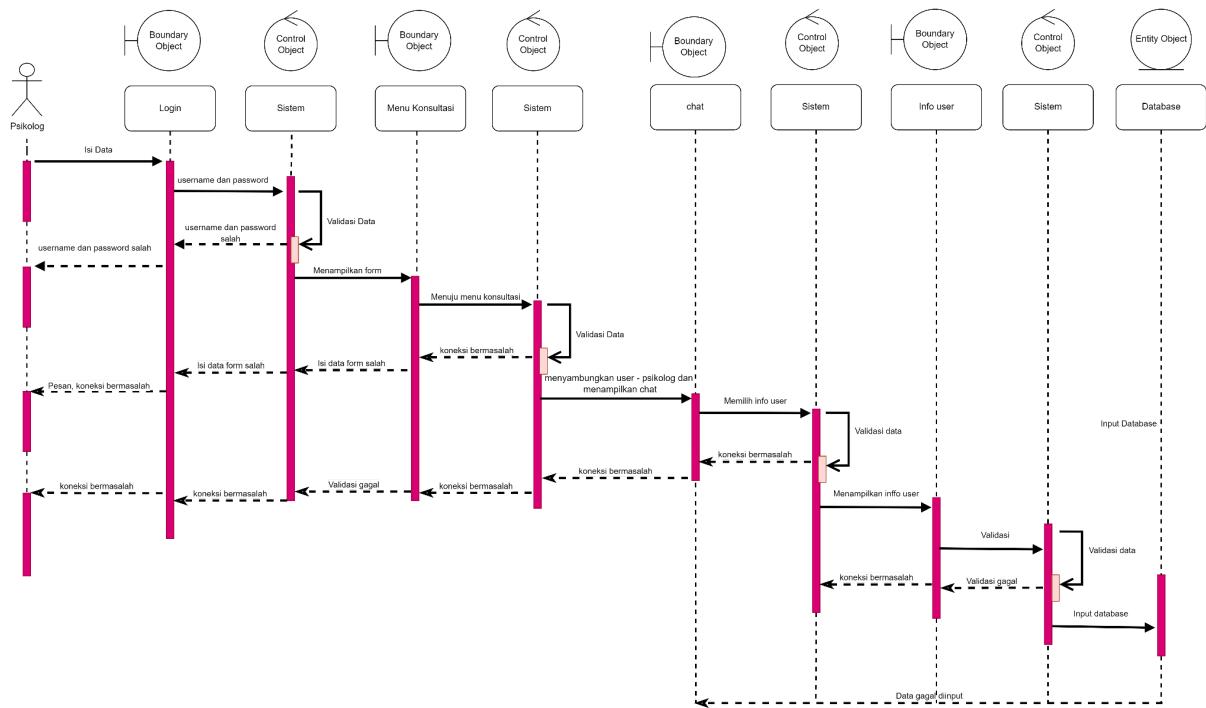
*Identifikasi kelas yang terkait dengan use case tersebut. Kelas di tahap perancangan dapat berbeda dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

No	Nama Kelas Perancangan	Tipe Kelas
1	Login	Boundary
2	Sistem	Controller
3	Menu Konsultasi	Boundary
4	Konsultasi	Boundary
5	Info User	Boundary
6	Database	Entity

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

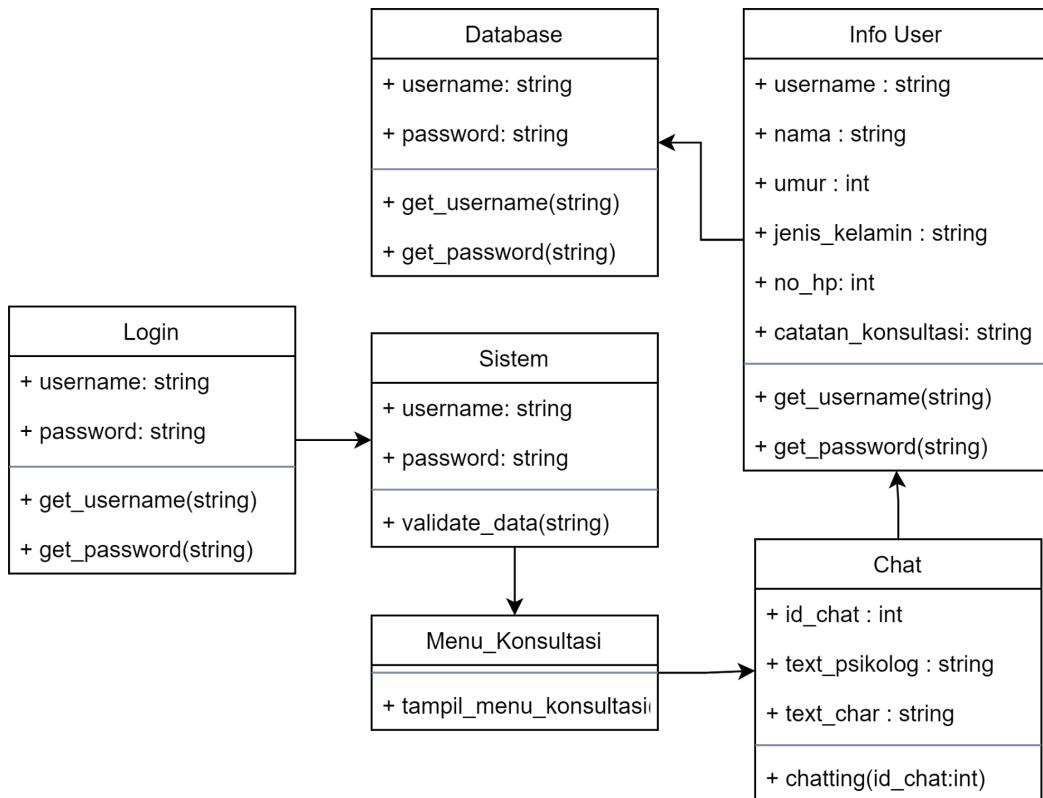
#### 3.1.3.2 Sequence Diagram

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



### 3.1.3.3 Diagram Kelas

Buatlah diagram kelas untuk use case tersebut. buat class diagram **BUKAN KESELURUHAN**, tapi **PER USE CASE**



### 3.1.4 Use Case Mengupload Medical Records

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

#### 3.1.4.1 Identifikasi Kelas

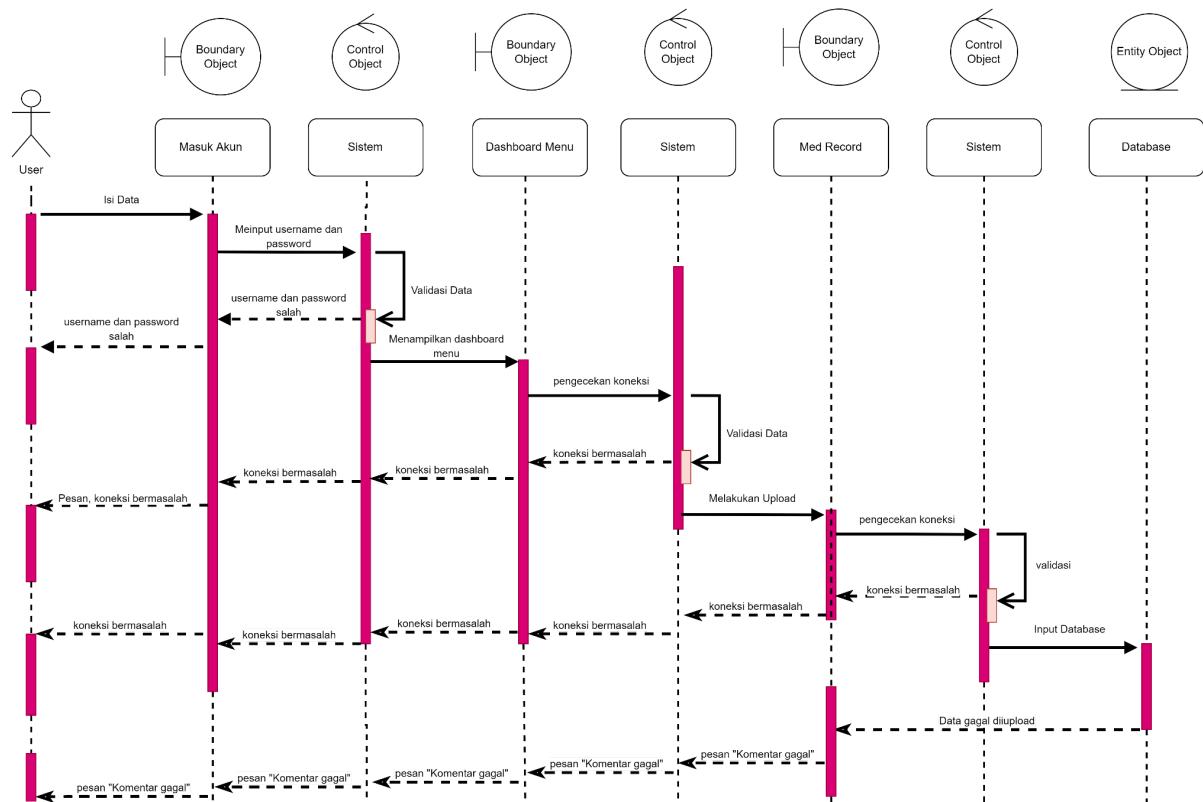
*Identifikasi kelas yang terkait dengan use case tersebut. Kelas di tahap perancangan dapat berbeda dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

No	Nama Kelas Perancangan	Tipe Kelas
1	Login	Boundary
2	Sistem	Control
3	Dashboard Menu	Boundary
4	Medical Record	Boundary
5	Database	Entity

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

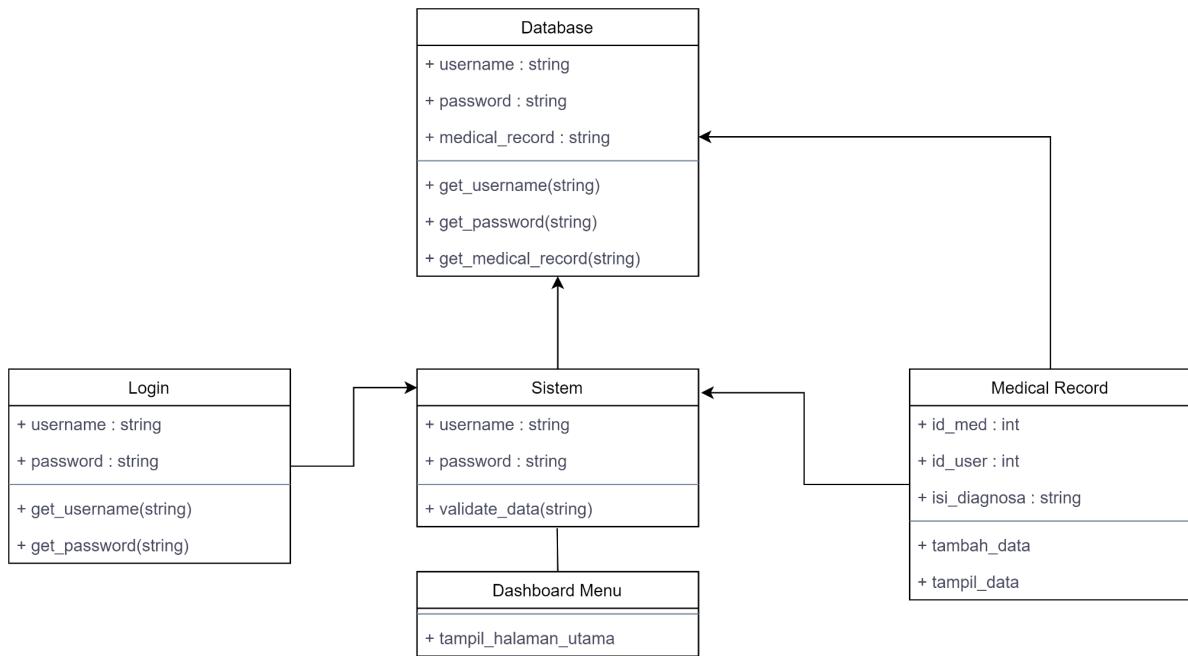
#### 3.1.4.2 Sequence Diagram

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



#### 3.1.4.3 Diagram Kelas

Buatlah diagram kelas untuk use case tersebut. buat class diagram **BUKAN KESELURUHAN**, tapi **PER USE CASE**



### 3.1.5 Use Case Manage System

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

#### 3.1.5.1 Identifikasi Kelas

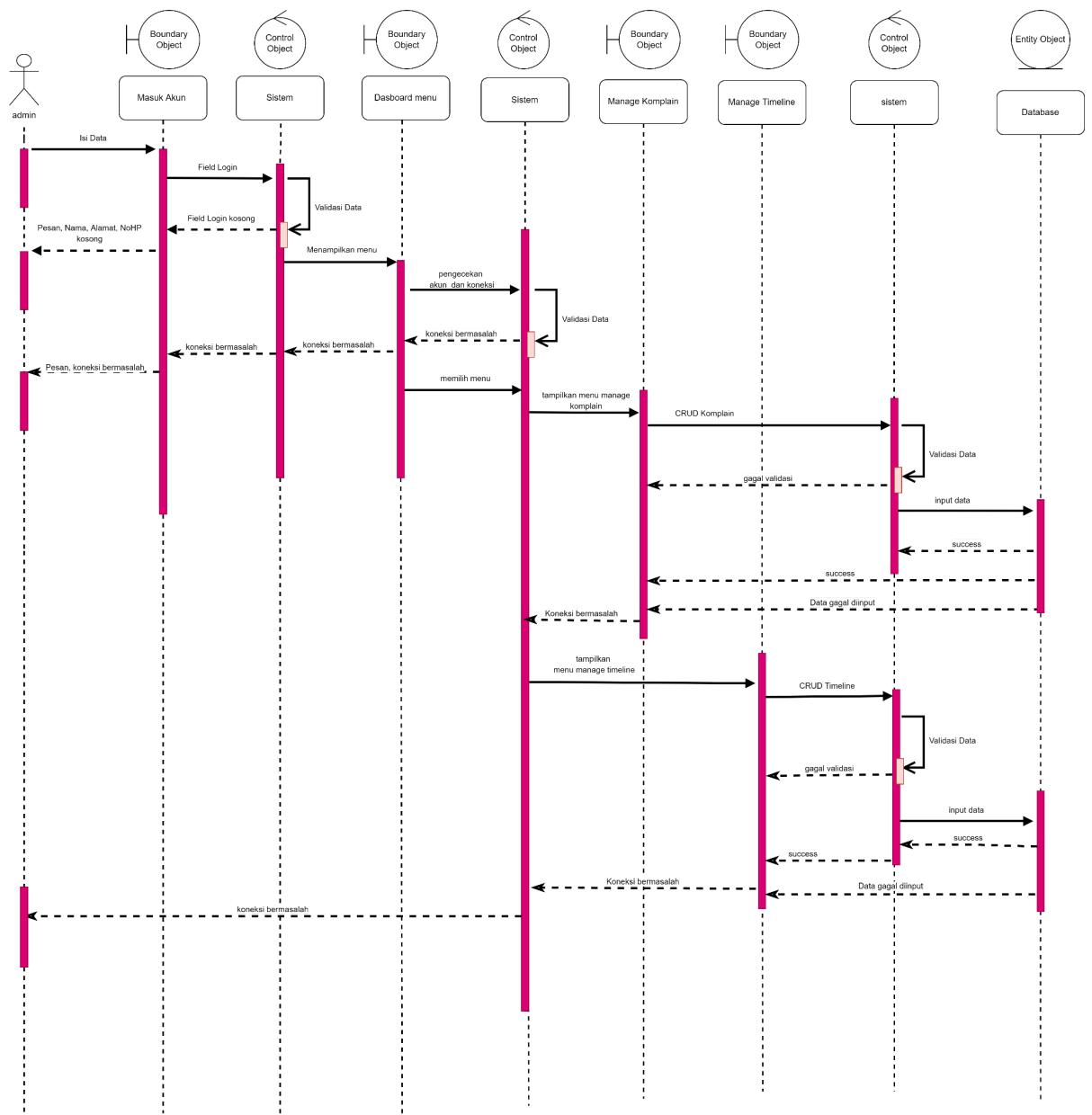
*Identifikasi kelas yang terkait dengan use case tersebut. Kelas di tahap perancangan dapat berbeda dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

No	Nama Kelas Perancangan	Tipe Kelas
	Login	Boundary
	Sistem	Control
	Dashboard	Boundary
	Manage Komplain	Boundary
	Manage Timeline	Boundary
	Database	Entity

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

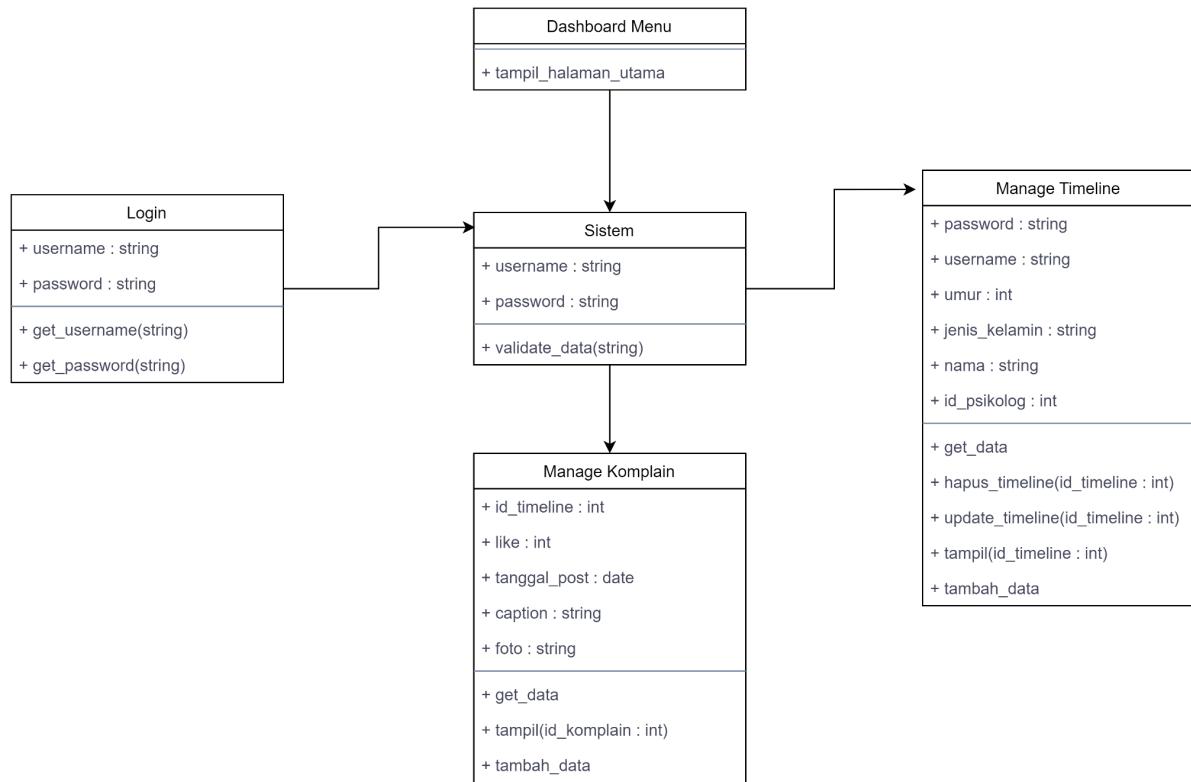
#### 3.1.5.2 Sequence Diagram

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



### 3.1.5.3 Diagram Kelas

Buatlah diagram kelas untuk use case tersebut. buat class diagram **BUKAN KESELURUHAN**, tapi **PER USE CASE**



### 3.1.6 Use Case Mengisi Biodata Lengkap User

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

#### 3.1.6.1 Identifikasi Kelas

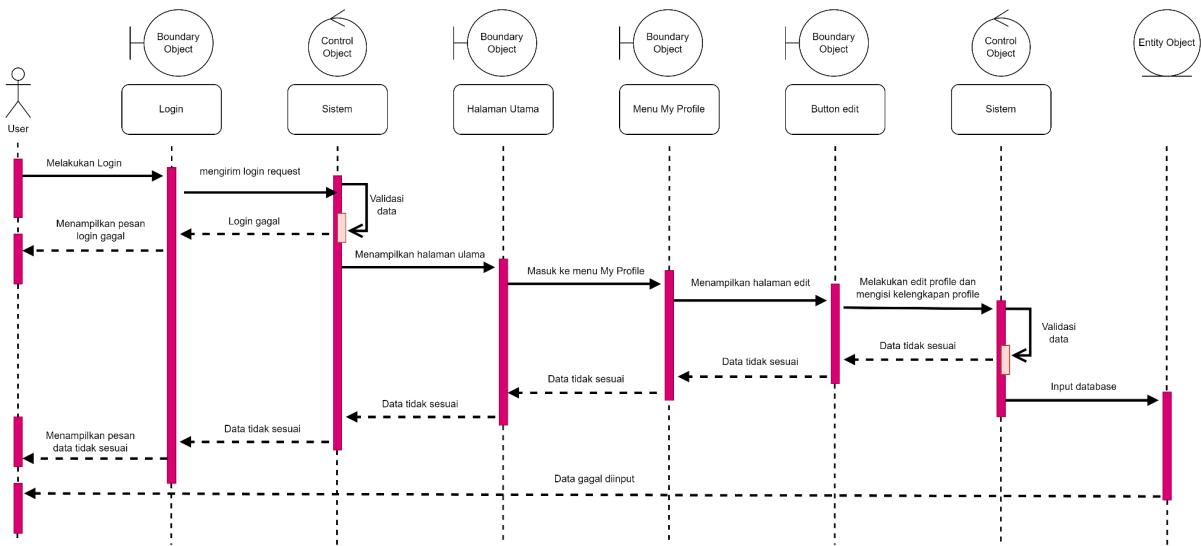
*Identifikasi kelas yang terkait dengan use case tersebut. Kelas di tahap perancangan dapat berbeda dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

No	Nama Kelas Perancangan	Tipe Kelas
1	Login	Boundary
2	Sistem	Control
3	Halaman Utama	Boundary
4	Menu My Profile	Boundary
5	Button edit	Boundary
6	Database	Entity

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

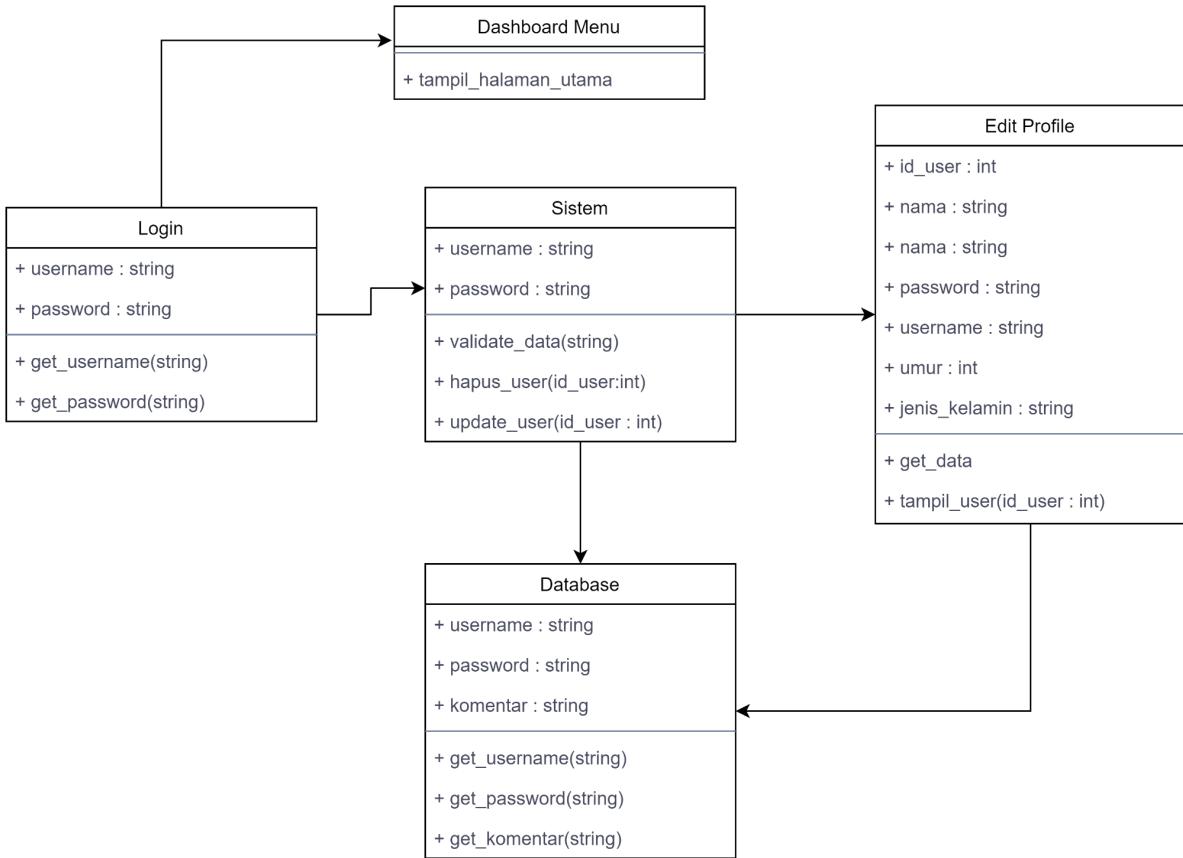
#### 3.1.6.2 Sequence Diagram

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



### 3.1.6.3 Diagram Kelas

Buatlah diagram kelas untuk use case tersebut. buat class diagram **BUKAN KESELURUHAN, tapi PER USE CASE**



### 3.1.7 Use Case Berkonsultasi

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

### 3.1.7.1 Identifikasi Kelas

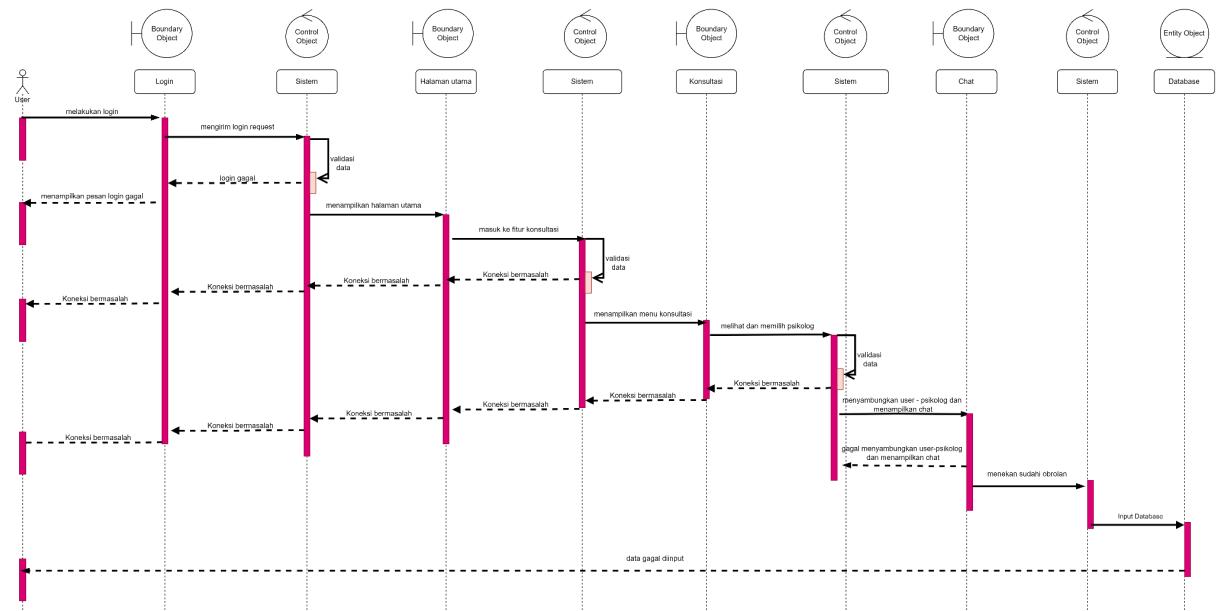
Identifikasi kelas yang terkait dengan use case tersebut. Kelas di tahap perancangan dapat berbeda dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:

No	Nama Kelas Perancangan	Tipe Kelas
1	Login	Boundary
2	Sistem	Controller
3	Halaman Utama	Boundary
4	Konsultasi	Boundary
5	Chat	Boundary
6	Database	Entity

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

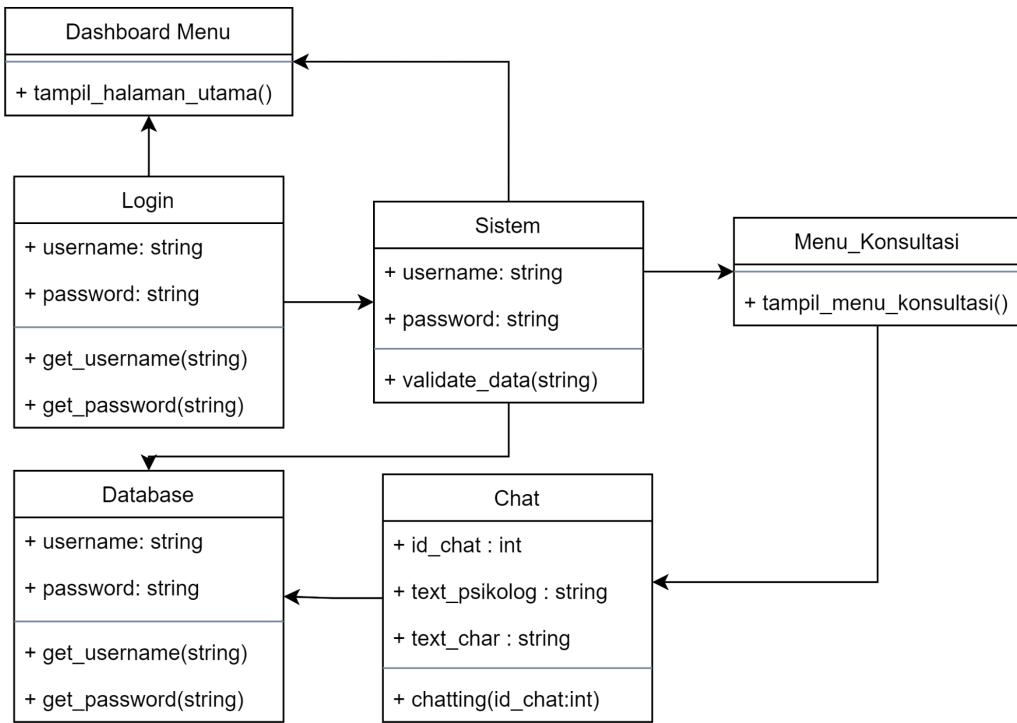
### 3.1.7.2 Sequence Diagram

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



### 3.1.7.3 Diagram Kelas

Buatlah diagram kelas untuk use case tersebut. buat class diagram **BUKAN KESELURUHAN**, tapi **PER USE CASE**



### 3.1.8 Use Case Melakukan Komplain

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

#### 3.1.8.1 Identifikasi Kelas

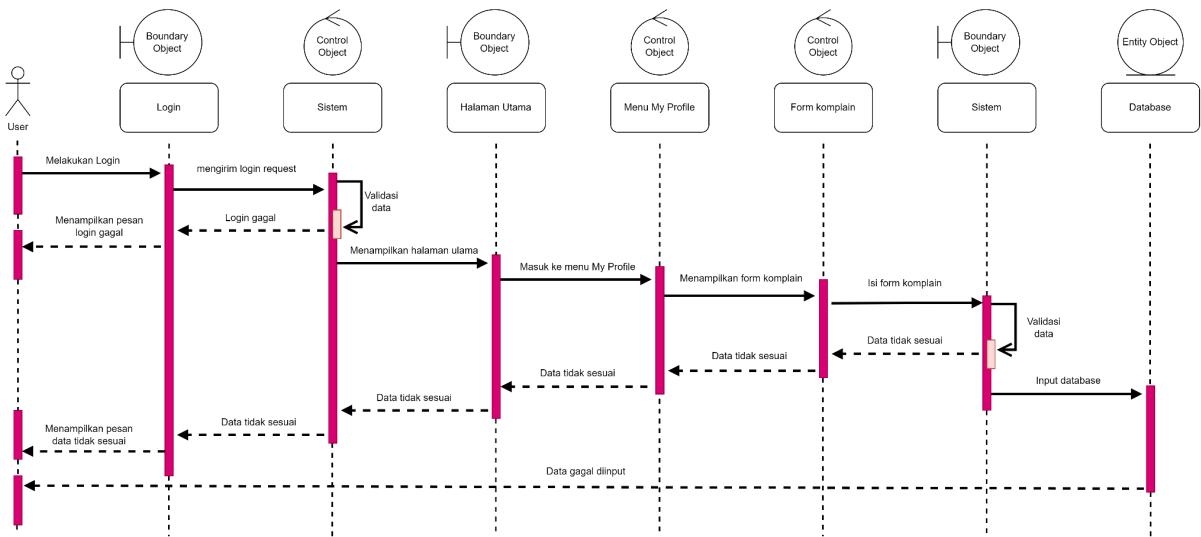
*Identifikasi kelas yang terkait dengan use case tersebut. Kelas di tahap perancangan dapat berbeda dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

No	Nama Kelas Perancangan	Tipe Kelas
1	Login	Boundary
2	Sistem	Controller
3	Halaman Utama	Boundary
4	Menu My Profile	Boundary
5	Form Komplain	Boundary
6	Database	Entity

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

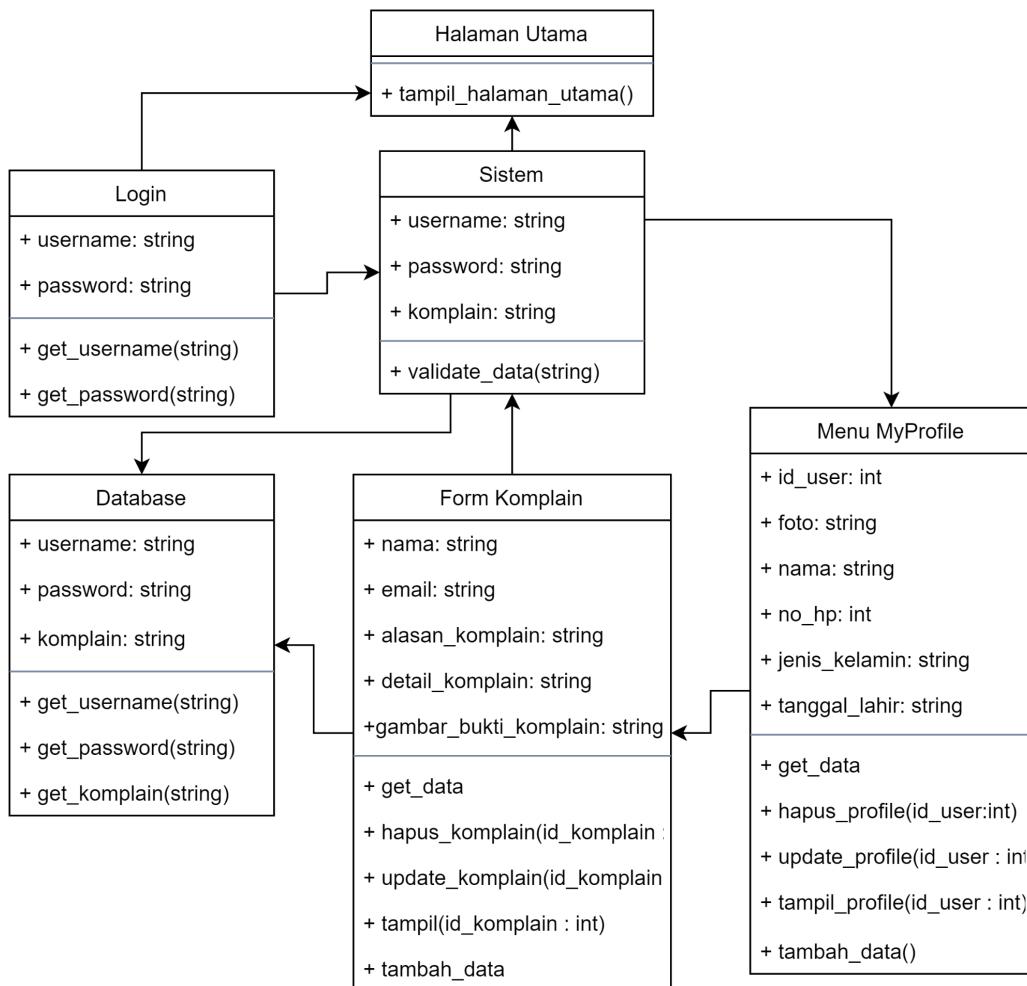
#### 3.1.8.2 Sequence Diagram

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



### 3.1.8.3 Diagram Kelas

Buatlah diagram kelas untuk use case tersebut. buat class diagram **BUKAN KESELURUHAN**, tapi **PER USE CASE**



### 3.1.9 Use Case Timeline

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

#### 3.1.9.1 Identifikasi Kelas

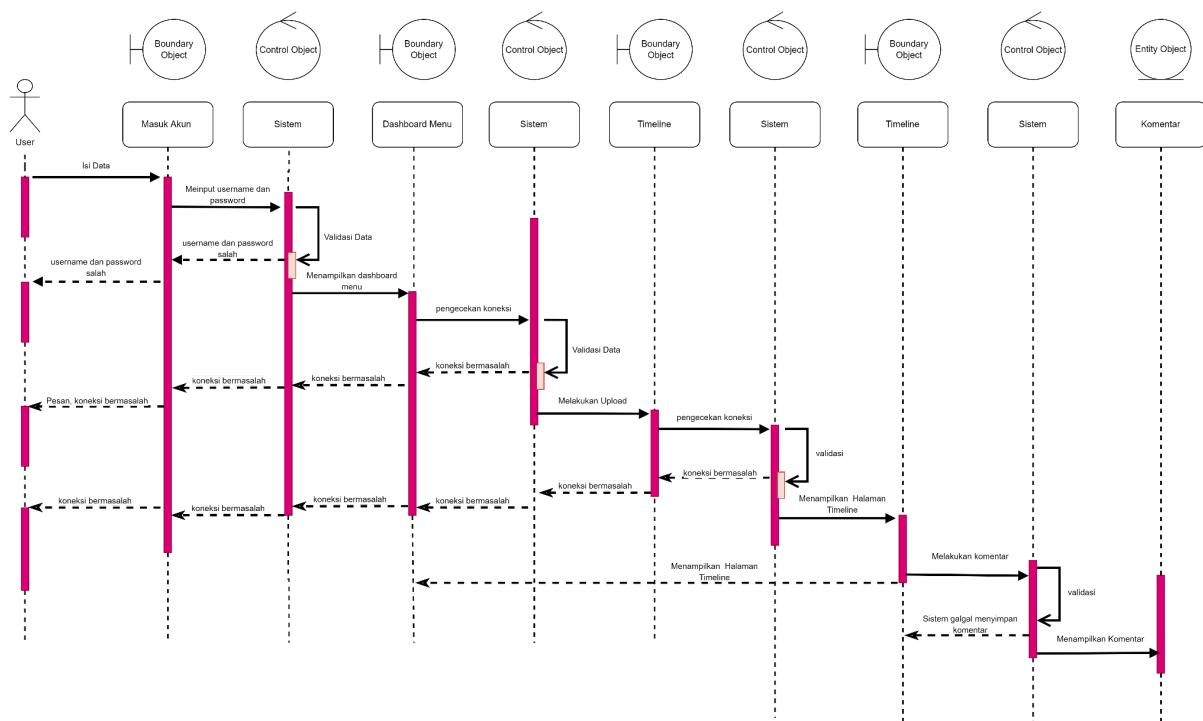
Identifikasi kelas yang terkait dengan use case tersebut. Kelas di tahap perancangan dapat berbeda dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:

No	Nama Kelas Perancangan	Tipe Kelas
1	Masuk Akun	Boundary
2	Sistem	Control
3	Dashboard Menu	Boundary
4	Timeline	Boundary
5	Komentar	Entity

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

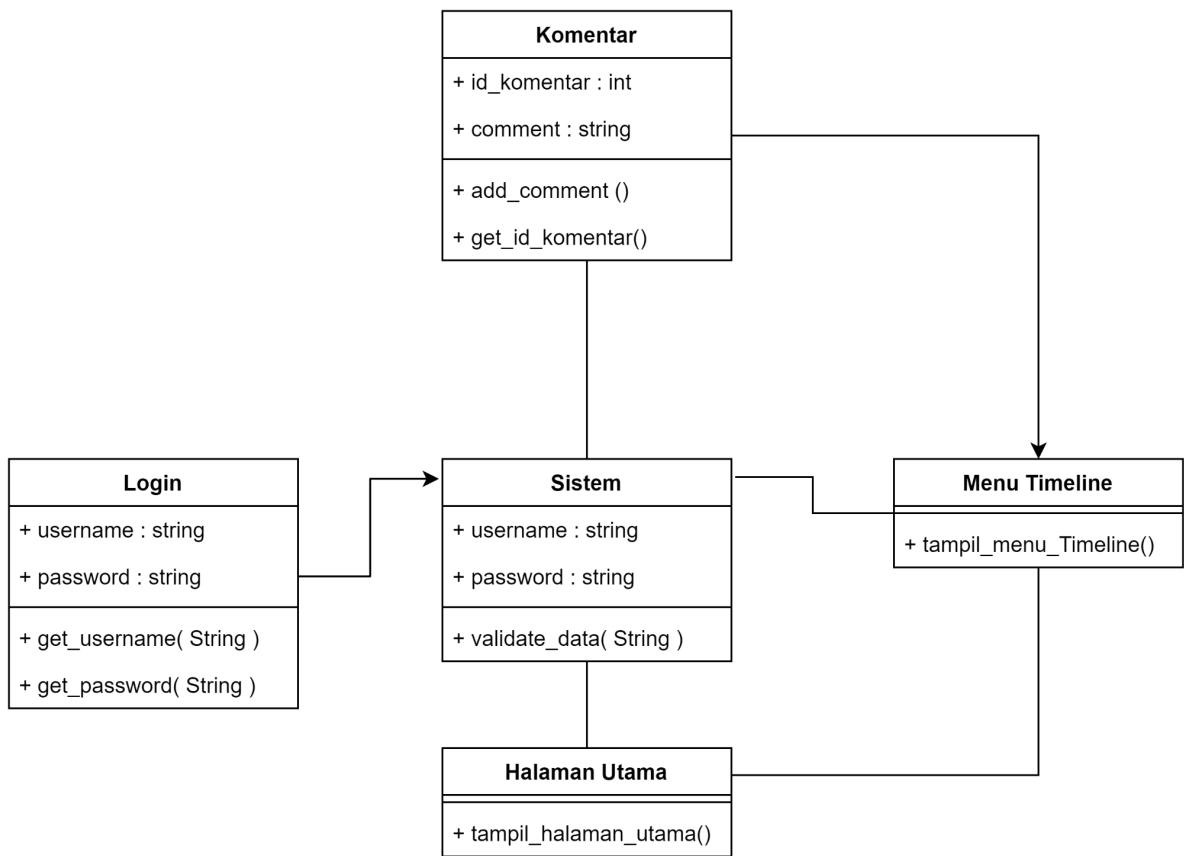
#### 3.1.9.2 Sequence Diagram

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



#### 3.1.9.3 Diagram Kelas

Buatlah diagram kelas untuk use case tersebut. buat class diagram **BUKAN KESELURUHAN**, tapi **PER USE CASE**



### 3.1.10 Use Case Melakukan Print Medical Record

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

#### 3.1.10.1 Identifikasi Kelas

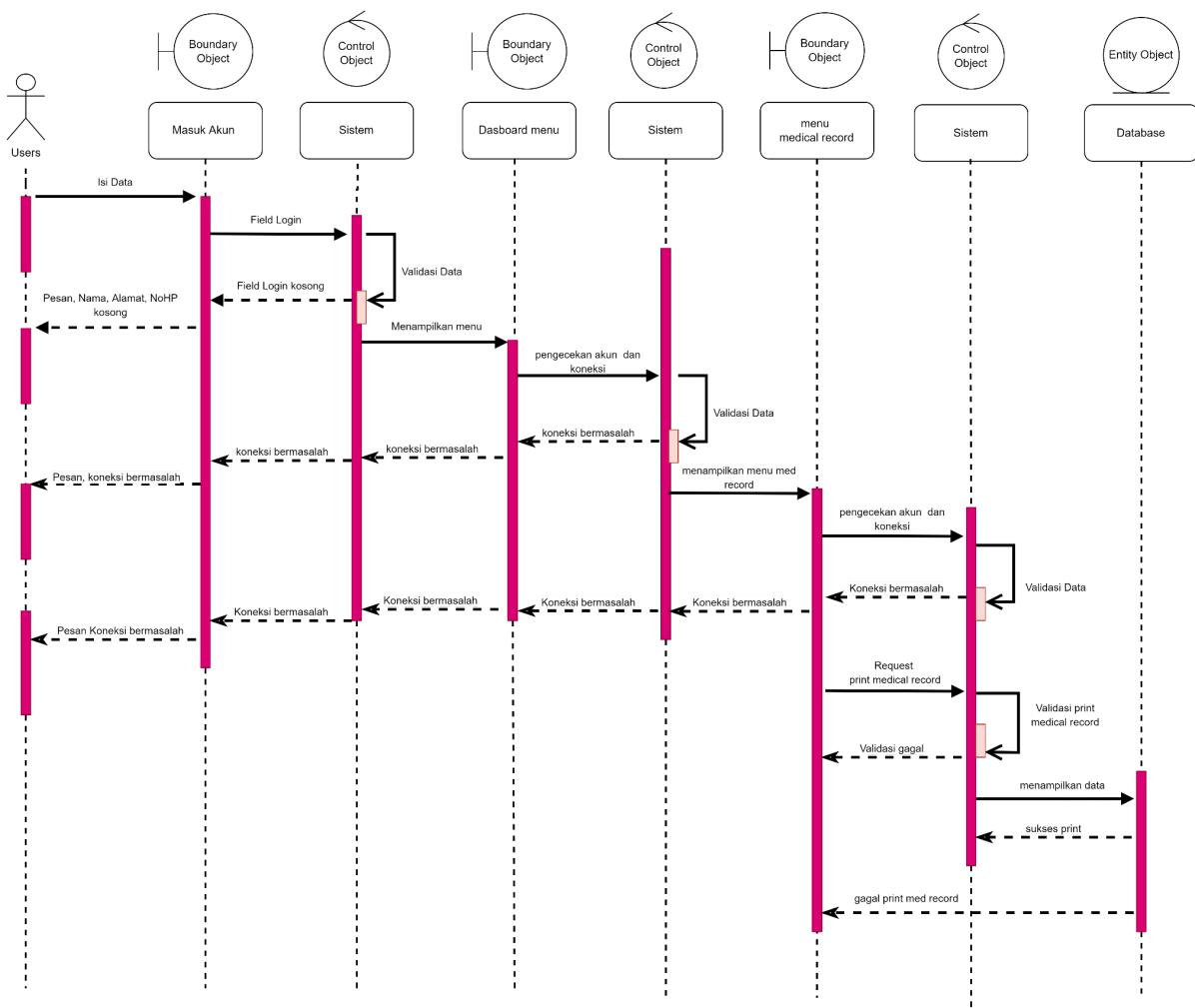
*Identifikasi kelas yang terkait dengan use case tersebut. Kelas di tahap perancangan dapat berbeda dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

No	Nama Kelas Perancangan	Tipe Kelas
1.	Login	Boundary
2	Sistem	Controller
3	Dashboard Menu	Boundary
4	Menu medical record	Boundary
5	database	Entity

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

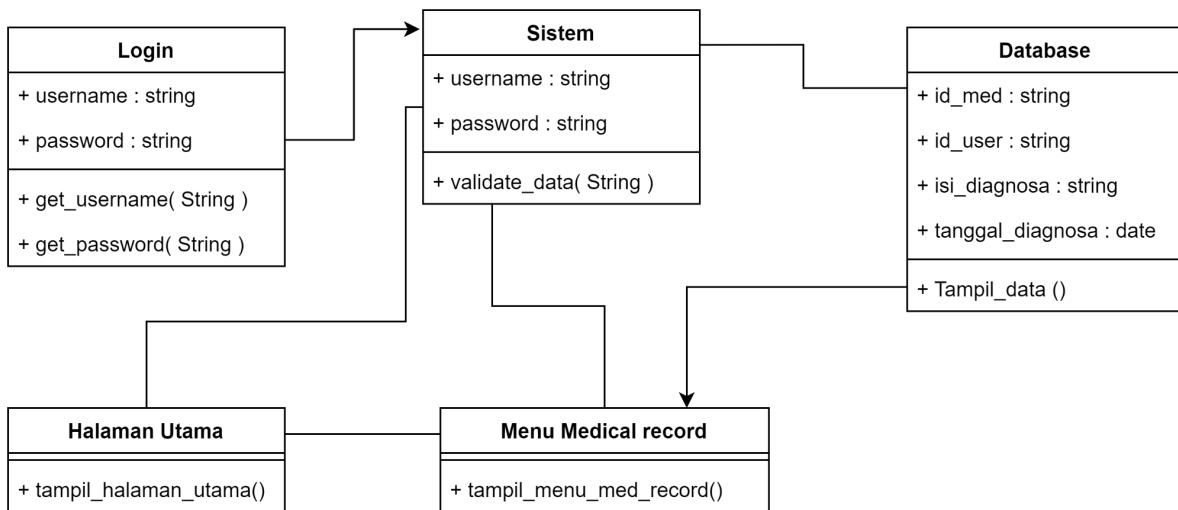
#### 3.1.10.2 Sequence Diagram

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



### 3.1.10.3 Diagram Kelas

Buatlah diagram kelas untuk use case tersebut. buat class diagram **BUKAN KESELURUHAN, tapi PER USE CASE**



### 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	Konsultasi, Komplai, Timeline, Medical Record
2	Admin	User, Komplain, Timeline
3	Psikolog	Medical Record, Konsultasi
4	Komplain	Admin, user
5	Medical Record	User, Psikolog
6	Timeline	User, Admin
7	Konsultasi	Chat, User, Psikolog
8	Chat	Konsultasi

Untuk setiap kelas:

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

#### 3.2.1 Kelas User

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

Nama Kelas : User

Nama Operasi	Visibility (private, public)	Keterangan
get data()	public	mengambil data
tambah data()	public	menambah data
tampil user()	public	menampilkan data user
update user()	public	update data user
hapus user()	public	hapus data user
Nama Atribut	Visibility (private, public)	Tipe
id user	public	integer
nama	public	string
jenis kelamin	public	string
umur	public	integer
username	public	string
password	public	string

#### 3.2.2 Kelas Admin

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

Nama Kelas : Admin

Nama Operasi	Visibility (private, public)	Keterangan
get data()	public	mengambil data
get id()	public	mengambil id
tambah data()	public	menambah data
update admin()	public	mengupdate data

<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>username</i>	<i>public</i>	<i>string</i>
<i>password</i>	<i>public</i>	<i>string</i>
<i>id admin</i>	<i>public</i>	<i>integer</i>

### 3.2.3 Kelas Psikolog

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

*Nama Kelas* : *Psikolog*

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
<i>get data()</i>	<i>public</i>	<i>mengambil data</i>
<i>tampil()</i>	<i>public</i>	<i>menampilkan data</i>
<i>tambah data()</i>	<i>public</i>	<i>menambah data</i>
<i>tampil psikolog()</i>	<i>public</i>	<i>menampilkan data psikolog</i>
<i>update psikolog()</i>	<i>public</i>	<i>mengupdate data psikolog</i>
<i>hapus psikolog()</i>	<i>public</i>	<i>menghapus data psikolog</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>id psikolog</i>	<i>public</i>	<i>integer</i>
<i>nama</i>	<i>public</i>	<i>string</i>
<i>jenis kelamin</i>	<i>public</i>	<i>string</i>
<i>umur</i>	<i>public</i>	<i>integer</i>
<i>username</i>	<i>public</i>	<i>string</i>
<i>password</i>	<i>public</i>	<i>string</i>

### 3.2.4 Kelas Komplain

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

*Nama Kelas* : .....

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
<i>get data()</i>	<i>public</i>	<i>mengambil data</i>
<i>tambah data()</i>	<i>public</i>	<i>menambah data</i>
<i>tampil komplain()</i>	<i>public</i>	<i>menampilkan data</i>
<i>update komplain()</i>	<i>public</i>	<i>mengupdate data</i>
<i>hapus komplain()</i>	<i>public</i>	<i>menghapus data</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>id komplain</i>	<i>public</i>	<i>int</i>
<i>id user</i>	<i>public</i>	<i>int</i>
<i>jenis komplain</i>	<i>public</i>	<i>string</i>
<i>deskripsi</i>	<i>public</i>	<i>string</i>
<i>status</i>	<i>public</i>	<i>boolean</i>

### 3.2.5 Kelas Medical Record

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

*Nama Kelas : Medical Record*

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
<i>get data()</i>	<i>public</i>	<i>mengambil data</i>
<i>tambah data()</i>	<i>public</i>	<i>menambah data</i>
<i>tampil()</i>	<i>public</i>	<i>menampilkan data</i>
<i>update diagnosa()</i>	<i>public</i>	<i>mengupdate data</i>
<i>hapus diagnosa()</i>	<i>public</i>	<i>menghapus data</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>id med</i>	<i>public</i>	<i>integer</i>
<i>id user</i>	<i>public</i>	<i>integer</i>
<i>isi diagnosa</i>	<i>public</i>	<i>string</i>
<i>tanggal diagnosa</i>	<i>public</i>	<i>date</i>

### 3.2.6 Kelas Timeline

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

*Nama Kelas : Timeline*

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
<i>get data()</i>	<i>public</i>	<i>mengambil data</i>
<i>tambah data()</i>	<i>public</i>	<i>menambah data</i>
<i>tampil komplain()</i>	<i>public</i>	<i>menampilkan data</i>
<i>update komplain()</i>	<i>public</i>	<i>mengupdate data</i>
<i>hapus komplain()</i>	<i>public</i>	<i>menghapus data</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>id timeline</i>	<i>public</i>	<i>int</i>
<i>id user</i>	<i>public</i>	<i>int</i>
<i>comment</i>	<i>public</i>	<i>string</i>
<i>caption</i>	<i>public</i>	<i>string</i>
<i>tanggal post</i>	<i>public</i>	<i>date</i>

### 3.2.7 Kelas Konsultasi

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

*Nama Kelas : konsultasi*

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
<i>get data()</i>	<i>public</i>	<i>mengambil data</i>
<i>tambah data()</i>	<i>public</i>	<i>menambah data</i>
<i>history konsultasi()</i>	<i>public</i>	<i>menampilkan data history konsultasi</i>

<i>tampil konsultasi()</i>	<i>public</i>	<i>menampilkan data</i>
<i>update konsultasi()</i>	<i>public</i>	<i>mengupdate data</i>
<i>hapus konsultasi()</i>	<i>public</i>	<i>menghapus data</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>id konsultasi</i>	<i>public</i>	<i>int</i>

### 3.2.8 Kelas Chat

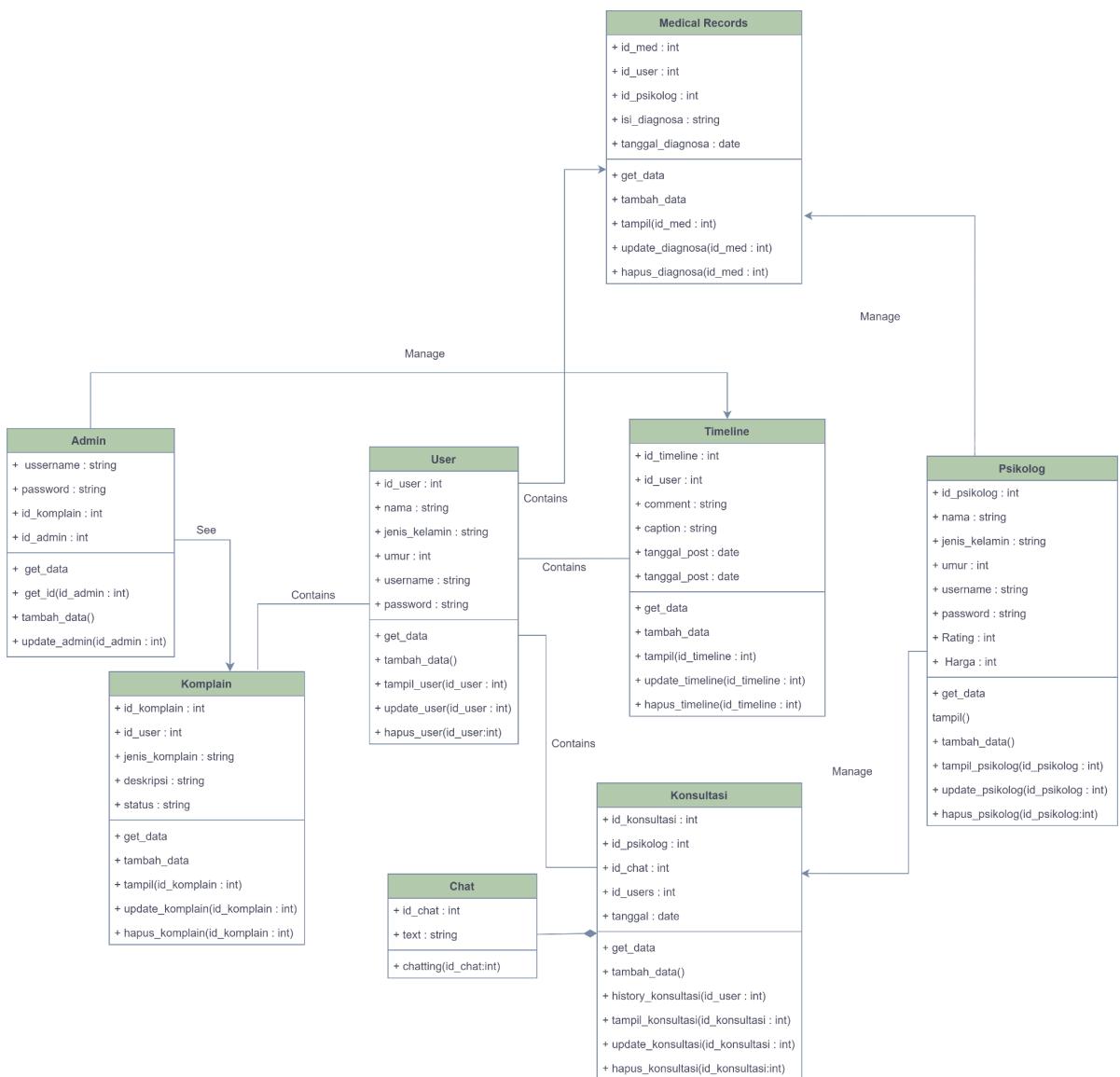
Bagian ini diisi dengan daftar operasi dan atribut. Buat untuk setiap kelas.

*Nama Kelas* : *chat*

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
<i>chatting()</i>	<i>public</i>	<i>tempat menampung chat konsultasi</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>id chat</i>	<i>public</i>	<i>int</i>
<i>text</i>	<i>public</i>	<i>string</i>

### 3.3 Diagram Kelas Keseluruhan

Bagian ini diisi dengan diagram kelas keseluruhan.



## 3.4 Algoritma/Query

### 3.4.1 Algoritma/Query Login

Nama Kelas : User

Nama Operasi : login

Algoritma :

(Algo-001)

input (username: string, password: string)

X = User

X = Users.find(\$and:[{username : value}, {password: value}])

return X

*Query* :

No Query	Query	Keterangan
Q-1	Insert	Untuk memasukan data ke database
Q-2	Find	Untuk mencari data dalam database
Q-3	Update	Untuk mengupdate data pada database
Q-4	Remove	Untuk menghapus data pada database

### 3.4.2 Algoritma/Query Register

*Nama Kelas* : User

*Nama Operasi* : register

*Algoritma* :

(Algo-002)

```
input (username: string, password: string)
```

```
X = User
```

```
X = Users.insert([{id_user: value}, {nama: value}, {jeniskelamin: value}, {umur: value}, {username: value}, {password: value}])
```

```
return X
```

*Query* :

No Query	Query	Keterangan
Q-1	Insert	Untuk memasukan data ke database
Q-2	Find	Untuk mencari data dalam database
Q-3	Update	Untuk mengupdate data pada database
Q-4	Remove	Untuk menghapus data pada database

### 3.4.3 Algoritma/Query Psikolog

*Nama Kelas* : Psikolog

*Nama Operasi* : model Psikolog

*Algoritma* :

(Algo-003)

```
class PsikologModel {
    int? id_psikolog, umur, rating, harga;
    String? nama, jenis_kelamin, username, password;

    PsikologModel({this.id_psikolog, this.nama, this.jenis_kelamin, this.umur, this.username, this.password,
    this.rating, this.harga});
    factory PsikologModel.fromMap(Map map) {
        return PsikologModel(
            id_psikolog: map['id_psikolog'], nama: map['nama'], jenis_kelamin: map['jenis_kelamin'], umur: map['umur'],
            username: map['username'], password: map['password'], rating: map['rating'], harga: map['harga']
        );
    }

    Map<String, dynamic> toMap() {
        return {
            'id_psikolog': id_psikolog, 'nama': nama, 'jenis_kelamin': jenis_kelamin, 'umur': umur, 'username': username,
            'password': password, 'rating': rating, 'harga': harga
        };
    }
}
```

*Query :*

No Query	Query	Keterangan
Q-1	Insert	Untuk memasukan data ke database
Q-2	Find	Untuk mencari data dalam database
Q-3	Update	Untuk mengupdate data pada database
Q-4	Remove	Untuk menghapus data pada database

### 3.4.4 Algoritma/Query Komplain

*Nama Kelas : Komplain  
Nama Operasi : Model Komplain  
Algoritma :*

(Algo-004)

```
class KomplainModel{
    String? idkomplain;
    String? iduser;
    String? jeniskomplain;
    String? deskripsi;
    String? status;

    KomplainModel({this.idkomplain, this.iduser, this.jeniskomplain, this.deskripsi, this.status});

    factory KomplainModel.fromMap(Map map){
        return KomplainModel(
            idkomplain: map['idkomplain'],
            iduser: map['iduser'],
            jeniskomplain: map['jeniskomplain'],
            deskripsi: map['deskripsi'],
            status: map['status']
        );
    }

    Map<String, dynamic> toMap(){
        return{
            'idkomplain': idkomplain,
            'iduser': iduser,
            'jeniskomplain': jeniskomplain,
            'deskripsi': deskripsi,
            'status': status
        };
    }
}
```

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

*Query :*

No Query	Query	Keterangan
Q-1	Insert	Untuk memasukan data ke database
Q-2	Find	Untuk mencari data dalam database
Q-3	Update	Untuk mengupdate data pada database

<i>Q-4</i>	<i>Remove</i>	<i>Untuk menghapus data pada database</i>
------------	---------------	---

### 3.4.5 Algoritma/Query Konsultasi

*Nama Kelas :Konsultasi*

*Nama Operasi : Model Konsultasi*

*Algoritma :*

*(Algo-005)*

```
class konsultasiModel {
    String? id_konsultasi;

    konsultasiModel({this.id_konsultasi});

    factory konsultasiModel.fromMap(Map map) {
        return konsultasiModel(
            id_konsultasi: map['id_konsultasi'],
        );
    }

    Map<String, dynamic> toMap() {
        return {
            'id_konsultasi': id_konsultasi,
        };
    }
}
```

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

*Query :*

<i>No Query</i>	<i>Query</i>	<i>Keterangan</i>
<i>Q-1</i>	<i>Insert</i>	<i>Untuk memasukan data ke database</i>
<i>Q-2</i>	<i>Find</i>	<i>Untuk mencari data dalam database</i>
<i>Q-3</i>	<i>Update</i>	<i>Untuk mengupdate data pada database</i>
<i>Q-4</i>	<i>Remove</i>	<i>Untuk menghapus data pada database</i>

### 3.4.6 Algoritma/Query Chat

*Nama Kelas :Chat*

*Nama Operasi :Model Chat*

*Algoritma :*

*(Algo-006)*

```
class chatModel {
    String? id_chat;
    String? text_psikolog;
    String? text_char;

    chatModel({this.id_chat, this.text_psikolog, this.text_char});

    factory chatModel.fromMap(Map map) {
        return chatModel(
            id_chat: map['id_chat'],
            text_psikolog: map['text_psikolog'],
        );
    }
}
```

```

        text_char: map('text_char'),
    );
}

Map<String, dynamic> toMap() {
    return {
        'id_chat': id_chat,
        'text_psikolog': text_psikolog,
        'text_char': text_char,
    };
}
}

```

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

*Query :*

No Query	Query	Keterangan
Q-1	Insert	Untuk memasukan data ke database
Q-2	Find	Untuk mencari data dalam database
Q-3	Update	Untuk mengupdate data pada database
Q-4	Remove	Untuk menghapus data pada database

### 3.4.7 Algoritma/Query Medical Record

*Nama Kelas :medical record*

*Nama Operasi :medRecord*

*Algoritma :*

(Algo-007)

```

class MedModel{
    int? idMed;
    int? idUser;
    String? isi_diagnosa;
    DateTime? date;

    MedModel({this.idMed, this.idUser, this.isi_diagnosa, this.date});

    factory MedModel.fromMap(map){
        return MedModel(
            idMed: map['idMed'],
           idUser: map['idUser'],
            isi_diagnosa: map['isi_diagnosa'],
        );
    }

    Map<String, dynamic> toMap(){
        return{
            'idMed':idMed,
            'idUser':idUser,
            'isi_diagnosa':isi_diagnosa
        };
    }
}

```

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

*Query* :

No Query	Query	Keterangan
Q-1	Insert	Untuk memasukan data ke database
Q-2	Find	Untuk mencari data dalam database
Q-3	Update	Untuk mengupdate data pada database
Q-4	Remove	Untuk menghapus data pada database

### 3.4.8 Algoritma/Query Timeline

*Nama Kelas* :Timeline

*Nama Operasi* :Model Timeline

*Algoritma* :

(Algo-008)

```

class TimelineModel {
    int? id_timeline;
    int? id_user;
    String? comment;
    String? caption;
    DateTime? tanggal_post;

    TimelineModel(
        {this.id_timeline,
        this.id_user,
        this.comment,
        this.caption,
        this.tanggal_post});

    factory TimelineModel.fromMap(Map map) {
        return TimelineModel(
            id_timeline: map['ID Timeline'],
            id_user: map['ID User'],
            comment: map['Comment'],
            caption: map['Caption'],
            tanggal_post: map['Tanggal Post']);
    }

    Map<String, dynamic> toMap() {
        return {
            'ID Timeline': id_timeline,
            'ID User': id_user,
            'Comment': comment,
            'Caption': caption,
            'Tanggal Post': tanggal_post
        };
    }
}

```

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

*Query* :

No Query	Query	Keterangan
Q-1	Insert	Untuk memasukan data ke database

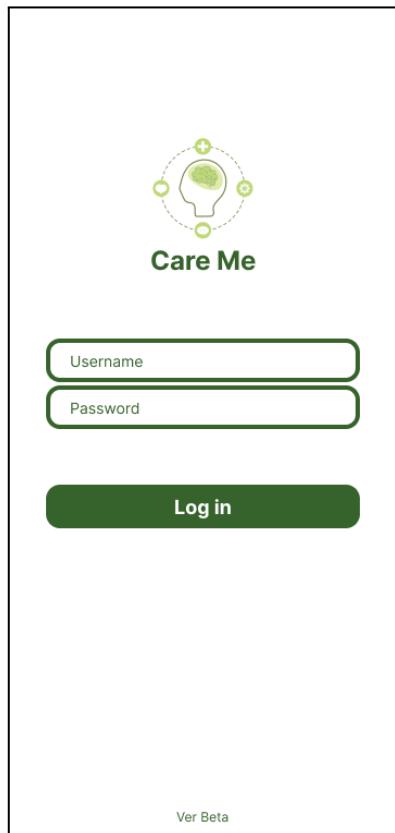
<i>Q-2</i>	<i>Find</i>	<i>Untuk mencari data dalam database</i>
<i>Q-3</i>	<i>Update</i>	<i>Untuk mengupdate data pada database</i>
<i>Q-4</i>	<i>Remove</i>	<i>Untuk menghapus data pada database</i>

### 3.5 Perancangan Antarmuka

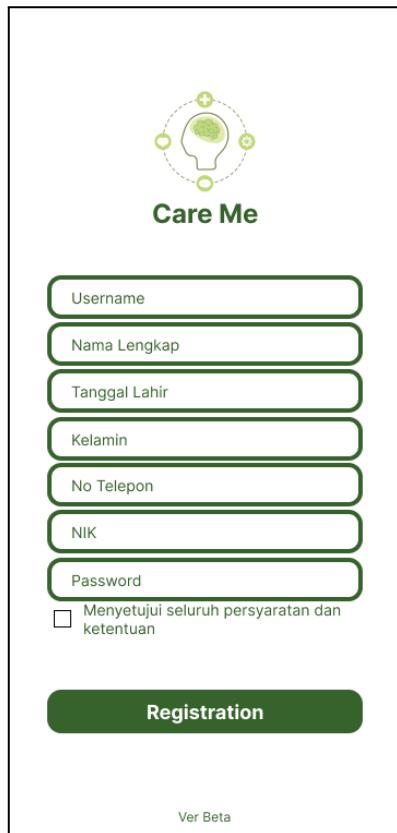
Bagian ini diisi dengan versi awal prototipe antarmuka.

Selanjutnya, untuk setiap antarmuka/layar, tuliskan spesifikasi detilnya, misalnya seperti dibawah ini:

*Antarmuka : 1*  
*Deskripsi : Halaman Login*

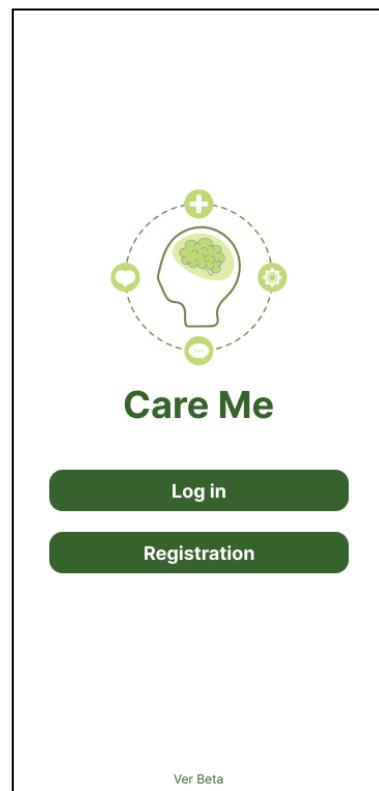


*Antarmuka : 2*  
*Deskripsi : Halaman Registrasi*



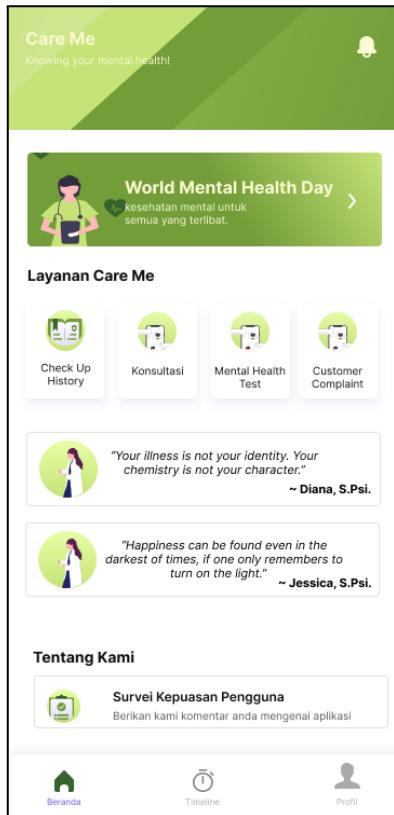
The registration form for the Care Me application. It features a logo at the top with a brain inside a lightbulb and the text "Care Me". Below the logo is a vertical stack of input fields with labels: "Username", "Nama Lengkap", "Tanggal Lahir", "Kelamin", "No Telepon", "NIK", and "Password". After the password field is a checkbox labeled "Menyetujui seluruh persyaratan dan ketentuan". At the bottom is a large green "Registration" button. In the bottom right corner, the text "Ver Beta" is visible.

*Antarmuka : 3  
Deskripsi : Halaman Identifier*



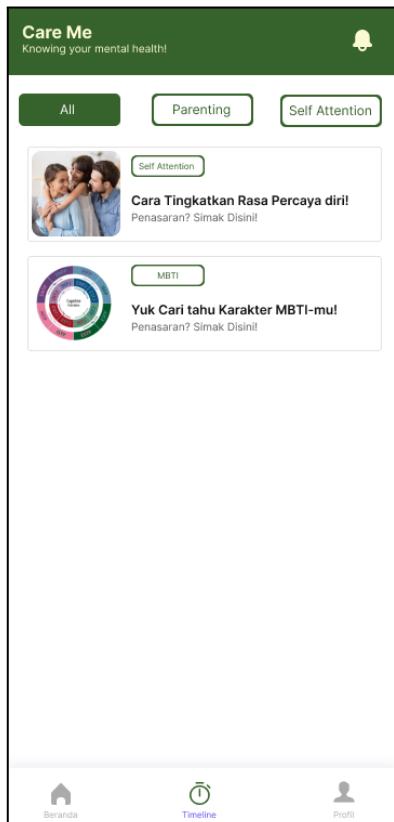
The identifier form for the Care Me application. It features a logo at the top with a brain inside a lightbulb and the text "Care Me". Below the logo are two large green buttons: "Log in" and "Registration". In the bottom right corner, the text "Ver Beta" is visible.

*Antarmuka : 4  
Deskripsi : Halaman Home*



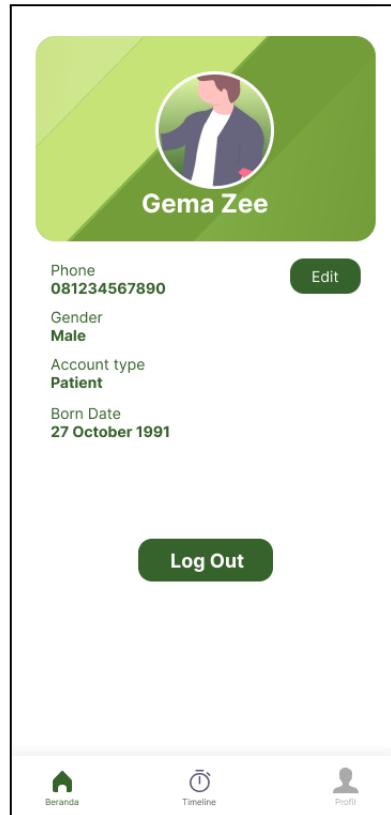
*Antarmuka*  
*Deskripsi*

: 5  
: Halaman Timeline



*Antarmuka* : 6

*Deskripsi : Halaman Profile*

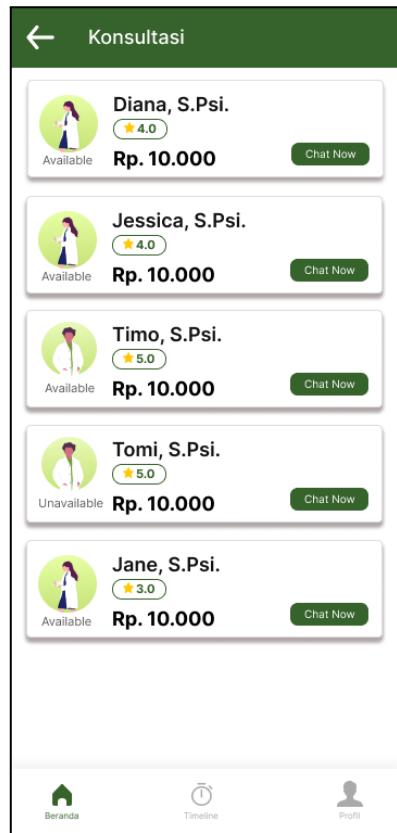


*Antarmuka  
Deskripsi*

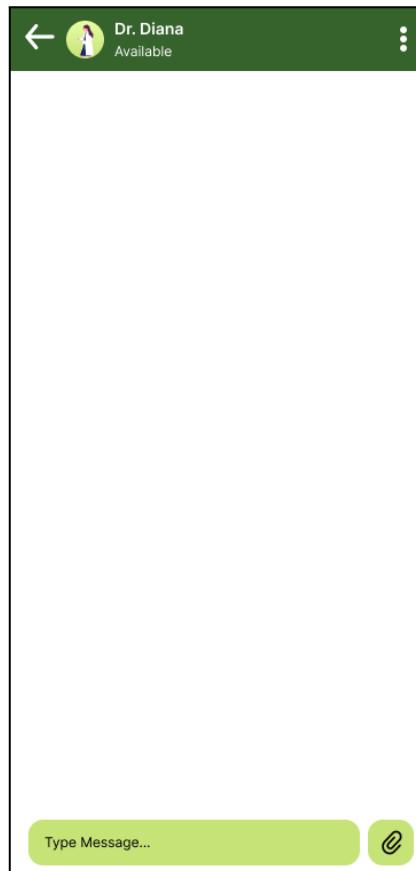
: 7  
: Halaman Komplain

*Antarmuka  
Deskripsi*

: 8  
: Halaman Konsultasi



*Antarmuka : 9  
Deskripsi : Halaman Chat*

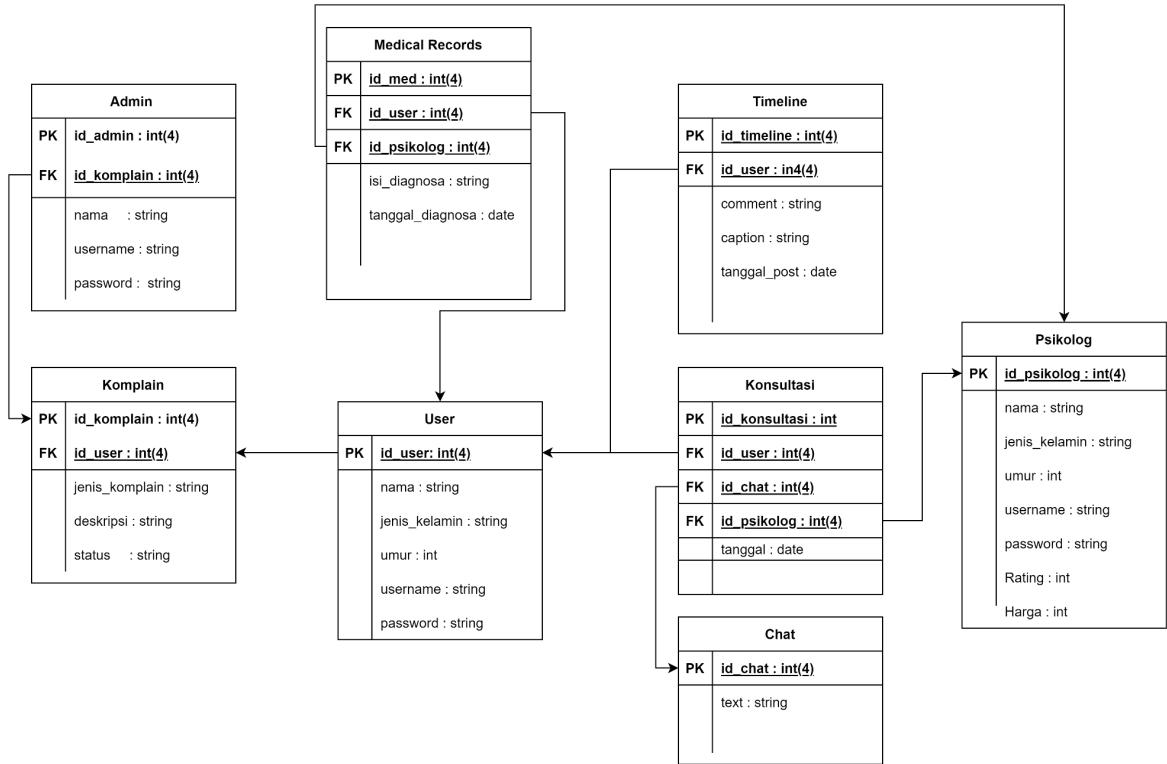


<b>Id Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
		<i>Diisi dengan string yg tampil pd layar</i>	<i>Diisi dengan penjelasan reaksi sistem, misalnya membuka layar apa, link kemana. Jika menyangkut suatu kode yang cukup rumit, acu algoritma yang telah diuraikan di atas.</i>
Button1	Button	OK	<i>Jika diklik, akan mengaktifkan Proses AlgoXXX.</i>
RTF1	RTF Box		<i>Isi Teks yang disimpan pada File xxx</i>

*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 basisdata dan traceability-nya terhadap kelas entity. (PEMBUATAN SKEMA RELASI)*



## 4 Matriks Kerunutan

Mapping use case dengan kelas-kelas terkait

Requirement	Usecase Terkait	Kelas
FR-01	Login dan register	Admin, user, psikolog, komplain, medical record, timeline, konsultasi, chat
FR-02	Memberikan konsultasi	Psikolog, konsultasi, chat
FR-03	Melihat info user	Psikolog
FR-04	Mengupload medical record	Psikolog
FR-05	Manage system	Admin, user, psikolog, komplain, medical record, timeline, konsultasi, chat
FR-06	Mengisi biodata user	User
FR-07	Berkonsultasi	User, konsultasi, chat
FR-08	Melakukan komplain	Komplain, user, admin
FR-09	Melihat Timeline	timeline, admin, user
FR-10	Melihat history	User, konsultasi
FR-11	Melakukan print medical record	User, medical record