

# **USER MANUAL**

## **SOCIAL LIFE**

Dibuat Oleh :

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



**Program Studi Teknik Informatika**

**Fakultas Informatika Universitas Telkom**

**2022**

## **DAFTAR ISI**

<b>DAFTAR ISI</b>	<b>2</b>
<b>1. Pendahuluan</b>	<b>4</b>
<b>2. Memulai Aplikasi</b>	<b>4</b>
<b>3. Panduan untuk Pengguna</b>	<b>5</b>
3.1 Sign Up	5
3.2 Login	7
3.3 Add Friend	10
3.4 Chat	12
3.5 Customization	14
3.6 Logout	16
3.7 Send Alert	16
3.8 Remind to Contact	16
<b>Halaman Utama</b>	<b>18</b>
<b>Penjelasan Bagian Side-Bar</b>	<b>19</b>
<b>Penjelasan Bagian Chats</b>	<b>21</b>
<b>Penjelasan Bagian ChatBox</b>	<b>23</b>



## 1. Pendahuluan

SociaLife merupakan sebuah aplikasi chat *real-time* berbasis website yang dibuat dengan MERN Stack dengan bantuan *socket.io* untuk kemudahan real-time chat antar user. Dengan aplikasi ini, user dapat melakukan register, login, menambah teman, mengupdate data seperti username dan password, dan tentunya chat dengan teman. Hal yang membedakan dari aplikasi chat lainnya ialah, aplikasi ini memiliki 2 fitur yang unik, yaitu peringatan pemakaian jika sudah 15 menit dalam aplikasi (*alert*) dan pengingat untuk men-chat teman jika sudah tidak chat selama 14 hari (*reminder*).

## 2. Memulai Aplikasi

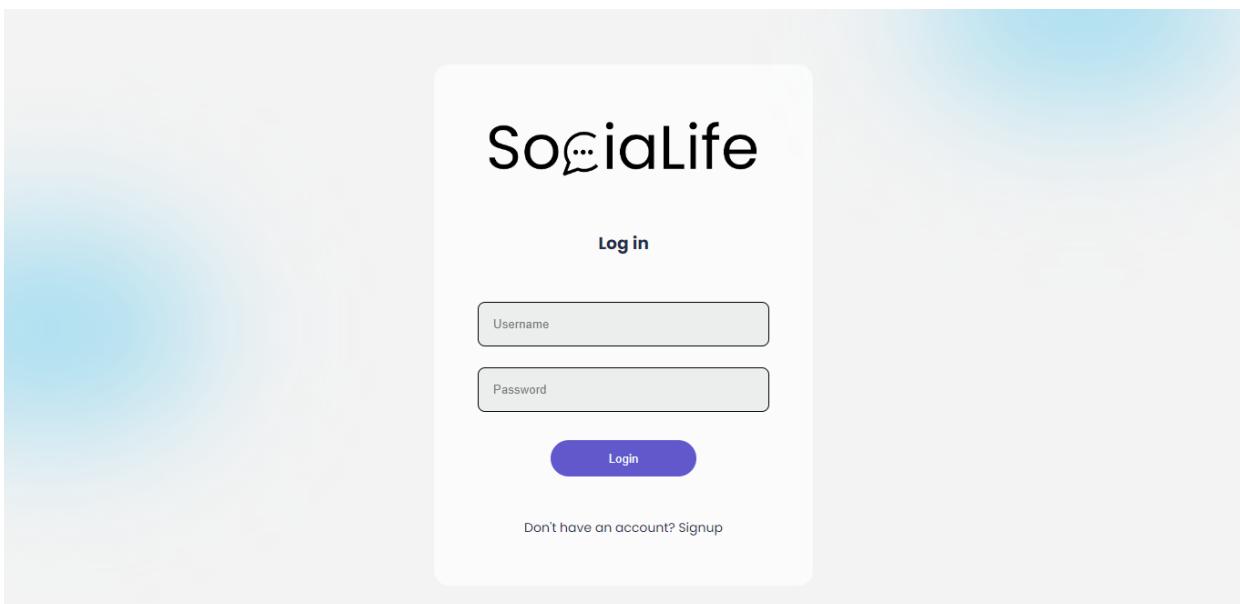
Sebelum memulai aplikasi, ada beberapa hal yang perlu dilakukan terlebih dahulu di komputer pengguna, yaitu:

### Persyaratan:

1. Git
2. NodeJS
3. NPM
4. Visual Studio Code

### Instalasi:

1. Clone project SociaLife (git clone <https://github.com/mufidu/socialife>) ke folder yang diinginkan
2. Masuk ke dalam direktori tersebut (cd socialife) dan buka dengan Visual Studio Code
3. Buat file baru bernama ‘.env’ dalam folder server dan copy datanya dari file ‘.env.example’ ke ‘.env’
4. Buat 3 terminal baru untuk masing-masing folder (client, server, socket) untuk instalasi dependensi dan penjalannya
5. Pada terminal pertama, masuk ke folder client (cd client) dan install dependensi dengan mengetik ‘npm install’
6. Pada terminal kedua, masuk ke folder server (cd server) dan install dependensi dengan mengetik ‘npm install’
7. Pada terminal ketiga, masuk ke folder socket(cd socket) dan install dependensi dengan mengetik ‘npm install’
8. Jalankan ketiga folder tersebut (terminal client, terminal server, terminal socket) dengan mengetik ‘npm start’ di terminal masing-masing
9. Akan terdapat indikasi bahwa ‘npm start’ berhasil di masing-masing terminal. Pada socket akan muncul ‘Connected Users’, pada server akan muncul ‘Listening at 5000’, dan pada client akan muncul ‘Starting Web Development Server’ dan langsung membuka browser yang mengarah ke ‘localhost:3000’ dan aplikasi siap digunakan.

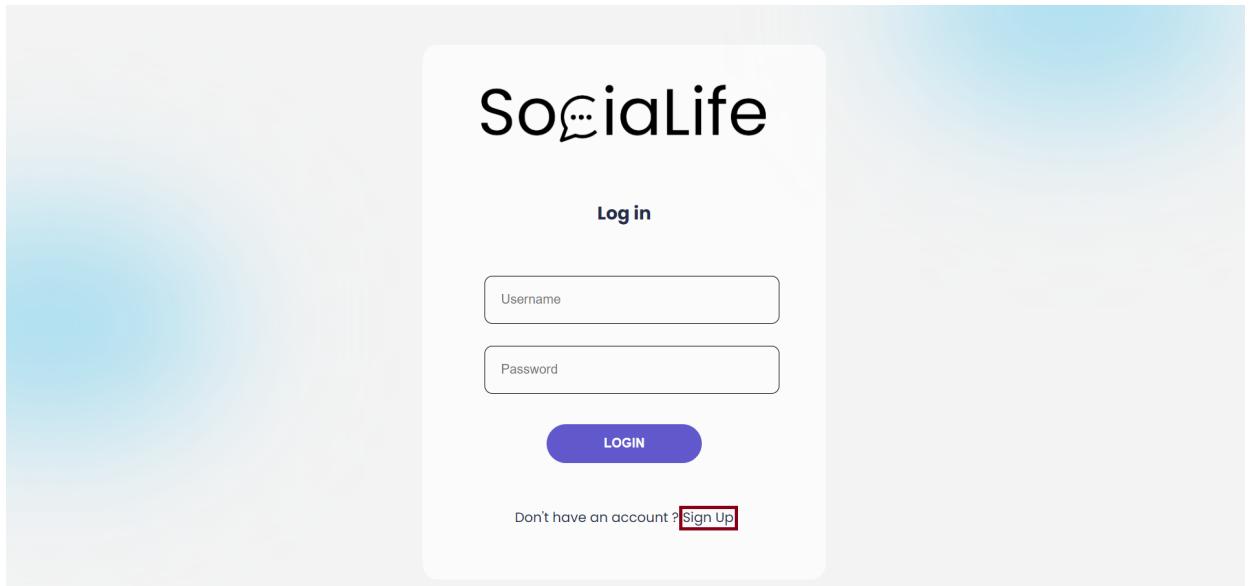


Gambar 1 : Tampilan Awal Website

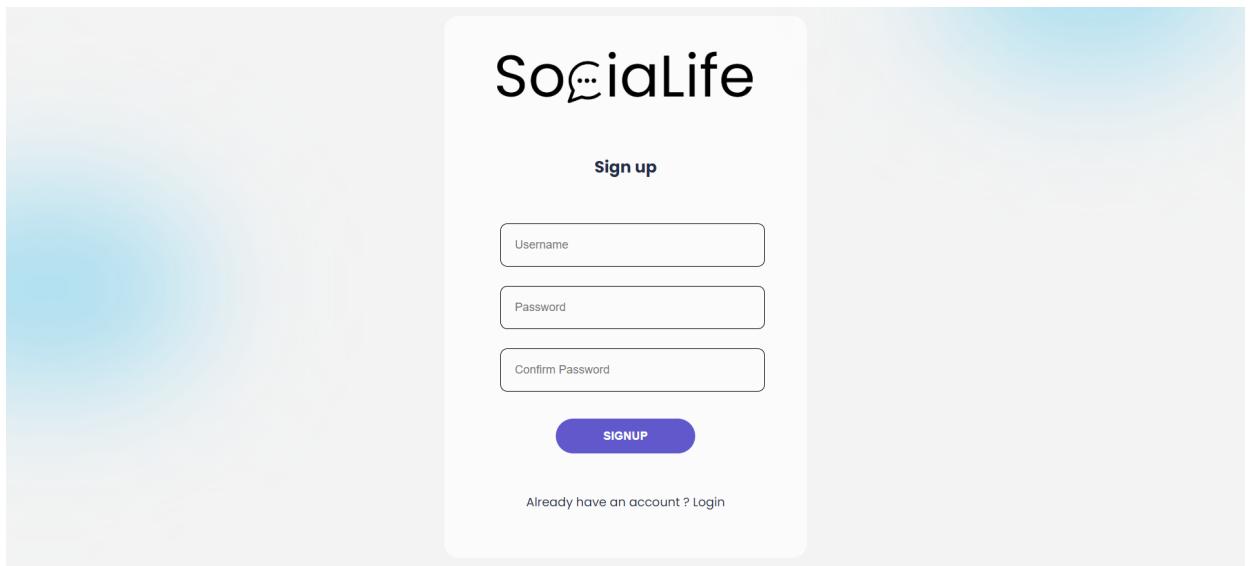
### 3. Panduan untuk Pengguna

#### 3.1 Sign Up

Pada halaman *Login*, terdapat suatu label untuk *user* yang belum memiliki dan akan membuat *account*. *User* dapat menekan label tersebut dan sistem akan menghubungkan ke halaman *Sign Up*. *Sign Up* berguna untuk menyimpan data *username* dan *password* yang dimasukkan *user* ke database *Users* sehingga *user* dapat mengakses aplikasi sesuai data yang disimpan.

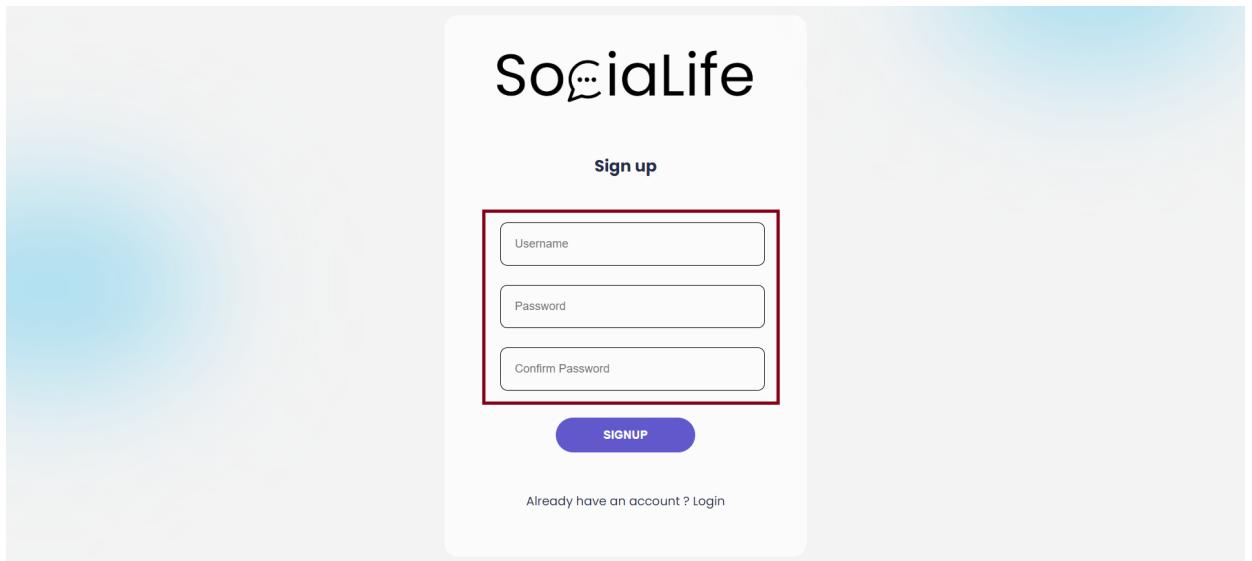


Gambar 2 : Link *Sign Up* untuk *Register Account*



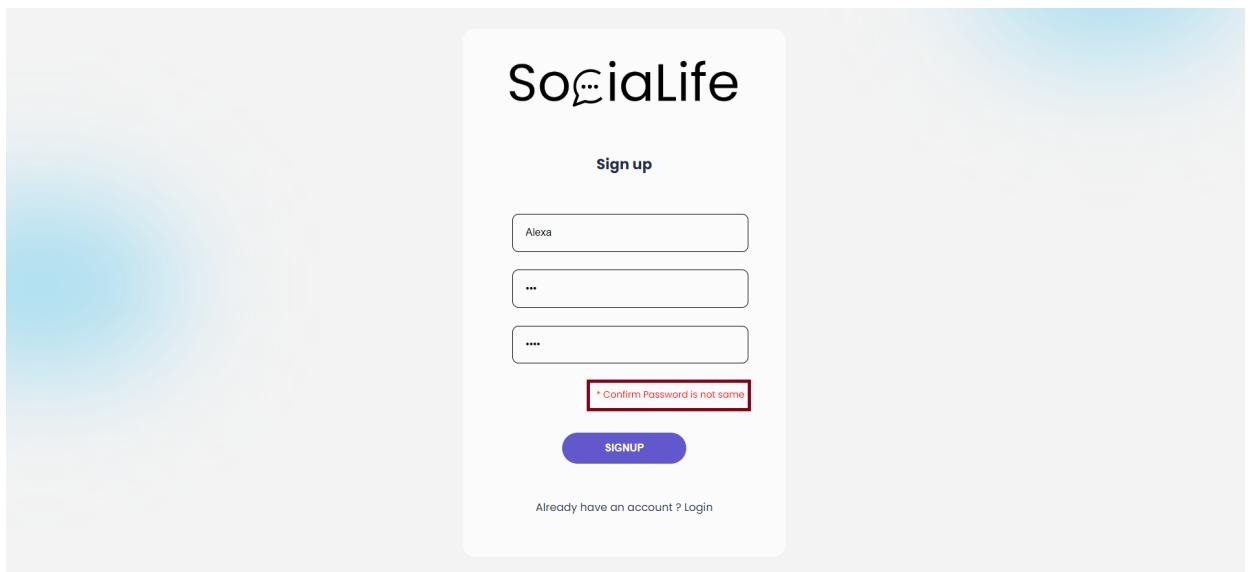
Gambar 3 : Halaman *Sign Up*

Pada halaman *Sign Up* terdapat terdapat logo SocialLife, form *Sign Up*, tombol *Sign Up*, dan link *Login*. Form *Sign Up* berisi input data yang dimiliki oleh *user*. *User* yang ingin membuat akun harus mengisi *field* data input tersebut seperti *username*, *password*, dan *confirm password*.



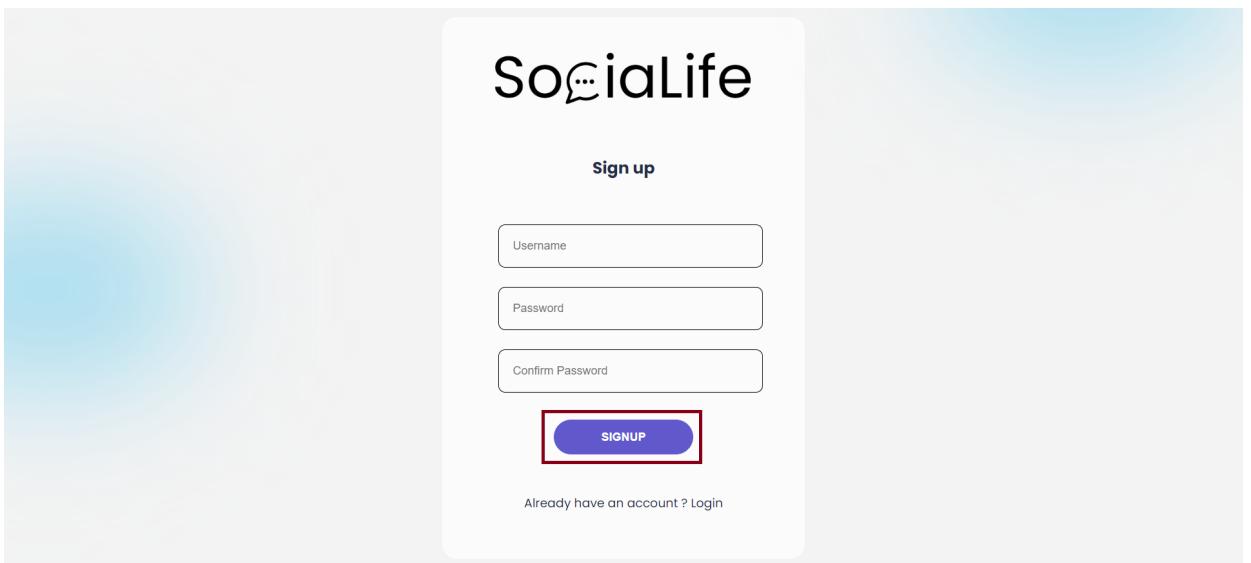
**Gambar 3 : Form *Sign Up* pada Halaman *Sign Up***

*Confirm password* harus sesuai dengan *password* yang dimasukkan sebelumnya untuk mengecek *user* tidak melakukan kesalahan. Jika tidak sesuai dengan *password* sebelumnya maka akan menampilkan tulisan “*Confirm Password is not same*”.



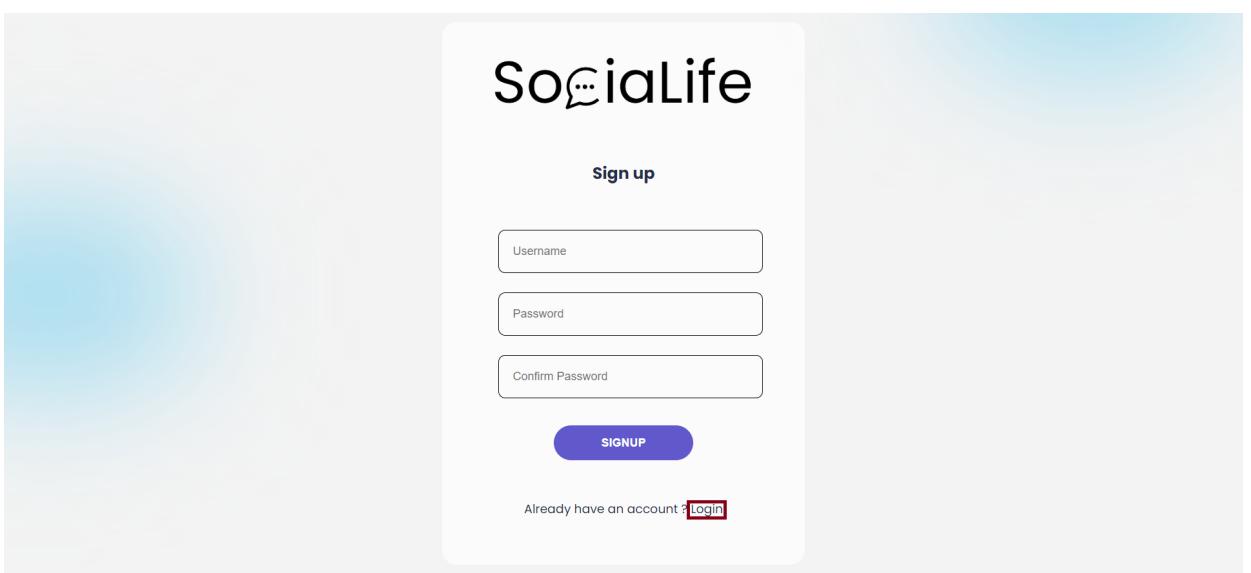
**Gambar 4 : *Confirm Password* Tidak Sesuai pada Halaman *Sign Up***

Jika semua *field* data tersebut sudah diisi maka *user* dapat menekan tombol *Sign Up* untuk mendaftarkan akun sesuai dengan data - data yang telah diisi.



**Gambar 5 : Tombol *Sign Up* pada Halaman *Sign Up***

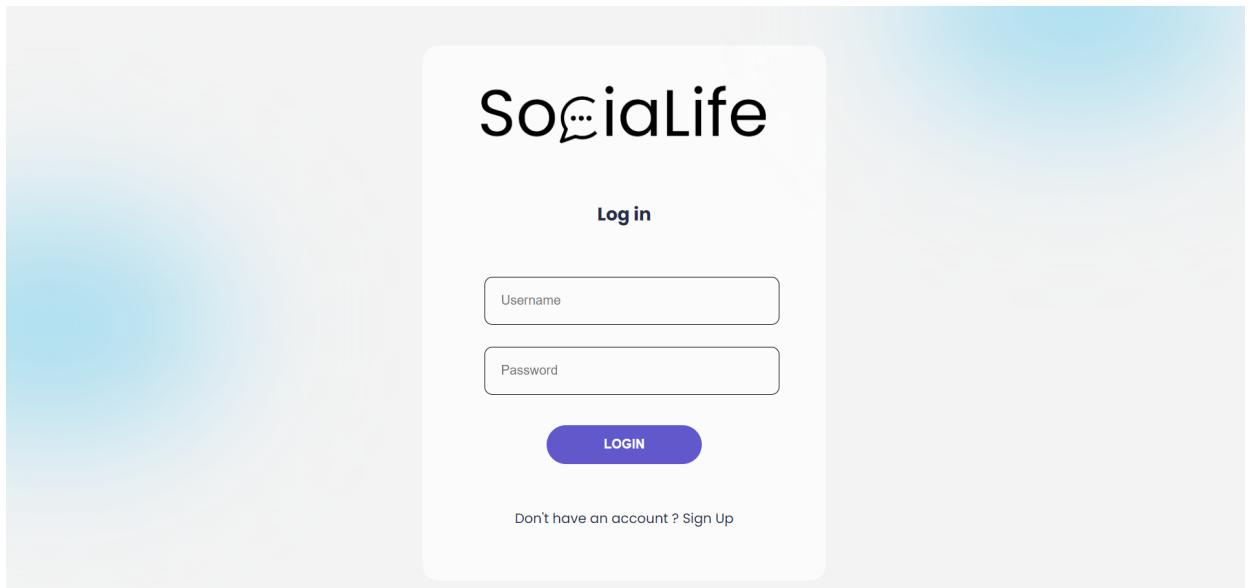
Jika *user* sudah memiliki akun dan akan melakukan *Login* maka dapat mengklik *Login*.



**Gambar 6 : Link *Login* pada Halaman *Sign Up***

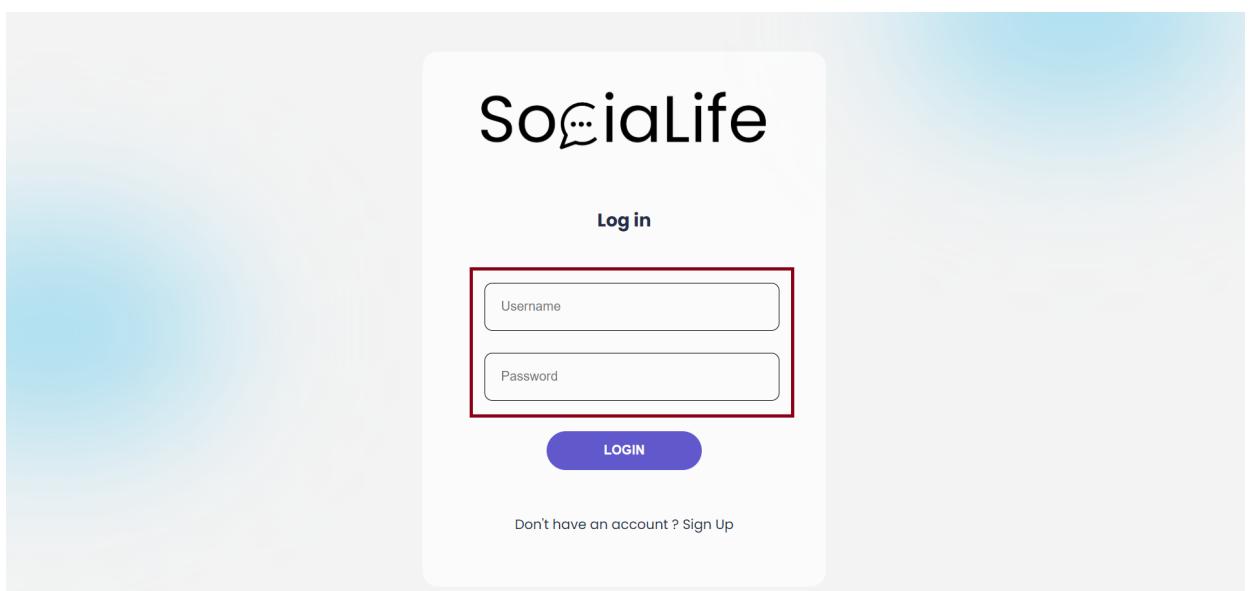
### 3.2 Login

Pada halaman *Login* terdapat logo SociaLife, form *Login*, tombol *Login*, dan link *Sign Up (Register)*. Jika *user* ingin masuk ke dalam aplikasi SociaLife maka *user* harus melakukan *login* terlebih dahulu. Namun, *user* harus sudah melakukan *register* sebelum *login* agar data *username* dan *password* tersimpan di dalam *database* *Users*.



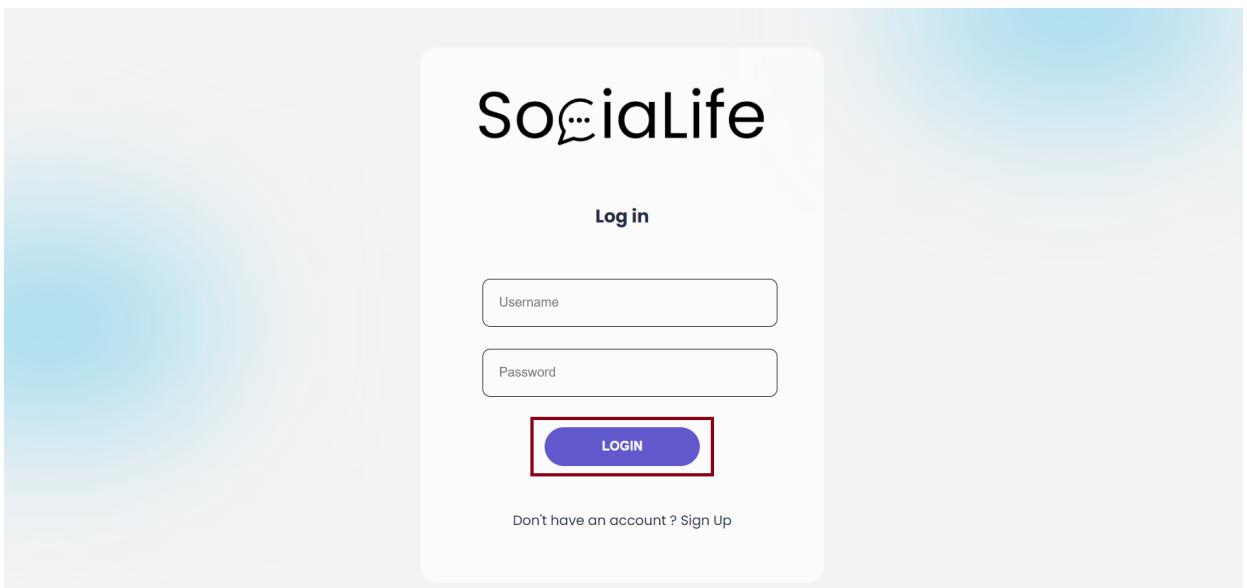
**Gambar 7 : Halaman Login**

Form *Login* berisi input data *username* dan *password* akun yang dimiliki oleh *user*. Jika *user* ingin masuk ke dalam aplikasi SociaLife maka *user* harus mengisi *field* data input tersebut.



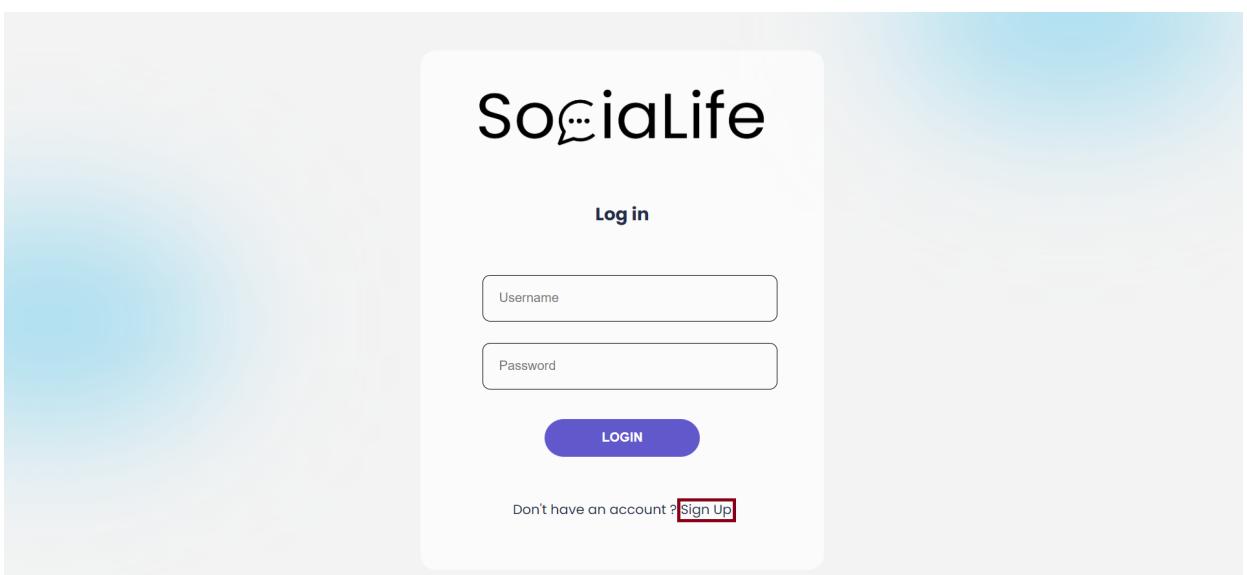
**Gambar 8 : Form *Login* pada Halaman *Login***

Jika *user* telah mengisi input *field* data *username* dan *password* yang telah teregister maka *user* dapat menekan tombol *Login* untuk mengakses akun yang dimiliki *user*.



Gambar 9 : Tombol *Login* pada Halaman *Login*

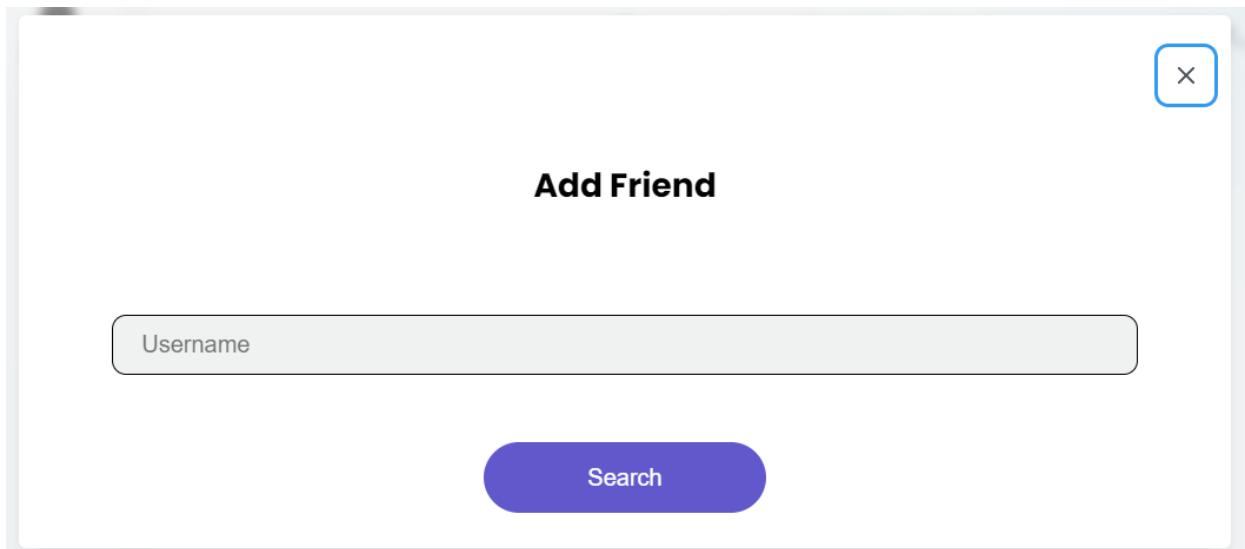
Jika *user* belum atau ingin melakukan *registrasi* maka dapat mengklik *Sign Up*.



Gambar 10 : Link *Sign Up* pada Halaman *Login*

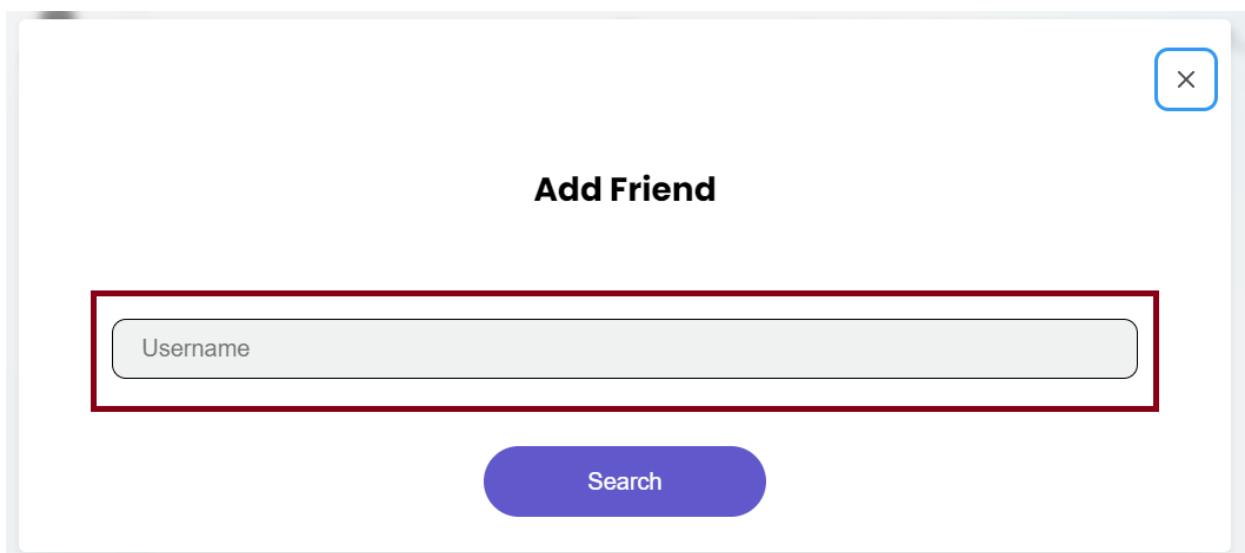
### 3.3 Add Friend

Untuk melakukan percakapan *user* dapat menambahkan teman yang ada dalam aplikasi.



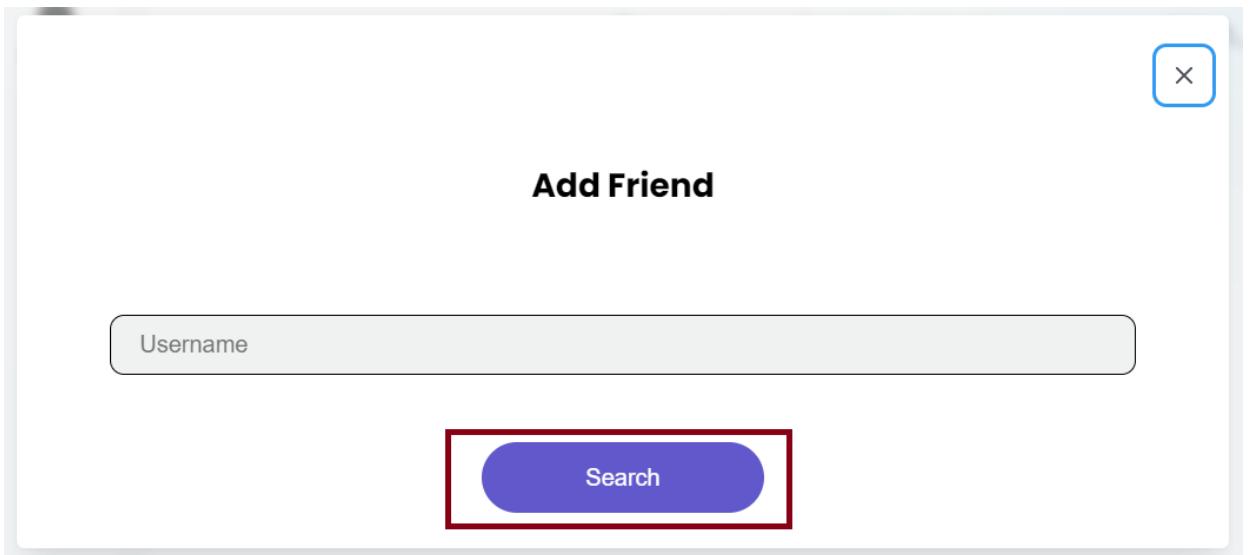
**Gambar 11 : Halaman Add Friend**

*User* mengisi *field* data input *username* teman yang sudah tercantum di dalam *database*.



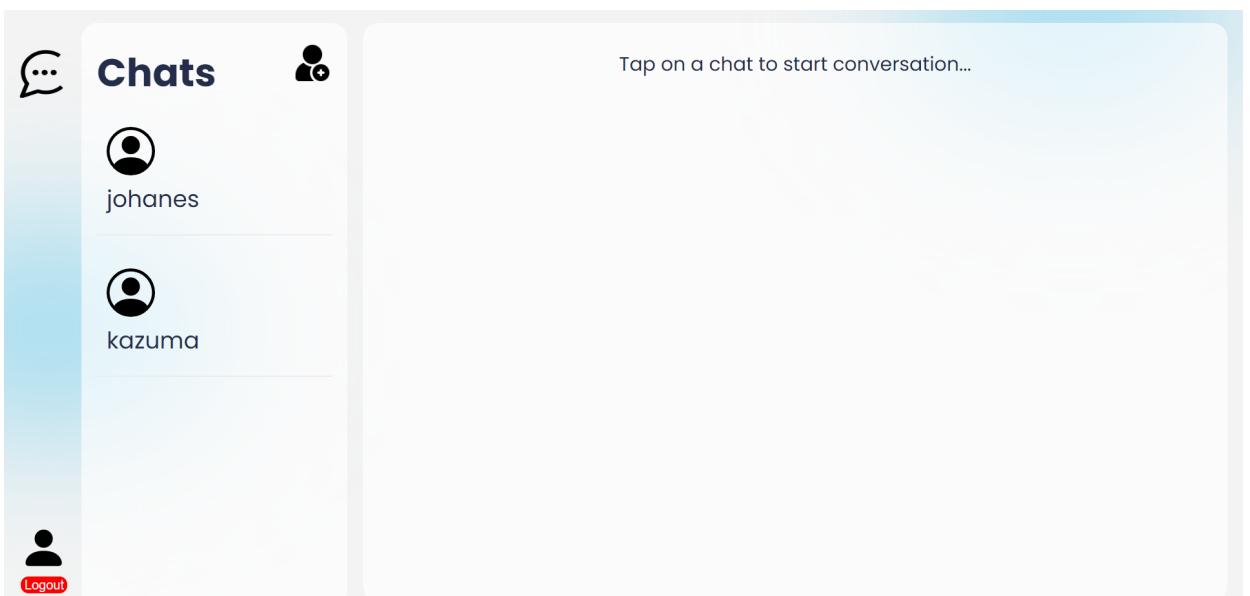
**Gambar 12 : Form Add Friend pada Halaman Add Friend**

Jika *user* telah mengisi input *field* data *username* teman maka *user* dapat menekan tombol *Search* untuk melakukan pencarian *username* teman yang sudah tercantum di dalam *database*.



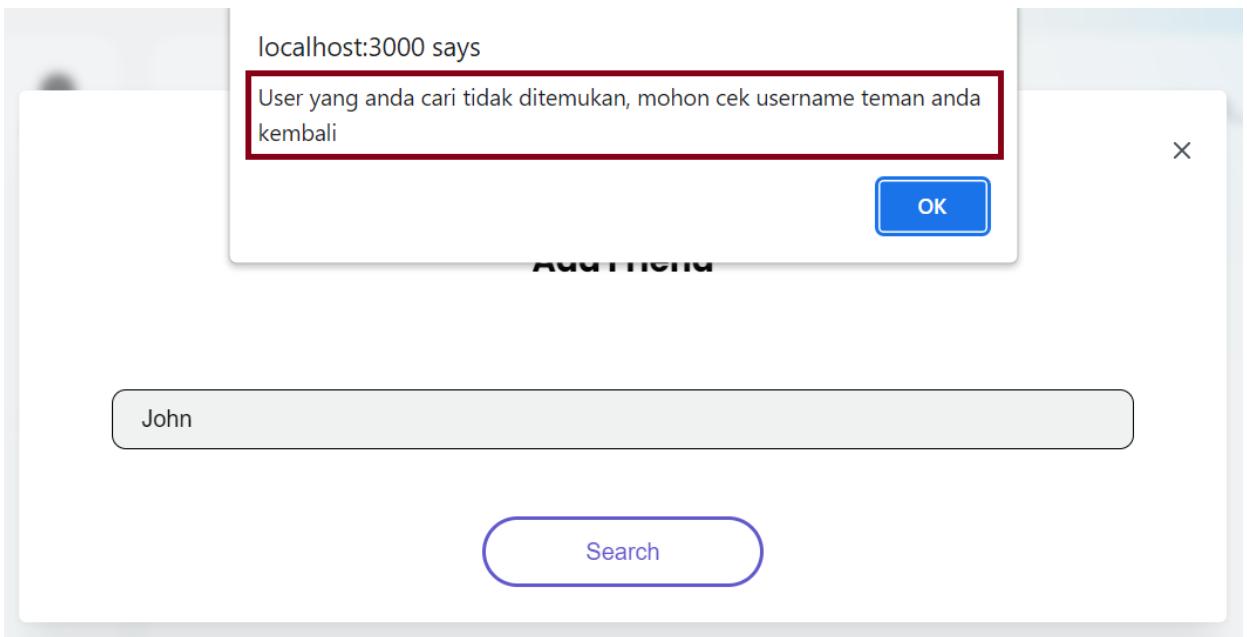
Gambar 13 : Tombol *Search* pada Halaman *Add Friend*

Apabila *username* teman sesuai tercantum dalam *database* maka akan secara langsung ditambahkan ke dalam list teman dan menampilkan tulisan “friend added successfully!!!”.



Gambar 14 : Berhasil Menambahkan Teman

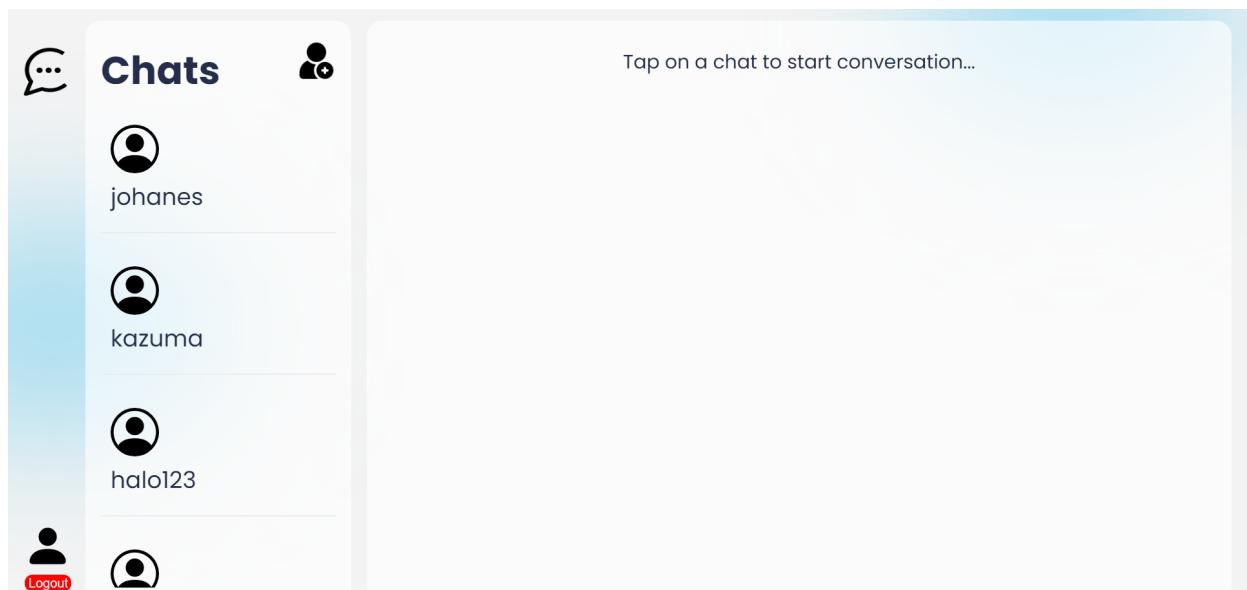
Jika *username* teman tidak sesuai dalam *database* maka akan menampilkan tulisan “User yang anda cari tidak ditemukan, mohon cek *username* teman anda kembali”.



**Gambar 15 : Gagal Menambahkan Teman**

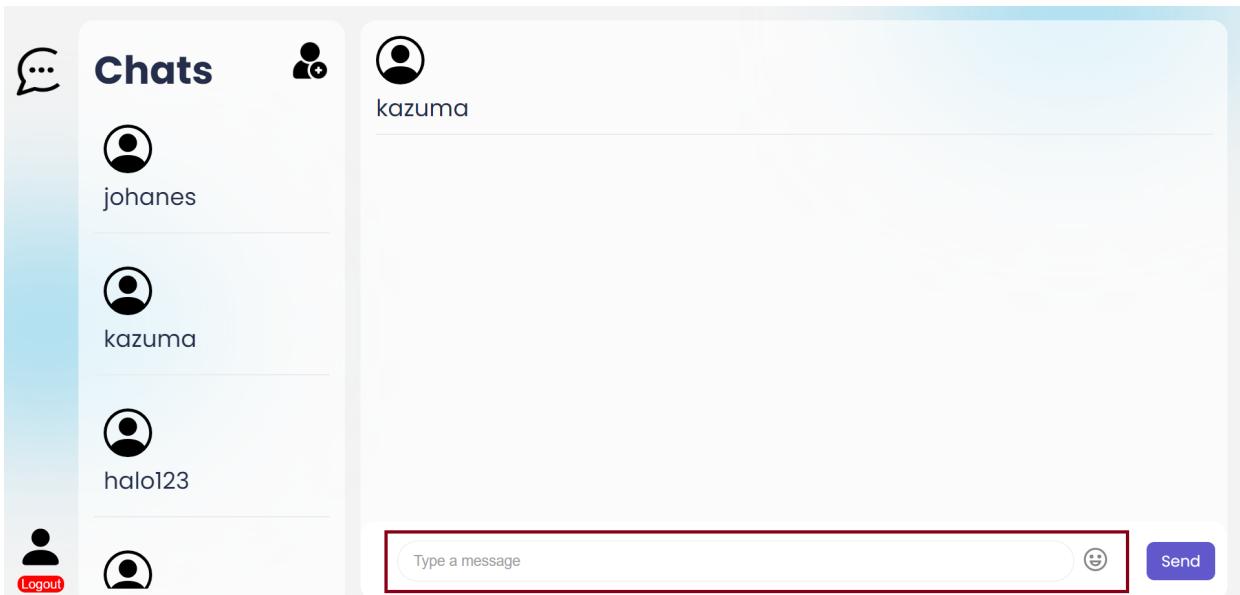
### 3.4 Chat

Pada halaman *Chat user* dapat mengirimkan pesan kepada teman yang dituju dan *user* juga dapat menerima pesan balasan dari teman tersebut, baik ketika berada di aplikasi maupun tidak berada di aplikasi.



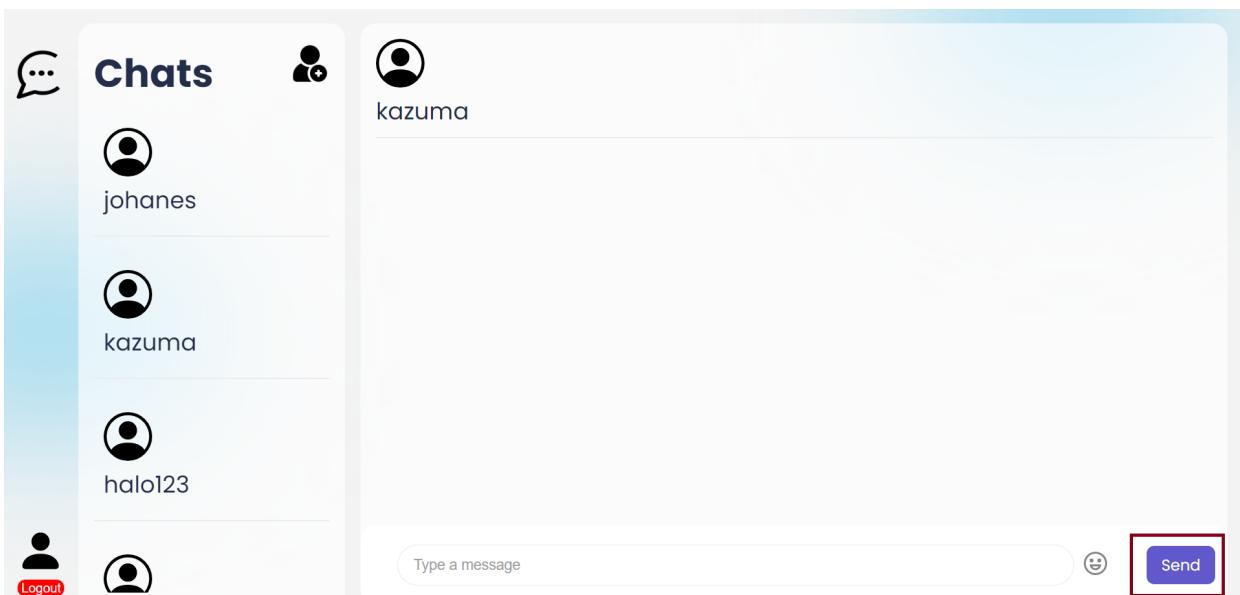
**Gambar 16 : Halaman Chat**

Untuk memulai percakapan *user* dapat mengirimkan pesan kepada teman yang dituju di dalam *field* “type a message”.

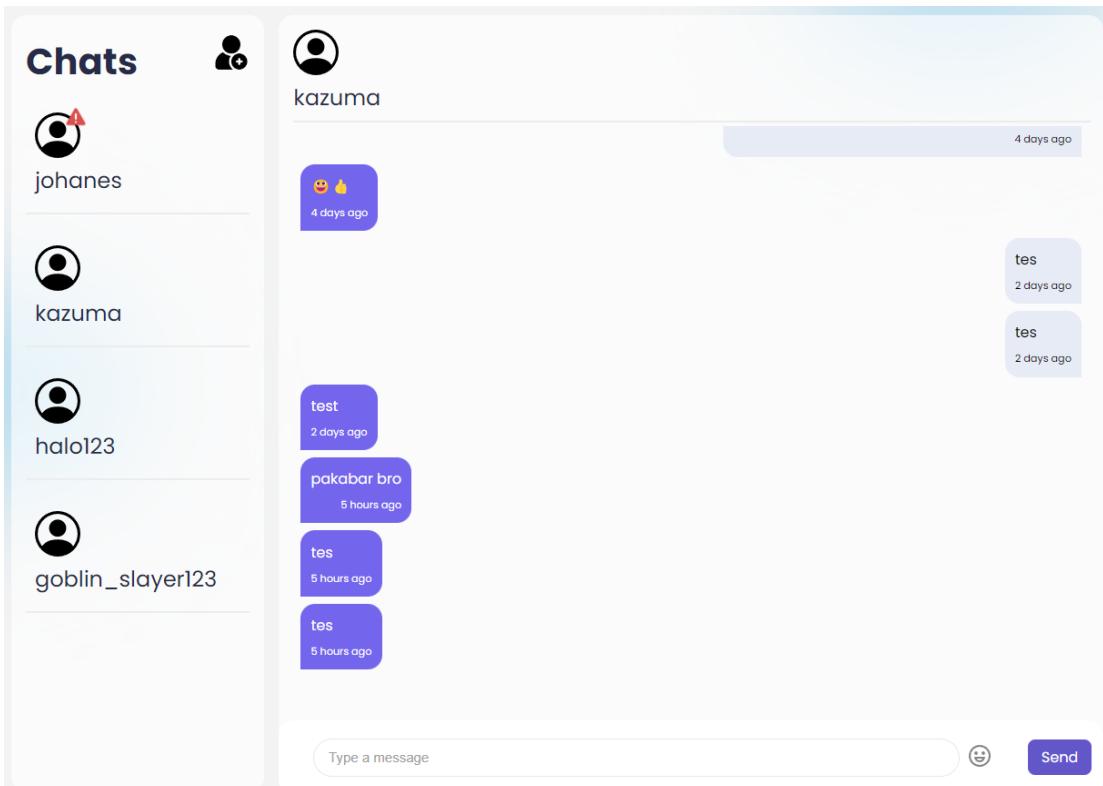


Gambar 17 : Menulis Pesan

Setelah *user* selesai mengetikan pesan, *user* dapat menekan tombol *Send* untuk mengirimkan pesan tersebut kepada teman yang dituju.



Gambar 17 : Mengirim Pesan



**Gambar 16 : Melakukan *Chatting***

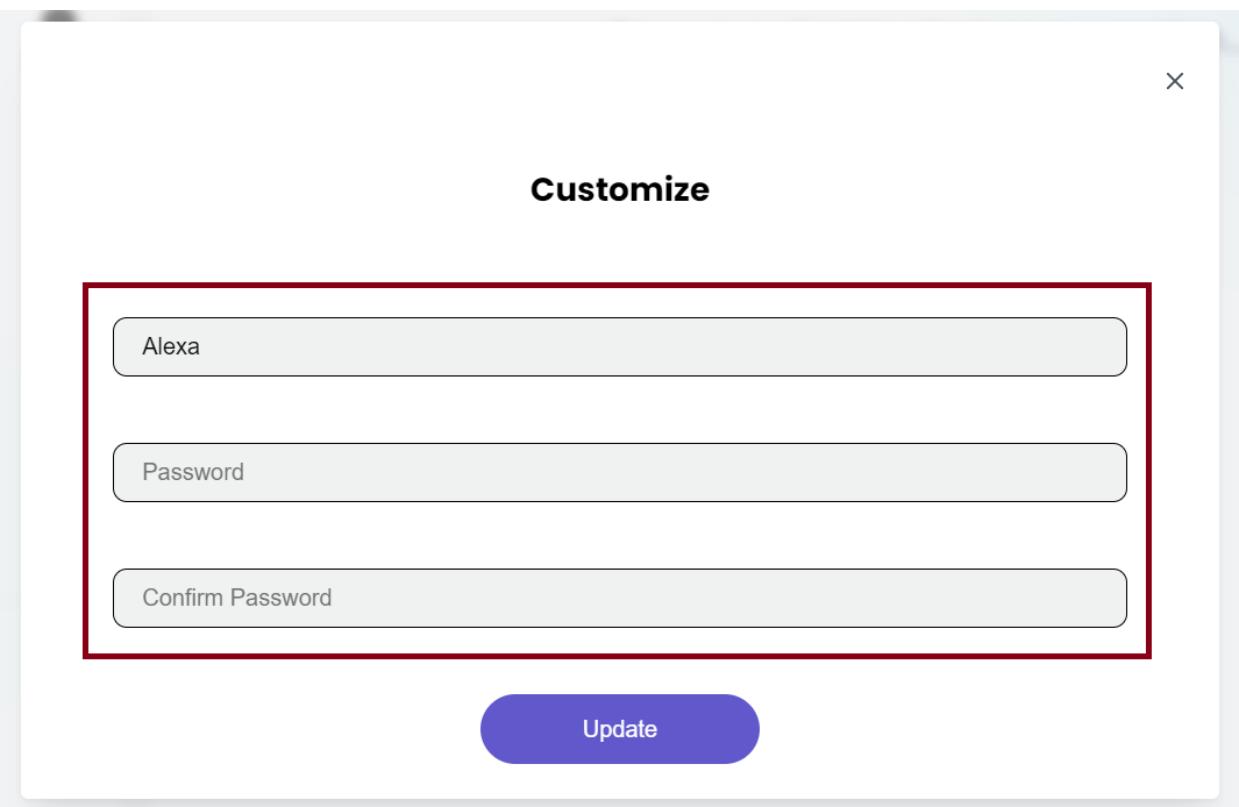
### 3.5 Customization

Pada halaman Form *Customize* berisi input data *username*, *password*, dan *confirm password* akun yang dimiliki oleh *user* apabila *user* ingin melakukan perubahan data.

A screenshot of a customization form. The title is 'Customize'. There are three input fields: 'Username' (containing 'Alexa'), 'Password', and 'Confirm Password'. Below the fields is a large blue 'Update' button.

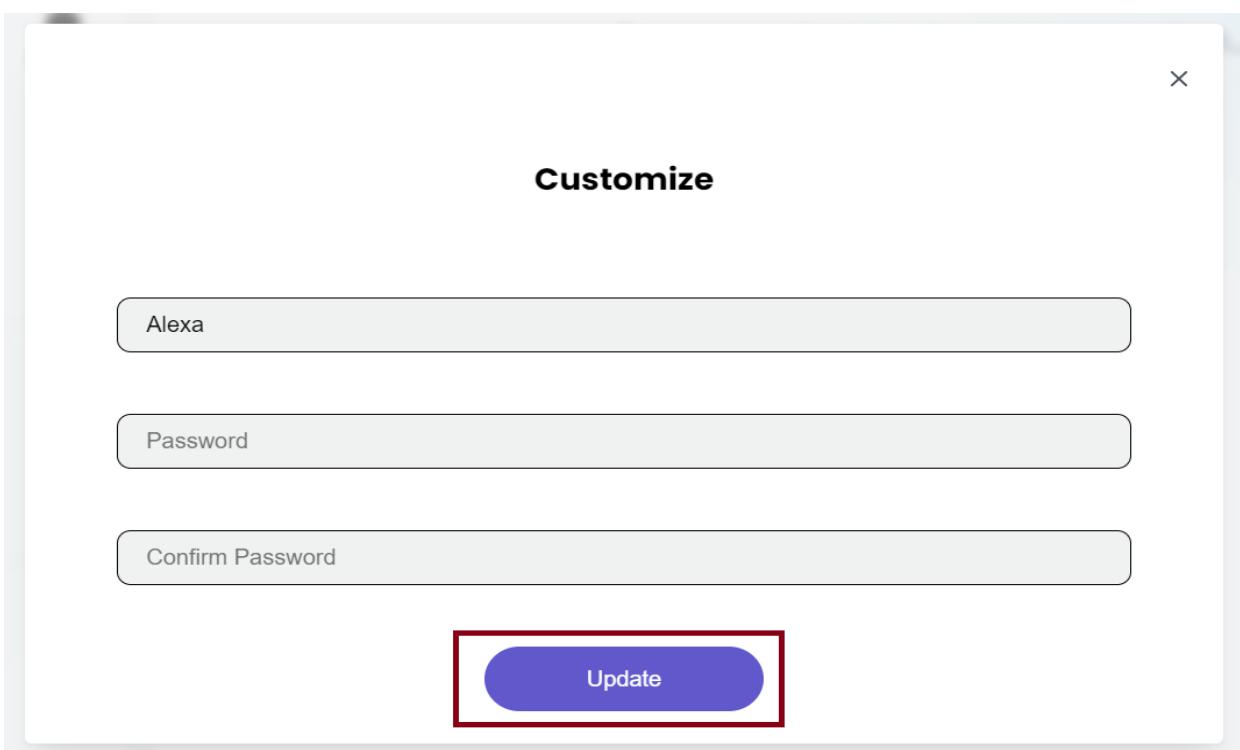
**Gambar 17 : Halaman *Customization***

Jika *user* ingin melakukan perubahan pada data *user* yang sudah teregistrasi seperti *username* dan *password* maka *user* dapat mengubahnya dengan mengisi field input seperti *username*, *password*, dan *confirm password*.



Gambar 18 : Form *Customize* pada Halaman *Customization*

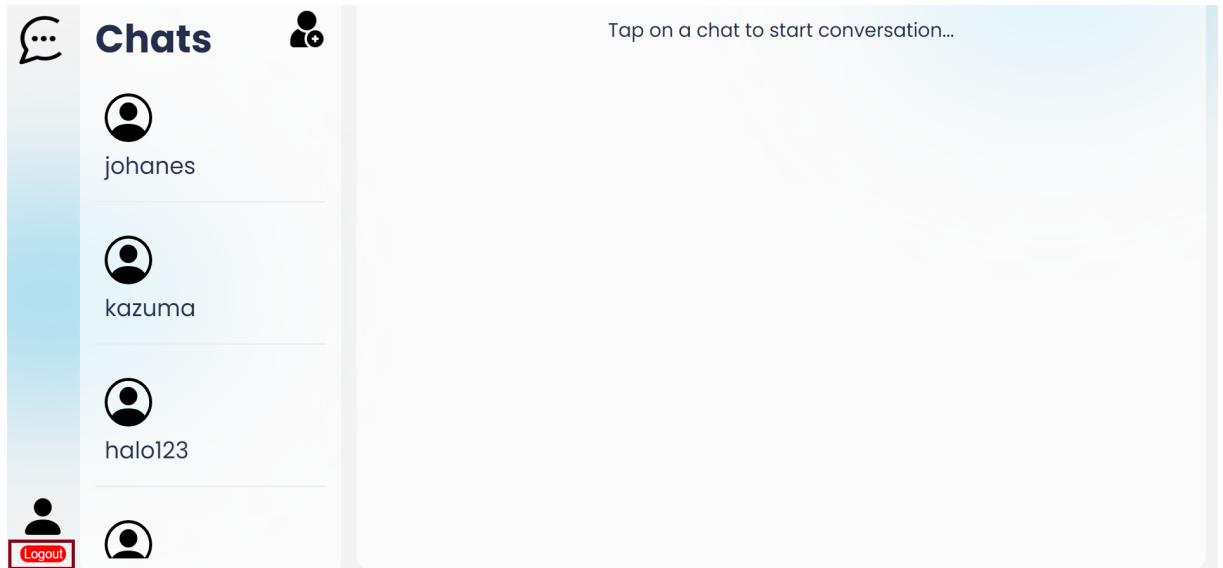
Jika *user* sudah yakin maka *user* dapat menekan tombol *Update* untuk melakukan pembaharuan. Data *user* sebelumnya akan tergantikan dengan data *user* yang baru setelah *user* melakukan *update*.



Gambar 19 : Tombol *Update*

### 3.6 Logout

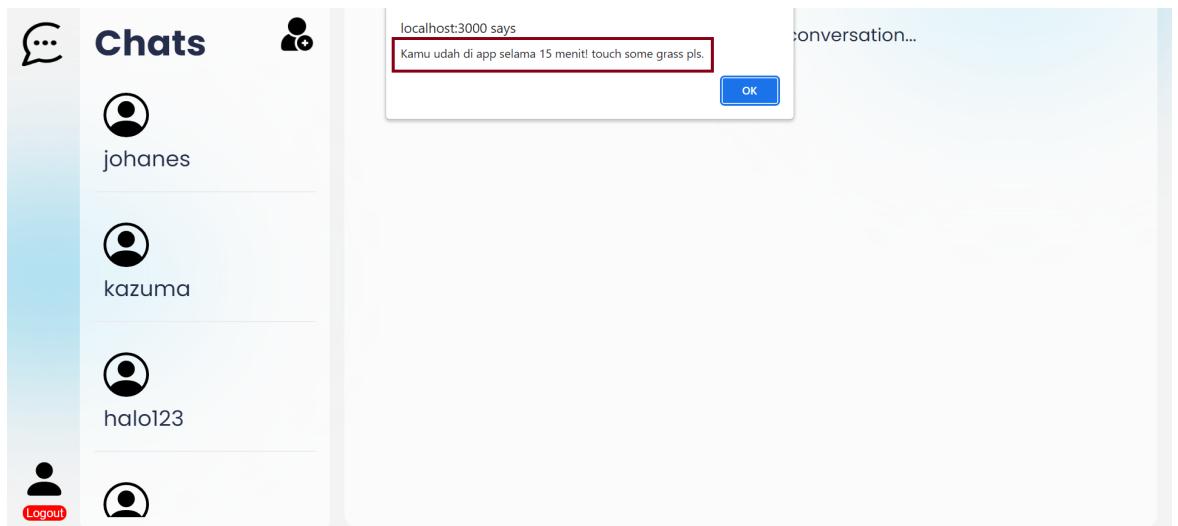
Jika *user* ingin keluar dan ingin menghapus session maka *user* dapat menekan tombol *Logout*. Setelah menekan tombol *Logout* *user* akan diarahkan ke halaman *Login*.



Gambar 20 : Tombol *Logout*

### 3.7 Send Alert

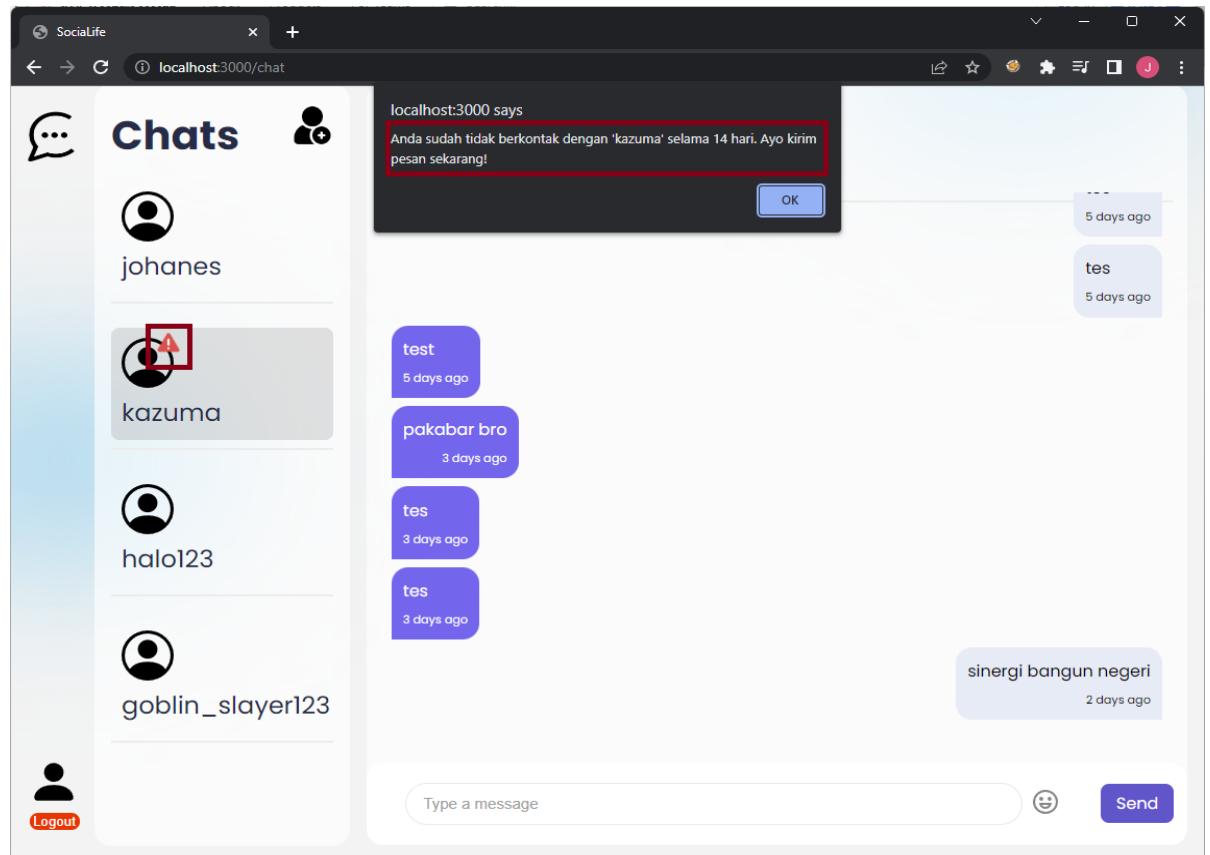
Jika *user* berada di aplikasi selama 15 menit maka aplikasi akan mengeluarkan peringatan bahwa *user* terlalu lama dalam aplikasi (15 menit), dan akan memerintahkan untuk *touch some grass*.



Gambar 21 : *Send Alert*

### 3.8 Remind to Contact

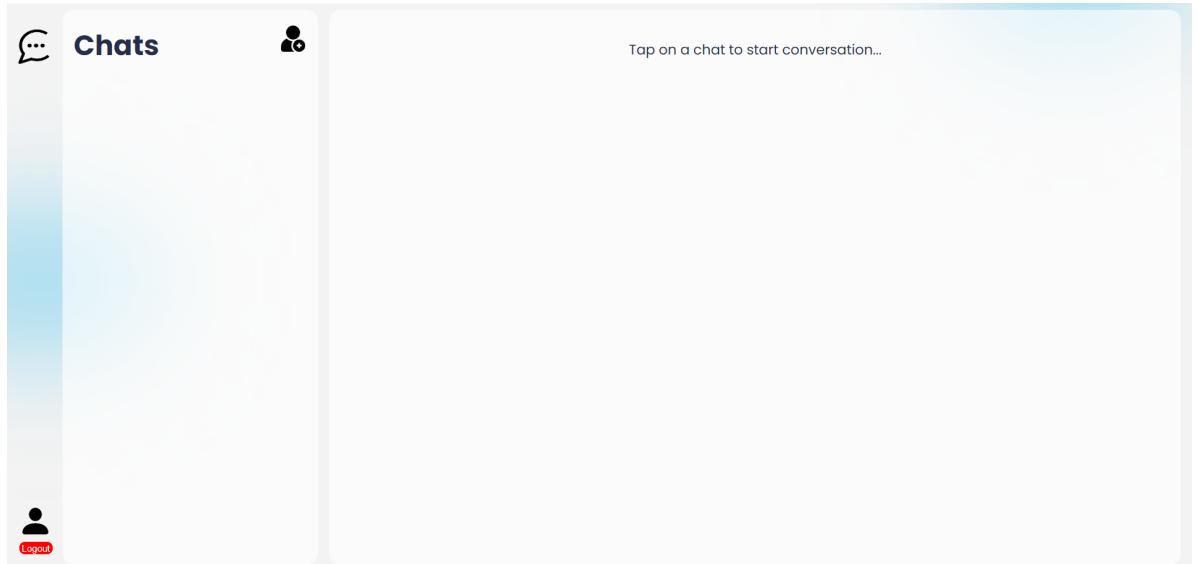
Jika *user* mengirim atau menerima pesan dan tidak ada pesan terbaru selama 14 hari berikutnya maka aplikasi akan mengeluarkan pengingat yang berupa *icon* di gambar kontak temannya.



Gambar 22 : *Remind to Contact*

## **Halaman Utama**

Halaman utama akan tampil setelah pengguna melakukan login ataupun register. Halaman utama ini dapat dibagi menjadi 3 bagian yang terletak seperti pada gambar dibawah berikut ini :



**Gambar 23 : Halaman Utama Aplikasi Chat**

## Penjelasan Bagian Side-Bar

Bagian ini berfungsi untuk menampilkan logo, customization, dan logout.



### a. Logo

Logo berfungsi sebagai tampilan untuk memberitahu bahwa *user* sedang berada pada aplikasi SociaLife.

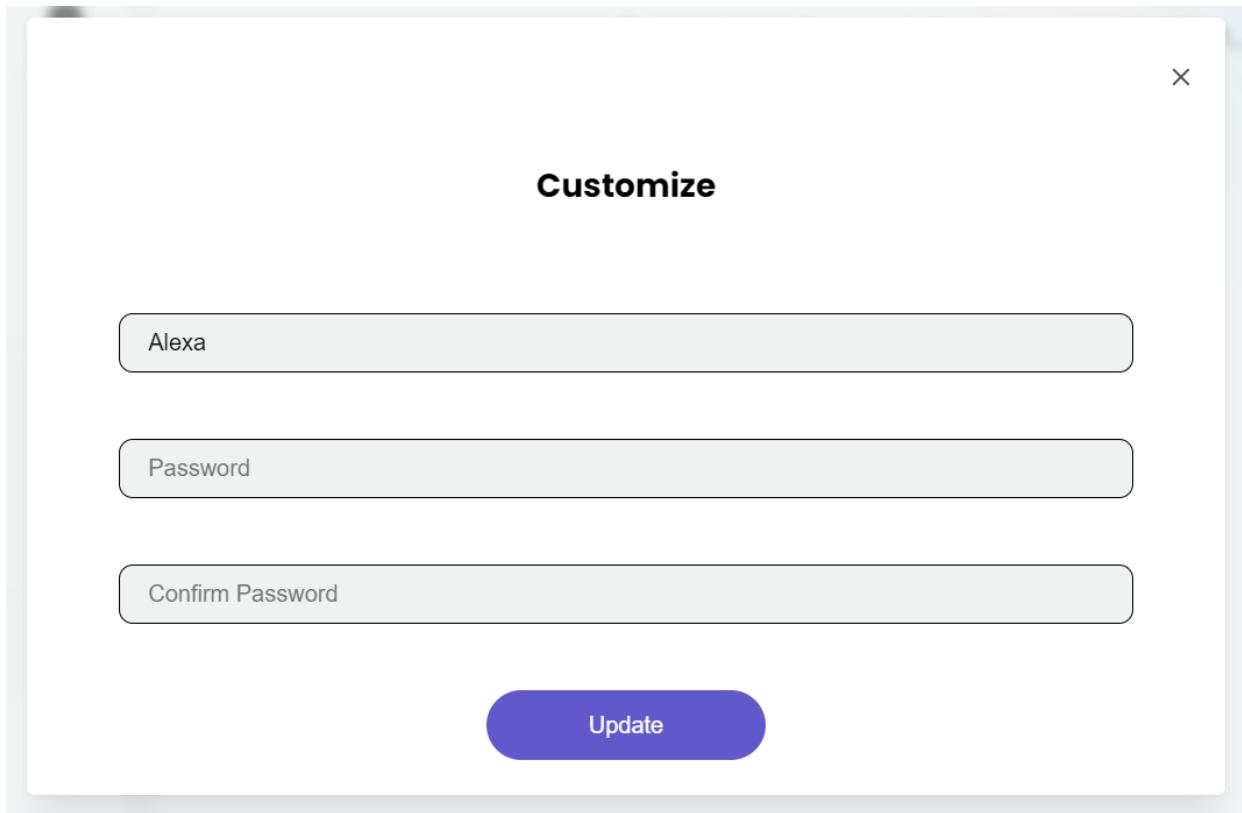


### b. Customization

User dapat mengubah data *username* atau *password* dengan mengklik *icon* profile *user*.



Setelah *icon* profile *user* ditekan, akan muncul modal form untuk *user* melakukan perubahan data *username* atau *password*.



User dapat mengisi *username* baru atau *password* baru. Setelah mengisi form kemudian user dapat menekan tombol *Update* :

Update

Maka data akan diperbarui pada *database* dan ketika user ingin *login* dengan *username* atau *password* yang lama, maka *login* akan gagal.

### c. Logout

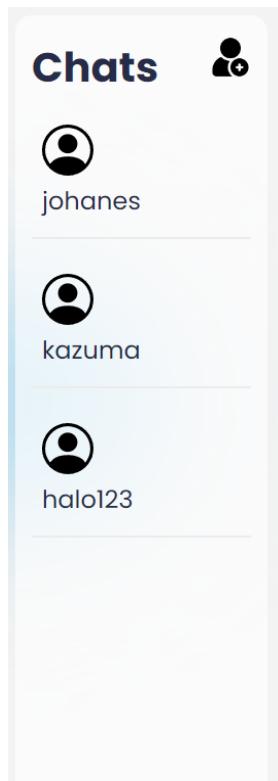
Jika user ingin keluar dari aplikasi maka user dapat menekan tombol *Logout* :

Logout

Setelah mengkonfirmasi keluar, maka akan diarahkan kembali ke halaman *login*.

## Penjelasan Bagian Chats

Bagian ini berfungsi untuk menambah teman dan menampilkan daftar teman.

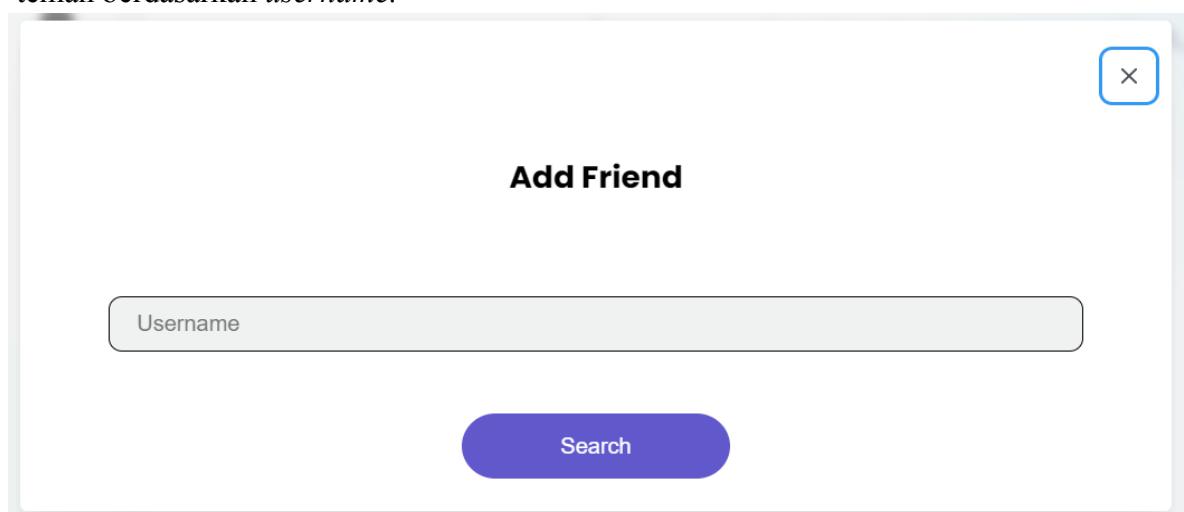


### a. Add Friend

Untuk menambah teman, *user* dapat menekan *icon* tambah profil di samping tulisan ‘Chats’.



Setelah *icon* tambah profil ditekan, akan muncul modal form untuk *user* menambah teman berdasarkan *username*.



User dapat mengisi *username* teman yang ingin ditambah. Setelah mengisi form kemudian tekan tombol *Search* :

Search

Maka daftar teman *user* akan bertambah di *database* dan nama teman akan muncul di bagian daftar teman, seperti berikut :



johanes



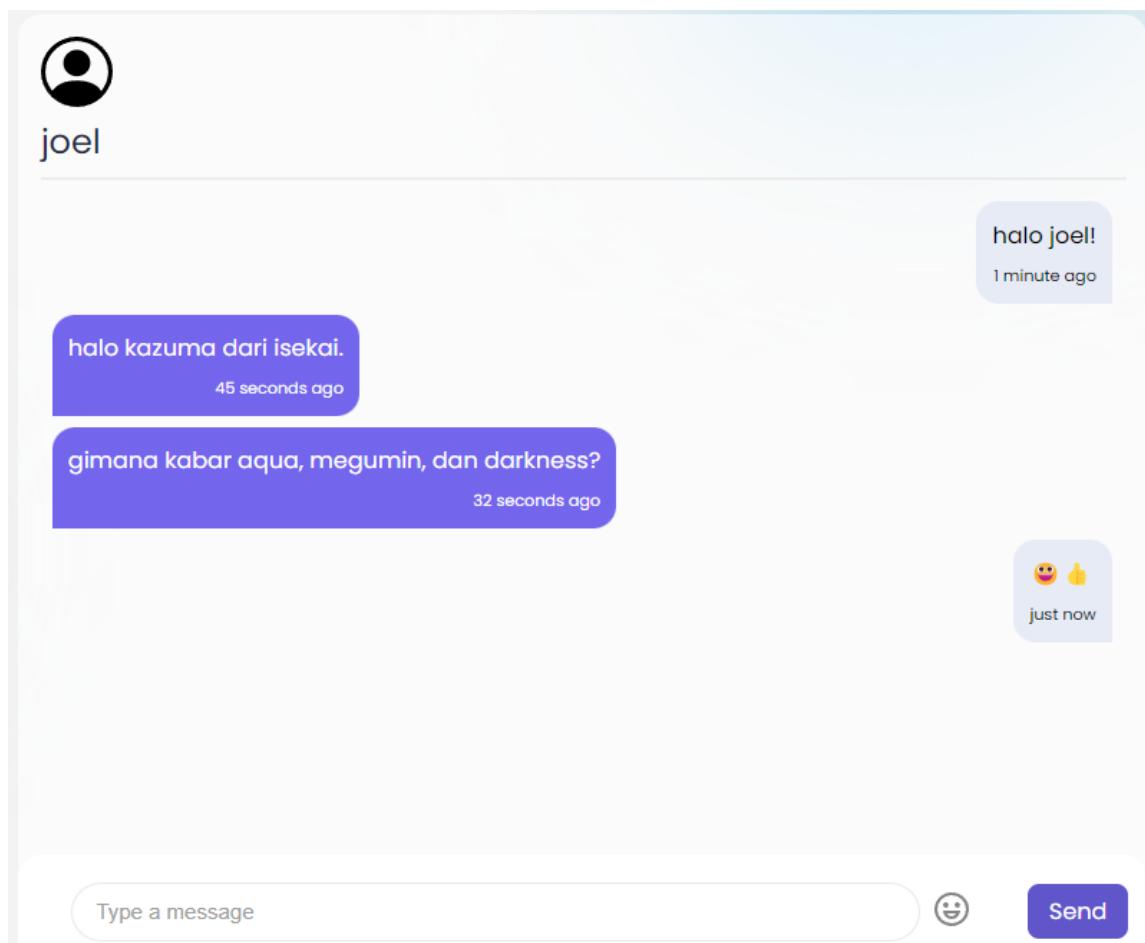
kazuma



halo123

## Penjelasan Bagian ChatBox

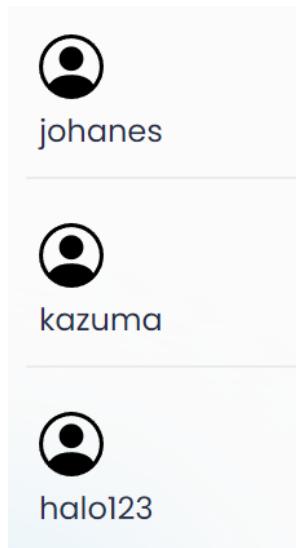
Menu ini berfungsi untuk melakukan *chatting* dengan teman yang berada di daftar teman.



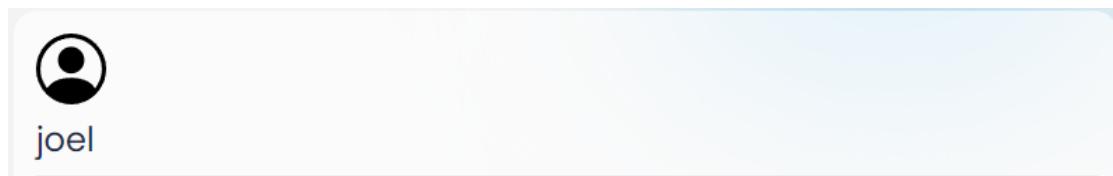
**Gambar 24 : Tampilan Chat**

### a. Chat

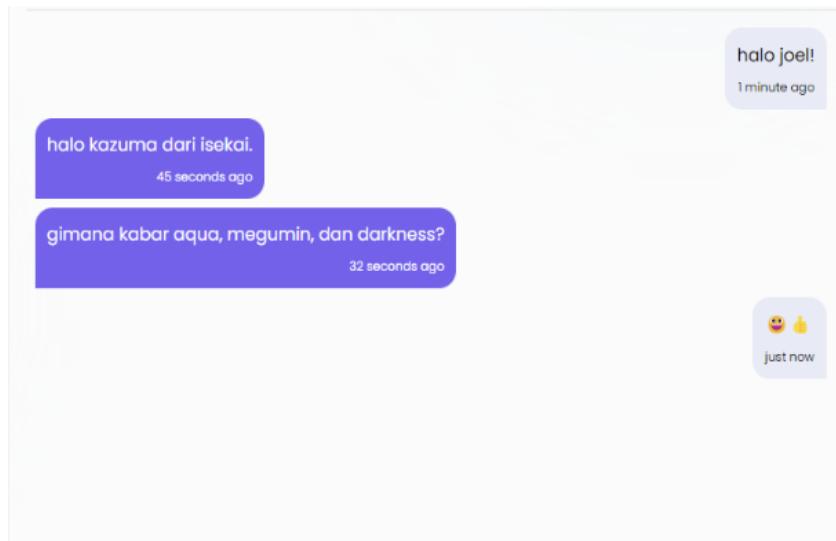
Untuk melakukan chat, User terlebih dahulu memilih teman dari daftar teman yang ingin di chat



Setelah itu, bagian *header* ChatBox akan langsung terisi dengan username teman yang dipilih, seperti berikut :



Jika teman yang dipilih belum melakukan chat maka kotak chat akan kosong. Tetapi jika sudah ada percakapan maka akan menampilkan percakapan sebelumnya.



Lalu, untuk mengetik pesan terdapat pada kolom pesan ‘type a message’ :

Type a message

Emote bersifat opsional untuk memberikan emote kepada teman.



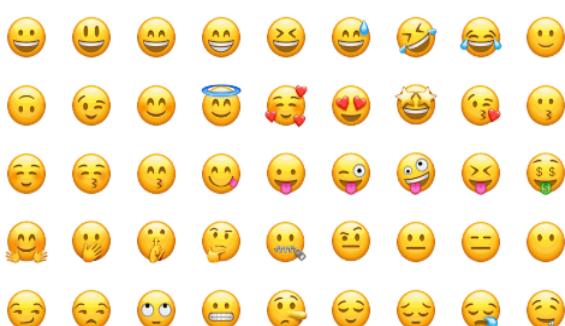
Search



Frequently Used



Smileys & People



Pesan yang sudah diketik di kolom pesan dapat dikirim dengan menekan tombol *Send* :

**Send**

Setelah menekan tombol *Send*, pesan akan masuk ke *database* Messages dan otomatis tertampil ke bagian kolom pesan, baik di *user* maupun di teman *user*.