

# Project Proposal

## Introduction

### a. Project Title

Aplikasi Pencarian Tempat Makan (Eatera)

### b. Short Description

Eatera merupakan aplikasi pencarian masakan/tempat makan yang memudahkan pengguna untuk menemukan pilihan makanan terbaik di berbagai tempat khususnya di sekitar kampus ITERA. Dengan fitur pencarian canggih dan rekomendasi personal, pengguna dapat dengan mudah menemukan tempat makan berdasarkan jenis masakan, jarak terdekat, dan tempat paling relevan. Eatera juga memungkinkan pengguna untuk melihat ulasan dan informasi lainnya yang berkaitan tentang tempat makan yang ingin dilihat. Kemudian pengguna dapat menyimpan daftar tempat makanan yang disukai ke daftar favorit.

### c. Project Team

- |                                 |           |
|---------------------------------|-----------|
| • Rizki Esa Fadillah            | 121140084 |
| • Fandu Dipo Anderson           | 121140033 |
| • Dimas Saputra                 | 121140059 |
| • Adriel Gideon Benedict Sirait | 121140196 |
| • Natanael Argajova             | 121140205 |

## Proposed Application

### a. Application Description

Aplikasi ini bertujuan untuk membantu pengguna menemukan tempat makan terdekat berdasarkan preferensi pengguna.

Halaman Beranda: Halaman ini adalah pusat informasi utama dalam Eatera. Di sini, pengguna akan menemukan pilihan restoran dan masakan yang disesuaikan dengan preferensi mereka. Pengguna juga dapat menjelajahi restoran dan masakan lainnya dengan menggunakan fitur pencarian atau menavigasi melalui kategori yang tersedia.

Halaman Detail Makanan: Halaman ini memberikan informasi lengkap tentang tempat makan tertentu. Ini mencakup nama tempat, alamat, jam operasional, ulasan, dan rating. Pengguna dapat melihat foto-foto tempat makan dan membaca ulasan dari pengguna lain. Halaman ini juga menyediakan tombol seperti "Save" untuk memudahkan pengguna untuk mengingat tempat makan tersebut.

Halaman Profil: Halaman profil adalah tempat di mana pengguna dapat melihat dan mengubah informasi pribadi. Ini mencakup foto profil dan data diri pengguna.

## b. Application Advantages

Aplikasi Eatera memiliki beberapa kelebihan yang membuatnya menjadi solusi ideal bagi para penjelajah kuliner. Pengguna dapat menikmati pilihan makanan yang disesuaikan dengan preferensi mereka. Dengan UI yang intuitif memastikan akses yang mudah ke informasi detail tentang setiap tempat makan, termasuk ulasan, dan rating pengguna. Aplikasi ini membantu pengguna menemukan lokasi tempat makan dengan mudah. Dengan kemampuan untuk menyimpan masakan/tempat makan favorit.

## c. Application Benefits

Dengan menggunakan Eatera pengguna dapat menghemat waktu dan usaha dalam mencari tempat makan yang sesuai dengan keinginan mereka. Pengguna juga dapat menjelajahi dan menikmati berbagai kuliner baru di sekitar ITERA, dan menjelajahi dunia kuliner.

# System Planning

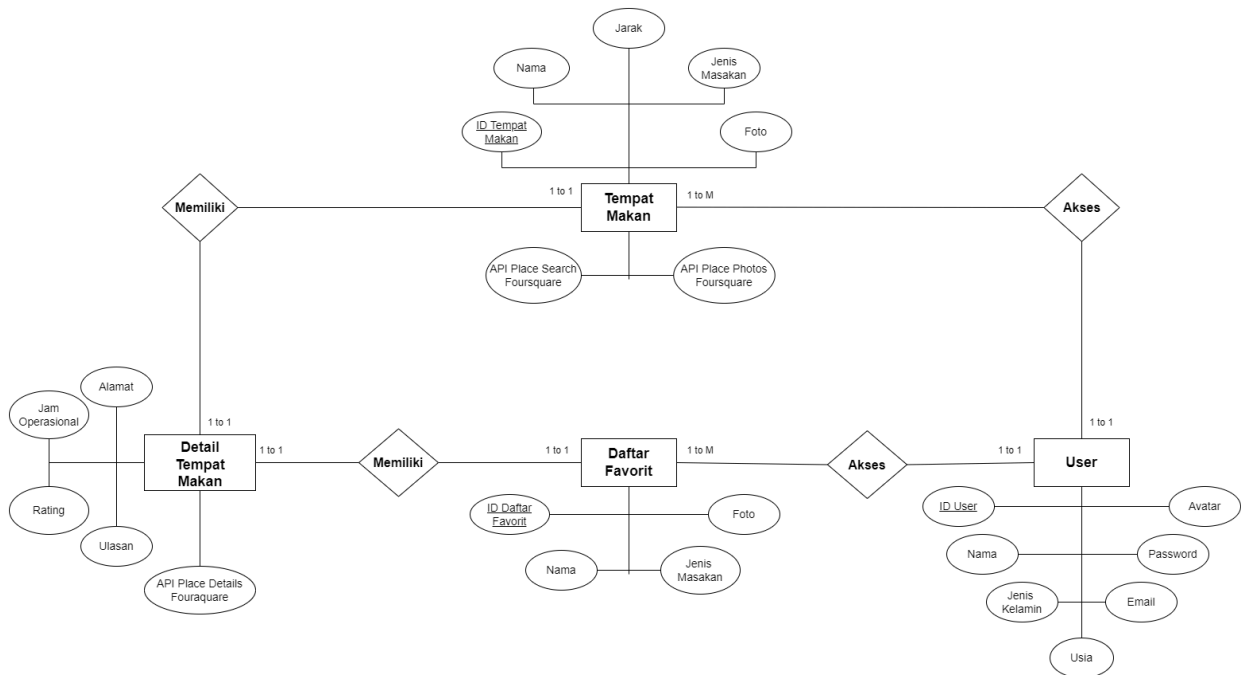
## a. Application Architecture

Aplikasi Eatera menggunakan arsitektur MVVM (Model-View-ViewModel) sebagai dasar aplikasinya. Dalam arsitektur ini, aplikasi terbagi menjadi tiga bagian utama:

- *Model*: Bagian ini bertanggung jawab atas logika bisnis dan data aplikasi. Model mewakili sumber daya data seperti tempat-tempat makan dan menyediakan operasi untuk mengambil, menyimpan, dan memperbarui data. Ini juga dapat mencakup integrasi dengan layanan pihak ketiga, seperti Foursquare API, untuk mengambil informasi tempat makan.
- *View*: Bagian ini adalah antarmuka pengguna (UI) yang dilihat dan digunakan oleh pengguna. View hanya menampilkan informasi dan mengirim perintah ke ViewModel. Dalam konteks aplikasi ini, View akan menampilkan daftar tempat makan, detail tempat makan, kolom pencarian, dan lainnya.
- *ViewModel*: Bagian ini bertindak sebagai perantara antara Model dan View. ViewModel bertanggung jawab untuk mempersiapkan dan menyajikan data dari Model ke View, serta menangani interaksi pengguna dan logika tampilan. Misalnya, ViewModel akan mengambil daftar tempat makan dari Model dan menyiapkannya untuk ditampilkan di View. Ini juga akan merespons perubahan yang dilakukan pengguna di antarmuka pengguna, seperti ketika pengguna memasukkan kriteria pencarian, dan memperbarui Model sesuai dengan itu.

Dengan menggunakan arsitektur MVVM, Eatera memisahkan peran antara logika aplikasi dan tampilan pengguna, sehingga memudahkan pengembangan tim yang terdiri dari desainer dan pengembang. Tim desainer dapat fokus pada desain antarmuka pengguna tanpa perlu memperhatikan logika aplikasi, sementara tim pengembang dapat berkonsentrasi pada implementasi logika aplikasi tanpa harus terganggu oleh tampilan pengguna. Hal ini meningkatkan efisiensi pengembangan aplikasi dan memungkinkan kolaborasi yang lebih baik antara anggota tim.

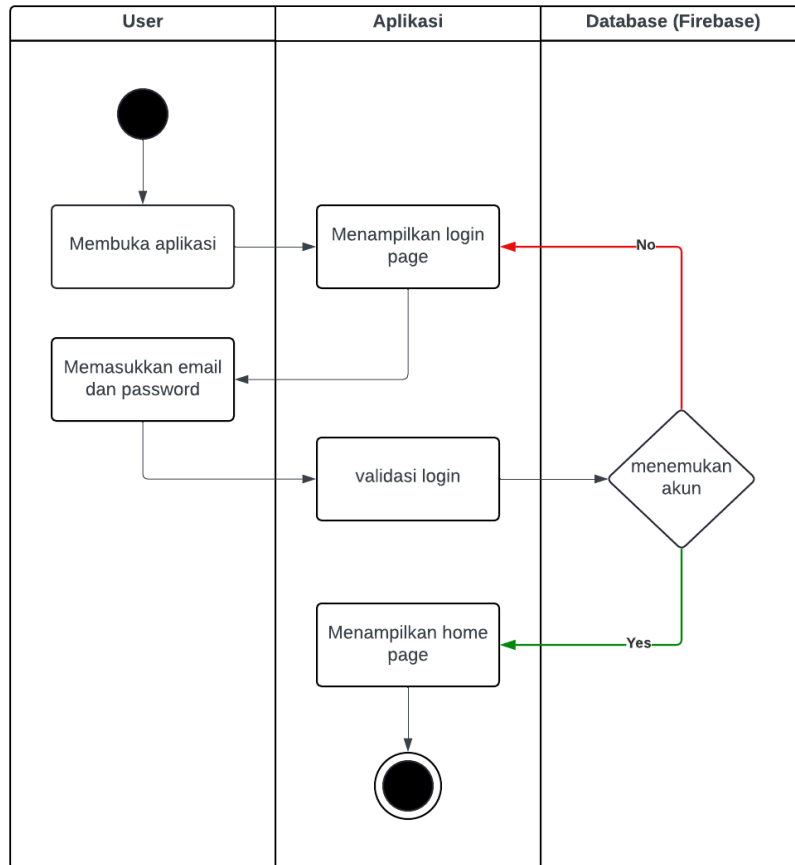
## b. Entity Relationship Diagram (ERD)



Dari diagram ER di atas, ada empat entitas utama yaitu "Tempat Makan", "Daftar Favorit", "User", dan "Detail Tempat Makan". Setiap entitas memiliki atributnya sendiri. "user" dalam hubungannya dengan "Tempat Makan" dan "Daftar Favorit" adalah "Akses", sementara "Tempat Makan" dan "Daftar Favorit" dalam hubungannya dengan "Detail Tempat Makan" adalah "Memiliki".

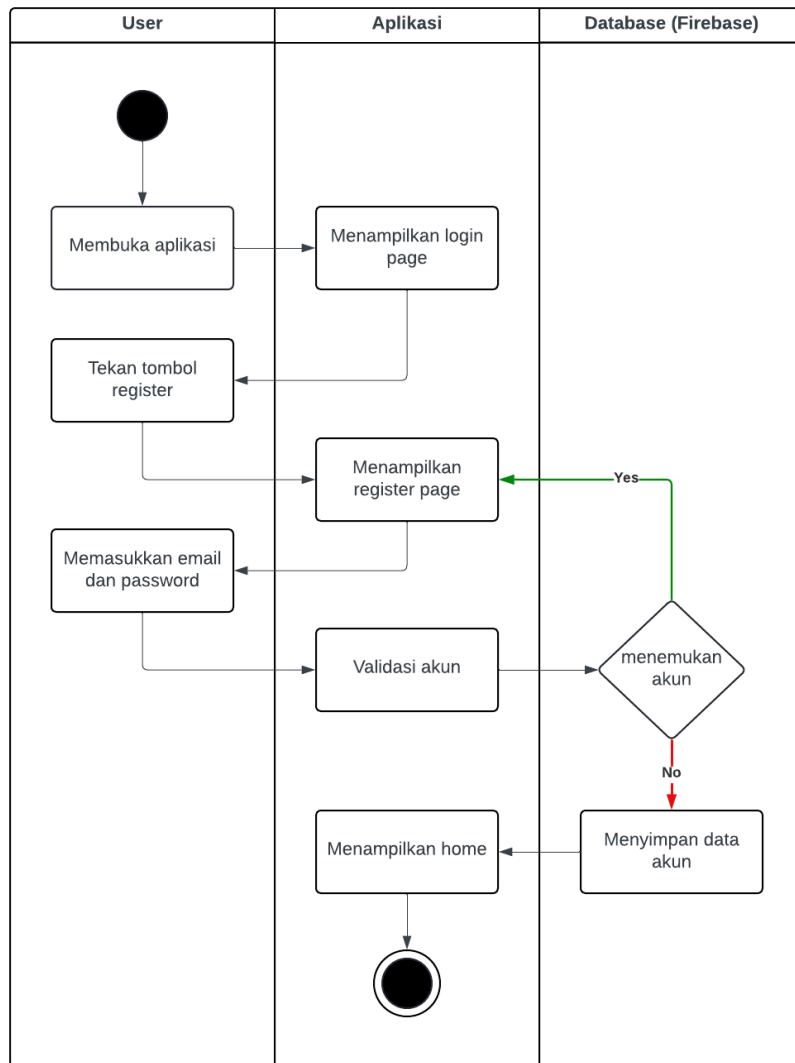
### c. UML Design

- Activity Diagram: [Write the description]
  - Login



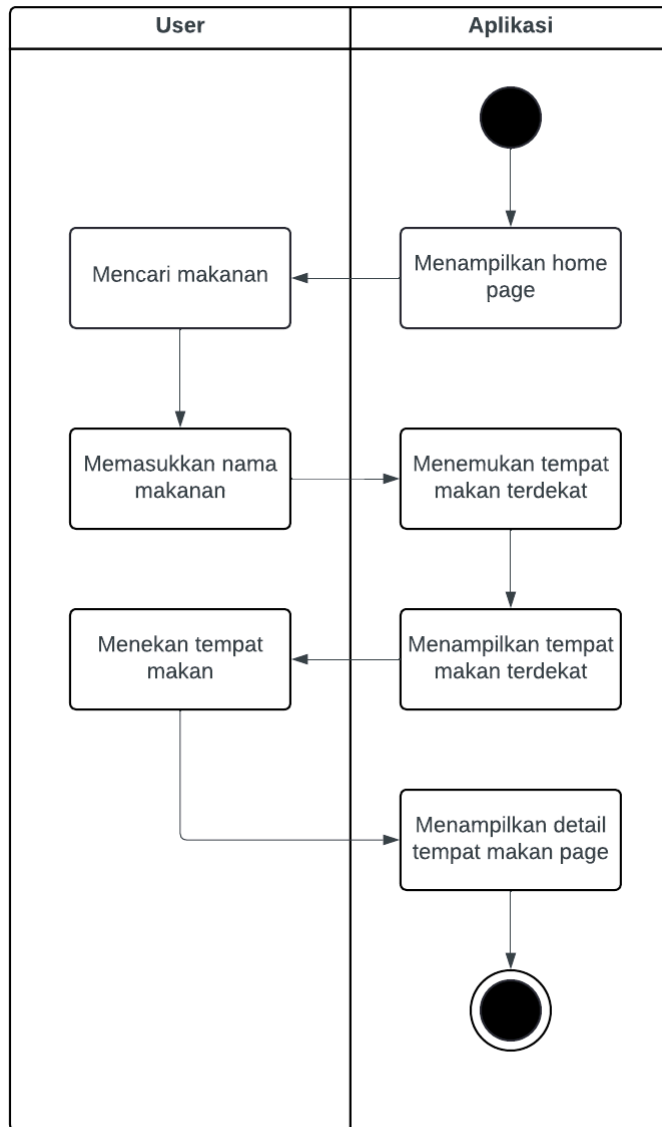
Keadaan awal dimulai ketika pengguna membuka aplikasi dan halaman login ditampilkan. Pengguna mengisi data yang diperlukan (email dan password) dan aplikasi memeriksa data login tersebut terhadap basis data firebase. Jika data tersebut tidak benar, maka pengguna akan diarahkan kembali ke halaman login. Jika data tersebut benar, pengguna akan diarahkan ke halaman utama sebagai keadaan selesai.

## 2. Register



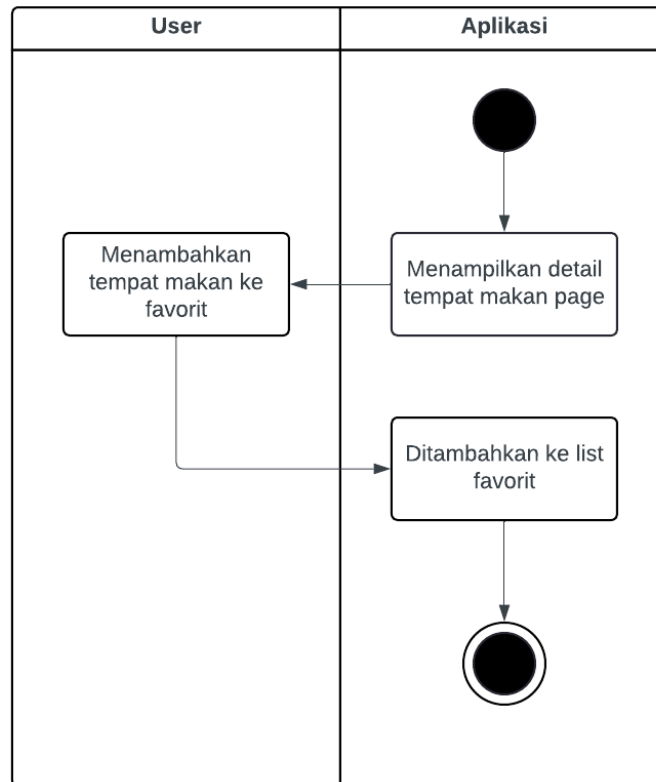
Keadaan awal dimulai ketika pengguna membuka aplikasi dan halaman login ditampilkan. Pengguna menekan tombol registrasi untuk mendaftar aplikasi, kemudian aplikasi akan menampilkan halaman register. Selanjutnya pengguna memasukkan email dan password untuk mendaftar, lalu aplikasi akan memvalidasi akun dengan memeriksa database. Jika akun tidak valid maka akan kembali ke halaman register, sedangkan jika valid akun akan tersimpan di database dan menampilkan halaman home/beranda.

### 3. Search



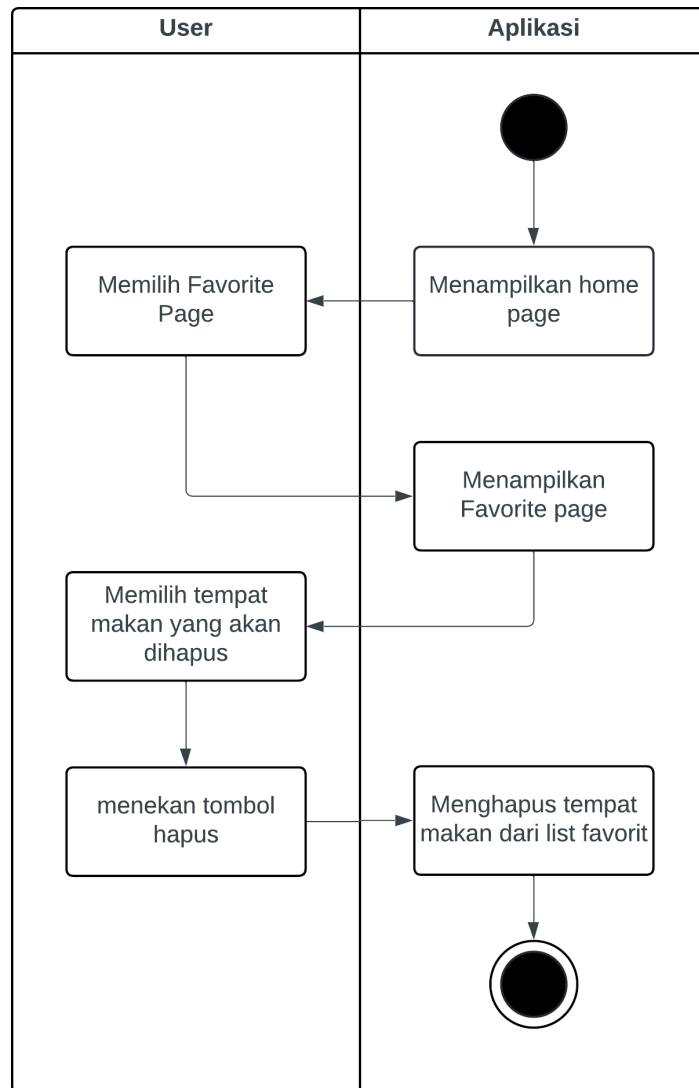
Keadaan awal dimulai ketika aplikasi menampilkan halaman home, ketika pengguna akan mencari tempat makan, pengguna akan memasukkan jenis makanan atau nama tempat. Kemudian aplikasi akan menemukan tempat makanan dan menampilkannya. Ketika pengguna menekan salah satu tempat di daftar makanan yang muncul, aplikasi akan menampilkan halaman detail untuk memberikan informasi lebih lengkap tentang tempat makan.

#### 4. Menambahkan tempat makan favorit



Keadaan awal dimulai ketika aplikasi menampilkan detail tempat makan, pada halaman detail tempat makan pengguna dapat menambahkannya ke tempat makan favorite dengan menekan tombol “Save” yang tersedia. Kemudian aplikasi akan menambahkan tempat makan tersebut ke halaman favorite list.

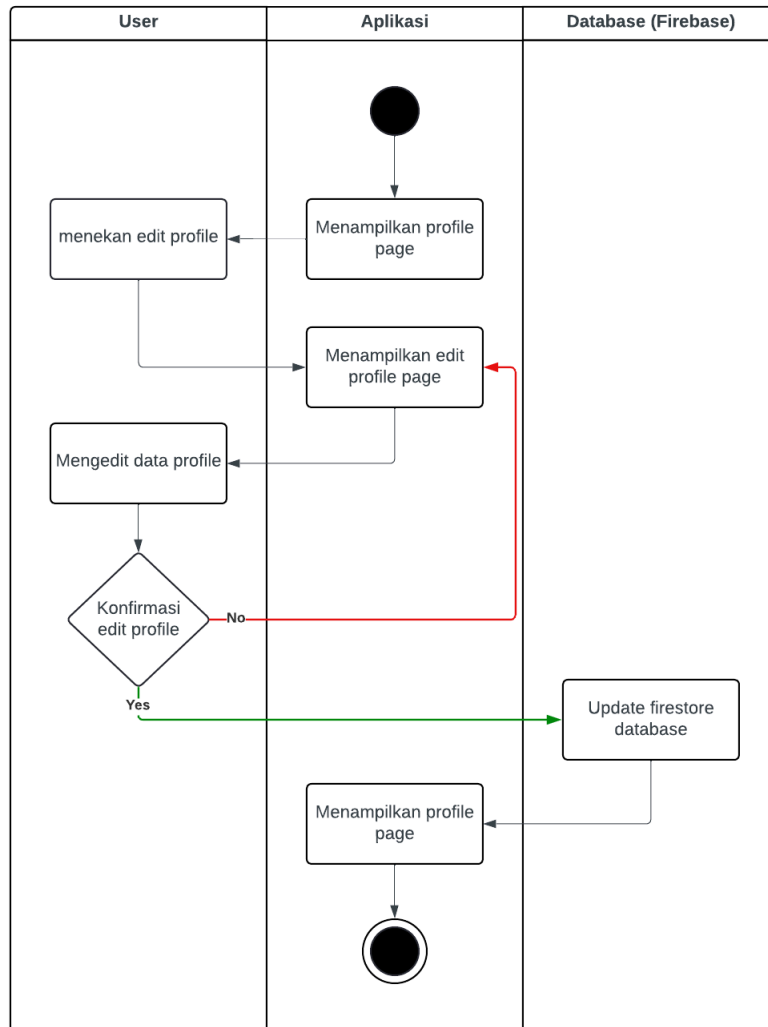
## 5. Menghapus tempat makan favorit



Keadaan awal dimulai ketika aplikasi menampilkan halaman beranda/home, pengguna memilih halaman favorite melalui navigasi yang tersedia. Kemudian aplikasi akan menampilkan halaman favorite, pengguna dapat memilih daftar tempat makan yang ingin dihapus. Ketika pengguna menekan tombol hapus, aplikasi akan menghapus tempat makan tersebut dari halaman favorite.

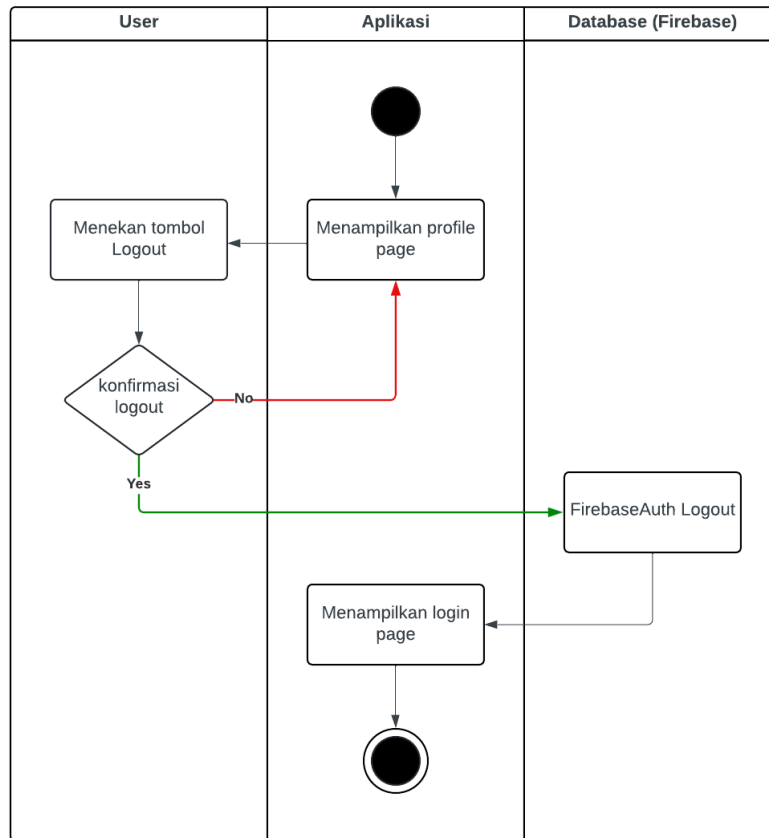


## 6. Edit Profile



Keadaan awal dimulai ketika aplikasi menampilkan halaman profil, ketika pengguna menekan tombol “Edit”, aplikasi akan menampilkan halaman edit profil. Kemudian pengguna dapat mengedit data - data pada halaman profil, lalu pengguna akan mengkonfirmasi perubahan dengan mengklik tombol “Simpan”. Kemudian jika berhasil dikonfirmasi database akan menyimpan dan mengupdate perubahan data dan menampilkannya di aplikasi pada halaman profil. Sedangkan jika gagal maka aplikasi akan kembali ke halaman edit profil.

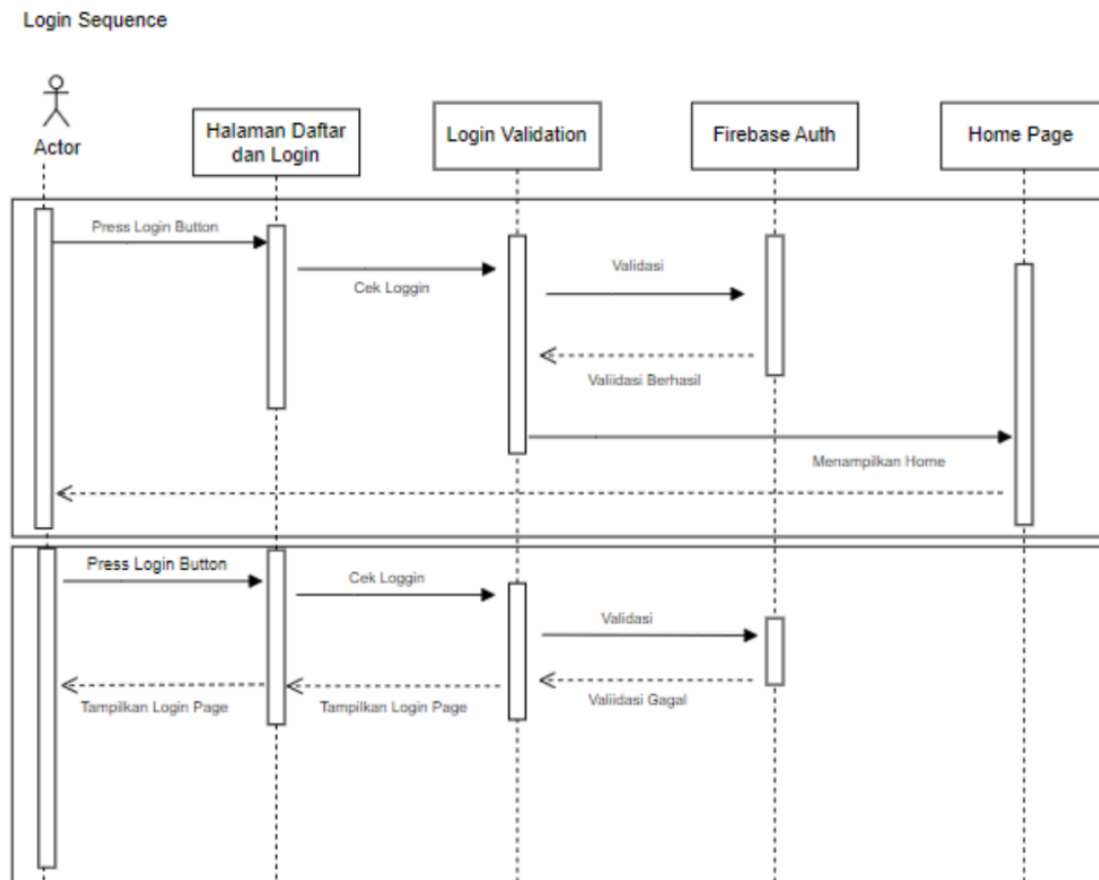
## 7. Logout



Keadaan awal dimulai ketika aplikasi menampilkan halaman profil, pada halaman profil pengguna dapat menekan tombol “Logout” dan mengkonfirmasi untuk logout. Jika tidak maka pengguna akan diarahkan kembali ke halaman profil, sedangkan jika ya maka firebase auth akan melakukan logout dan aplikasi akan menampilkan halaman login.

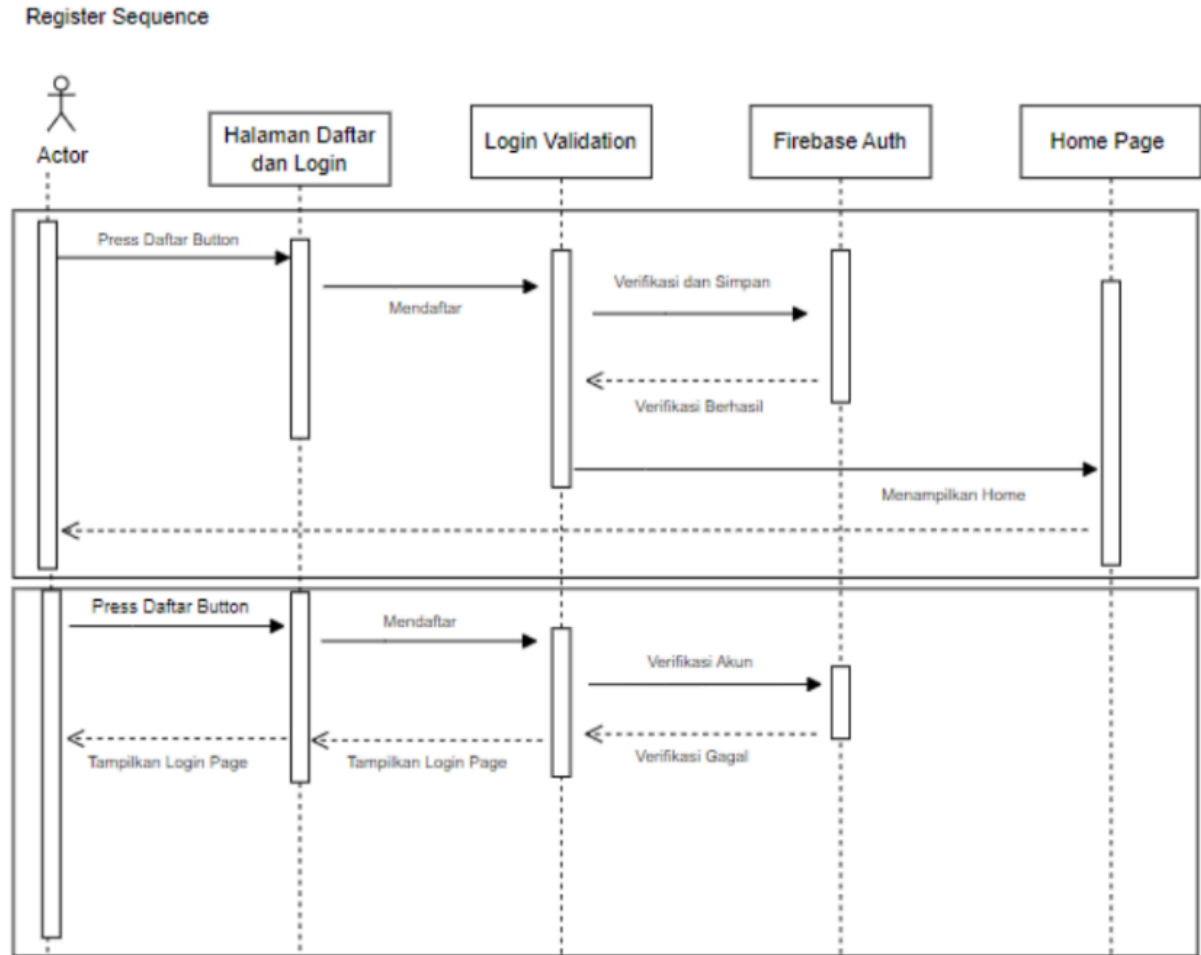
- Sequence Diagram:

- Login Sequence



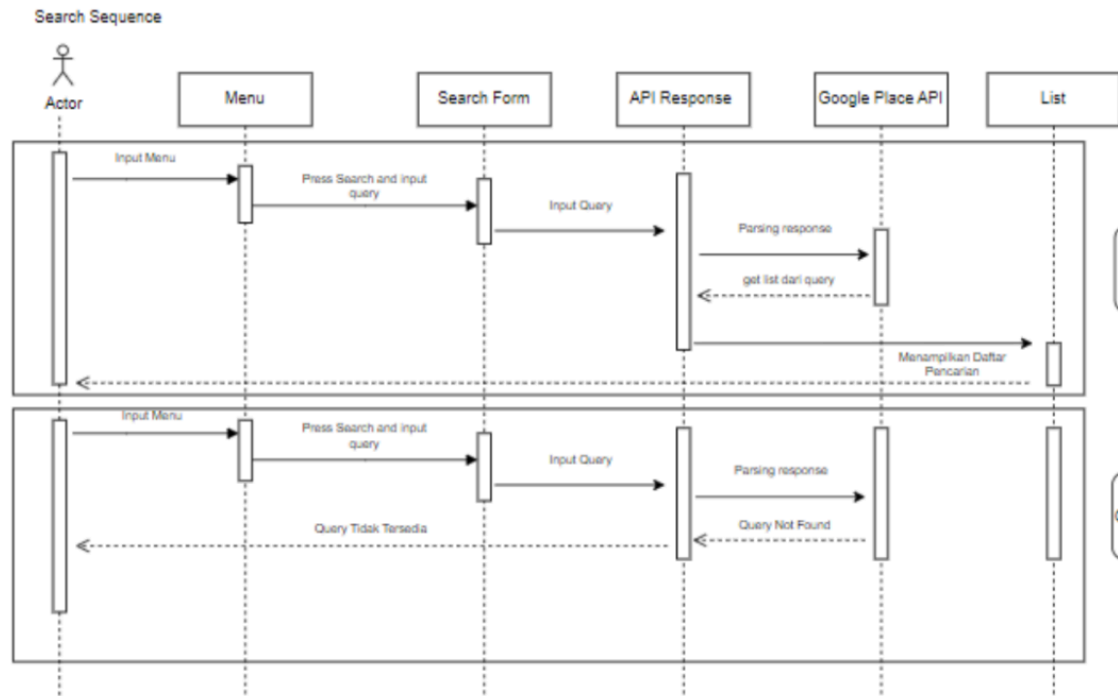
Proses dimulai dengan seorang 'Aktor' yang berinteraksi dengan sistem, dalam konteks ini adalah pengguna yang mencoba untuk masuk. Pengguna disajikan dengan halaman yang berisi dua opsi: Daftar dan Login. Ketika pengguna memutuskan untuk login, sistem akan memvalidasi kredensial pengguna menggunakan Firebase Auth. Firebase Auth adalah layanan yang ditawarkan oleh Google untuk mengautentikasi pengguna dengan berbagai metode, seperti email & password. Setelah validasi berhasil, sistem memberikan akses ke Halaman Beranda. Jika Aktor ingin login lagi, mereka akan menekan tombol Login. Sistem kemudian memeriksa apakah pengguna sudah masuk. Jika pengguna sudah masuk, validasi berhasil. Jika pengguna belum masuk, mereka akan menekan tombol Login lagi. Sistem kembali memeriksa apakah pengguna sudah login dan memvalidasi kredensial pengguna sekali lagi. Jika kredensial pengguna salah, validasi gagal dan sistem menampilkan Halaman Login lagi, meminta pengguna untuk memasukkan kredensial mereka.

## 2. Register Sequence



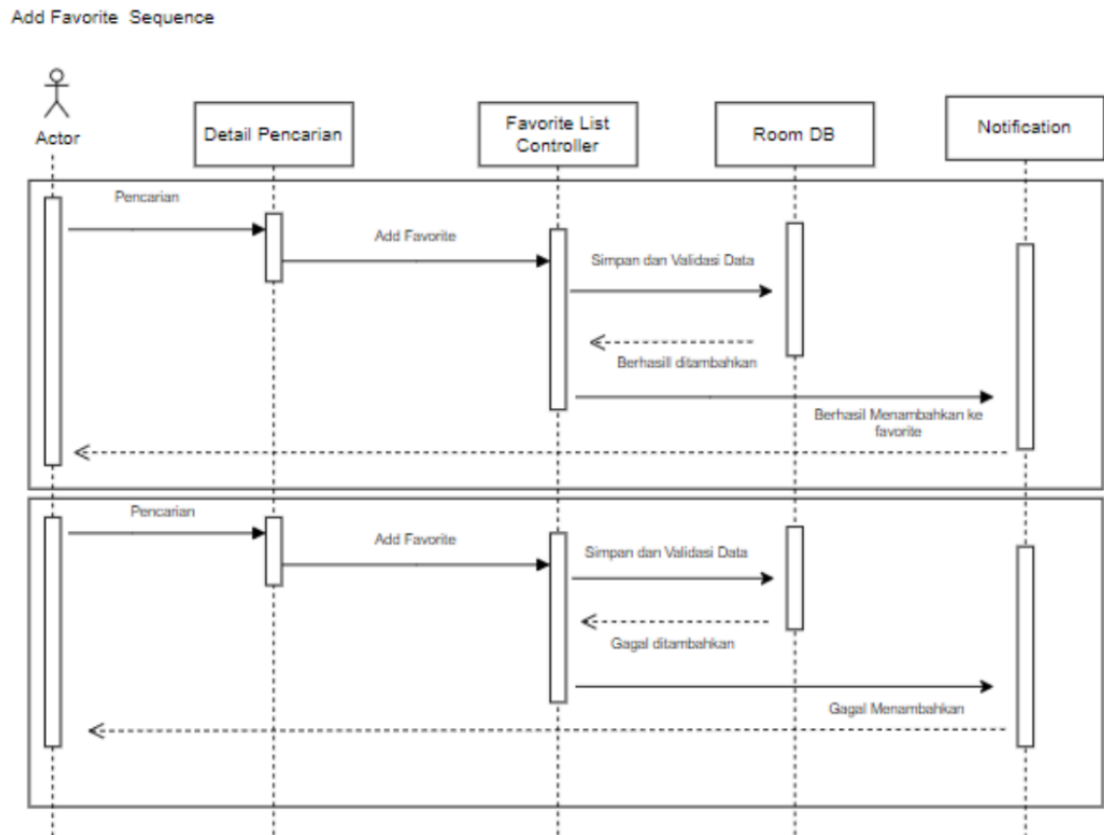
Halaman Beranda adalah layar awal. Pengguna menekan tombol "Daftar" (Register). Sistem memvalidasi input pengguna dan menyimpan data tersebut. Pengguna kemudian terdaftar. Sistem berhasil memverifikasi pendaftaran. Pengguna diminta untuk menekan tombol "Daftar" lagi, mungkin untuk mengonfirmasi tindakan tersebut. Jika pengguna mengonfirmasi, mereka didaftarkan kembali, yang bisa menjadi langkah redundan dalam urutan ini. Sistem kemudian memverifikasi akun pengguna dan menampilkan halaman login.

### 3. Search Sequence



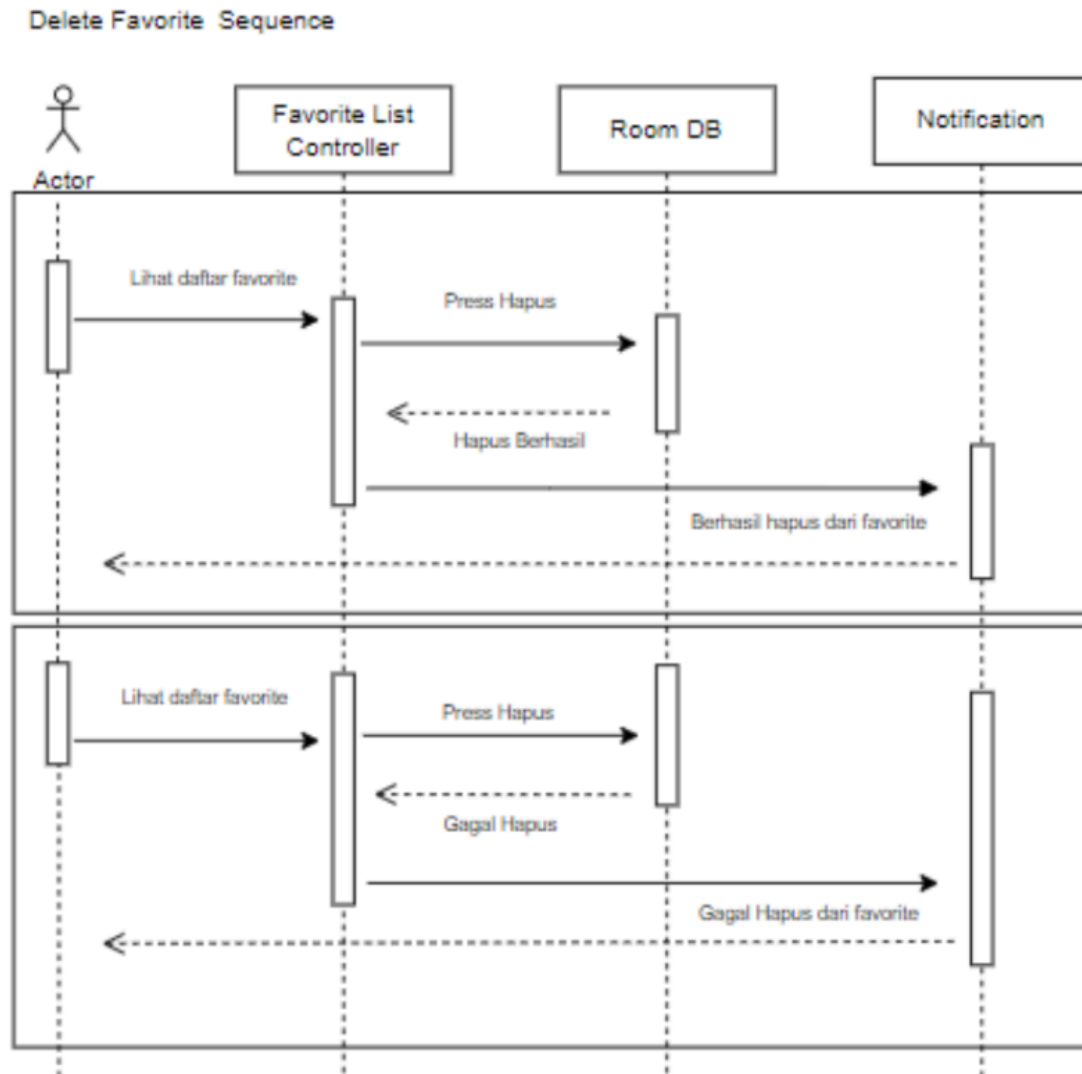
Pengguna memilih aktor, namun dalam contoh ini, tidak ada aktor yang dipilih. Pengguna kemudian memilih opsi "Input Query" di dalam menu Input dan diminta untuk mengisi form pencarian dengan query. Aplikasi mengirim query tersebut ke Google Place API dan menerima respons dari API. Aplikasi menampilkan daftar hasil pencarian yang sesuai dengan query tersebut. Pengguna diminta untuk mencari ulang dan memasukkan query baru jika diperlukan. Setelah memasukkan ulang query pencarian, aplikasi memproses respons dari Google Place API dan membuat daftar hasil pencarian yang baru. Jika query tidak dapat ditemukan, aplikasi akan menampilkan pesan "Query Not Found". Akhirnya, aplikasi menampilkan daftar hasil pencarian yang sesuai dengan query yang dimasukkan.

#### 4. Add Favorite Sequence



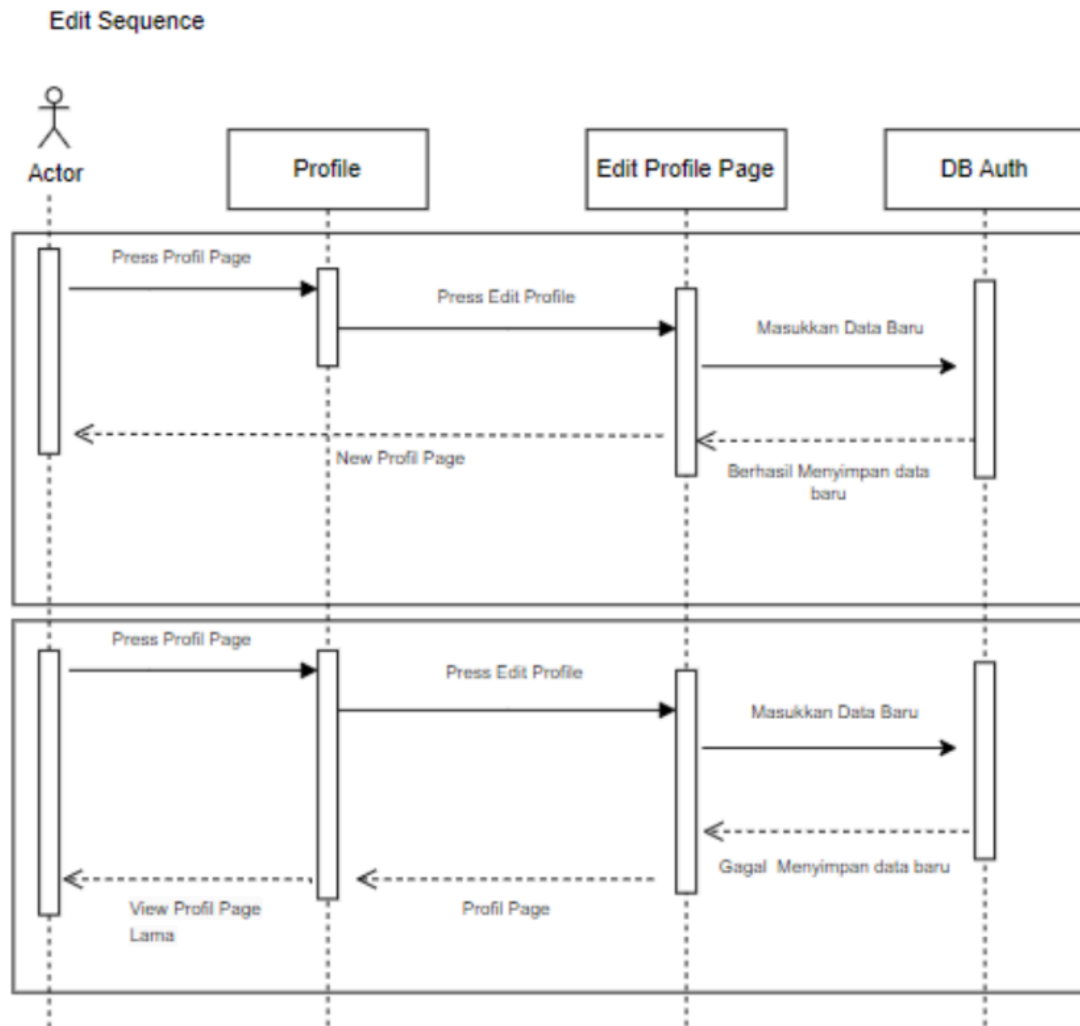
Pengguna memulai urutan "Tambah Favorit". Sistem mencari item yang diinginkan menggunakan fungsi "Pencarian". Sistem menampilkan hasil pencarian di halaman "Detail Pencarian". Pengguna memilih item untuk ditambahkan ke favorit mereka. Sistem mencoba menambahkan item tersebut ke "Daftar Favorit" menggunakan "Controller" dan "Room DB" (database). Jika penambahan berhasil, sistem menampilkan notifikasi "Berhasil ditambahkan". Jika penambahan gagal, sistem menampilkan notifikasi "Gagal ditambahkan". Sebagai alternatif, sistem juga bisa menampilkan notifikasi "Berhasil Menambahkan ke favorite" atau "Gagal Menambahkan".

## 5. Deleted Favorite Sequence



Pengguna mengakses daftar favorit, yang merupakan kumpulan item yang telah ditandai sebagai favorit oleh pengguna. Pengguna memilih item dari daftar favorit yang ingin dihapus. Pengguna menekan tombol "Hapus", yang merupakan tombol untuk menghapus item. Sistem kemudian mencoba menghapus item yang dipilih dari daftar favorit. Jika penghapusan berhasil, sistem menampilkan pesan yang menunjukkan bahwa item berhasil dihapus dari daftar favorit. Jika penghapusan gagal, sistem menampilkan pesan yang menunjukkan bahwa penghapusan tidak berhasil. Proses ini melibatkan beberapa komponen, termasuk Aktor (pengguna), Daftar Favorit (kumpulan item favorit), Kontroller (komponen yang menangani input pengguna dan mengelola daftar favorit), Room DB (database yang menyimpan daftar favorit), dan Notifikasi (komponen yang menampilkan pesan kepada pengguna).

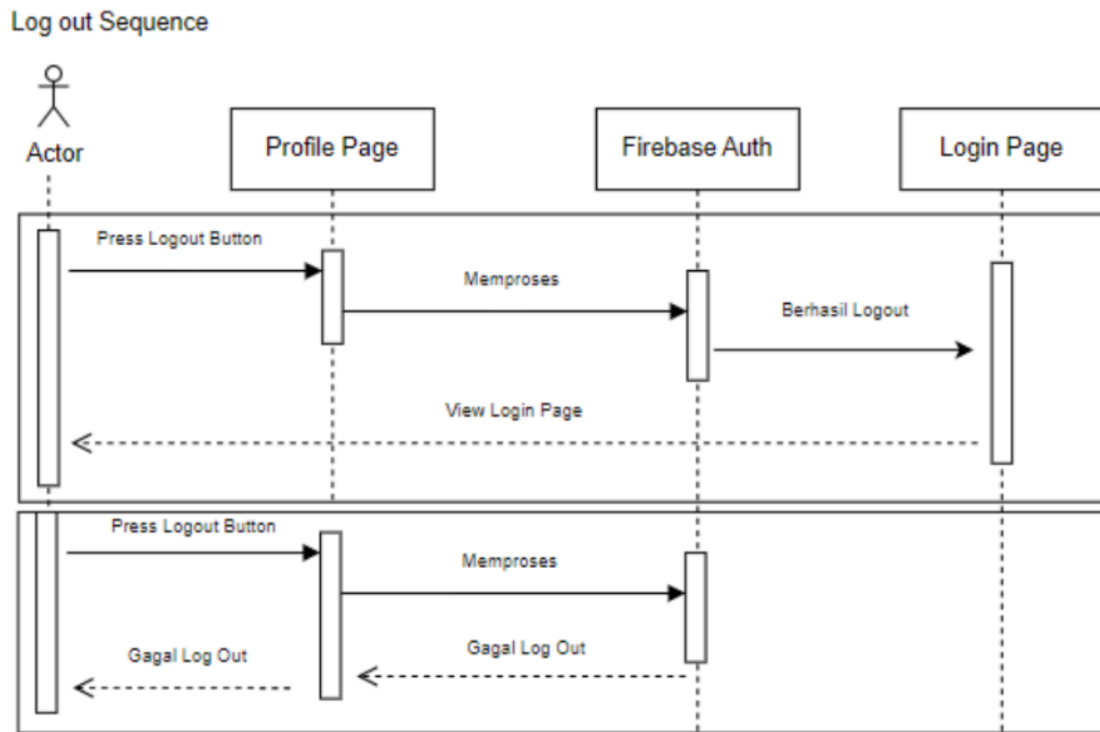
## 6. Edit Profile Sequence



Aktor menekan tombol "Profil Page", kemudian halaman "Profil" ditampilkan. Aktor menekan tombol "Edit Profile", lalu halaman "Edit Profile" ditampilkan. Sistem kemudian meminta "DB Auth" (autentikasi database). Aktor memasukkan data baru. Halaman "Profil" baru ditampilkan, menunjukkan bahwa data baru berhasil disimpan ("Berhasil Menyimpan data baru"). Aktor menekan tombol "Edit Profile" lagi. Aktor mencoba memasukkan data baru. Sistem gagal menyimpan data baru ("Gagal Menyimpan data baru"). Halaman "View Profil" ditampilkan, menunjukkan data lama. Urutan kemudian kembali ke "Profil Page", yang menunjukkan bahwa aktor mungkin mencoba mengedit informasi profil berulang kali.

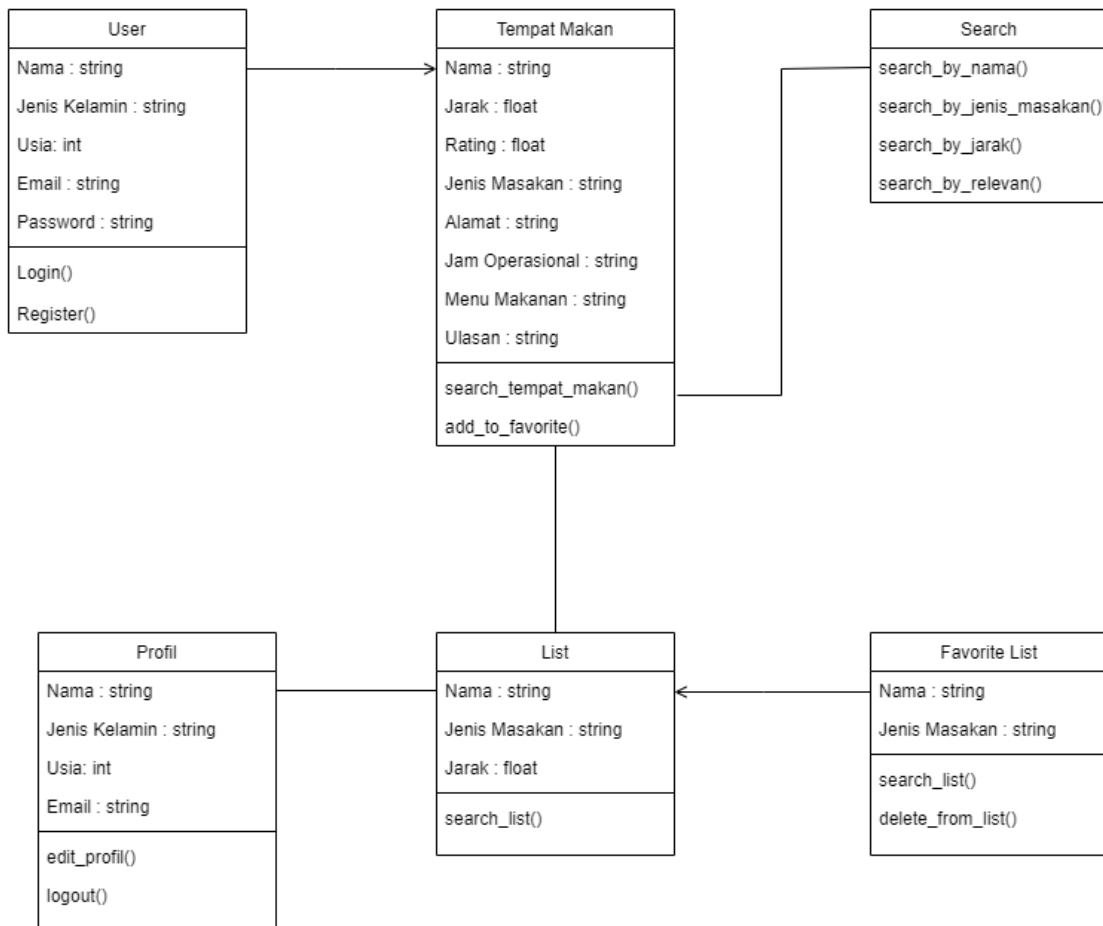


## 7. Logout Sequence



Pengguna, sebagai aktor, memulai proses logout dari akun mereka. Mereka mengklik tombol logout yang biasanya ditemukan di halaman profil atau menu pengaturan. Halaman profil adalah tempat di mana informasi akun dan pengaturan pengguna ditampilkan. Saat tombol logout di klik, sistem mulai memproses permintaan logout. Setelah beberapa saat, pengguna diarahkan ke halaman login. Ini menunjukkan bahwa permintaan logout sedang diproses oleh sistem. Namun, untuk mengonfirmasi tindakan logout, pengguna harus mengklik tombol logout sekali lagi. Sistem kemudian memproses konfirmasi logout ini. Jika terjadi kesalahan selama proses ini, sistem akan menampilkan pesan "Gagal Log Out". Hal ini menunjukkan bahwa ada masalah dalam proses logout. Firebase Auth, layanan yang disediakan oleh Firebase untuk menangani autentikasi pengguna, kemungkinan besar bertanggung jawab untuk menangani proses logout ini. Jika semua berjalan lancar, pengguna akan diarahkan kembali ke halaman login, di mana mereka dapat memasukkan kredensial login mereka untuk mengakses akun kembali. Akhirnya, jika proses logout berhasil, sistem akan menampilkan pesan "Berhasil Logout", menandakan bahwa pengguna telah berhasil keluar dari akun mereka. Dengan demikian, urutan logout melibatkan pengguna yang mengklik tombol logout, sistem yang memproses permintaan, mengarahkan pengguna ke halaman login, dan kemudian mengkonfirmasi tindakan logout. Jika tidak ada kesalahan yang terjadi, pengguna akan berhasil keluar dari akun mereka.

- Class Diagram: [Write the description]



Dalam Class Diagram ini terdapat 6 kelas, antara lain : User, Tempat Makan, Search, Profil, List, dan Favorite List.

1. Kelas *User* memiliki lima atribut antara lain atribut Nama, Jenis Kelamin, Usia, Email dan Password. Dengan menggunakan atribut-atribut ini, kita dapat menjalankan 2 metode, yaitu login atau register.
2. Kelas *Tempat Makan* memiliki atribut-atribut yang digunakan untuk menjelaskan detail informasi seputar tempat makan yang sedang dilihat, mulai dari Nama, Jarak, Rating, Jenis Masakan, Alamat, Jam Operasional, Menu Makanan, dan Ulasan. Dengan atribut-atribut tersebut, akan ada 2 metode yang dapat dijalankan:
  - a. `Search_tempat_makan`: untuk mencari tempat makan sesuai preferensi user.
  - b. `Add_to_favorite`: untuk menambahkan tempat makan tersebut ke daftar favorit user.
3. Kelas *Search* akan digunakan untuk menjalankan 4 fungsi antara lain: `search_by_nama` (berdasarkan nama tempat makan), `search_by_jenis_masakan` (berdasarkan jenis masakan), `search_by_relevan` (berdasarkan tempat paling

relevan) dan `search_by_jarak` (berdasarkan jarak antara pengguna dengan tempat makanan). Fungsi - fungsi ini dapat digunakan untuk mencari tempat makan yang sesuai dengan preferensi user dengan menggunakan kolom pencarian pada aplikasi.

4. Kelas *List* akan berisi semua tempat makan yang telah ditambahkan oleh user ke daftar favorite. Kelas ini akan berisi beberapa atribut tentang tempat makan yang telah ditambahkan ke dalam daftar favorit, seperti nama, jenis masakan, dan jarak. User juga dapat menggunakan atribut ini untuk mencari tempat makan tersebut di dalam daftar (`search_list`) atau untuk menghapusnya dari daftar (`delete_from_list`). Kelas ini juga memiliki kelas turunan yaitu Favorite List.
5. Kelas Profile akan terdiri dari beberapa atribut tentang pengguna, seperti Nama, Jenis Kelamin, Usia, Email dan Password. Dalam kelas ini pengguna dapat menjalankan beberapa metode seperti:
  - a. `Edit_profile`: untuk mengubah data profil user.
  - b. `Logout`: untuk keluar dari akun kita.

#### d. UI Design (Figma)

Link Desain Figma :

<https://www.figma.com/design/WForkLw5Ki6mkdIJ7UXVwA/Eatera?node-id=0-1&t=26cAitNiMxrHI8hc-o>

Link Prototype Figma :

<https://www.figma.com/proto/WForkLw5Ki6mkdIJ7UXVwA/Eatera?node-id=335-336&t=i2cFhZiL6fcBgz6L-1&scaling=scale-down&page-id=0%3A1&starting-point-node-id=3%3A8>

## Application Development Stage

### a. Implementation Stage

Pada pembuatan aplikasi Eatera, kami menggunakan metode waterfall yang dijabarkan dalam berbagai tahap berikut:

Perencanaan: Pertama kami akan melakukan analisis kebutuhan pengguna, hasil dari tahap ini adalah dokumen spesifikasi yang mencakup semua fitur dan fungsionalitas yang dibutuhkan oleh pengguna. Fitur utama yang akan diperlihatkan seperti pencarian masakan sesuai preferensi, menu, daftar favorit, serta ulasan pengguna.

Desain: Berdasarkan spesifikasi kebutuhan yang kami dapatkan, kami akan menciptakan desain UI/UX menggunakan figma, tertata agar desain mudah dan dinamis digunakan pengguna.

Pengembangan program: Disini kami akan menggunakan arsitektur MVVM, program akan dibuat sesuai spesifikasi desain.

Pengujian: Setelah pengembangan selesai, pengujian yang ketat dilakukan untuk memastikan bahwa semua fitur berfungsi dengan benar. Ini termasuk pengujian Unit menggunakan Mockito dengan Fake, Mock, dan Dummy, serta Pengujian UI alur dengan API menggunakan MockWebServer.

Pengimplementasian : Setelah pengujian berhasil, aplikasi diimplementasikan dan dibuat tersedia untuk pengguna. Ini termasuk pembuatan APK untuk aplikasi.

## b. Time Plan

Proyek dimulai pada tanggal 11 April 2024 dan akan berakhir pada tanggal 23 Mei 2024. Pengembangan Aplikasi Eatera akan dimulai pada tanggal 25 April 2024 dan diperkirakan akan memakan waktu sekitar 5 minggu untuk diselesaikan.

No	Task	Week 1	Week 2	Week 3	Week 4	Week 5
1	Build Retrofit dan Room DB					
2	Build Login dan Register					
3	Build Home, Profile, Explore, dan Favorit List					
4	Build Motion Layout, dan Splash Screen					
5	Implementasi Testing dan Build APK					