

# **SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK**

## **SocialLife Aplikasi Social Media**

Oleh :

Diva Annisa Febecca - 1301204302  
Johannes Raphael Nandaputra - 1301204243  
Muhammad Naufal Abdillah - 1301201586  
Muhammad Mufid Utomo - 1301204441  
Ricardo Hamonangan - 1301204201

Kelompok 3  
IF-44-01



**PROGRAM STUDI INFORMATIKA**

**FAKULTAS INFORMATIKA**

**2022**

# Daftar Isi

<b>Daftar Isi</b>	<b>2</b>
<b>1. Pendahuluan</b>	<b>6</b>
1.1 Tujuan Penulisan Dokumen	6
1.2 Cakupan Produk	6
1.3 Konvensi Dokumen	6
1.4 Referensi	7
1.5 Gambaran Umum	7
<b>2. Overall Description</b>	<b>7</b>
2.1 Perspektif Produk	7
2.2 Fungsi Produk	7
2.3 Kelas dan Karakteristik Pengguna	8
2.4 Lingkungan Operasi	8
2.5 Batasan Perancangan dan Implementasi	8
2.6 Asumsi dan Dependensi	8
<b>3. Functional Requirements</b>	<b>9</b>
<b>4. Fitur Sistem (Use Cases)</b>	<b>14</b>
<b>4.1 Use Case 1</b>	<b>14</b>
4.1.2 Tujuan	14
4.1.3 Input	14
4.1.4 Output	14
4.1.5 Skenario Utama	14
4.1.6 Prakondisi	14
4.1.7 Langkah-langkah:	14
4.1.8 Pascakondisi	14
4.1.9 Skenario eksepsional 1	14
4.1.10 Contoh	14
<b>4.2 Use Case 2</b>	<b>15</b>
4.2.1 Nama Use Case:	15
4.2.2 Tujuan	15
4.2.3 Input	15
4.2.4 Output	15
4.2.5 Skenario Utama	15

4.2.6 Prakondisi	15
4.2.7 Langkah-langkah:	15
4.2.8 Pascakondisi	15
4.2.9 Skenario eksepsional 1	15
4.2.10 Contoh	15
4.3 Use Case 3	16
4.3.1 Nama Use Case:	16
4.3.2 Tujuan	16
4.3.3 Input :	16
4.3.4 Output	16
4.3.5 Skenario Utama:	16
4.3.6 Prakondisi	16
4.3.7 Langkah-langkah:	16
4.3.8 Pascakondisi	16
4.3.9 Skenario eksepsional 1	16
4.3.10 Contoh	16
4.4 Use Case 4	16
4.4.1 Nama Use Case:	17
4.4.2 Tujuan:	17
4.4.3 Input:	17
4.4.4 Output	17
4.4.5 Skenario Utama	17
4.4.6 Prakondisi	17
4.4.7 Langkah-langkah:	17
4.4.8 Pascakondisi	17
4.4.9 Skenario eksepsional 1	17
4.4.10 Contoh	17
4.5 Use Case 5	17
4.5.1 Nama Use Case:	18
4.5.2 Tujuan	18
4.5.3 Input	18
4.5.4 Output	18
4.5.5 Skenario Utama:	18
4.5.6 Prakondisi	18
4.5.7 Langkah-langkah:	18
4.5.8 Pascakondisi	18
4.5.9 Skenario eksepsional 1	18

4.5.10 Contoh	18
4.6 Use Case 6	18
4.6.1 Nama Use Case:	19
4.6.2 Tujuan	19
4.6.3 Input	19
4.6.4 Output	19
4.6.5 Skenario Utama	19
4.6.6 Prakondisi	19
4.6.7 Langkah-langkah:	19
4.6.8 Pascakondisi	19
4.6.9 Skenario eksepsional 1	19
4.6.10 Contoh	19
4.7 Use Case 7	19
4.7.1 Nama Use Case:	19
4.7.2 Tujuan	20
4.7.3 Input: -	20
4.7.4 Output	20
4.7.5 Skenario Utama	20
4.7.6 Prakondisi	20
4.7.7 Langkah-langkah:	20
4.7.8 Pascakondisi	20
4.7.9 Skenario eksepsional 1	20
4.7.10 Contoh	20
4.8 Use Case 8	20
4.8.1 Nama Use Case:	20
4.8.2 Tujuan	20
4.8.3 Input	21
4.8.4 Output	21
4.8.5 Skenario Utama	21
4.8.6 Prakondisi	21
4.8.7 Langkah-langkah:	21
4.8.8 Pascakondisi	21
4.8.9 Skenario eksepsional 1	21
4.8.10 Contoh	21
5. Requirements Non Fungsional	21
Lampiran A: Daftar Kata-Kata Sukar	22



## **1. Pendahuluan**

### **1.1 Tujuan Penulisan Dokumen**

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini ditulis dengan tujuan untuk mendeskripsikan semua kebutuhan yang diperlukan dalam pengembangan aplikasi “SocialLife” secara jelas dan tepat. Dokumen ini menjelaskan kebutuhan antarmuka, kebutuhan non fungsional, dan kebutuhan lain yang menjadi acuan untuk pengembang dalam membuat aplikasi ini.

### **1.2 Cakupan Produk**

Cakupan yang dibahas dalam dokumen ini adalah semua kebutuhan yang diperlukan dalam pengembangan aplikasi “SocialLife”, dengan manfaat, tujuan, dan sasaran sebagai berikut.

#### **1.2.1 Manfaat Produk**

SocialLife diharapkan bisa membantu pengguna untuk berkomunikasi dengan teman atau keluarganya dengan medium aplikasi modern, tanpa membuat pengguna candu terhadap media sosial.

#### **1.2.2 Tujuan Produk**

SocialLife dibuat dengan tujuan untuk menjadi alternatif dari social media yang sudah ada, agar pengguna tidak terlalu bergantung dengan social media yang membuat candu.

#### **1.2.3 Sasaran Produk**

SocialLife ditujukan untuk mereka yang sering berkecimpung di media sosial dan merasa kalau media sosial yang sudah ada sangat membuat candu dan menghabiskan waktu. Aplikasi SocialLife bisa menjadi alternatif yang baik untuk mereka.

### **1.3 Konvensi Dokumen**

Dokumen SKPL SocialLife ini dibuat dengan menggunakan font Times New Roman, dengan ukuran 12 di *body*, 18 di *heading*, dan 14 di *subheading*.

## 1.4 Referensi

Dokumen ini memiliki referensi dokumen berikut:

- TACHOnet Software Requirements Specification - DG TREN

## 1.5 Gambaran Umum

Dokumen ini memiliki berapa bagian.

1. Kebutuhan antarmuka pengguna menjelaskan bagaimana antarmuka aplikasi akan dibuat.
2. Use case akan menjelaskan bagaimana aplikasi seharusnya bekerja.
3. Kebutuhan non fungsional akan menjelaskan hal-hal yang harus bisa dilakukan oleh aplikasi ini.

## 2. Overall Description

### 2.1 Perspektif Produk

Aplikasi SocialLife dibuat sebagai alternatif dari media sosial yang sudah ada, di mana aplikasi ini tidak akan mengeksploitasi pengguna sebagaimana media sosial yang lain. Sehingga SocialLife dibuat sebagai produk baru, karena aplikasi ini bukan menggantikan aplikasi yang sudah ada, melainkan menjadi alternatif baru. Antarmuka aplikasi ini bisa diakses lewat web sehingga bisa diakses lewat device yang bisa membuka browser. Lalu, aplikasi akan terhubung dengan satu database yang menyimpan semua data pengguna, dan data yang penting akan dienkripsi untuk melindunginya dari pihak yang tidak bertanggungjawab.

### 2.2 Fungsi Produk

Aplikasi SocialLife ini memiliki fungsi layaknya media sosial yang lain, seperti user bisa membagikan postingan, memberi komentar, dan menghubungi temannya via chat atau telepon. Fungsi unggulan dari aplikasi ini adalah SocialLife akan mengingatkan pengguna apabila sudah menghabiskan 2 dalam di dalam aplikasi, sehingga pengguna akan terhindar dari kecanduan. Pengguna juga akan diingatkan apabila ada teman yang sudah tidak dihubungi lebih dari 14 hari, sehingga pengguna bisa terhubung dengan teman dan keluarga dengan optimal.

## 2.3 Kelas dan Karakteristik Pengguna

Aplikasi ini hanya memiliki satu kelas pengguna. Pengguna yang belum mendaftar bisa mendaftar di aplikasi, lalu setelahnya bisa mengakses semua fitur yang ada di SocialLife.

## 2.4 Lingkungan Operasi

Aplikasi SocialLife adalah aplikasi yang berbasis web, yang bisa dijalankan di mobile, PC, dan device lain yang bisa mengakses browser.

## 2.5 Batasan Perancangan dan Implementasi

Aplikasi SocialLife akan diimplementasikan dengan menggunakan MERN stack dengan bahasa pemrograman JavaScript.

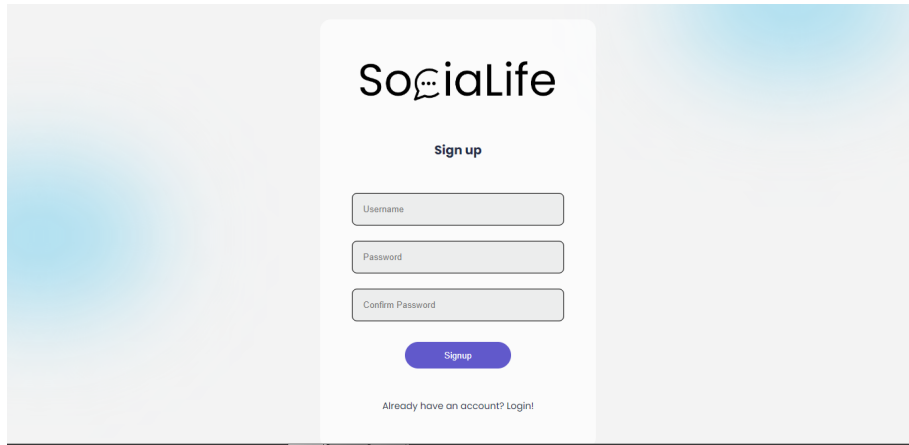
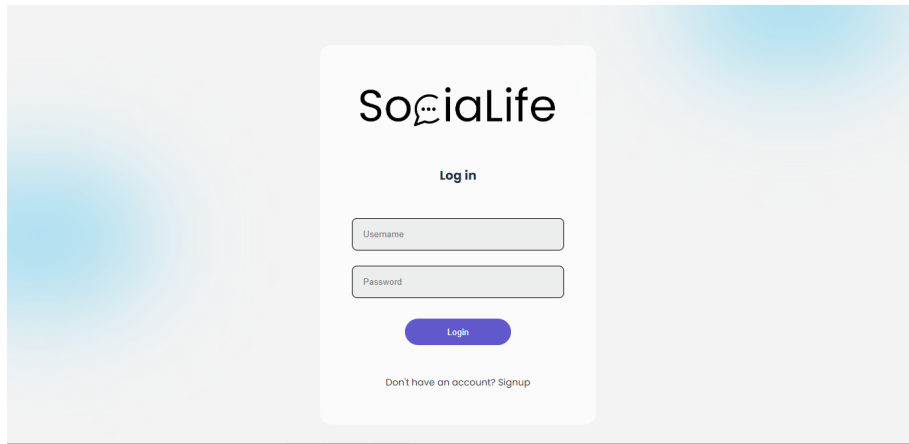
## 2.6 Asumsi dan Dependensi

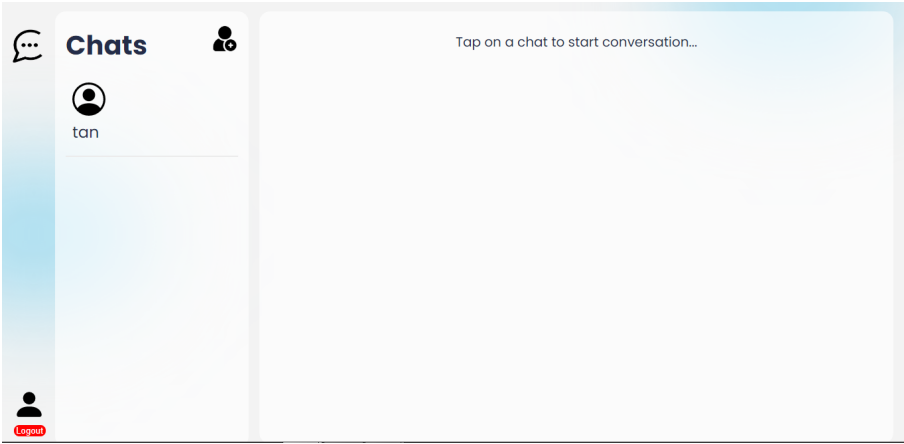
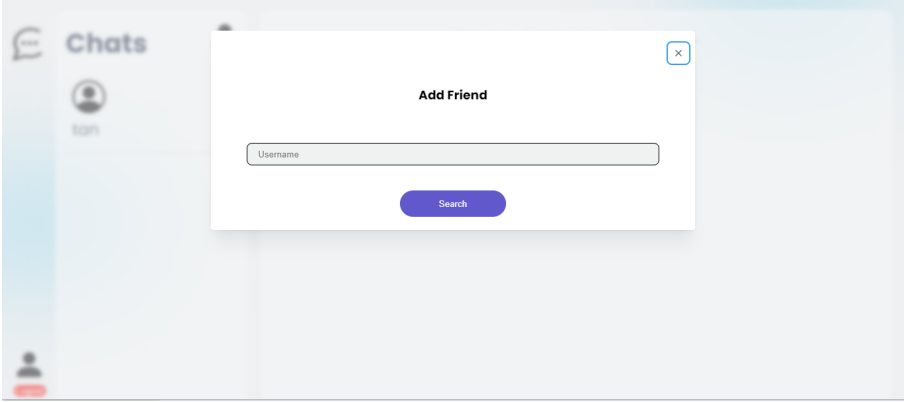
Asumsi berikut dibuat dalam perealisasiian aplikasi SocialLife:

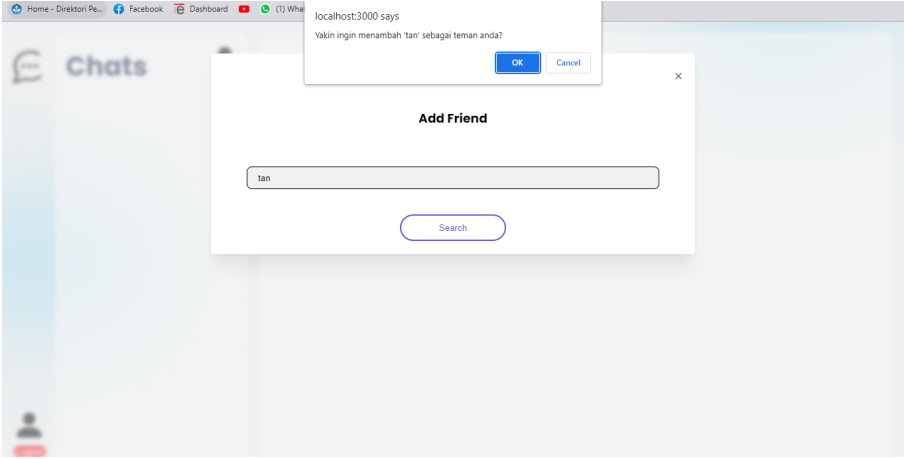
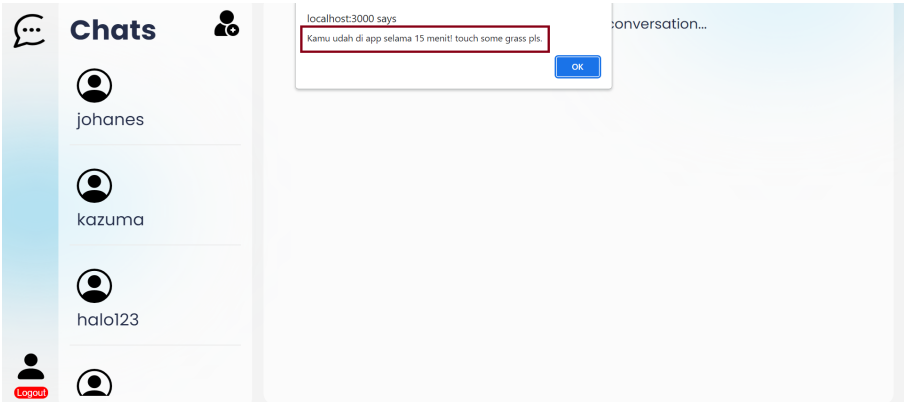
1. Pengguna aplikasi SocialLife harus terhubung ke internet yang memadai ketika menjalankan aplikasi.
2. Browser yang digunakan untuk mengakses SocialLife versi web harus sudah mendukung JavaScript ES6 (2015).
3. SocialLife versi Beta seharusnya dirilis 3 bulan setelah versi Alpha.
4. SocialLife versi Stable dirilis 6 bulan setelah versi Beta, dengan harapan semua *bug* sudah diperbaiki.

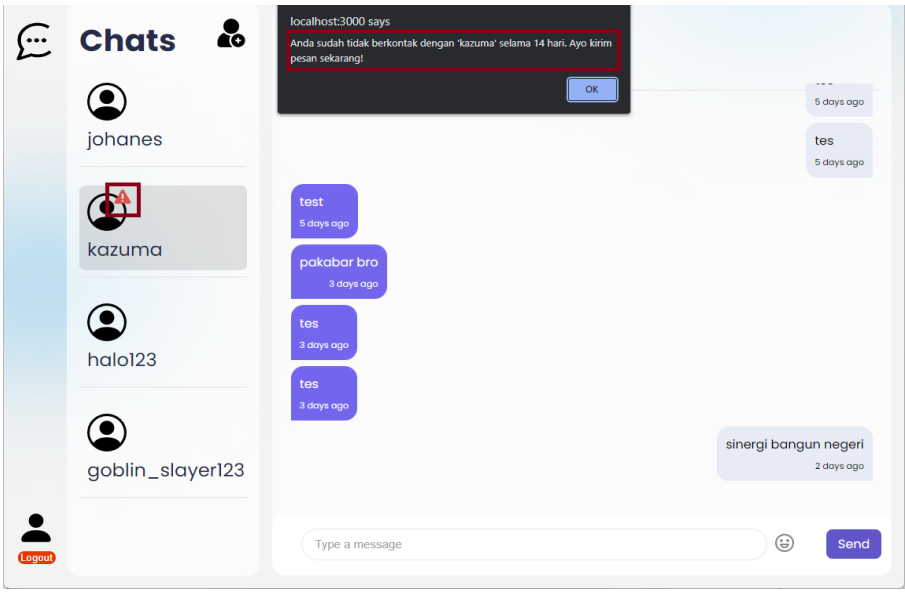
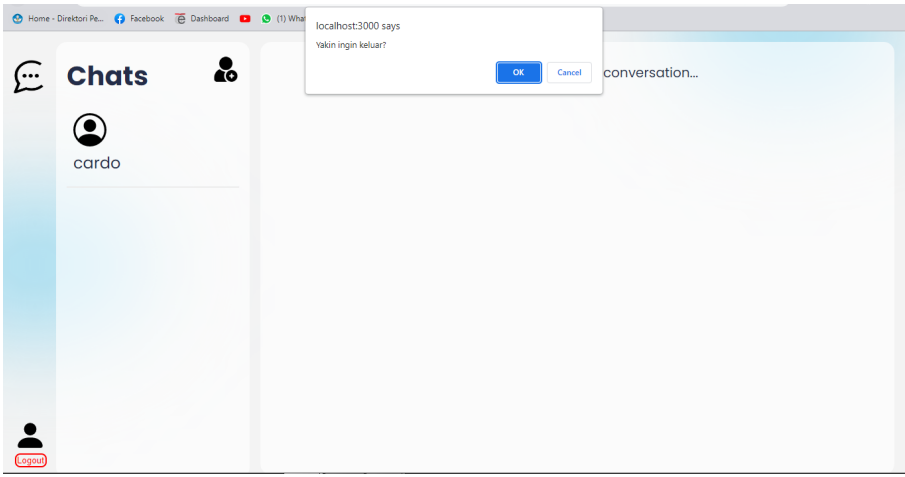


### 3. Functional Requirements

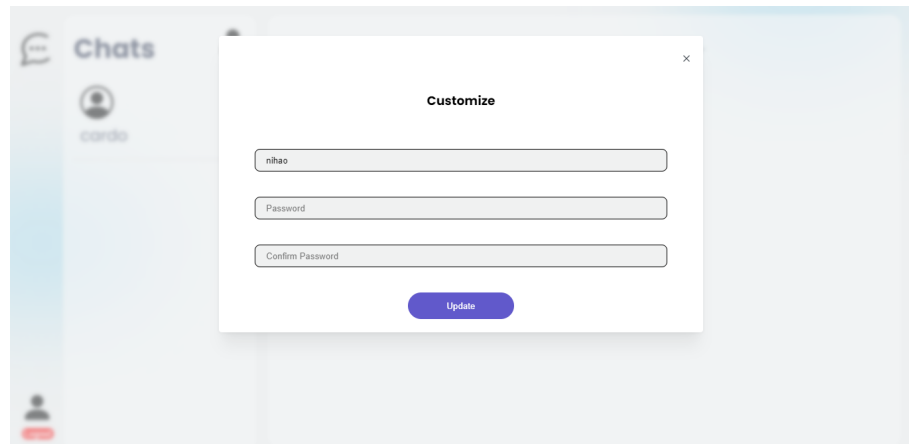
ID	Nama Requirement	Deskripsi
SKPL-P01	Register	<p>Di halaman Register <i>user</i> diminta untuk menginputkan <i>username</i>, <i>password</i> dan juga <i>confirm password</i>. Jika <i>user</i> sudah memiliki akun maka <i>user</i> dapat menekan tombol <i>login</i> sehingga akan masuk kedalam halaman <i>login</i>.</p> <p>Sketsa Interface :</p> 
SKPL-P02	Login	<p>Pada saat Login, <i>user</i> menginputkan <i>username</i> dan <i>password</i> akun untuk melanjutkan ke aplikasi. Jika <i>user</i> belum memiliki akun maka <i>user</i> akan menekan tombol <i>sign up</i> sehingga akan masuk kedalam halaman <i>register</i>.</p> <p>Sketsa Interface :</p> 

SKPL -P03	Chat	<p>Main interfacenya mempunyai 4 bagian :</p> <ul style="list-style-type: none"> <li>- AddFriend</li> <li>- Chat</li> <li>- Customization</li> <li>- Logout</li> </ul> <p>Sketsa Interface :</p> 
SKPL -P04	AddFriend	<p>Pada bagian addFriend, akan ditampilkan <i>username</i> yang mana <i>username</i> yang diisi adalah <i>username</i> teman kita lalu kita akan menekan tombol <i>search</i>.</p> <p>Sketsa Interface :</p>  <p>Setelah <i>user</i> menekan tombol <i>search</i> maka akan muncul <i>alert</i> 'Yakin ingin menambahkan '<i>username</i>' sebagai teman kamu'.</p>

		<p>Sketsa Interface :</p> 
SKPL-P05	Send Alert	<p>Jika <i>user</i> telah menggunakan aplikasi selama 15 menit maka akan muncul notifikasi bahwa <i>user</i> telah terlalu lama memakai aplikasi.</p> <p>Sketsa Interface :</p> 
SKPL-P06	Remind to Contact	<p>Jika <i>user</i> tidak pernah menghubungi suatu teman selama 14 hari maka <i>user</i> akan dikirimkan notifikasi bahwa <i>user</i> tidak pernah mengontak temannya.</p> <p>Sketsa Interface :</p>

		
SKPL-P07	Log Out	<p>User bisa <i>log out</i> dari aplikasi dengan cara menekan tombol <i>logout</i> lalu akan muncul <i>alert</i> ‘Yakin ingin keluar?’. Jika kita menekan tombol ‘OK’ maka akan keluar. Namun, jika menekan tombol ‘Cancel’ maka akan tetap di aplikasi.</p> <p>Sketsa Interface :</p> 
SKPL-P08	Customize	<p>Pada bagian gambar diatas <i>logout</i> jika di tekan maka kita akan masuk ke halaman <i>customize</i> yang dimana <i>user</i> dapat mengganti <i>username</i> dan juga <i>passwordnya</i>.</p>

Sketsa Interface :



## 4. Fitur Sistem (Use Cases)

### 4.1 Use Case 1

4.1.1 Nama Use Case  
Register

4.1.2 Tujuan

Mendaftarkan akun *user* ke *database*.

4.1.3 Input

Akun *user* yang akan ditambahkan.

4.1.4 Output

*System* akan menambahkan akun *user* ke *database*.

4.1.5 Skenario Utama

*User* mendaftarkan akun lewat *form* yang disediakan oleh *system*.

4.1.6 Prakondisi

*User* ingin melakukan pendaftaran akun.

4.1.7 Langkah-langkah :

Langkah 1 : *User* membuka aplikasi.

Langkah 2 : *User* memilih menu registrasi.

Langkah 3 : *System* akan menampilkan *form* registrasi.

Langkah 4 : *User* memberikan data registrasi.

Langkah 5 : *System* menyimpan data *user* ke *database*.

Langkah 6 : *System* selesai melakukan registrasi.

4.1.8 Pascakondisi

*User* berhasil terdaftar, akun tersimpan di *database*.

4.1.9 Skenario eksepsional 1

*User* gagal registrasi.

#### 4.1.10 Contoh

Setelah melakukan registrasi data *user* tidak masuk ke dalam *database*.

## 4.2 Use Case 2

### 4.2.1 Nama Use Case

Log In

### 4.2.2 Tujuan

Memasukan akun *user*.

### 4.2.3 Input

Akun *user* yang akan dimasukkan.

### 4.2.4 Output

*System* akan memeriksa akun *user* yang ada di *database*.

### 4.2.5 Skenario Utama

*User* memasukan data akun yang disediakan *system*.

### 4.2.6 Prakondisi

*Username* dan *password user* sudah terdaftar.

### 4.2.7 Langkah-langkah

Langkah 1 : *User* menekan tombol *login* yang tersedia.

Langkah 2 : *System* menampilkan halaman *login*.

Langkah 3 : *User* memasukan *username* dan *password*.

### 4.2.8 Pascakondisi

*System* menerima *username* dan *password* dari *user* kemudian melakukan validasi ke *database* dan *system* akan merespon dengan memunculkan halaman dashboard.

### 4.2.9 Skenario eksepsional 1

*User* gagal login.

#### 4.2.10 Contoh

*User* tidak memasukan *username* dan *password* sesuai dengan yang tersimpan di *database*.

### 4.3 Use Case 3

#### 4.3.1 Nama Use Case

Log Out

#### 4.3.2 Tujuan

*User* mengeluarkan akun dari aplikasi.

#### 4.3.3 Input

-

#### 4.3.4 Output

*User* berhasil mengeluarkan akun dari aplikasi.

#### 4.3.5 Skenario Utama

*User* menekan tombol *log out*, kemudian akun akan dikeluarkan dalam aplikasi.

#### 4.3.6 Prakondisi

*User* ingin mengeluarkan akun dari aplikasi.

#### 4.3.7 Langkah-langkah

Langkah 1 : *User* menekan tombol log out.

Langkah 2 : *System* mengkonfirmasi apakah *user* ingin log out.

Langkah 3 : *System* mengeluarkan akun *user* dari aplikasi.

#### 4.3.8 Pascakondisi

*User* berhasil mengeluarkan akun dari aplikasi.

#### 4.3.9 Skenario eksepsional 1

*User* gagal log out.

#### 4.3.10 Contoh



*User* kehilangan jaringan internet ketika sedang mencoba untuk *log out*.

## 4.4 Use Case 4

### 4.4.1 Nama Use Case

Chat

### 4.4.2 Tujuan

*User* dapat mengirim pesan ke teman yang sudah ditambahkan *user*.

### 4.4.3 Input

Isi text.

### 4.4.4 Output

*System* mengirim isi *text* tersebut ke teman *user*.

### 4.4.5 Skenario Utama

*User* mengklik *chat bubble* teman *user*, kemudian mengisi pesan yang ingin di kirim di dalam chat box untuk dikirim.

### 4.4.6 Prakondisi

*User* sudah mengetik isi *text* di dalam chat box.

### 4.4.7 Langkah-langkah

Langkah 1 : *User* membuka chat window teman *user*.

Langkah 2 : *User* mengisi kolom chat.

Langkah 3 : *User* menekan tombol *send*.

Langkah 4 : Sistem mengirim *text* ke teman *user*.

### 4.4.8 Pascakondisi

*User* berhasil mengirim *text*.

### 4.4.9 Skenario eksepsional 1

*User* gagal memberikan respon.

### 4.4.10 Contoh

*User* kehilangan jaringan internet ketika sedang mengirimkan respon.

## 4.5 Use Case 5

### 4.5.1 Nama Use Case:

Add Friend

### 4.5.2 Tujuan

*User* dapat mencari teman melalui *username*.

### 4.5.3 Input

*Username* teman *user*.

### 4.5.4 Output

*System* menambahkan teman.

### 4.5.5 Skenario Utama

*User* ingin mencari teman *user* yang baru, kemudian *user* akan memasukkan *username* teman *user* di *search engine*. Jika ada maka *user* akan men-add teman dengan suatu tombol.

### 4.5.6 Prakondisi

Mencari teman melalui *username*.

### 4.5.7 Langkah-langkah

Langkah 1 : *User* mengetik *username*.

Langkah 2 : *System* memunculkan orang yang dicari dari *username*.

Langkah 3 : *User* mengklik tambah teman.

Langkah 4 : *System* menambahkan teman.

### 4.5.8 Pascakondisi

*User* sukses mencari teman melalui *username*.

### 4.5.9 Skenario eksepsional 1

*User* tidak menemukan *username* teman tersebut.

### 4.5.10 Contoh

Ketika *user* mencari teman melalui *username* tidak ditemukan di *database*.

## 4.6 Use Case 6

### 4.6.1 Nama Use Case

Send Alert

### 4.6.2 Tujuan

Memberikan peringatan terhadap *user* karena sudah menggunakan aplikasi selama 15 menit.

### 4.6.3 Input

-

### 4.6.4 Output

*System* memberikan peringatan ke *user*.

### 4.6.5 Skenario Utama

*User* akan diberikan peringatan jika sudah membuka aplikasi selama 15 menit.

### 4.6.6 Prakondisi

*User* telah membuka aplikasi selama 15 menit.

### 4.6.7 Langkah-langkah

Langkah 1 : *System* memberikan peringatan kepada *user*.

Langkah 2 : *User* menerima pesan dan menutup aplikasi.

### 4.6.8 Pascakondisi

*System* berhasil mengirim peringatan kepada *user*.

### 4.6.9 Skenario eksepsional 1

*System* tidak berhasil mengirim peringatan pada *user*.

### 4.6.10 Contoh

Adanya kesalahan *system* ketika pengiriman peringatan.

## 4.7 Use Case 7

### 4.7.1 Nama Use Case

Remind to Contact

### 4.7.2 Tujuan

Mengingatkan *user* untuk menghubungi temannya yang belum di chat selama 14 hari atau lebih.

### 4.7.3 Input

-

### 4.7.4 Output

*System* menampilkan peringatan.

### 4.7.5 Skenario Utama

*User* akan mendapat notifikasi pengingat / *reminder* bahwa *user* belum mengontak suatu teman selama 14 hari.

### 4.7.6 Prakondisi

*User* tidak menghubungi teman selama 14 hari.

### 4.7.7 Langkah-langkah

Langkah 1 : *System* memberikan peringatan kepada *user* untuk menghubungi teman.

Langkah 2 : *User* menerima peringatan.

Langkah 3 : *User* mengirimkan pesan terhadap teman yang sudah lama tidak dihubungi.

Langkah 4 : *System* dapat mengirim pesan.

### 4.7.8 Pascakondisi

*System* berhasil memperingati *user* agar menghubungi temannya.

### 4.7.9 Skenario eksepsional 1

*System* gagal memberikan peringatan terhadap *user*.

### 4.7.10 Contoh

Adanya kesalahan *system* ketika pengiriman peringatan.

## 4.8 Use Case 8

### 4.8.1 Nama Use Case

Customization

### 4.8.2 Tujuan

Mengubah *username* dan / atau *password*.

### 4.8.3 Input

-

### 4.8.4 Output

*User* mengubah *username* dan / atau *password*.

### 4.8.5 Skenario Utama

*User* mengklik gambar yang ada diatas *logout*.

### 4.8.6 Prakondisi

*User* menekan tombol *customize*.

### 4.8.7 Langkah-langkah

Langkah 1 : *User* membuka menu *customize*.

Langkah 2 : *User* mengubah *username* dan / atau *passwordnya*.

Langkah 3 : *System* menanyakan apakah *user* yakin jika ingin merubah *username* dan / atau *passwordnya*.

Langkah 4 : *System* mengubah pengaturan sesuai perubahan yang dilakukan oleh *user*.

### 4.8.8 Pascakondisi

*System* berhasil mengubah *username* dan / atau *password*.

### 4.8.9 Skenario eksepsional 1

*System* gagal merubah *username* dan / atau *password*.

### 4.8.10 Contoh

Adanya kesalahan dalam perubahan *username* dan / atau *password* atau koneksi yang putus.

## 5. Requirements Non Fungsional

ID	Parameter	Kebutuhan
SKPL-N01	Performance	<i>Software</i> bisa mengirim pesan hampir seketika sehingga <i>user</i> bisa membaca pesan tanpa harus menunggu.
SKPL-N02	Performance	<i>Software</i> tidak memakai lebih dari 10% dari RAM ketika pemakaian berlangsung agar tidak terlalu membebani <i>device user</i> .
SKPL-N03	Performance	<i>Software startup</i> tidak melebihi dari 5 detik agar <i>user</i> tidak terlalu lama dalam menunggu startup app.
SKPL-N04	Response time	<i>Software response time</i> seperti menekan chat dan meng-input kalimat tidak melebihi 3 detik.
SKPL-N07	Security	<i>User</i> diharuskan untuk mempunyai akun agar menghindari <i>spam</i> , dan mencegah adanya penyalahgunaan aplikasi.
SKPL-N09	Portability	<i>Software</i> dibuat dengan tujuan untuk bisa dipasarkan sebagai <i>web-based application</i> , sehingga bisa terjangkau oleh berbagai kalangan.
SKPL-N10	Usability	<i>Software</i> mempunyai <i>User Interface</i> yang nyaman bagi <i>user</i> . <i>User interface</i> nya harus menarik, mudah untuk dipelajari, konsisten, ramah untuk segala umur, dan tidak membingungkan.

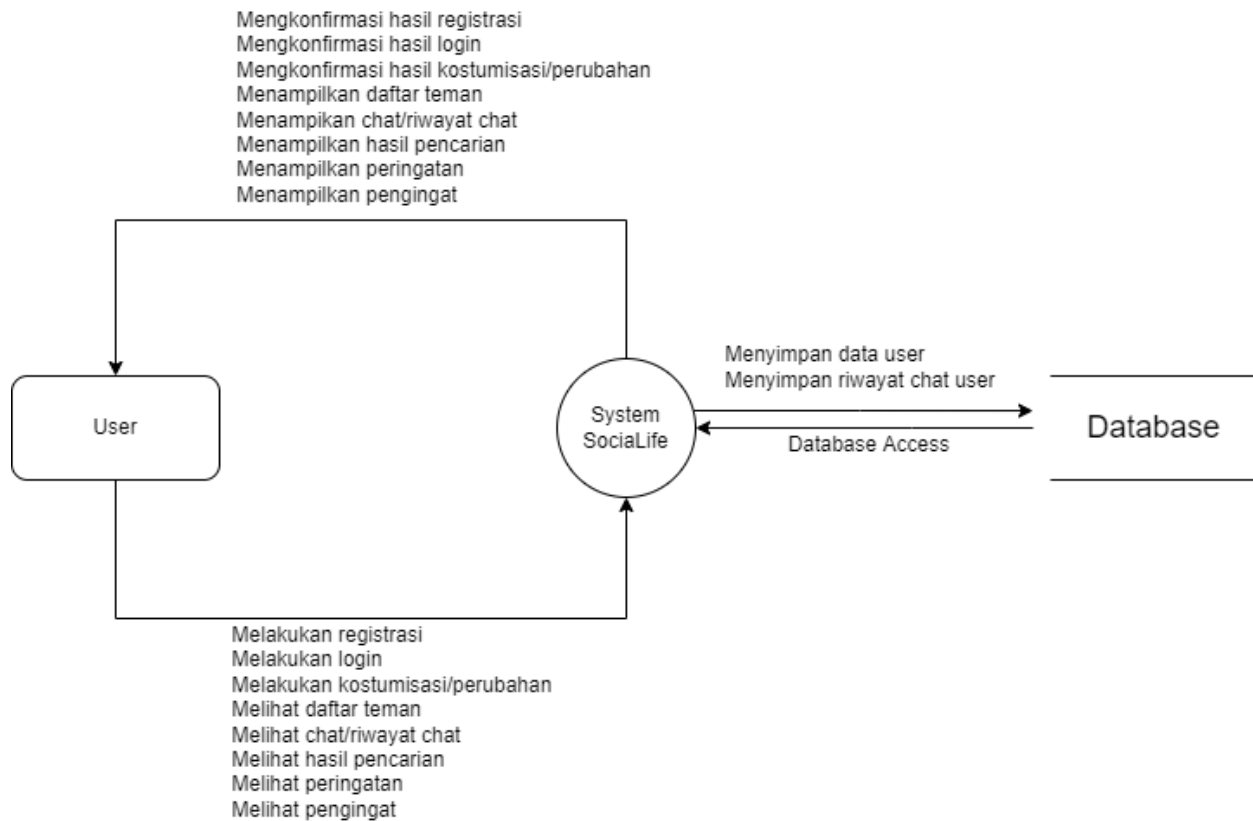
## Lampiran A: Daftar Kata-Kata Sukar

Kata Sukar	Keterangan
CAPTCHA	Jenis tindakan keamanan yang dikenal sebagai autentikasi tantangan-tanggapan.
Encryption	Proses mengamankan suatu informasi dengan membuat informasi tersebut tidak dapat dibaca tanpa bantuan pengetahuan khusus.
Log In	Sebuah proses untuk masuk ke dalam sebuah layanan <i>online</i> dengan memasukkan kode identitas pengguna (ID dan Password).
Log Out	Sebuah proses untuk mengakhiri koneksi.
OS	<i>Operating System</i> merupakan perangkat lunak yang berfungsi sebagai penghubung antara pengguna dan perangkat keras.
Password	Sebuah kata sandi atau kode rahasia yang digunakan untuk masuk ke dalam sebuah akun.
RAM	<i>Random Access Memory</i> merupakan sebuah perangkat penyimpanan data.
Register	Pendaftaran terhadap suatu program yang berfungsi untuk menghubungkan data pribadi dengan program tersebut.
User	Pemakai atau pengguna sebuah aplikasi.

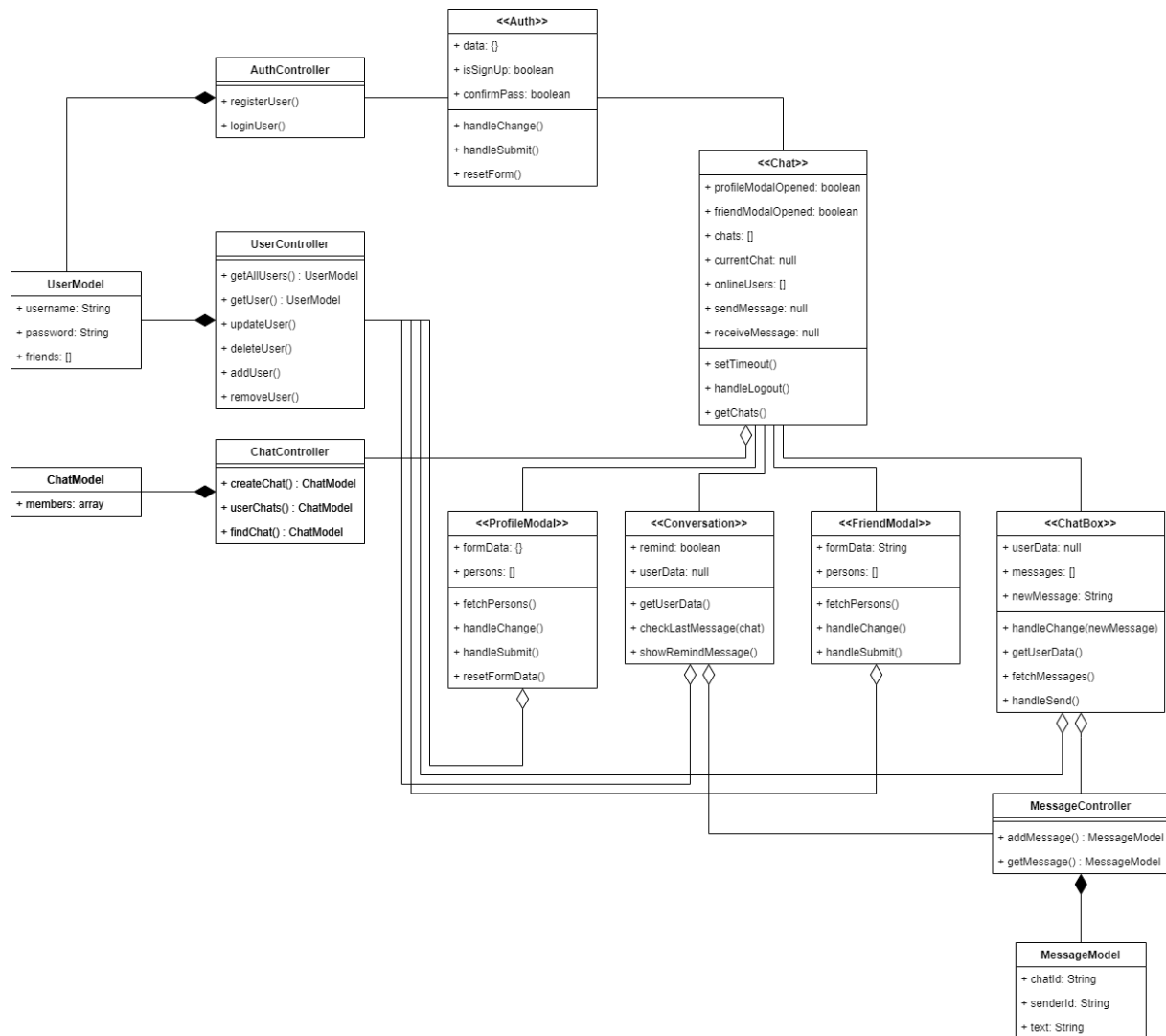


## Lampiran B: Diagram

### 1. Data Flow Diagram



## 2. Class Diagram



### 3. Use Case Diagram

