

BAB IV

PERANCANGAN DAN DESAIN

4.1. Perancangan Sistem

Perancangan sistem dalam pembuatan aplikasi ini merupakan proses awal untuk membangun kinerja sistem dari aplikasi yang akan dibuat. Sistem yang ada pada aplikasi *emergency call* berbasis *mobile hybrid* di Kota Tegal yang dibangun dengan *ionic framework* sebagai aplikasi yang digunakan dalam pembuatan aplikasi sekaligus sebagai *frontend* dan menyesuaikan tampilan antarmuka antara pengguna dengan sistem. Proses sistem ini dilakukan dengan cara membuat *Usecase Diagram*, *Activity Diagram*, *Sequence Diagram* dan *Class Diagram*.

4.1.1. Usecase Diagram

Usecase Diagram memperlihatkan hubungan diantara aktor dan *usecase*. Aktor merepresentasikan seorang *user* atau subsistem lain yang akan berinteraksi dengan sistem. Sedangkan *usecase* merupakan urutan kejadian yang menggambarkan interaksi antara *user* dengan sistem. Fungsionalitas sistem didefinisikan ke dalam *usecase* dari sudut eksternal sistem yang berguna untuk uji kelayakan sistem.

1. Identifikasi Aktor

Identifikasi aktor mendeskripsikan interaksi aktor dengan sistem yang ada di aplikasi *forntend* pada *emergency call* berbasis *mobile hybrid* di Kota Tegal. Sistem yang ada dapat di definisikan sebagai berikut:

Tabel 4.1 Identifikasi Aktor Frontend User

| No | Aktor | Nama Usecase |
|----|-------------|--|
| 1. | <i>User</i> | a. Registrasi akun b. <i>Login</i> c. Pelaporan aplikasi d. Edit profil akun <i>user</i> e. Melihat profil instansi f. Melihat tentang aplikasi g. Melihat cara pemakaian aplikasi h. <i>Logout</i> |

2. Identifikasi Diagram Usecase

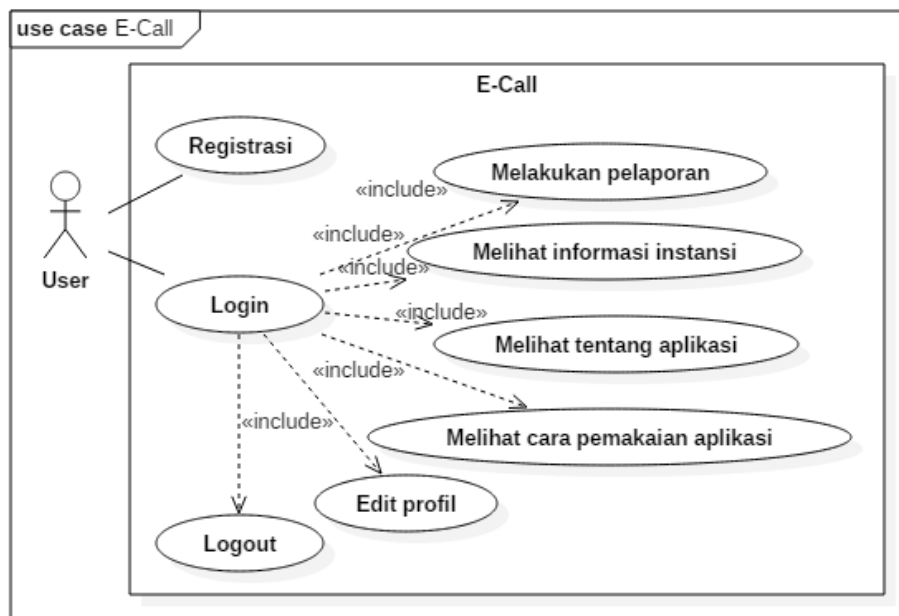
Identifikasi Diagram *Usecase* menggambarkan interaksi antara aktor dengan *system*.

Tabel 4.2. Identifikasi Diagram Usecase Frontend User

| No. | Nama Usecase | Deskripsi | Aktor |
|-----|------------------------------|--|-------------|
| 1. | Registrasi | Melakukan pendaftaran akun <i>user</i> | <i>User</i> |
| 2. | <i>Login</i> | Proses validasi akun <i>user</i> dan masuk ke dalam aplikasi | <i>User</i> |
| 3. | Pelaporan aplikasi | Melakukan pelaporan kejadian darurat | <i>User</i> |
| 4. | Edit profil akun <i>user</i> | Melakukan ubah data profil <i>user</i> | <i>User</i> |

| | | | |
|----|---------------------------------|---|------|
| 5. | Melihat profil instansi | Mengetahui informasi instansi | User |
| 6. | Melihat tentang aplikasi | Melihat informasi tentang aplikasi | User |
| 7. | Melihat cara pemakaian aplikasi | Melihat informasi cara pemakaian aplikasi | User |
| 8. | Logout | Keluar akun dari aplikasi | User |

Berikut adalah gambar model *Usecase Diagram* aplikasi *forntend* pada *emergency call* berbasis *mobile hybrid* di Kota Tegal :

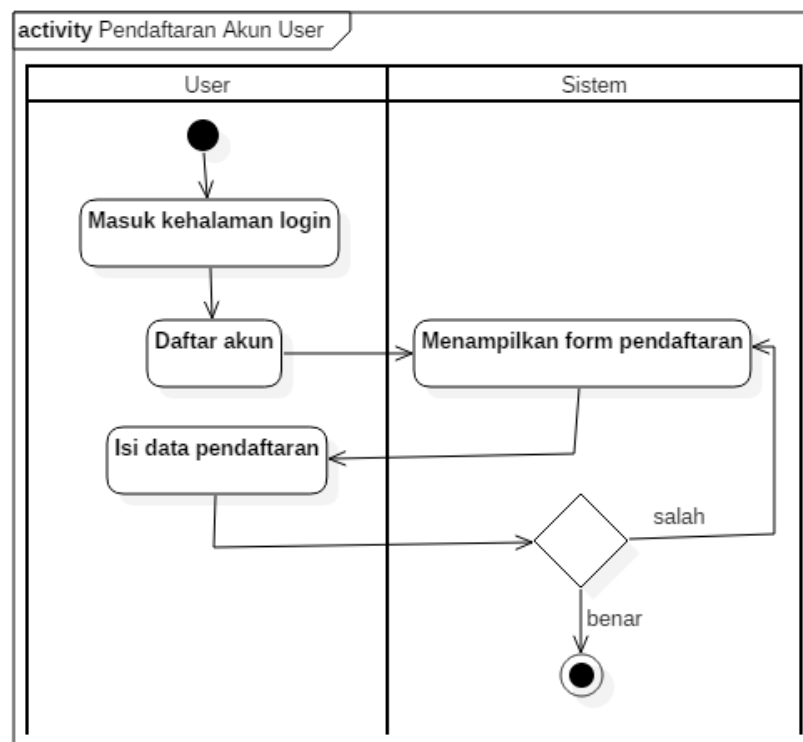


Gambar 4.1. Usecase Diagram Frontend Emergency Call

4.1.2. Activity Diagram

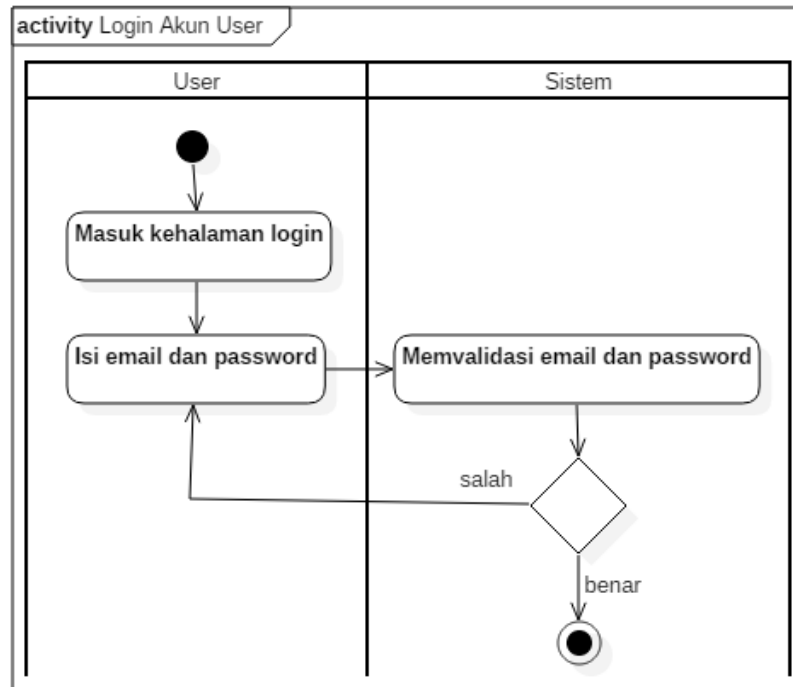
Activity diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana alur berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

a. Activity Diagram Pendaftaran Akun User



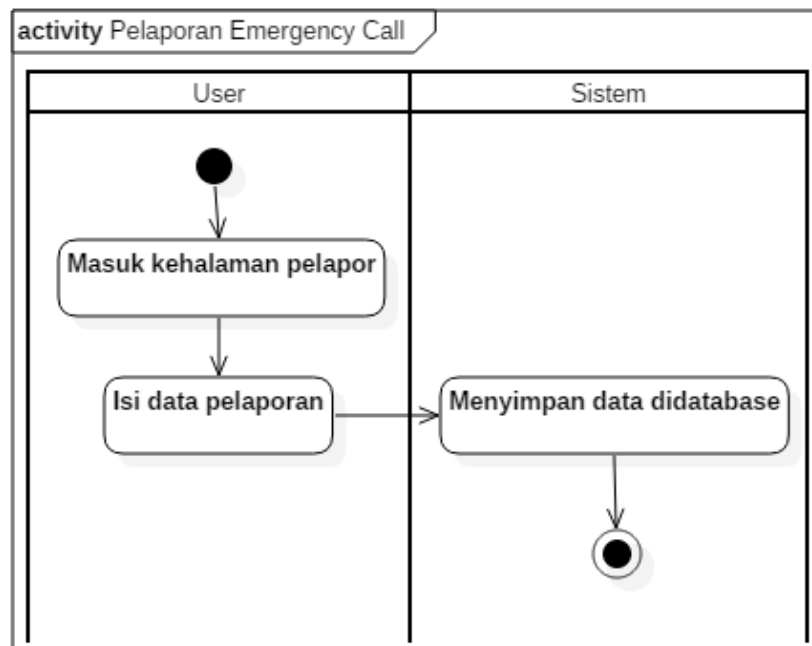
Gambar 4.2. Activity Diagram Pendaftaran Akun User

b. *Activity Diagram Login Akun User*

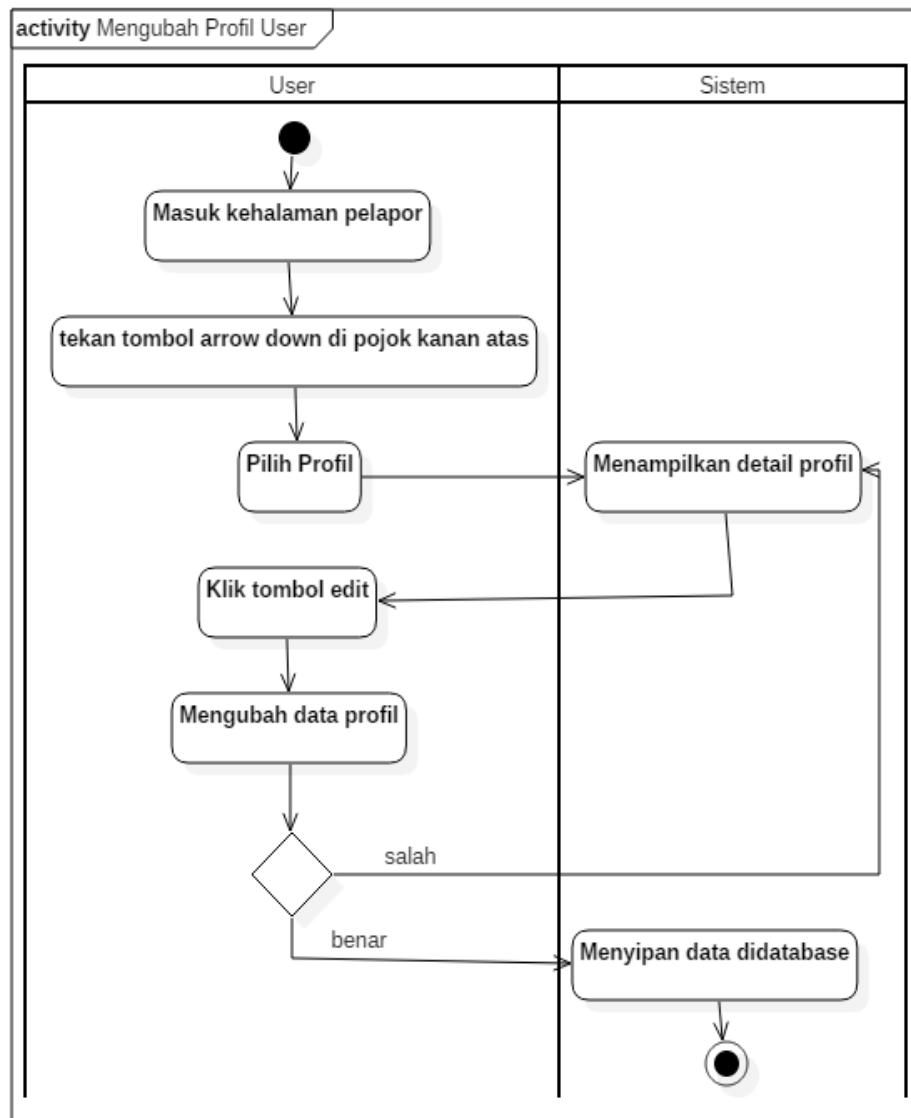


Gambar 4.3. Activity Diagram Login Akun User

c. *Activity Diagram Pelaporan*

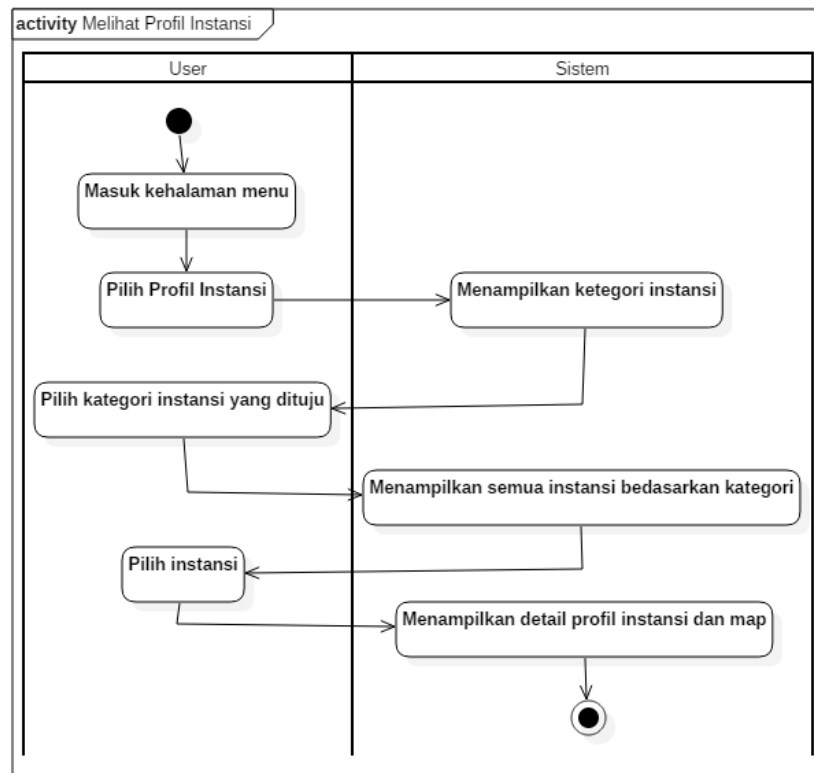


Gambar 4.4. Activity Diagram Pelaporan

d. *Activity Diagram Edit Profil User*

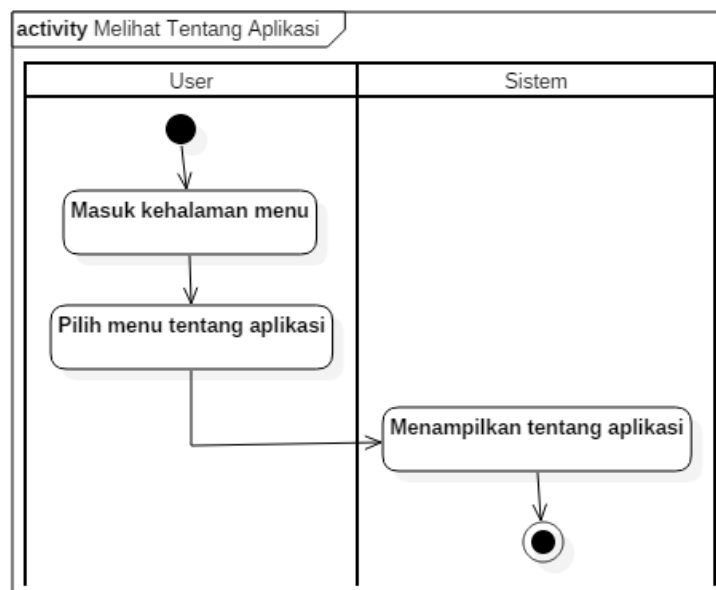
Gambar 4.5. Activity Diagram Edit Profil User

e. *Activity Diagram* Melihat Profil Instansi



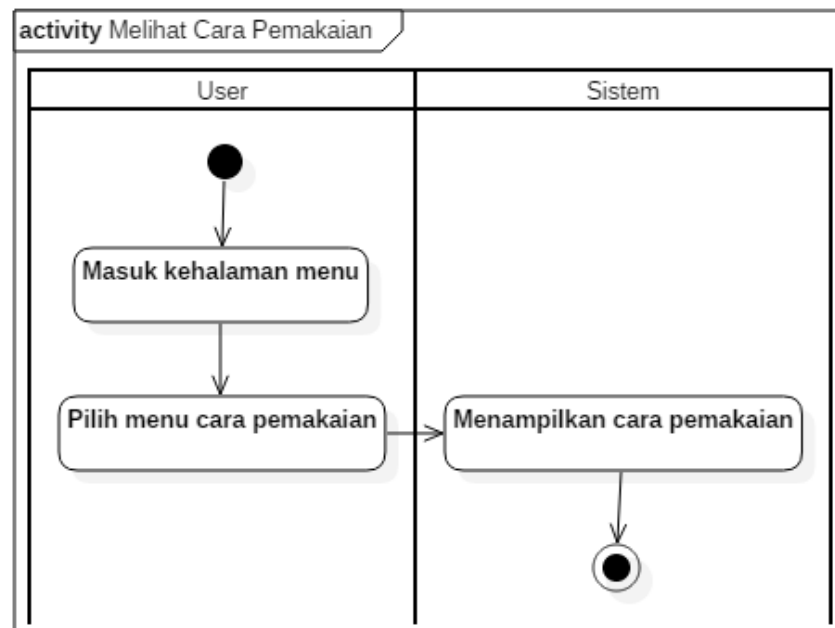
Gambar 4.6. Activity Diagram Melihat Profil Instansi

f. *Activity Diagram* Melihat Tentang Aplikasi



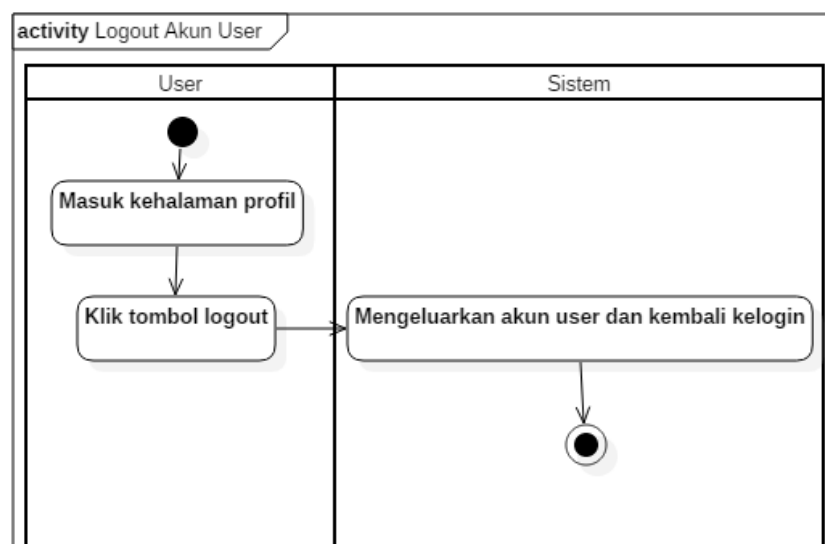
Gambar 4.7. Activity Diagram Melihat Tentang Aplikasi

g. *Activity Diagram* Melihat Cara Pemakaian Aplikasi



Gambar 4.8. Activity Diagram Melihat Cara Pemakaian Aplikasi

h. *Activity Diagram* Logout Akun User



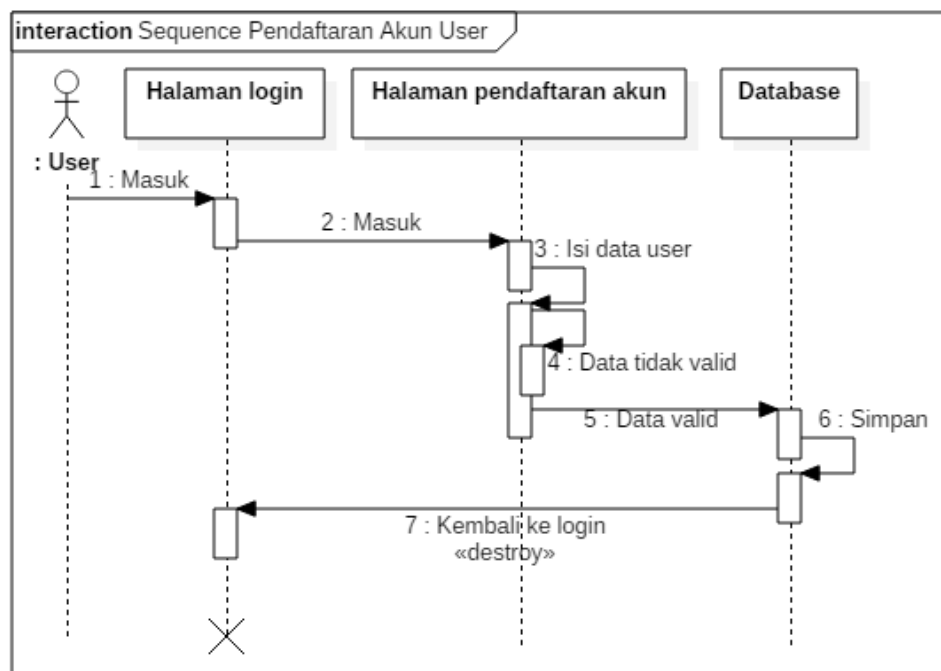
Gambar 4.9. Activity Diagram Logout Akun User

4.1.3. Sequence Diagram

Sequence diagram digunakan untuk menggambarkan interaksi antar objek dalam waktu yang berurutan. Tetapi pada dasarnya *sequence diagram* selain digunakan dalam lapisan abstraksi model objek. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama pesan diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan proses vertikal.

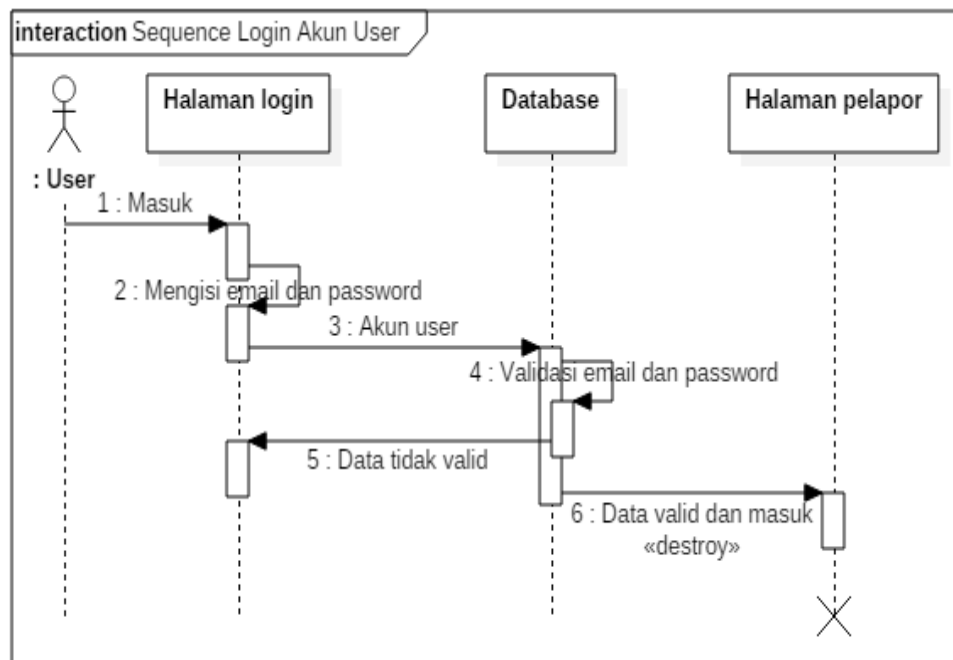
Berikut adalah *sequence diagram* yang ada pada aplikasi *frontend* pada *emergency call* berbasis *mobile hybrid* di Kota Tegal, yaitu:

a. Sequence Diagram Pendaftaran Akun User



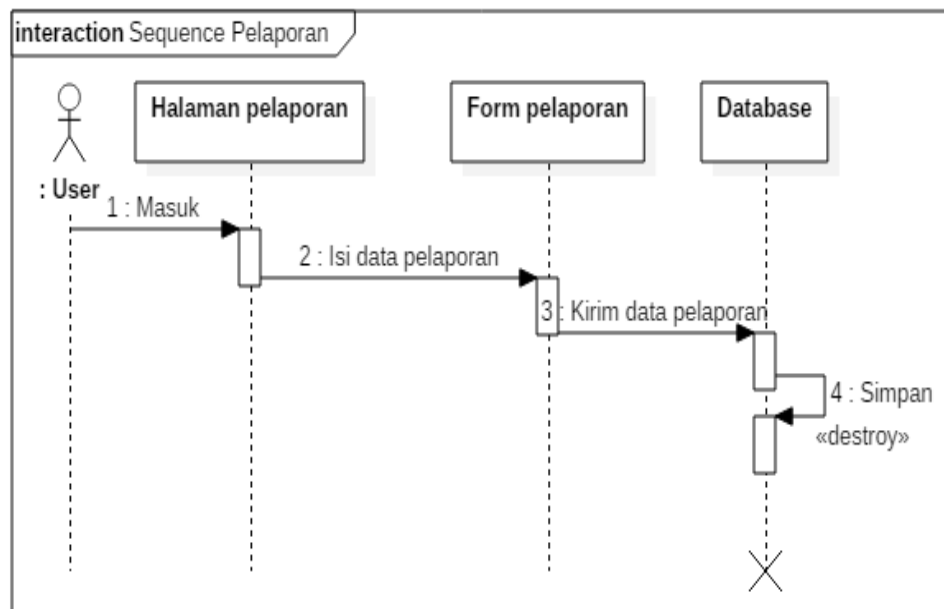
Gambar 4.10. Sequence Diagram Pendaftaran Akun User

b. *Sequence Diagram Login Akun User*



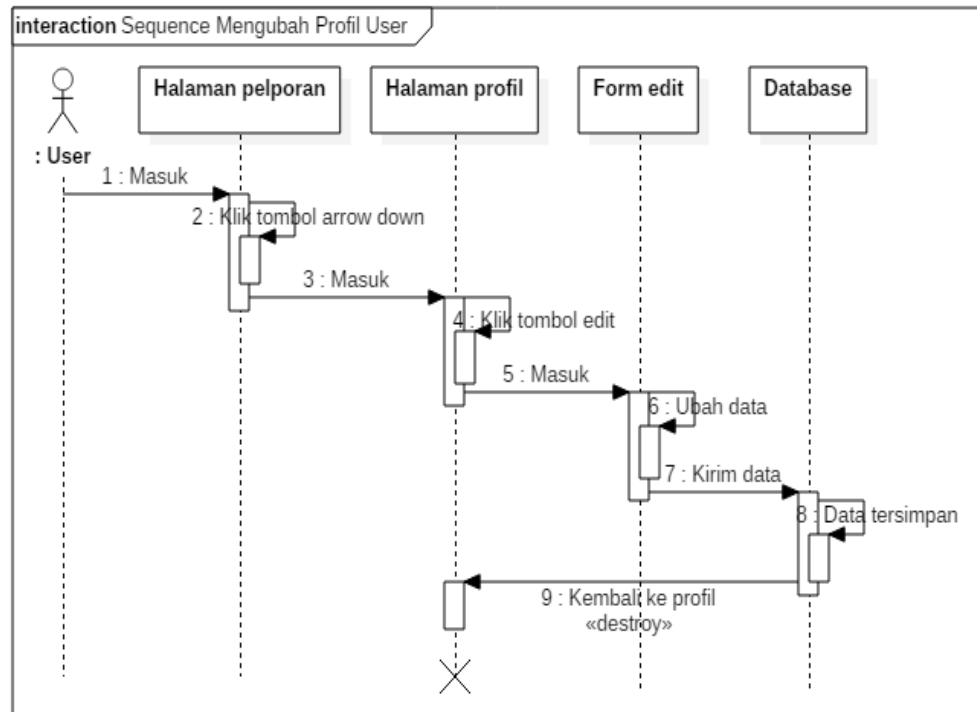
Gambar 4.11. Sequence Diagram Login Akun User

c. *Sequence Diagram Pelaporan*



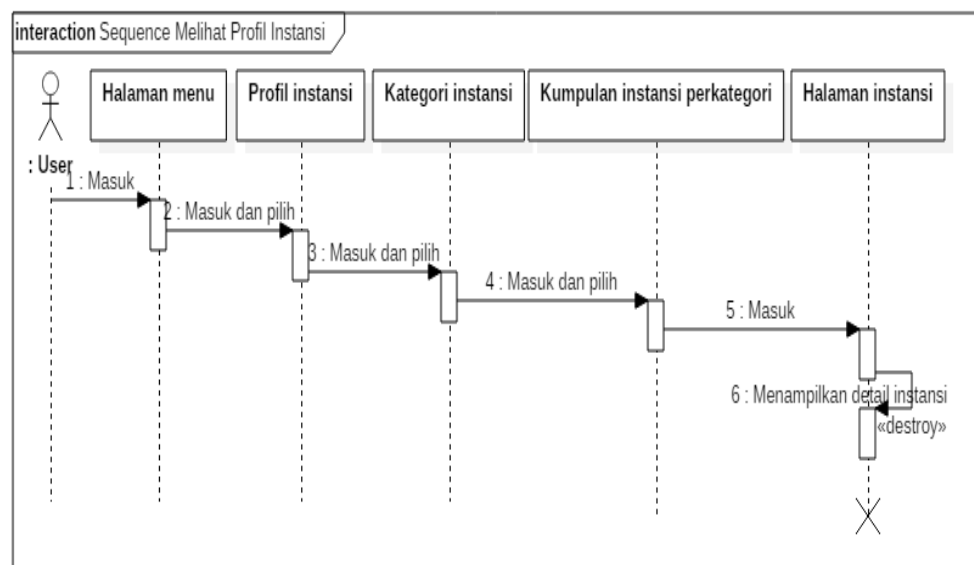
Gambar 4.12. Sequence Diagram Pelaporan

d. *Sequence Diagram Edit Profil User*



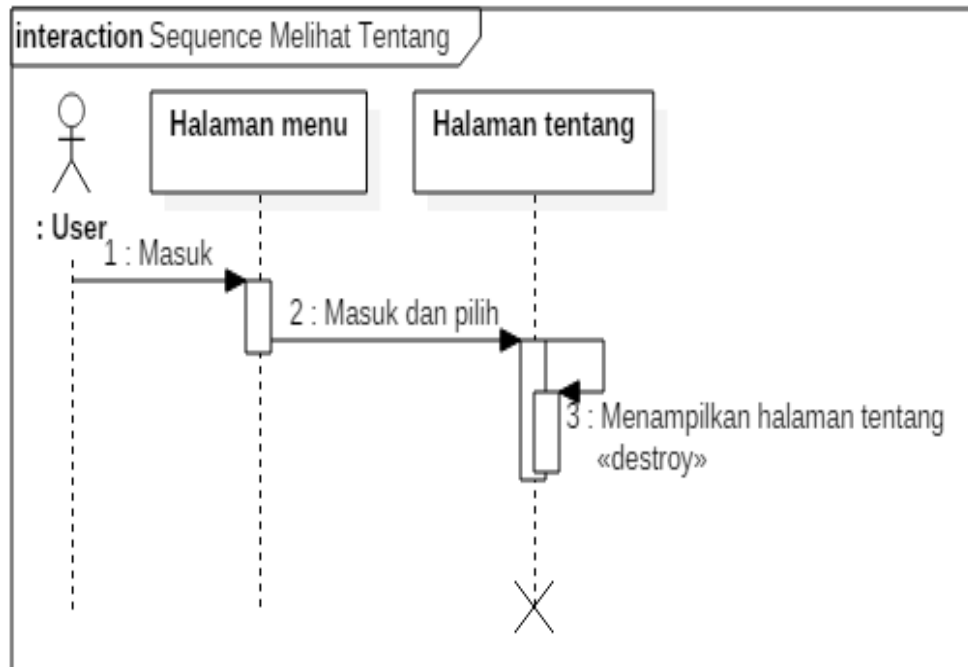
Gambar 4.13. Sequence Diagram Edit Profil User

e. *Sequence Diagram Melihat Profil Instansi*



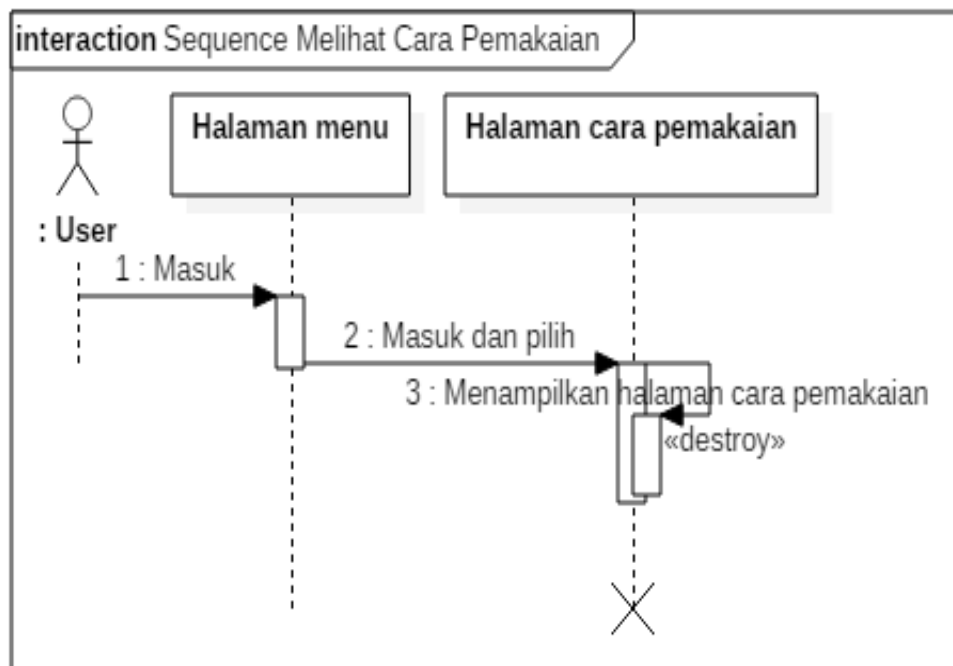
Gambar 4.14. Sequence Diagram Melihat Profil Instansi

f. *Sequence Diagram* Melihat Tentang Aplikasi



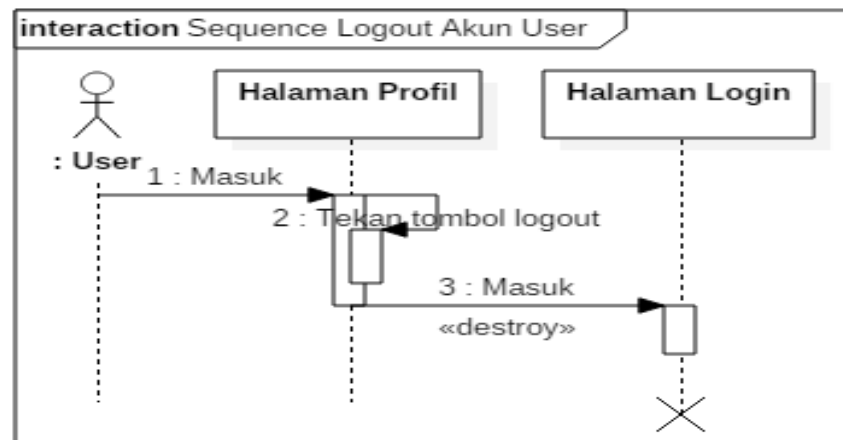
Gambar 4.15. Sequence Diagram Melihat Tentang Aplikasi

g. *Sequence Diagram* Melihat Cara Pemakaian Aplikasi



Gambar 4.16. Sequence Diagram Melihat Cara Pemakaian Aplikasi

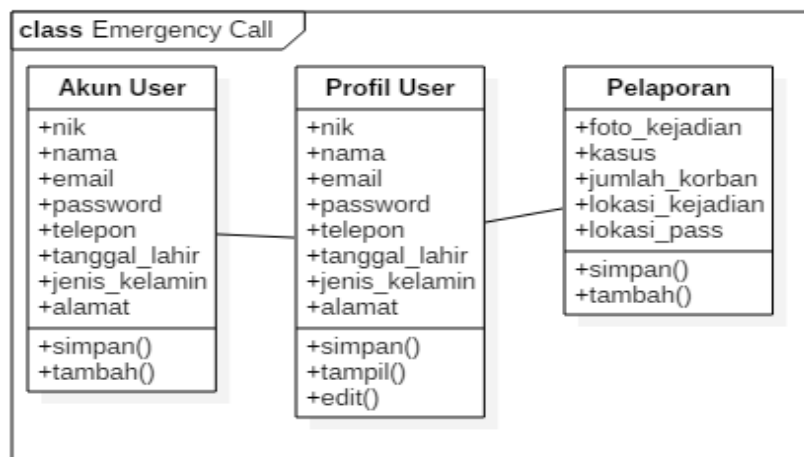
h. Sequence Diagram Logout Akun User



Gambar 4.17. Sequence Diagram Logout Akun User

4.1.4. Class Diagram

Class Diagram menggambarkan struktur statis *class* di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. Dengan melihat karakteristik sistem mulai dari bagian pendaftaran sampai dengan pelaporan kejadian darurat beserta proses-proses yang terjadi, maka dapat dibuat *class diagram*. Berikut *class diagram* aplikasi *frontend* pada *emergency call* berbasis *mobile hybrid* di Kota Tegal.



Gambar 4.18. Class Diagram Frontend Emergency Call

4.2. Desain Layout Aplikasi

Berikut ini merupakan rancangan desain aplikasi *frontend* pada *emergency call* berbasis *mobile hybrid* di Kota Tegal.

a. Desain *Layout* Halaman *Login Akun User*

The wireframe shows a mobile app interface for user login. It features a central container with a placeholder for a profile picture labeled 'No Image'. Below this are two input fields for 'Email' and 'Password'. A rounded button labeled 'Masuk' (Login) is positioned below the password field. Underneath the button is a link for 'Lupa Password ?' (Forgot Password?). At the bottom of the container is a link for 'Belum Punya Akun ? Daftar' (Don't have an account? Register).

Gambar 4.19. Desain Layout Halaman Login Akun User

b. Desain *Layout* Halaman Pendaftaran Akun User

The wireframe shows a mobile app interface for user registration. It features a central container with a vertical stack of input fields for 'NIK', 'Nama' (Name), 'Email', 'Password', 'Telepon' (Phone), 'Tanggal Lahir' (Date of Birth), 'Jenis Kelamin' (Gender), and 'Alamat' (Address). At the bottom of the container are two rounded buttons: 'Hapus' (Delete) and 'Daftar' (Register).

Gambar 4.20. Desain Layout Halaman Pendaftaran Akun User

c. Desain *Layout* Halaman Pelaporan

A mobile app layout for reporting an incident. It features a central form with the following fields: 'Foto Kejadian' (Incident Photo) with a camera icon, 'Kasus' (Case) with a dropdown arrow, 'Jumlah Korban' (Number of Victims), 'Lokasi Kejadian' (Incident Location), 'Alamat Kejadian' (Incident Address), and 'Kecamatan' (District) with a dropdown arrow. At the bottom of the form is a 'Kirim' (Send) button.

Gambar 4.21. Desain Layout Halaman Pelaporan

d. Desain *Layout* Halaman *Edit Profil User*

Three mobile app layouts for editing a user profile. The first layout shows a 'No Image' placeholder and a list of fields: NIK, Nama, Email, Telepon, Tanggal Lahir, and Jenis Kelamin. The second layout shows the same fields with a 'C' icon above them and 'Edit' and 'Logout' buttons at the bottom. The third layout shows the same fields with an 'Ambil Foto' (Take Photo) button at the top right and a 'Simpan' (Save) button at the bottom.

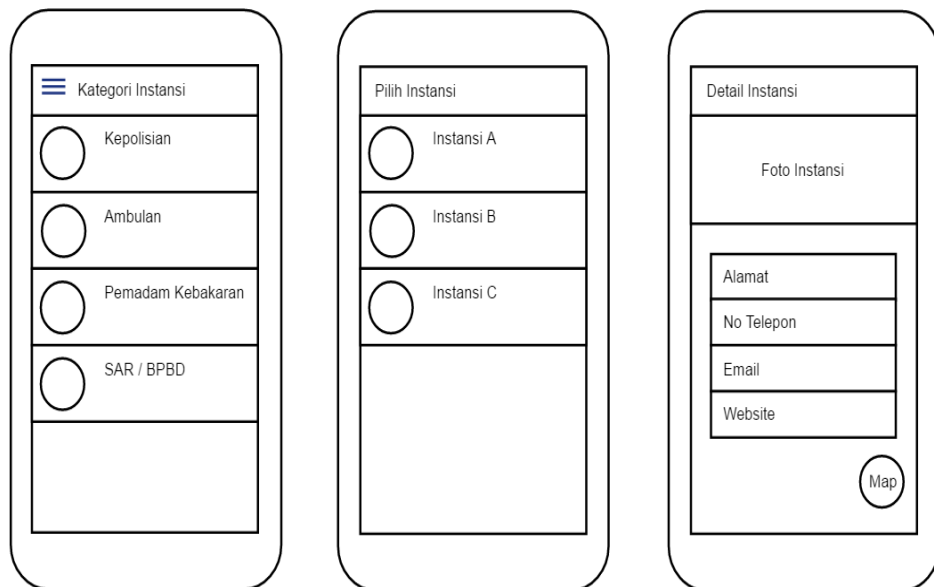
Gambar 4.22. Desain Layout Halaman Edit Profil User

e. Desain *Layout* Halaman Menu



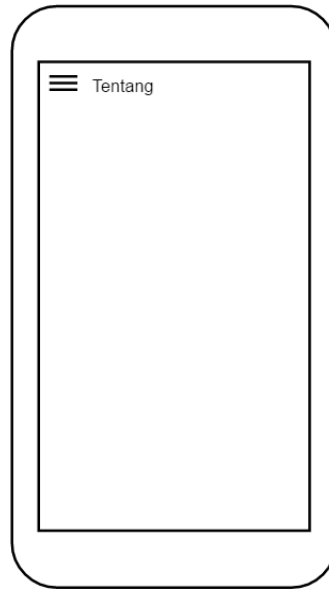
Gambar 4.23. Desain *Layout* Halaman Menu

f. Desain *Layout* Halaman Melihat Profil Instansi



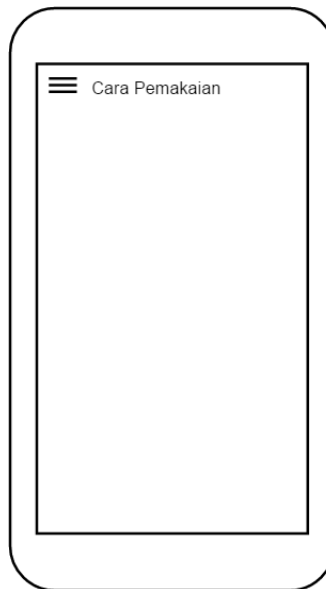
Gambar 4.24. Desain *Layout* Halaman Melihat Profil Instansi

g. Desain *Layout* Halaman Tentang Aplikasi



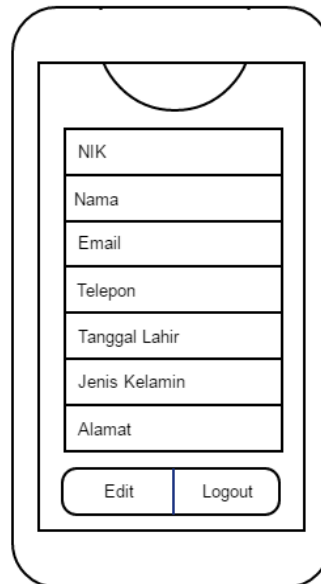
Gambar 4.25. Desain Layout Halaman Tentang

h. Desain *Layout* Halaman Cara Pemakaian Aplikasi



Gambar 4.26. Desain Layout Halaman Cara Pemakaian

i. Desain *Layout* Halaman *Logout*



NIK

Nama

Email

Telepon

Tanggal Lahir

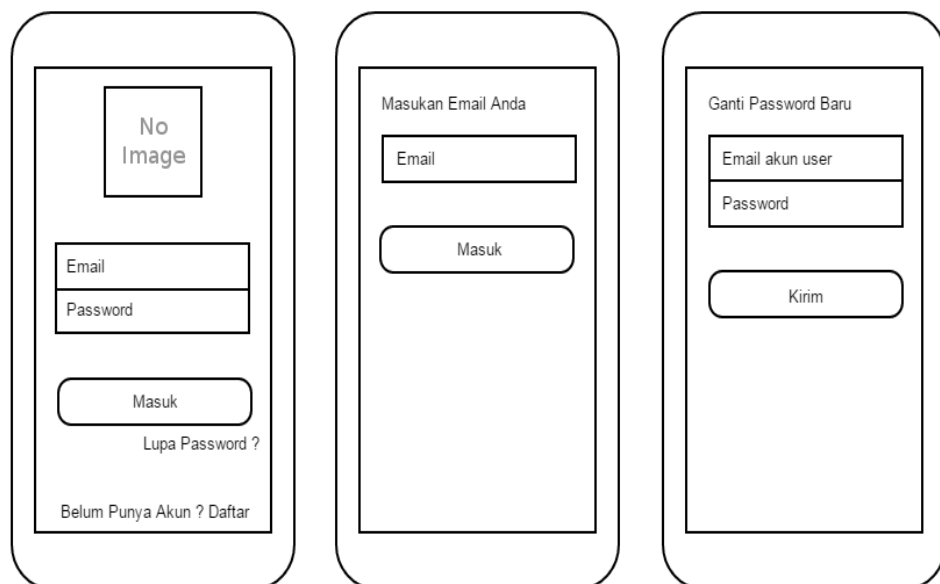
Jenis Kelamin

Alamat

Edit Logout

Gambar 4.27. Desain Layout Halaman Logout

j. Desain Layout Halaman Lupa *Password*



No Image

Email

Password

Masuk

Lupa Password ?

Belum Punya Akun ? Daftar

Masukan Email Anda

Email

Masuk

Ganti Password Baru

Email akun user

Password

Kirim

Gambar 4.28. Desain Layout Halaman Lupa Password