

## **BAB 3**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **3.1 Analisis Sistem**

Analisis sistem merupakan penjabaran dari suatu sistem yang utuh kedalam komponen-komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan maupun hambatan yang akan timbul pada tahap pembangunan sistem.

Analisis bertujuan untuk mendapatkan pemahaman secara keseluruhan tentang sistem, Serta untuk menentukan kebutuhan dari sistem yang dibangun. Analisis sistem merupakan tahap yang penting, karena kesalah dalam tahap ini akan menyebabkan pada tahap selanjutnya.

##### **3.1.1 Analisis Masalah**

Jatuh akan sangat berbahaya apabila dialami oleh seorang lansia maupun ibu hamil, karena dapat mengakibatkan gangguan kesehatan, cedera, bahkan sampai mengakibatkan kematian .

Kejadian jatuh sering kali tidak disadari oleh keluarga karena keluarga tidak dapat mendampingi setiap waktu. Dengan membangun aplikasi yang menyediakan informasi secara realtime, yang memberikan notifikasi ketika pengguna mengalami insiden jatuh, maka para pengguna lain yang memantaunya dapat segera memberikan pertolongan ketika seseorang mengalami kejadian jatuh sehingga dapat meminimalisir dampak yang akan diterima oleh penderita.

##### **3.1.2 Analisis Sistem yang Dibangun**

Analisis sistem yang dibangun merupakan gambaran lengkap sistem yang akan dibangun. Analisis sistem yang dibangun ini juga berdasarkan pada analisis masalah yang sudah dilakukan sebelumnya. Sehingga dari hasil analisis tersebut akan didapatkan gambaran dari sistem yang akan dibangun beserta prosedur dari aplikasi yang dibangun.

Pada penelitian ini aplikasi yang dibangun diberi nama *smart fall alert*. Sistem yang dibangun harus memenuhi kebutuhan dan solusi dari masalah yang ada yaitu memberikan informasi kepada keluarga ketika ada anggota keluarga (yang diawasi) memiliki resiko tinggi saat mengalami kejadian jatuh. Dari masalah utama tersebut maka gambaran aplikasi yang dibangun akan memiliki fitur untuk mendeteksi kejadian jatuh dan memberikan notifikasi kepada keluarga ketika kejadian tersebut terjadi.

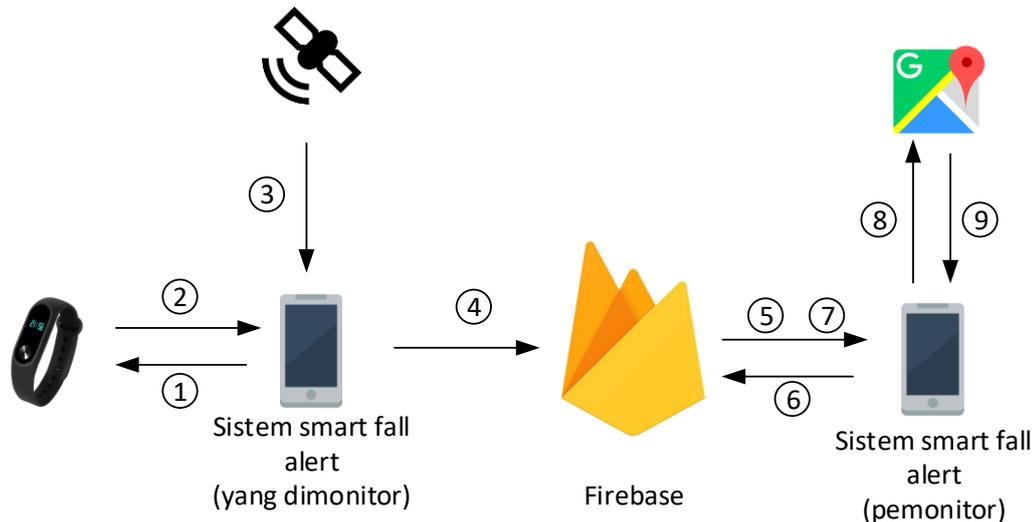
Adapun prosedur dari analisis sistem yang di bangun adalah sebagai berikut yang merupakan gambaran umum dari sistem yang dibangun:

- 1) Pengguna melakukan pendaftaran atau masuk akun.
- 2) Pengguna membuat grup monitoring
- 3) Pengguna (yang dimonitoring) mengaktifkan fitur pendeteksi jatuh pada aplikasi dan menghubungkan perangkat miband ke aplikasi.
- 4) Pengguna menambahkan id akun yang ingin di monitoring kedalam grup monitoring.
- 5) Menghubungkan miband dengan aplikasi, selanjutnya miband akan mengirimkan nilai *heart rate* yang diperoleh dari *wearable device*.
- 6) Jika sistem mendeteksi kejadian jatuh, sistem mengirimkan data *longitude* dan *latitude* dari GPS ke *firebase*.
- 7) Pengguna (pemonitoring) mendapat pemberitahuan bahwa pengguna (yang dimonitoring) mengalami kejadian jatuh, lalu menampilkan lokasinya pada peta *google*.
- 8) Pengguna (pemonitoring) dapat menghubungi pengguna (pemonitoring) lain menggunakan fitur *direct whatsapp* yang dimiliki oleh aplikasi.
- 9) Memberikan informasi rumah sakit terdekat ke pengguna dengan memanfaatkan *location based service*.
- 10) Pengguna dapat melihat riwayat kejadian jatuh.

### **3.1.3 Analisis Arsitektur Sistem**

Tahap analisis ini bertujuan untuk mengidentifikasi arsitektur sistem yang akan dibangun. Baik itu gambaran pengguna hingga transaksi data yang akan

dibangun. Gambaran arsitektur sistem dari aplikasi yang akan dibangun dapat dilihat pada Gambar 3.1 Arsitektur Sistem.



**Gambar 3.1 Arsitektur Sistem**

Berikut merupakan penjelasan dari arsitektur sistem yang akan dibangun :

1. Sistem meminta data heart dari Mi band.
2. Miband mengirimkan data heart rate yang di minta oleh sistem.
3. Sistem mendeteksi lokasi user saat itu dengan fasilitas GPS melalui satelit.
4. Sistem mengirimkan data kejadian jatuh ke *firebase database*.
5. *Firestore* cloud messaging mengirimkan notifikasi ke sistem aplikasi pemonitor.
6. Sistem pemonitor meminta data kejadian jatuh ke *firebase*.
7. *Firestore* mengirimkan data yang diminta oleh sistem pemonitor.
8. Sistem mengirimkan data *longitude* dan *latitude* ke *google maps API*.
9. *Google maps API* menampilkan lokasi kejadian jatuh dari data *longitude* dan *latitude* yang telah dikirimkan.

### 3.1.4 Analisis Deteksi Jatuh

Analisis deteksi jatuh dilakukan dengan tujuan untuk melakukan perhitungan percepatan dari data yang diperoleh oleh sensor *accelerometer*. Gerakan tubuh manusia bisa tercermin dari akselerasi. Ketika orang-orang berjalan, turun, naik, melompat, berlari, atau jatuh, data akselerasi bervariasi. Algoritma pada

penelitian ini menggunakan nilai puncak *Signal Vector Magnitude* (SVM) untuk mengenali kejadian jatuh [28].

Nilai SVM akan dihitung sebagai berikut :

$$SVM = \sqrt{(Ax)^2 + (Ay)^2 + (Az)^2}$$

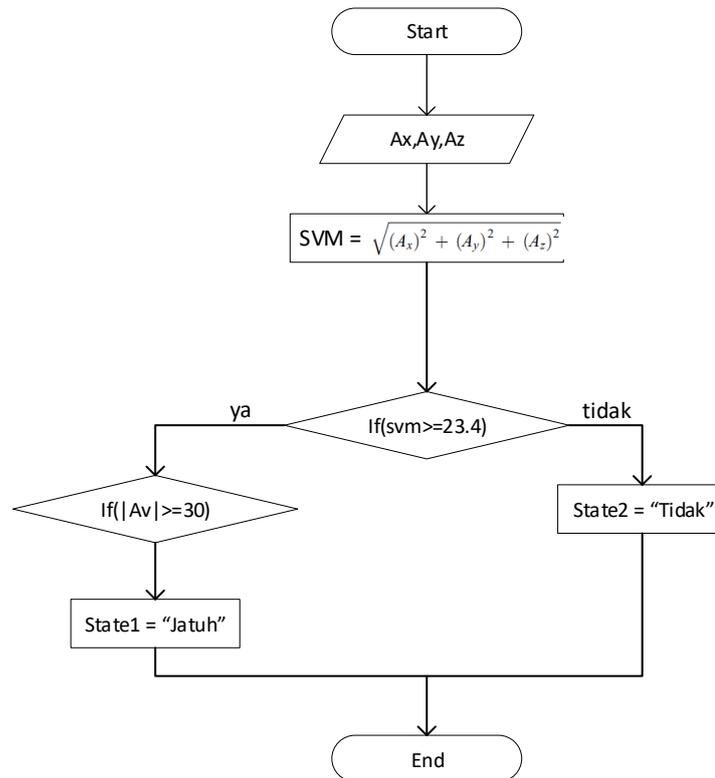
Di mana Ax, Ay, dan Az mewakili kecepatan percepatan masing-masing sumbu X, Y, dan Z dari *accelerometer*. Sebagian besar kejadian jatuh memiliki nilai SVM lebih tinggi daripada aktivitas sehari-hari. Ketika nilai SVM lebih dari sama dengan nilai threshold yang telah ditentukan yaitu 23,4 (m/s<sup>2</sup>), maka sistem akan memberikan statement bahwa kejadian jatuh telah terdeteksi.

Selain itu, algoritma pendeteksian ini jatuh juga menambahkan akselerasi vertikal sebagai prosedur untuk memverifikasi. Akselerasi vertikal nilai  $|A_v|$  yang dihitung sebagai berikut:

$$|A_v| = |A_x \sin \theta_z + A_y \sin \theta_y - A_z \cos \theta_y \cos \theta_z|$$

Dan akselerasi vertikal juga dapat meningkatkan akurasi di mana  $\theta_y$  dan  $\theta_z$  menunjukkan nilai pitch and roll-nya ditentukan oleh *accelerometer* dan *magnetometer* dari sensor *smartphone*. Dalam perhitungan ini, nilai threshold  $Th|A_v|$  yang telah ditentukan adalah 30.

Alur dari algoritma pendeteksi jatuh yang akan digunakan, dapat dilihat pada Gambar 3.2 *Flow Chart* Algoritma Deteksi Jatuh.



**Gambar 3.2 Flow Chart Algoritma Deteksi Jatuh**

### 3.1.5 Analisis Teknologi

Analisis teknologi digunakan untuk mengetahui teknologi apa saja yang dibutuhkan dalam pembangunan aplikasi pendeteksi jatuh.

#### 3.1.5.1 Analisis Sensor *Accelerometer*

Sensor *accelerometer* merupakan perangkat elektronik yang digunakan untuk mengukur percepatan dari suatu objek [9]. Accelerometer ini mengukur percepatan, bahwa perangkat mengalami perubahan yang sesuai dengan nilai dari tiga sumbu X,Y,Z lalu data dari tiga sumbu tersebut akan dihitung untuk mengetahui percepatan atau perlambatan yang terjadi. Percepatan adalah suatu keadaan yang mengakibatkan perubahan bertambahnya kecepatan terhadap waktu, sedangkan apabila perubahan berkurangnya kecepatan terhadap waktu disebut perlambatan. *Accelerometer* yang diletakkan di permukaan bumi dapat mendeteksi percepatan 1g (ukuran gravitasi bumi) pada titik vertikalnya. Sedangkan percepatan yang arah pergerakannya secara horizontal, sensor *accelerometer* akan mengukur

percepatannya secara langsung ketika bergerak secara horizontal [10]. Perhitungan nilai *accelerometer* akan difokuskan pada setiap sumbunya, karena nilai keseluruhan *accelerometer* berpengaruh untuk mendeteksi kejadian jatuh.

### 3.1.5.2 Analisis Pengambilan dan Pemanfaatan Mi Band 2

Aplikasi yang dibangun menggunakan miband sebagai alat bantu untuk mengukur nilai *heart rate* dari orang yang dimonitoring. Wristband ini sendiri dapat digunakan di tangan kanan maupun di tangan kiri, yang akan berfungsi untuk mendeteksi jumlah darah yang mengalir. Mi band menggunakan lampu LED hijau yang dipasangkan dengan fotodioda yang peka terhadap cahaya untuk mendeteksi jumlah darah yang mengalir di pergelangan tangan. Saat darah mengalir di pergelangan tangan akan memantulkan cahaya, dan cahaya tersebut akan dibaca oleh foto dioda, darah yang tidak memantulkan cahaya berarti volume darah lebih tinggi. Melalui pantulan cahaya tersebut, sensor akan menganalisa data dan menghitung nilai *heart rate*. Dengan mengedipkan lampu LED ratusan kali perdetik, Mi Band dapat menghitung frekuensi jantung berdenyut. Pengambilan denyut jantung dapat dilihat pada Gambar 3.3 Pengambilan Menggunakan Mi Band.



**Gambar 3.3 Pengambilan Denyut Menggunakan Mi Band**

Sumber : <https://www.mi.co.id/id/miband2/>

Data nilai *heart rate* yang diperoleh akan dikirimkan ke aplikasi yang telah terpasang di *smartphone* menggunakan koneksi Bluetooth 4.0 yang terdapat pada Mi Band. selanjutnya nilai *heart rate* tersebut akan digunakan sebagai pengukur nilai detak jantung orang yang di monitoring. Untuk memberikan informasi detak jantung kepada pemonitoring, setelah orang yang dimonitoring mengalami kejadian jatuh. Data nilai *heart rate* akan selalu diperbarui per 10 detik. Apabila nilai *heart*

rate < 60 Bpm maka sistem akan menyatakan orang yang di monitoring mengalami kondisi bradikardia [14].

Miband membutuhkan autentikasi untuk mendapatkan nilai heart rate yang berupa UUID (Universally Unique Identifier). UUID merupakan kode identifikasi unik yang diberikan oleh sistem, Maksud dari UUID adalah untuk memungkinkan sistem untuk secara unik mengidentifikasi informasi. Berikut UUID (Universally Unique Identifier) yang akan digunakan sebagai komunikasi antara Mi Band dengan aplikasi yang akan dibangun:

1. *UUID of service*  
0000fee0-0000-1000-8000-00805f9b34fb
2. *Hardware service (HRDW) UUID*  
0000fee0-0000-1000-8000-00805f9b34fb
3. *Heart Monitor Service (HMS) UUID*  
0000180d-0000-1000-8000-00805f9b34fb
4. *Descriptor UUID*  
00002902-0000-1000-8000-00805f9b34fb
5. *Device information service UUID*  
0000180a-0000-1000-8000-00805f9b34fb
6. *Alert notification service UUID*  
00001811-0000-1000-8000-00805f9b34fb

### **3.1.5.3 Analisis Terjemahan Google Place API**

Pada penelitian ini *Places* API akan digunakan di dalam sistem yang dibangun. Pemilihan dan penggunaan *Places* API ini berdasarkan pada fungsionalitas sistem yang akan dibangun untuk menyediakan informasi mengenai informasi rumah sakit terdekat. Dengan penggunaan *Places* API ini aplikasi yang akan dibangun dapat menampilkan informasi lokasi rumah sakit terdekat dari pengguna yang mengalami kejadian jatuh.

Cara kerja dan penggunaan *Places* API ini sebetulnya berkaitan dengan platform web yang difungsikan sebagai API pada sistem yang dibangun berikut detail penggunaannya.

1. Buat request untuk meminta informasi rumah sakit terdekat.

```
https://maps.googleapis.com/maps/api/place/nearbysearch/json?location
=-
6.883820,107.612500&radius=10000&type=hospital&sensor=true&key
=AIzaSyAiG-qLaa9q7ezcuCGA9XNpl8Ty7vaq0vg
```

2. Berikut adalah contoh respond JSON berhasil dari *Places* API.

```
{
  "html_attributions" : [],
  "next_page_token"      :
  "CqQCHwEAAEBzIROrWQL0Ic3_zb6NaOyXj5ktCVvGPCGp
kwPhvZnmXKbONNudNxF7H5XmKv6oGOQYsL4UV6pl2O8
FVs3SCETPQd8iycJiOLrOJMRgXAnbFfPIBxUwGDX5CKMa
mGxZJ7J_0aYWHkvUzuykobLWuJmkeDqOWklqpuuco4rNwY
Zi0LFtqgnbz1RR2YL8Mppzr_QECFZf_owvGNQ6fZs1nZRnlp
MFdgATs3TbXL0Td_jdB0dvhsvD3mAQFIwhknqubQ7M6Sn-
Me_IZ6wmPAyqEB4O3OFpZ_V4-Iza1Hb-
PoFE1dbwrRVHm85j6LorypcmU2qLooK9UcmzxGuTmVTijyc
aCtFQU-
tTab6705mcxzZdyi2ORMNqILRfFZ8H4TK6zRIQONKu3i-
VV1SimiE6wOGa4BoUgf_eVBmsVosC440UbsPxeTBXXOc",
  "results" : [
    {
      "geometry" : {
        "location" : {
          "lat" : -6.898102199999999,
          "lng" : 107.5981391
        },
        "viewport" : {
          "northeast" : {
            "lat" : -6.894116499999999,
```

```

        "lng" : 107.5996799
      },
      "southwest" : {
        "lat" : -6.902093699999999,
        "lng" : 107.5965927
      }
    }
  },
  "icon" :
  "https://maps.gstatic.com/mapfiles/place_api/icons/doctor-71.png",
  "id" : "6328f356e3af1a027e5c6020d656a9bfe967c0db",
  "name" : "Dr. Hasan Sadikin Central General Hospital",
  "opening_hours" : {
    "open_now" : true
  },
  "photos" : [
    {
      "height" : 4000,
      "html_attributions" : [
        "\u003ca
href=\"https://maps.google.com/maps/contrib/113547426646689361389/photos\" \u003eanggi fernandi\u003c/a\u003e"
      ],
      "photo_reference" : "CmRaAAAATa9BOOKWMMoHs-t4HX8tNz5p7wnlh-gZXyuAlSDqTKaZWWNNVAC4Kfg9bCdZQeaUt4do1XFiLyKLIajqm5VLTC4XO0JQIueLoV21edpngW6Vk5IAT8pnJ1CJKysXUikNPEhDMRyn74q-Np-z3gY4qCWweGhTnSZLxnIfzFI4Ux2kPLXM0uxEqCQ",
      "width" : 3000
    }
  ]
}

```

```

    }
  ],
  "place_id" : "ChIJWTAYp0LmaC4R5WtgJNL_etA",
  "plus_code" : {
    "compound_code" : "4H2X+Q7 Sukajadi, Bandung City,
West Java, Indonesia",
    "global_code" : "6P594H2X+Q7"
  },
  "rating" : 3.5,
  "reference" : "ChIJWTAYp0LmaC4R5WtgJNL_etA",
  "scope" : "GOOGLE",
  "types" : [ "hospital", "point_of_interest", "establishment" ],
  "user_ratings_total" : 727,
  "vicinity" : "Jalan Pasteur No.38, Pasteur"
}
],
"status" : "OK"
}

```

### 3.1.6 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak dibagi menjadi dua kebutuhan yaitu kebutuhan non fungsional dan fungsional. Kebutuhan fungsional dapat dilihat pada Tabel 3.1 Spesifikasi Kebutuhan Fungsional.

**Tabel 3.1 Spesifikasi Kebutuhan Fungsional**

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-001	Sistem menyediakan fungsi registrasi
SKPL-F-002	Sistem menyediakan fungsi <i>login</i>
SKPL-F-003	Sistem menyediakan fungsi pembuatan grup
SKPL-F-004	Sistem menyediakan fungsi melihat daftar grup
SKPL-F-005	Sistem menyediakan fungsi penambahan anggota grup

SKPL-F-006	Menyediakan fungsi untuk menerima permintaan gabung grup
SKPL-F-007	Sistem menyediakan fungsi untuk mengaktifkan fitur deteksi jatuh.
SKPL-F-008	Sistem menyediakan fungsi untuk menghubungkan aplikasi dengan perangkat Mi Band.
SKPL-F-009	Sistem menyediakan fungsi tampil daftar anggota grup
SKPL-F-010	Sistem menyediakan fasilitas untuk menampilkan lokasi kejadian jatuh
SKPL-F-011	Sistem menyediakan informasi nilai heart rate pengguna yang mengalami kejadian jatuh
SKPL-F-012	Sistem menyediakan fungsi untuk keluar dari grup
SKPL-F-013	Sistem menyediakan fungsi riwayat kejadian jatuh
SKPL-F-014	Sistem menyediakan informasi lokasi rumah sakit terdekat
SKPL-F-015	Sistem menyediakan fasilitas perubah data akun pengguna
SKPL-F-016	Sistem menyediakan fungsi untuk keluar dari aplikasi

Kebutuhan non fungsional pada perangkat lunak yang dibangun dapat dilihat pada Tabel 3.2 Spesifikasi Kebutuhan Non Fungsional

**Tabel 3.2 Spesifikasi Kebutuhan Non Fungsional**

Kode SKPL	Spesifikasi Kebutuhan perangkat Lunak
SKPL-NF-001	Sistem yang dibangun berbasis android
SKPL-NF-002	Sistem yang dibangun minimal sistem operasi android versi 5.0 <i>Lollipop</i> ke atas.
SKPL-NF-003	Sistem yang dibangun menggunakan sensor <i>accelerometer</i>
SKPL-NF-004	Desain user interface pada sistem yang dibangun mengacu pada <i>guideline design android google</i>
SKPL-NF-005	Menggunakan <i>wearable device</i> Mi Band

### 3.1.7 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi

elemen-elemen apa saja yang dibutuhkan oleh sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan oleh sistem, keluaran yang akan dihasilkan oleh sistem serta proses yang dibutuhkan untuk mengolah masukan dan menghasilkan keluaran sesuai dengan apa yang dibutuhkan.

Analisis kebutuhan non fungsional pada sistem ini meliputi analisis perangkat lunak, perangkat keras, dan perangkat pikir.

#### 1. *Product Requirement*

Berikut adalah kebutuhan non fungsional berdasarkan klasifikasi *product requirement*.

1. SKPL-NF-001 Sistem yang dibangun menggunakan mobile
2. SKPL-NF-002 Sistem yang dibangun menggunakan minimal sistem operasi android versi 5.0 *lollipop*, dan dibangun dengan kebutuhan perangkat lunak minimal :

**Tabel 3.3 SKPL-NF-002**

No.	Kebutuhan Perangkat Lunak
1	Sistem Operasi ( <i>Platform</i> bebas)
2	<i>Integrated development environment</i> (IDE) berupa android studio
3	<i>Android development tools</i> versi 21 sebagai <i>pulgins</i> android studio
4	<i>Android software development kit</i> sebagai <i>framework</i> android
5	<i>Text Editor</i>
6	<i>Web Browser</i>

3. SKPL-NF-003 Sistem yang dibangun menggunakan sensor *accelerometer*.

#### 2. *Organisational Requirement*

Berikut adalah kebutuhan non fungsional berdasarkan klasifikasi *organisational requirement*.

SKPL-NF-004 Desain *user interface* pada perangkat lunak yang akan dibangun mengacu pada desain *guidelines* android *google*.

### 3. *External Requirement*

Berikut adalah kebutuhan non fungsional berdasarkan klasifikasi *external requirement*.

SKPL-NF-005 Sistem membutuhkan perangkat *wearable device* Mi band

### 4. Analisis Kebutuhan Non Fungsional Perangkat Pikir

Analisis kebutuhan non fungsional perangkat pikir adalah batasan-batasan dari layanan-layanan dan fungsi-fungsi yang dibutuhkan dari sebuah sistem dari segi perangkat pikir. Dalam penelitian ini dikelompokkan menjadi 3 kategori yaitu *User Knowledge and Experience*, *User Job and Task Characteristic* and *User Physical Characteristic*.

#### 1. *User Knowledge and Experience*

*User knowledge and experience* yang diharapkan dari *user* perangkat lunak pada penelitian ini dapat dilihat pada Tabel 3.4 *User Knowledge and Experience*

**Tabel 3.4 *User Knowledge and Experience***

Knowledge and Experiences	
Komputer <i>Literacy</i>	Cukup sampai tinggi
Sistem <i>Experiences</i>	Rendah sampai tinggi
Application <i>Experiences</i>	Rendah sampai tinggi
Task <i>Experiences</i>	Rendah sampai tinggi
Other Sistem Use	<i>Frequent</i>
Education	SMP sampai Perguruan Tinggi
Reading Level	Sedang sampai tinggi
Typing Skill	Sedang sampai tinggi
Native Language or Culture	Indonesia

#### 2. *User Job and Task Characteristic*

*User Job and Task Characteristic* yang diharapkan dari *user* perangkat lunak pada penelitian ini dapat dilihat pada Tabel 3.5 *User Job and Task Characteristic*.

**Tabel 3.5 *User Job and Task Characteristic***

<i>User Job and Task Characteristic</i>
---

<i>Task Structure</i>	Tinggi
<i>Social Interactions</i>	Perlu
<i>Primary Training</i>	None
<i>Job Category</i>	None
<i>Frequency of Use</i>	Rendah
<i>Task or Need Importance</i>	Medium

### 3. *User Physical Characteristic*

*User Physical Characteristic* yang diharapkan dari *user* perangkat lunak pada penelitian ini dapat dilihat pada Tabel 3.6 *User Physical Characteristic*.

**Tabel 3.6 *User Physical Characteristic***

<i>User Physical Characteristic</i>	
<i>Age</i>	<i>Middle aged, and elderly</i>
<i>Gender</i>	<i>Male and Female</i>
<i>Handedness</i>	<i>Both</i>
<i>Disabilities</i>	<i>None</i>

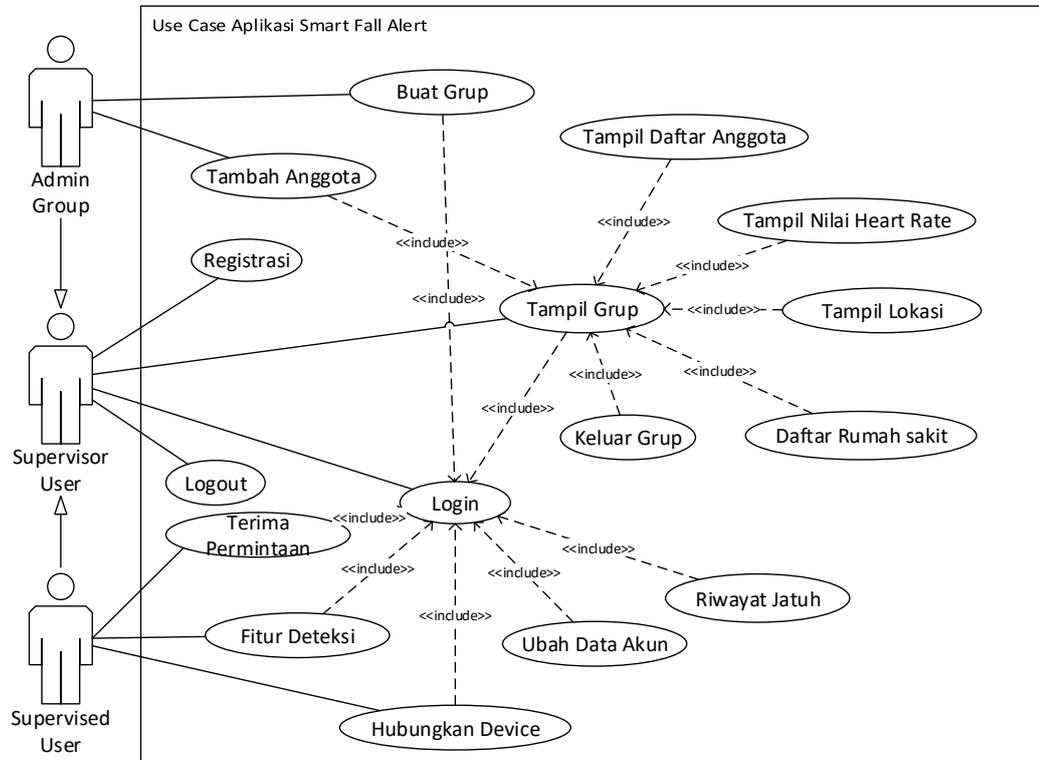
### 3.1.8 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan analisis yang menjelaskan penggambaran, perencanaan dan pembuatan dan pembuatan sketsa dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Analisis kebutuhan fungsional menentukan bagaimana suatu sistem akan menyelesaikan apa yang mesti diselesaikan.

#### 3.1.8.1 *Usecase Diagram*

*Usecase diagram* merupakan alat yang digunakan dalam pengembangan perangkat lunak atau sistem untuk menangkap kebutuhan fungsional dari sistem terkait. *Use case diagram* menggambarkan sekelompok *use case* dan aktor yang disertai dengan hubungan diantaranya. *Use case diagram* ini menjelaskan dan menerangkan kebutuhan yang diinginkan/ dikehendaki user/pengguna, serta sangat berguna dalam menentukan struktur organisasi dan model dari sebuah sistem. Pada

sistem yang akan dibuat menggunakan *use case* diagram yang dapat dilihat pada Gambar 3.4 *Usecase Diagram*.



**Gambar 3.4 Use Case Diagram**

Definisi *actor* dari pembangunan aplikasi *smart fall alert* dapat dilihat pada Tabel 3.7 Definisi *Actor*.

**Tabel 3.7 Definisi Actor**

No.	Actor	Deskripsi
1.	<i>Supervisor user</i>	Actor dengan role ini adalah pengguna yang memonitoring/mengawasi <i>supervised user</i> dan memiliki wewenang pada level <i>user interface</i>
2.	<i>Admin Group</i>	Actor dengan role ini memiliki wewenang sebagai admin dari grup yang dibuat
3.	<i>Supervised user</i>	Actor dengan role ini adalah pengguna yang dimonitoring/diawasi oleh <i>supervisor user</i> dan memiliki wewenang pada level <i>user interface</i>

Definisi use case dari pembangunan aplikasi *smart fall alert* dapat dilihat pada Tabel 3.8 Definisi Use Case.

**Tabel 3.8 Definisi Use Case**

No.	Use Case	Deskripsi
1	Registrasi	Merupakan proses pembuatan akun yang dilakukan oleh pengguna.
2	<i>Login</i>	Merupakan proses yang dilakukan oleh pengguna untuk mengakses aplikasi
3	Buat Grup	Merupakan proses pembuatan grup
4	Tampil Grup	Merupakan proses untuk menampilkan daftar grup yang di ikuti
5	Tambah Pengguna	Merupakan proses untuk menambah pengguna di grup
6	Terima Permintaan	Merupakan proses untuk menerima permintaan ketika ditambahkan di grup
7	Fitur Deteksi	Merupakan fungsi untuk mengaktifkan sistem deteksi jatuh
8	Hubungkan Perangkat	Merupakan proses yang dilakukan untuk menghubungkan perangkat MiBand ke aplikasi
9	Tampil Daftar Anggota	Merupakan proses untuk melihat daftar anggota grup
10	Tampil Lokasi	Merupakan proses untuk melihat informasi lokasi setiap pengguna pada peta <i>google</i>
11	Tampil Nilai <i>Heart Rate</i>	Menampilkan nilai detak jantung pengguna yang di monitoring
12	Keluar Grup	Merupakan fungsi yang dilakukan oleh pengguna untuk keluar dari grup
13	Riwayat Jatuh	Merupakan proses untuk menampilkan informasi riwayat jatuh yang terjadi

14	Daftar Rumah Sakit	Merupakan proses untuk menampilkan informasi rumah sakit terdekat
15	Ubah Data Akun	Merupakan proses yang dilakukan oleh pengguna untuk merubah data personal
16	<i>Logout</i>	Merupakan proses yang dilakukan oleh pengguna untuk keluar akun dari aplikasi

### 3.1.8.2 Skenario *Use Case*

Use case skenario merupakan *flow of event* untuk *use case* utama yang dapat menggambarkan urutan interaksi aktor dengan *use case* tersebut dimulai dari awal aktor berinteraksi hingga selesai.

1. Skenario *use case* registrasi digunakan ketika pengguna ingin membuat akun baru. Dapat dilihat pada Tabel 3.9 Skenario *Use Case* Registrasi.

**Tabel 3.9 Skenario *Use Case* Registrasi**

<i>Use Case Name</i>	Registrasi	
<i>Goal Context</i>	<i>User</i> berhasil melakukan pembuatan akun	
<i>Preconditions</i>	<i>User</i> belum memiliki akun	
<i>Successful End Condition</i>	Sistem berhasil menyimpan data <i>user</i>	
<i>Failed End Condition</i>	Sistem gagal menyimpan data <i>user</i>	
<i>Primary Actors</i>	<i>Supervisor user</i>	
<i>Trigger</i>	<i>User</i> menekan tombol daftar	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1.	<i>User</i> menekan tombol daftar
	2.	Sistem menampilkan form pendaftaran
	3.	<i>User</i> mengisi form yang disediakan
	4.	<i>User</i> menekan tombol daftar
	5.	Sistem memvalidasi form daftar
	6.	Menampilkan pesan pendaftaran <i>user</i> berhasil tersimpan
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>

	4.1	Menampilkan pesan kesalahan pendaftaran
--	-----	---

2. Skenario *use case login* digunakan ketika pengguna ingin melakukan *login*. Dapat dilihat pada Tabel 3.10 Skenario *Use Case Login*.

**Tabel 3.10 Skenario Use Case Login**

<i>Use Case Name</i>	<i>Login</i>	
<i>Goal Context</i>	<i>User</i> berhasil melakukan <i>login</i>	
<i>Preconditions</i>	<i>User</i> membuka aplikasi	
<i>Successful End Condition</i>	Berhasil <i>login</i> kedalam sistem	
<i>Failed End Condition</i>	Menampilkan pesan kesalahan	
<i>Primary Actors</i>	<i>Supervisor user</i>	
<i>Trigger</i>	<i>User</i> menekan tombol masuk	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1.	Sistem menampilkan form <i>login</i>
	2.	<i>User</i> mengisi form <i>login</i>
	3.	<i>User</i> menekan tombol <i>login</i>
	4.	Sistem melakukan validasi form <i>login</i>
	5.	Sistem melakukan autektikasi akun
	6.	Sistem menampilkan halaman utama
<i>Extensions</i>	<i>Step</i>	<i>Branching Action</i>
	4.1	Menampilkan pesan kesalahan
	5.1	Menampilkan pesan kesalahan autentikasi

3. Skenario *use case* buat grup digunakan ketika pengguna ingin membuat grup monitoring. Dapat dilihat pada Tabel 3.11 Skenario *Use Case Buat Grup*.

**Tabel 3.11 Skenario Use Case Buat Grup**

<i>Use Case Name</i>	<i>Buat Grup</i>
<i>Goal Context</i>	<i>User</i> berhasil membuat grup
<i>Preconditions</i>	<i>User</i> belum memiliki grup
<i>Successful End Condition</i>	<i>User</i> berhasil membuat grup
<i>Failed End Condition</i>	Menampilkan pesan kegagalan

<i>Primary Actors</i>	<i>Admin Group</i>	
<i>Trigger</i>	<i>User menekan tombol buat</i>	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1.	User menekan tombol buat
	2.	Sistem menampilkan form buat grup
	3.	User mengisi nama grup
	4.	User menekan tombol buat grup
	5.	Sistem memvalidasi nama form buat grup
	6.	Sistem menampilkan pesan grup berhasil dibuat
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	5.1	Menampilkan pesan kesalahan

4. Skenario *use case* tampil grup digunakan ketika pengguna ingin melihat daftar grup yang diikuti. Dapat dilihat pada Tabel 3.12 Skenario *Use Case* Tampil Grup.

**Tabel 3.12 Skenario *Use Case* Tampil Grup**

<i>Use Case Name</i>	<b>Tampil Grup</b>	
<i>Goal Context</i>	User dapat melihat daftar grup monitoring yang diikuti	
<i>Preconditions</i>	List daftar grup ditampilkan	
<i>Successful End Condition</i>	Berhasil menampilkan <i>list</i> daftar grup	
<i>Failed End Condition</i>	List daftar grup tidak berhasil ditampilkan	
<i>Primary Actors</i>	<i>Supervisor user</i>	
<i>Trigger</i>	<i>User menekan tombol grup</i>	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1.	<i>User menekan tombol grup</i>
	2.	Menampilkan halaman grup
	3.	Sistem melakukan <i>request</i> data ke <i>firebase</i>
	4.	Sistem menampilkan daftar grup

<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	4.1	Menampilkan pesan “belum memiliki grup”

5. Skenario *use case* tambah pengguna digunakan ketika pengguna ingin menambah pengguna lain di grup. Dapat dilihat pada Tabel 3.13 Skenario *Use Case* Tambah Pengguna.

**Tabel 3.13 Skenario *Use Case* Tambah Pengguna**

<i>Use Case Name</i>	<b>Tambah Pengguna</b>	
<i>Goal Context</i>	Menambahkan <i>user</i> ke grup monitoring	
<i>Preconditions</i>	Belum menambahkan user ke grup	
<i>Successful End Condition</i>	Berhasil menambahkan user ke grup	
<i>Failed End Condition</i>	Tidak dapat menambahkan user ke grup	
<i>Primary Actors</i>	<i>Admin Group</i>	
<i>Trigger</i>	<i>User</i> menekan tombol tambah pengguna	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1.	<i>User</i> menekan tombol tambah
	2.	Sistem menampilkan form tambah pengguna
	3.	<i>User</i> mengisikan email pengguna yang ingin ditambahkan
	4.	<i>User</i> menekan tombol cari
	5.	Sistem mencari email user
	6.	Sistem menampilkan nama pengguna
	7.	<i>User</i> menekan tombol tambahkan
	8.	Sistem menampilkan pesan berhasil mengirim permintaan
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	5.1	Sistem menampilkan pesan email tidak ditemukan

6. Skenario *use case* terima permintaan digunakan ketika pengguna ingin menerima permintaan untuk bergabung di grup. Dapat dilihat pada Tabel 3.14 Skenario *Use Case* Terima Permintaan.

**Tabel 3.14 Skenario Use Case Terima Permintaan**

<i>Use Case Name</i>	<b>Terima Permintaan</b>	
<i>Goal Context</i>	Menerima permintaan gabung grup monitoring	
<i>Preconditions</i>	Belum bergabung di grup	
<i>Successful End Condition</i>	Berhasil bergabung di grup	
<i>Failed End Condition</i>	Tidak dapat bergabung ke grup	
<i>Primary Actors</i>	1. <i>Supervisor user</i> 2. <i>Supervised user</i>	
<i>Trigger</i>	<i>User</i> menekan tombol terima	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1.	<i>User</i> menekan tombol permintaan.
	2.	Sistem menampilkan halaman permintaan
	3.	Sistem melakukan <i>request</i> data ke <i>firebase</i> .
	4.	Sistem menampilkan list permintaan gabung grup
	3.	<i>User</i> menekan tombol terima .
	4.	Sistem menampilkan pesan berhasil bergabung kedalam grup.
<i>Extension</i>	<i>Step</i>	<i>Action</i>
	4.1	Sistem menampilkan pesan tidak ada permintaan gabung grup

7. Skenario *use case* fitur deteksi digunakan ketika pengguna ingin mengaktifkan fitur deteksi jatuh. Dapat dilihat pada Tabel 3.15 Skenario *Use Case* Fitur Deteksi.

**Tabel 3.15 Skenario Use Case Fitur Deteksi**

<i>Use Case Name</i>	<b>Fitur Deteksi</b>
<i>Goal Context</i>	User berhasil mengaktifkan fitur deteksi jatuh
<i>Preconditions</i>	Fitur deteksi jatuh belum aktif

<i>Successful End Condition</i>	Fitur deteksi jatuh berhasil diaktifkan	
<i>Failed End Condition</i>	Tidak berhasil mengaktifkan fitur deteksi jatuh	
<i>Primary Actors</i>	<i>Supervised user</i>	
<i>Trigger</i>	<i>User</i> menekan tombol <i>switch</i>	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1.	<i>User</i> menekan tombol akun
	2.	Sistem menampilkan halaman akun
	3.	<i>User</i> menekan tombol <i>switch</i>
	4.	Sistem menjalankan fungsi deteksi jatuh
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	3.1	Menampilkan pesan deteksi jatuh berhasil diaktifkan

8. Skenario *use case* hubungan perangkat digunakan ketika pengguna ingin menghubungkan *wearable device* ke aplikasi. Dapat dilihat pada Tabel 3.16 Skenario *Use Case* Hubungan Perangkat.

**Tabel 3.16 Skenario *Use Case* Hubungan Perangkat**

<i>Use Case Name</i>	<b>Hubungan Perangkat</b>	
<i>Goal Context</i>	<i>Wearable device</i> berhasil dihubungkan	
<i>Preconditions</i>	<i>Wearable device</i> belum terhubung	
<i>Successful End Condition</i>	Sistem berhasil terhubung dengan <i>wearable device</i>	
<i>Failed End Condition</i>	Sistem tidak berhasil terhubung dengan <i>wearable device</i>	
<i>Primary Actors</i>	<i>Supervised user</i>	
<i>Trigger</i>	<i>User</i> menekan tombol akun	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1.	<i>User</i> menekan tombol akun
	2.	Sistem menampilkan halaman akun
	3.	<i>User</i> menekan tombol hubungan perangkat
	4.	Sistem menampilkan form masukan

	5.	User mengisikan mac address wearable device
	6.	User menekan tombol hubungkan
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	6.1	Menampilkan pesan perangkat terhubung

9. Skenario *use case* tampil daftar anggota digunakan ketika pengguna ingin melihat daftar anggota grup. Dapat dilihat pada Tabel 3.17 Skenario *Use Case* Tampil Daftar Anggota.

**Tabel 3.17 Skenario *Use Case* Tampil Daftar Anggota**

<i>Use Case Name</i>	<b>Tampil Daftar Anggota</b>	
<i>Goal Context</i>	Menampilkan daftar anggota grup monitoring	
<i>Preconditions</i>	Sistem menampilkan daftar grup	
<i>Successful End Condition</i>	Berhasil menampilkan daftar anggota grup	
<i>Failed End Condition</i>	Tidak dapat menampilkan daftar anggota grup	
<i>Primary Actors</i>	<i>Supervisor user</i>	
<i>Trigger</i>	<i>User</i> menekan tombol anggota grup	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1.	<i>User</i> menekan tombol anggota grup
	2.	Sistem melakukan <i>request</i> data ke <i>firebase</i>
	3.	Sistem menampilkan daftar anggota grup

10. Skenario *use case* tampil lokasi digunakan ketika pengguna ingin melihat lokasi anggota grup. Dapat dilihat pada Tabel 3.18 Skenario *Use Case* Tampil Lokasi.

**Tabel 3.18 Skenario *Use Case* Tampil Lokasi**

<i>Use Case Name</i>	<b>Tampil Lokasi</b>	
<i>Goal Context</i>	Menampilkan lokasi user yang mengalami kejadian jatuh pada sebuah grup di peta <i>google</i>	
<i>Preconditions</i>	Sistem menampilkan daftar grup	
<i>Successful End Condition</i>	Berhasil menampilkan lokasi user di peta <i>google</i>	

<i>Failed End Condition</i>	Tidak dapat menampilkan lokasi user di peta <i>google</i>	
<i>Primary Actors</i>	<i>Supervisor user</i>	
<i>Trigger</i>	<i>User</i> memilih pemberitahuan	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1.	User memilih grup monitoring yang ingin dilihat lokasi anggotanya
	2.	User memilih pemberitahuan kejadian jatuh.
	3.	Sistem melakukan <i>request</i> data lokasi ke <i>firebase</i>
	4.	Sistem menampilkan lokasi pada peta <i>google</i>

11. Skenario *use case* tampil nilai *heart rate* digunakan ketika pengguna ingin mengetahui nilai *heart rate* pengguna yang dimonitoring. Dapat dilihat pada Tabel 3.19 Skenario *Use Case* Tampil Nilai *Heart Rate*.

**Tabel 3.19 Skenario *Use Case* Tampil Nilai *Heart Rate***

<i>Use Case Name</i>	<b>Tampil Nilai <i>Heart Rate</i></b>	
<i>Goal Context</i>	Menampilkan nilai <i>heart rate</i> user yang mengalami kejadian jatuh	
<i>Preconditions</i>	Sistem menampilkan nilai <i>heart rate</i>	
<i>Successful End Condition</i>	Berhasil menampilkan nilai <i>heart rate</i>	
<i>Failed End Condition</i>	Tidak dapat menampilkan nilai <i>heart rate</i>	
<i>Primary Actors</i>	<i>Supervisor user</i>	
<i>Trigger</i>	<i>User</i> memilih grup monitoring	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1.	<i>User</i> memilih grup monitoring
	2.	User memilih pemberitahuan kejadian jatuh
	3.	Sistem melakukan request data nilai <i>heart rate</i> ke <i>firebase</i>
	4.	Sistem menampilkan nilai <i>heart rate</i>

12. Skenario *use case* keluar grup digunakan ketika user ingin keluar dari grup monitoring. Dapat dilihat pada Tabel 3.20 Skenario *Use Case* Keluar Grup.

**Tabel 3.20 Skenario *Use Case* Keluar Grup**

<i>Use Case Name</i>	<b>Keluar Grup</b>	
<i>Goal Context</i>	Keluar dari grup monitoring	
<i>Preconditions</i>	Sistem menampilkan daftar grup	
<i>Successful End Condition</i>	Berhasil keluar dari grup	
<i>Failed End Condition</i>	Tidak berhasil keluar dari grup	
<i>Primary Actors</i>	<i>Supervisor user</i>	
<i>Trigger</i>	<i>User</i> memilih grup	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1.	<i>User</i> memilih grup monitoring
	2.	<i>User</i> menekan tombol pilihan
	3.	<i>User</i> menekan tombol keluar grup
	4.	Sistem menampilkan halaman daftar grup
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	4.1	Menampilkan pesan berhasil keluar dari grup

13. Skenario *use case* riwayat jatuh digunakan ketika pengguna ingin melihat daftar riwayat jatuh. Dapat dilihat pada Tabel 3.21 Skenario *Use Case* Tampil Riwayat Jatuh.

**Tabel 3.21 Skenario *Use Case* Riwayat Jatuh**

<i>Use Case Name</i>	<b>Riwayat Jatuh</b>
<i>Goal Context</i>	Menampilkan riwayat kejadian jatuh
<i>Preconditions</i>	Sistem menampilkan halaman riwayat kejadian jatuh
<i>Successful End Condition</i>	Sistem berhasil menampilkan riwayat kejadian jatuh
<i>Failed End Condition</i>	Tidak berhasil menampilkan riwayat kejadian jatuh
<i>Primary Actors</i>	<i>Supervisor user</i>

<i>Trigger</i>	<i>User</i> menekan riwayat	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1.	User menekan tombol riwayat
	2.	Menampilkan halaman riwayat
	3.	Sistem melakukan request data riwayat jatuh ke <i>firebase</i>
	4.	Sistem menampilkan riwayat kejadian jatuh

14. Skenario *use case* daftar rumah sakit digunakan ketika pengguna ingin melihat daftar rumah sakit terdekat. Dapat dilihat pada Tabel 3.22 Skenario *Use Case* Daftar Rumah Sakit.

**Tabel 3.22 Skenario *Use Case* Daftar Rumah Sakit**

<i>Use Case Name</i>	<b>Daftar Rumah Sakit</b>	
<i>Goal Context</i>	Menampilkan daftar rumah sakit	
<i>Preconditions</i>	Sistem menampilkan daftar anggota grup	
<i>Successful End Condition</i>	Sistem berhasil menampilkan daftar rumah sakit terdekat dari lokasi user yang jatuh	
<i>Failed End Condition</i>	Tidak berhasil menampilkan daftar rumah sakit	
<i>Primary Actors</i>	<i>Supervisor user</i>	
<i>Trigger</i>	<i>User</i> menekan tombol rumah sakit	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1.	User menekan tombol rumah sakit
	2.	Sistem akan menghitung jarak rute terdekat rumah sakit dari lokasi user yang jatuh
	3.	Sistem menampilkan daftar rumah dari jarak yang terdekat

15. Skenario *use case* ubah data akun digunakan ketika pengguna ingin merubah data pribadi pengguna. Dapat dilihat pada Tabel 3.23 Skenario *Use Case* Ubah Data Akun.

**Tabel 3.23 Skenario *Use Case* Ubah Data Akun**

<i>Use Case Name</i>	<b>Ubah Data Akun</b>	
<i>Goal Context</i>	User berhasil merubah data akun	
<i>Preconditions</i>	Sistem menampilkan data akun	
<i>Successful End Condition</i>	Sistem berhasil menyimpan perubahan data akun	
<i>Failed End Condition</i>	Sistem tidak berhasil menyimpan perubahan data akun	
<i>Primary Actors</i>	<i>Supervisor user</i>	
<i>Trigger</i>	<i>User</i> menekan tombol akun	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1.	<i>User</i> menekan tombol akun
	2.	Sistem menampilkan halaman akun
	3.	<i>User</i> menekan tombol ubah
	4.	Sistem menampilkan form perubahan data
	5.	<i>User</i> mengisi form perubahan data
	6.	<i>User</i> menekan tombol simpan
	7.	Sistem memvalidasi form perubahan data
	8.	Sistem menyimpan perubahan data akun
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	7.1	Menampilkan pesan kesalahan
	8.1	Menampilkan pesan data akun berhasil diubah

16. Skenario *use case* logout digunakan ketika pengguna ingin keluar dari aplikasi.

Dapat dilihat pada Tabel 3.24 Skenario *Use Case Logout*.

**Tabel 3.24 Skenario Use Case Logout**

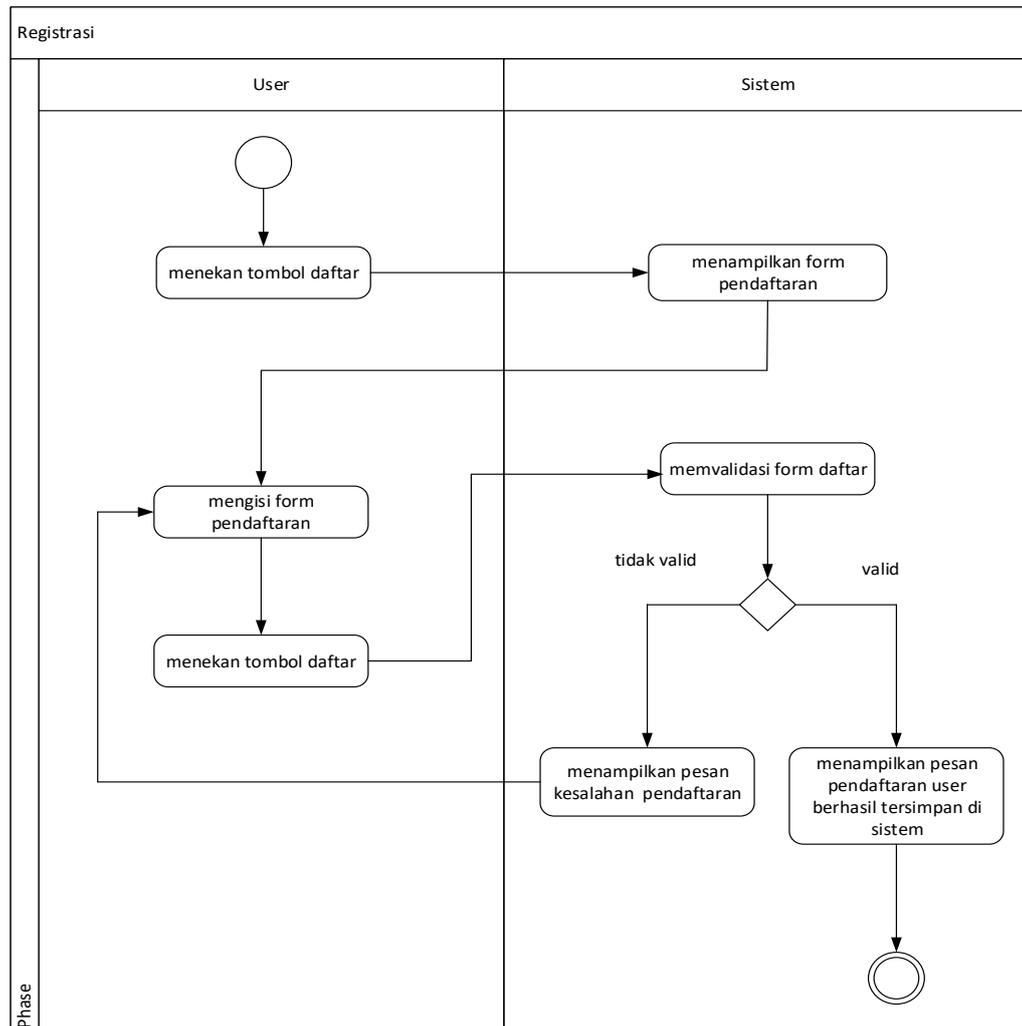
<i>Use Case Name</i>	<b>Logout</b>
<i>Goal Context</i>	<i>User</i> logout dari aplikasi
<i>Preconditions</i>	<i>User</i> sudah <i>login</i> kedalam aplikasi
<i>Successful End Condition</i>	<i>User</i> berhasil melakukan <i>logout</i> dari aplikasi
<i>Failed End Condition</i>	<i>User</i> tidak berhasil <i>logout</i> dari aplikasi

<i>Primary Actors</i>	<i>Supervisor user</i>	
<i>Trigger</i>	<i>User menekan tombol akun</i>	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1.	<i>User menekan tombol akun</i>
	2.	Sistem menampilkan halaman akun
	3.	<i>User menekan tombol keluar</i>
	4.	Sistem menampilkan halaman <i>login</i>

### 3.1.8.3 Activity Diagram

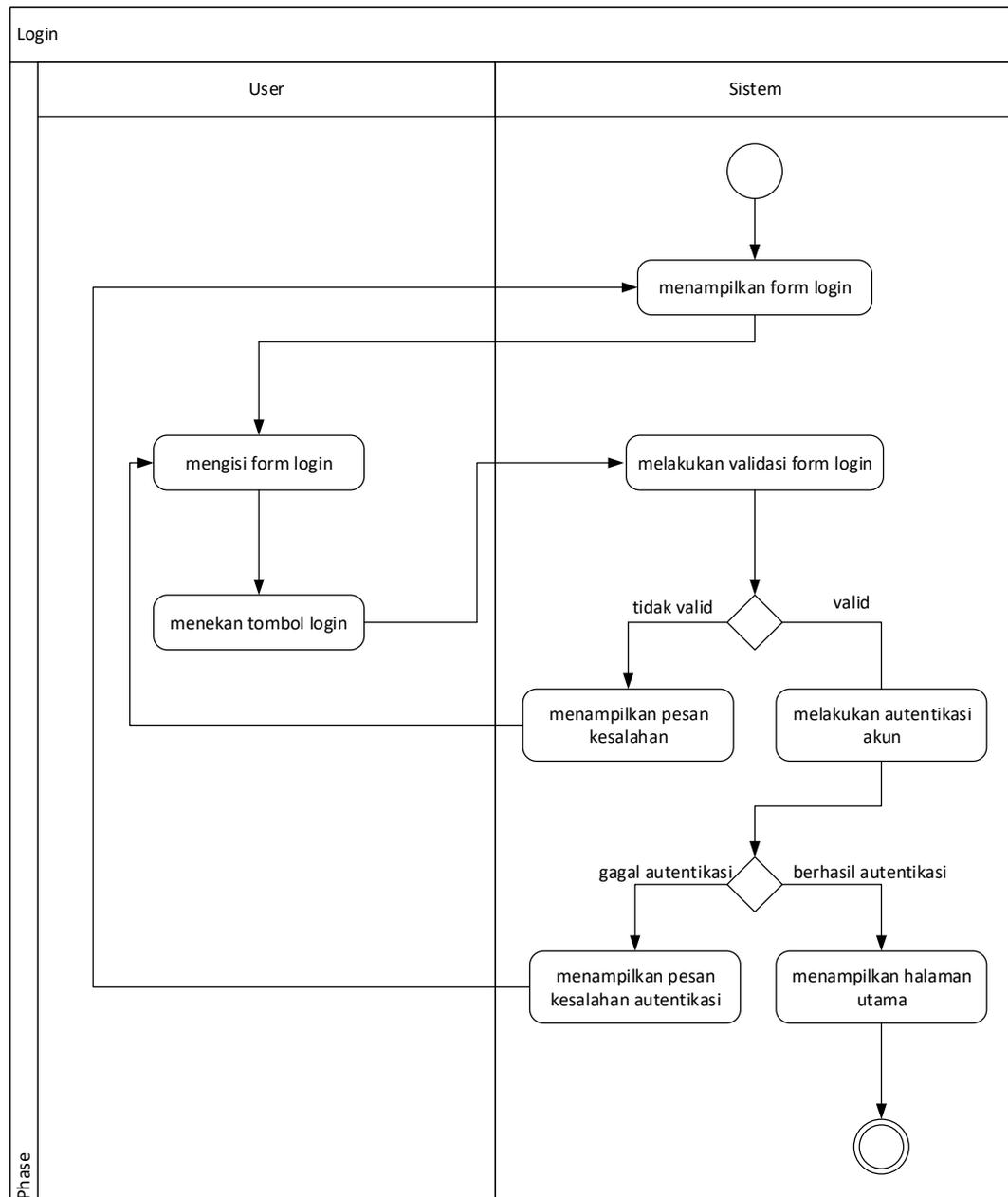
*Activity* diagram memodelkan alur kerja (*work flow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah *flowchart* karena *user* dapat memodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari satu aktivitas ke dalam keadaan sesaat (*state*). Diagram *Activity* berfokus pada aktifitas - aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas - aktifitas tersebut bergantung satu sama lain. Berikut adalah *activity* diagram yang terbentuk dalam aplikasi :

1. *Activity* diagram registrasi digunakan ketika pengguna ingin membuat akun baru. Dapat dilihat pada Gambar 3.5 *Activity* Diagram Registrasi.



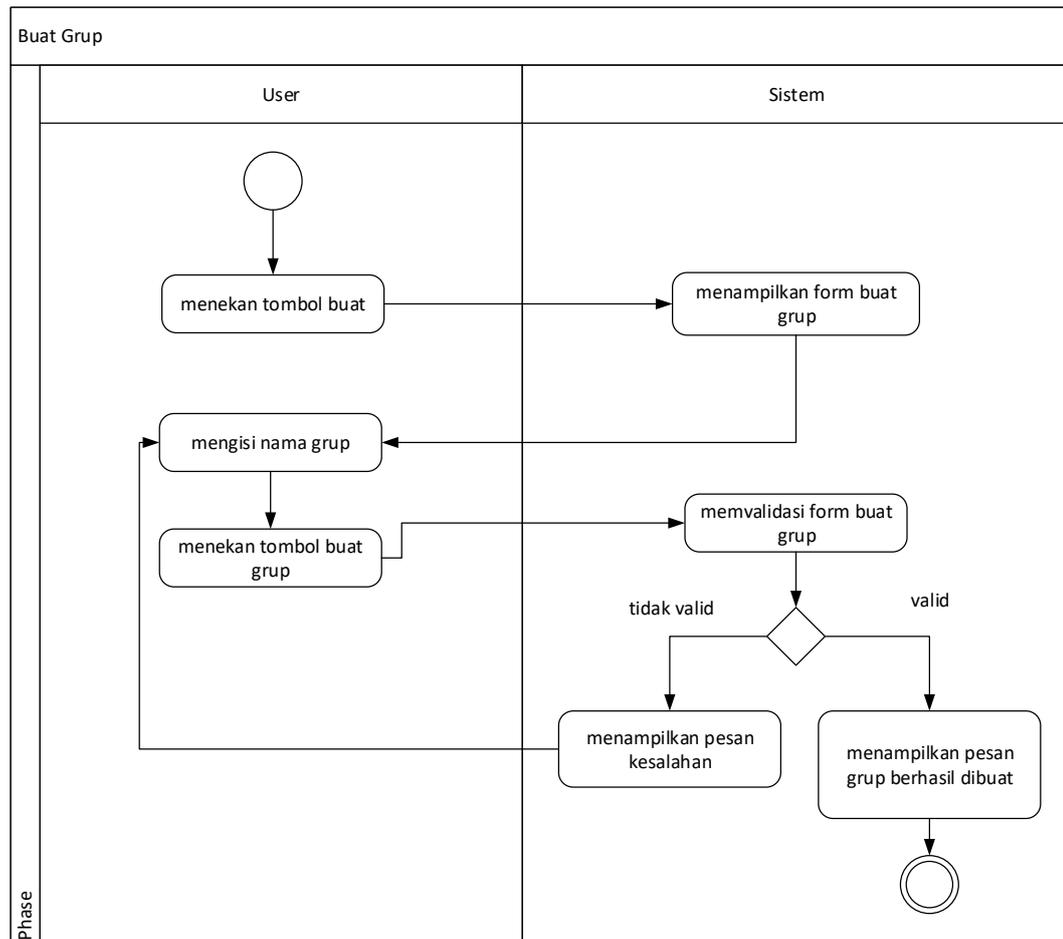
**Gambar 3.5 Activity Diagram Registrasi**

2. Activity diagram *login* digunakan ketika pengguna ingin melakukan *login*. Dapat dilihat pada Gambar 3.6 Activity Diagram *Login*.



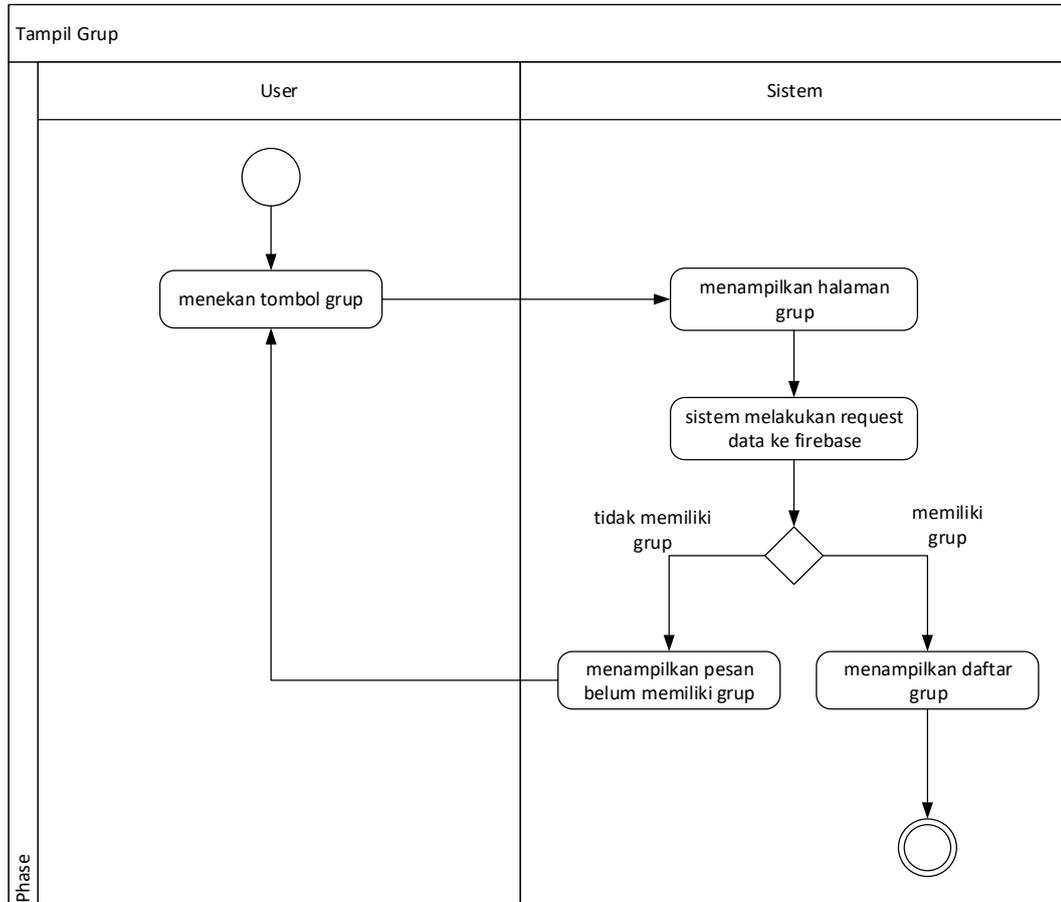
**Gambar 3.6 Activity Diagram Login**

3. Activity diagram buat grup digunakan ketika pengguna ingin membuat grup monitoring. Dapat dilihat pada Gambar 3.7 Activity Diagram Buat Grup.



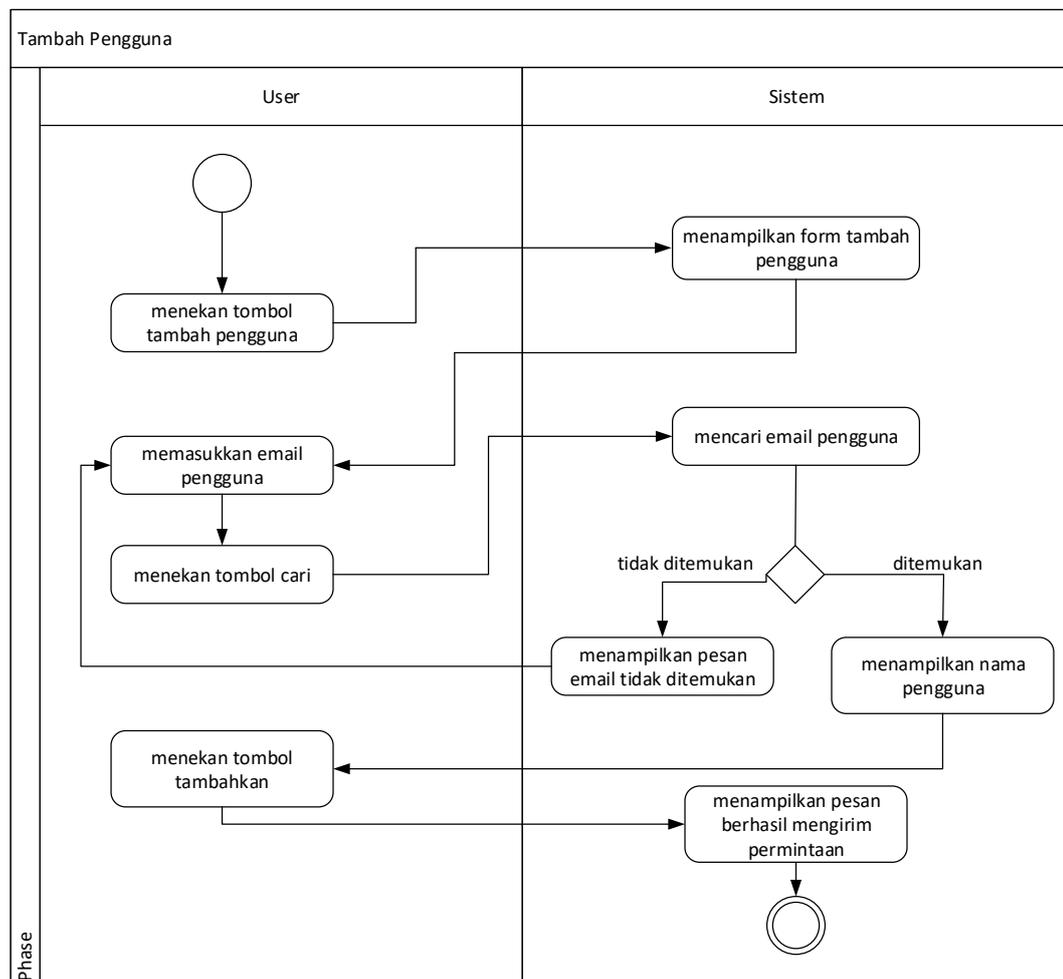
**Gambar 3.7 Activity Diagram Buat Grup**

4. Activity diagram tampil grup digunakan ketika pengguna ingin melihat daftar grup yang diikuti. Dapat dilihat pada Gambar 3.8 Activity Diagram Tampil Grup.



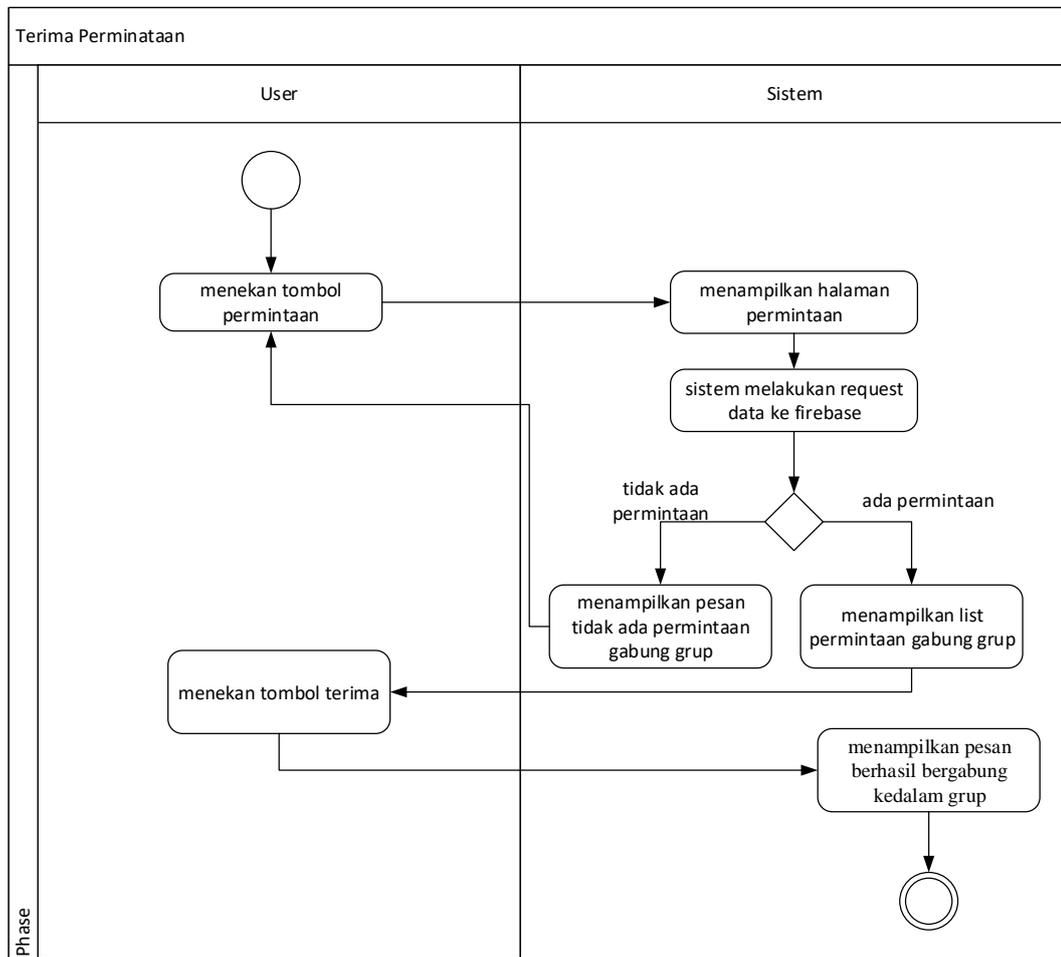
**Gambar 3.8 Activity Diagram Tampil Grup**

5. *Activity* Diagram tambah pengguna digunakan ketika pengguna ingin menambah pengguna lain di grup monitoring. Dapat dilihat pada Gambar 3.9 Activity Diagram Tambah Pengguna.



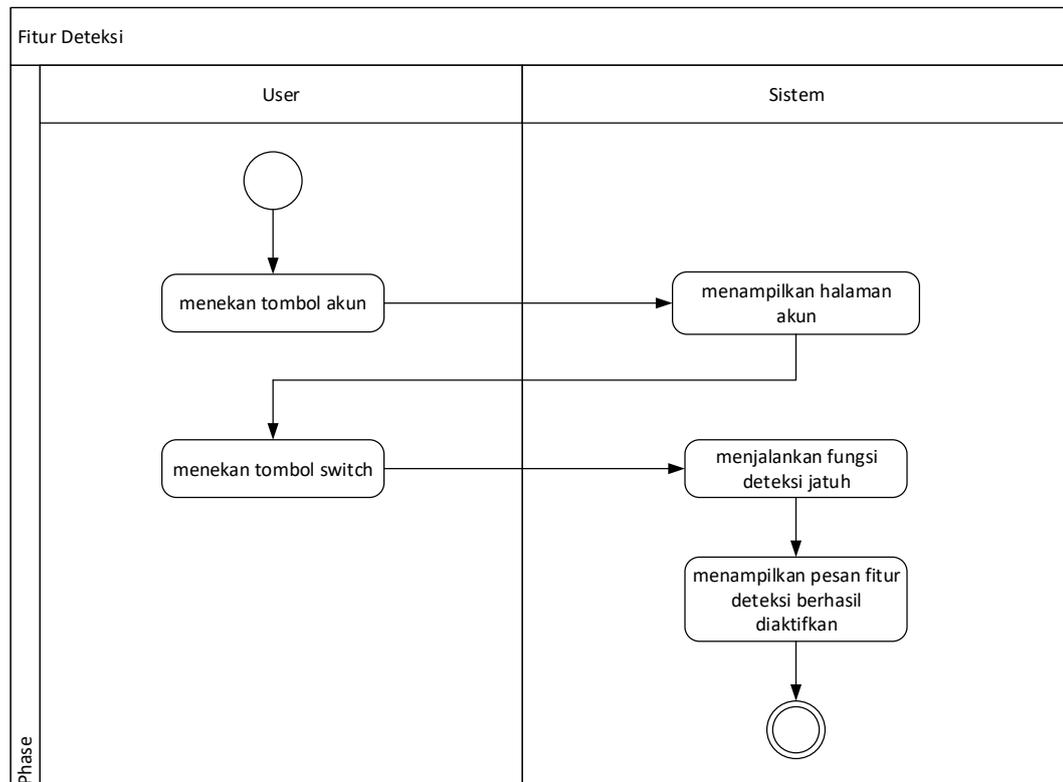
**Gambar 3.9 Activity Diagram Tambah Pengguna**

6. *Activity Diagram* terima permintaan digunakan ketika pengguna ingin menerima permintaan gabung grup . Dapat dilihat pada Gambar 3.10 *Activity Diagram* Terima Permintaan.



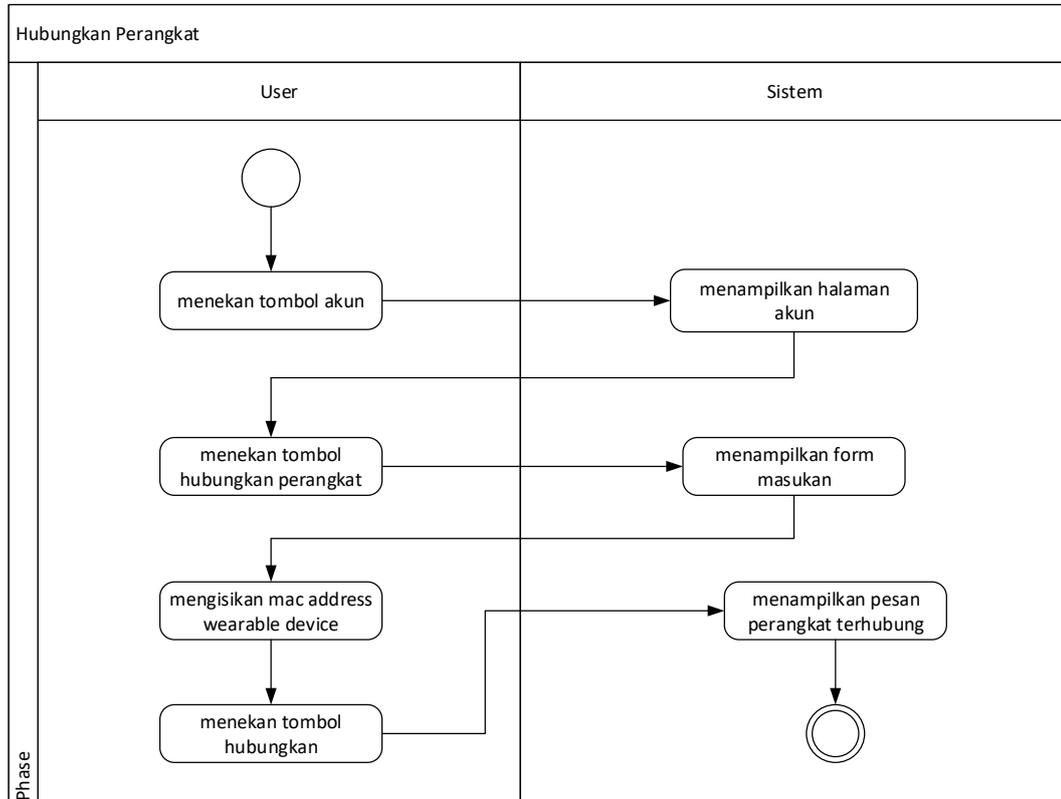
**Gambar 3.10 Activity Diagram Terima Permintaan**

7. *Activity* diagram fitur deteksi digunakan ketika pengguna ingin mengaktifkan fitur deteksi jatuh. Dapat dilihat pada Gambar 3.11 *Activity* Diagram Fitur Deteksi.



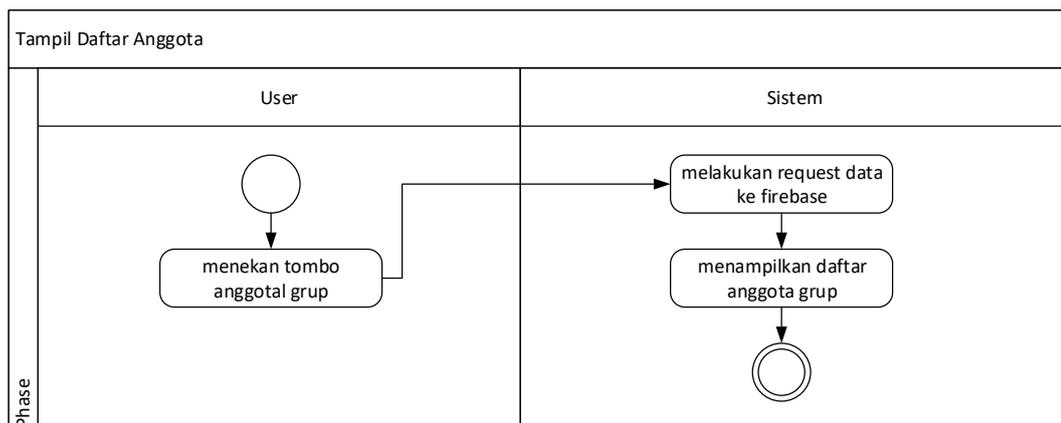
**Gambar 3.11 Activity Diagram Fitur Deteksi**

8. Activity diagram hubungan perangkat digunakan ketika pengguna ingin menghubungkan *wearable device* ke aplikasi. Dapat dilihat pada Gambar 3.12 Activity Diagram Hubungan Perangkat.



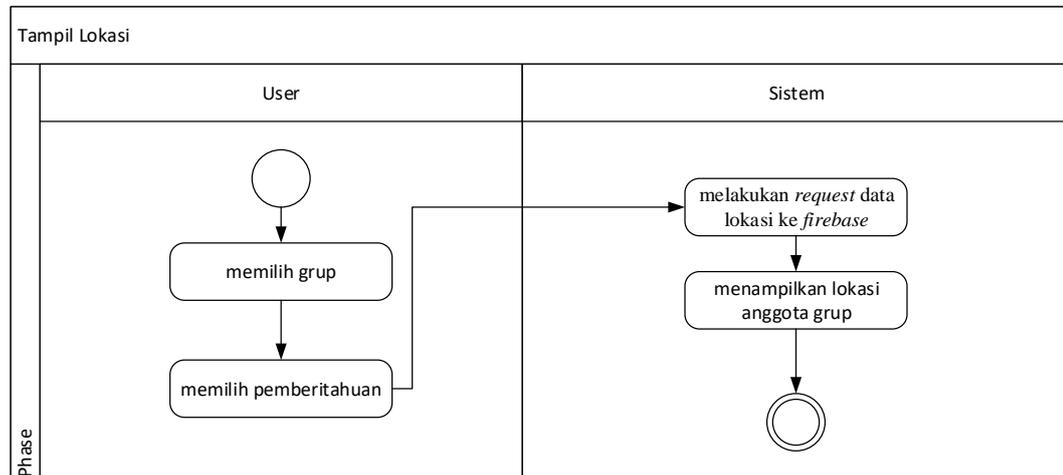
Gambar 3.12 Activity Diagram Hubungkan Perangkat

9. Activity Diagram tampil daftar anggota digunakan ketika pengguna ingin melihat daftar anggota grup. Dapat dilihat pada Gambar 3.13 Activity Diagram Tampil Daftar Anggota.



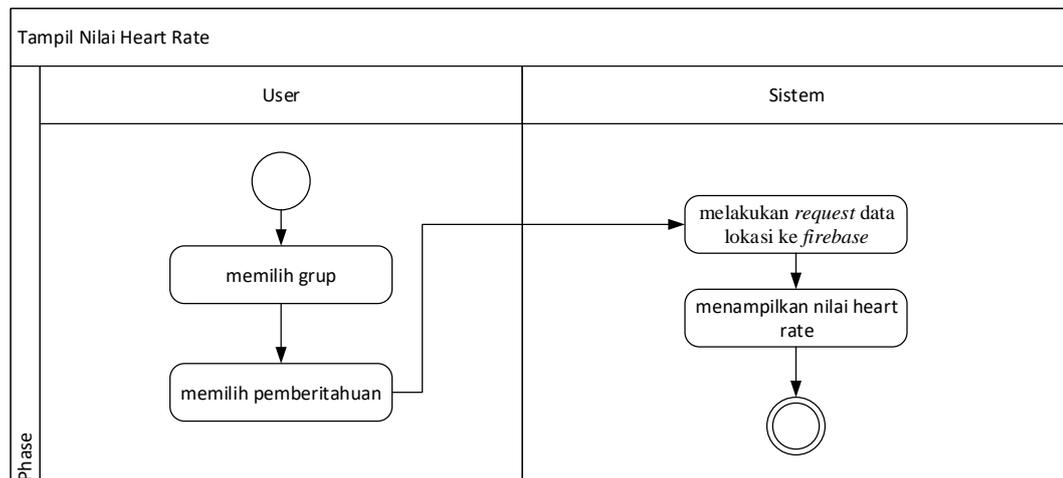
Gambar 3.13 Activity Diagram Tampil Daftar Anggota

10. *Activity* diagram tampil lokasi digunakan ketika pengguna ingin melihat lokasi anggota grup. Dapat dilihat pada Gambar 3.14 *Activity* Diagram Tampil Lokasi.



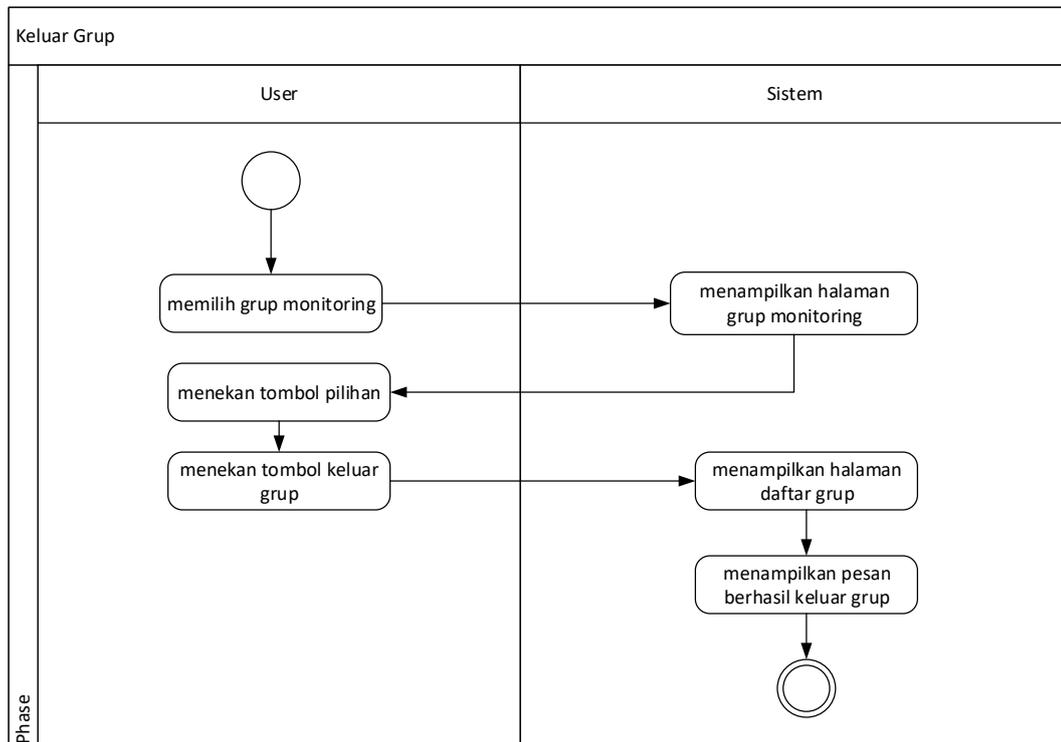
**Gambar 3.14 Activity Diagram Tampil Lokasi**

11. *Activity* diagram tampil nilai *heart rate* digunakan ketika pengguna ingin mengetahui nilai *heart rate* pengguna yang dimonitoring. Dapat dilihat pada Gambar 3.15 *Activity* Diagram Tampil Nilai *Heart Rate*.



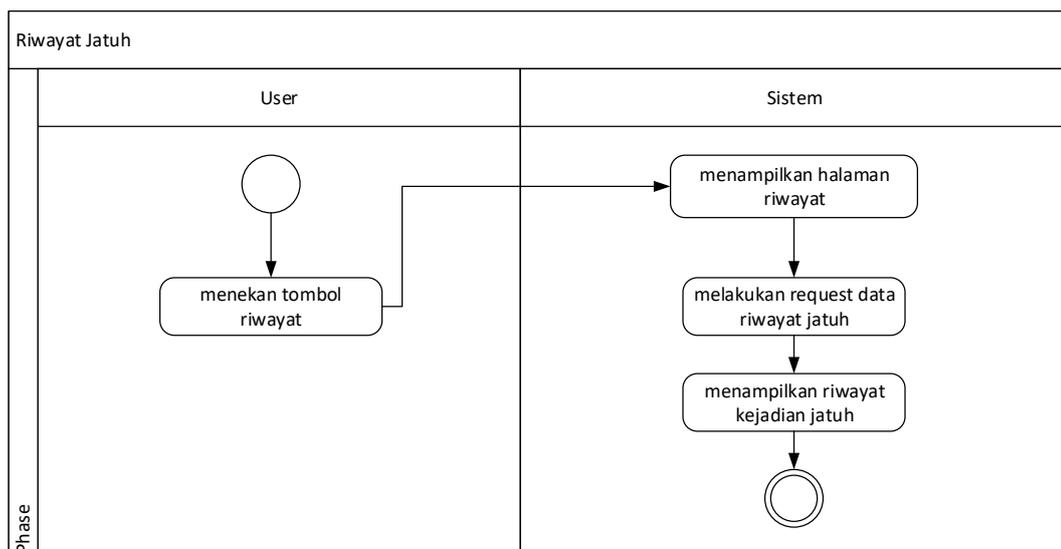
**Gambar 3.15 Activity Diagram Tampil Nilai Heart Rate**

12. *Activity* diagram keluar grup digunakan ketika user ingin keluar dari grup monitoring. Dapat dilihat pada Gambar 3.16 *Activity* Diagram Keluar Grup.



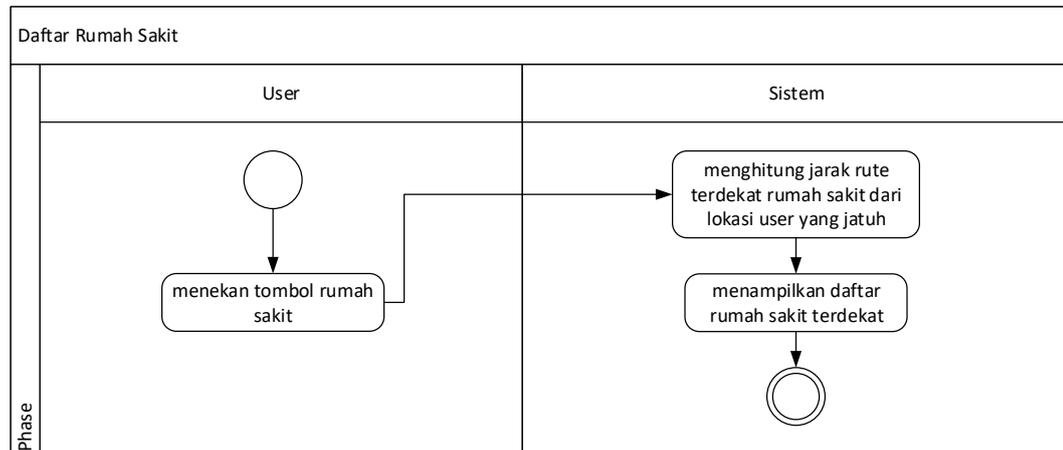
**Gambar 3.16 Activity Diagram Keluar Grup**

13. Activity diagram riwayat jatuh digunakan ketika pengguna ingin melihat daftar riwayat jatuh. Dapat dilihat pada Gambar 3.17 Activity Diagram Tampil Riwayat Jatuh.



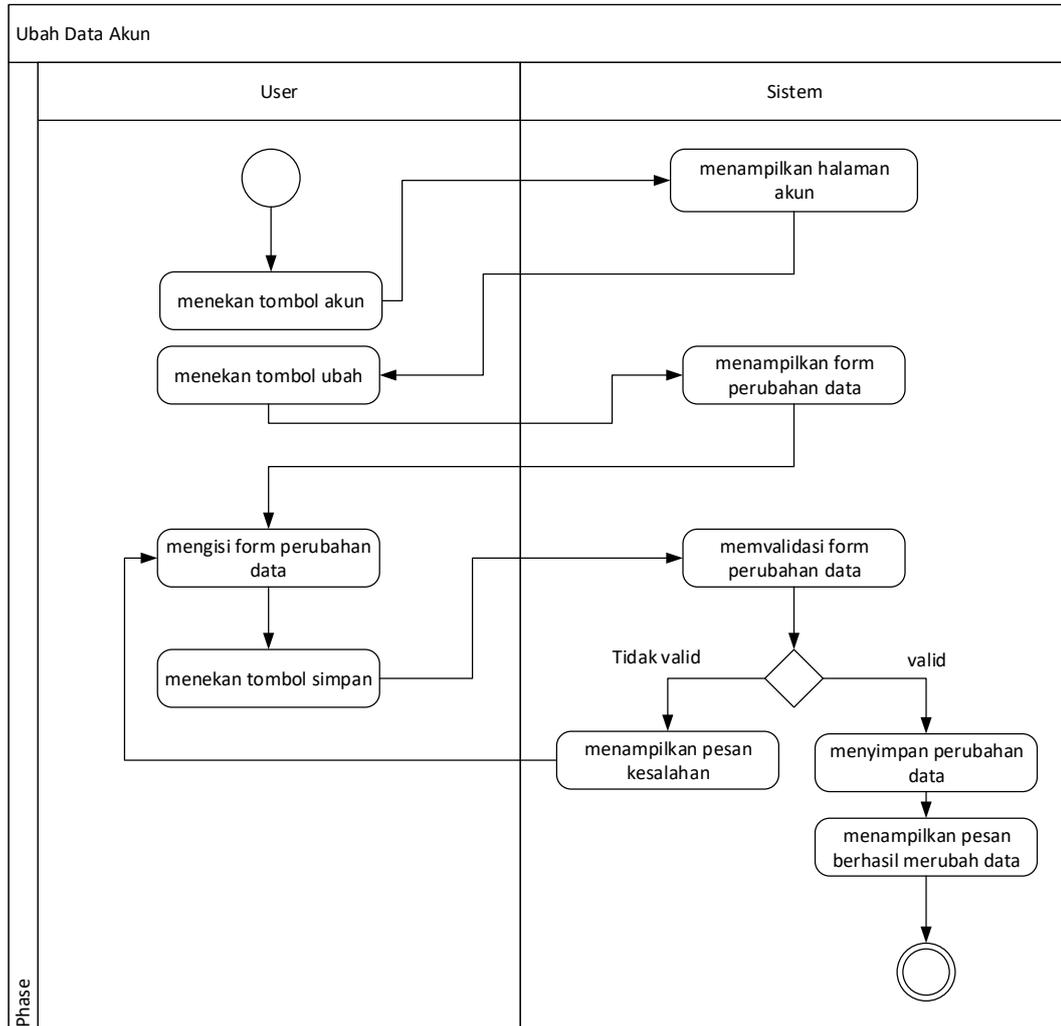
**Gambar 3.17 Activity Diagram Riwayat Jatuh**

14. *Activity* Diagram daftar rumah sakit digunakan ketika pengguna ingin melihat daftar rumah sakit terdekat. Dapat dilihat pada Gambar 3.18 *Activity* Diagram Daftar Rumah Sakit.



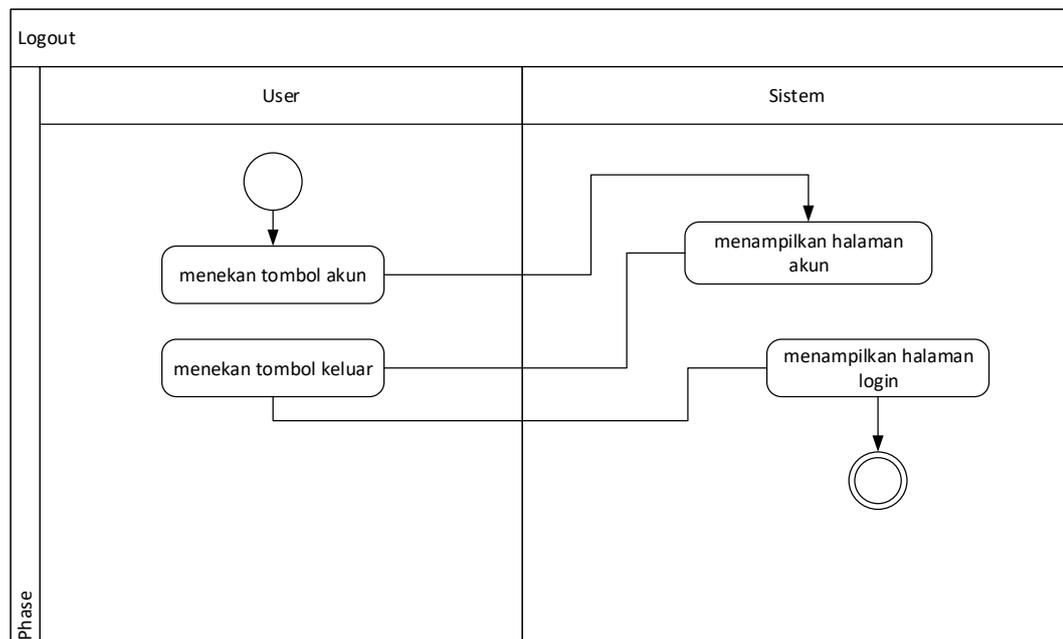
**Gambar 3.18 *Activity* Diagram Daftar Rumah Sakit**

15. *Activity* diagram ubah data akun digunakan ketika pengguna ingin merubah data pribadi pengguna. Dapat dilihat pada Gambar 3.19 *Activity* Diagram Ubah Data Akun.



**Gambar 3.19 Activity Diagram Ubah Data Akun**

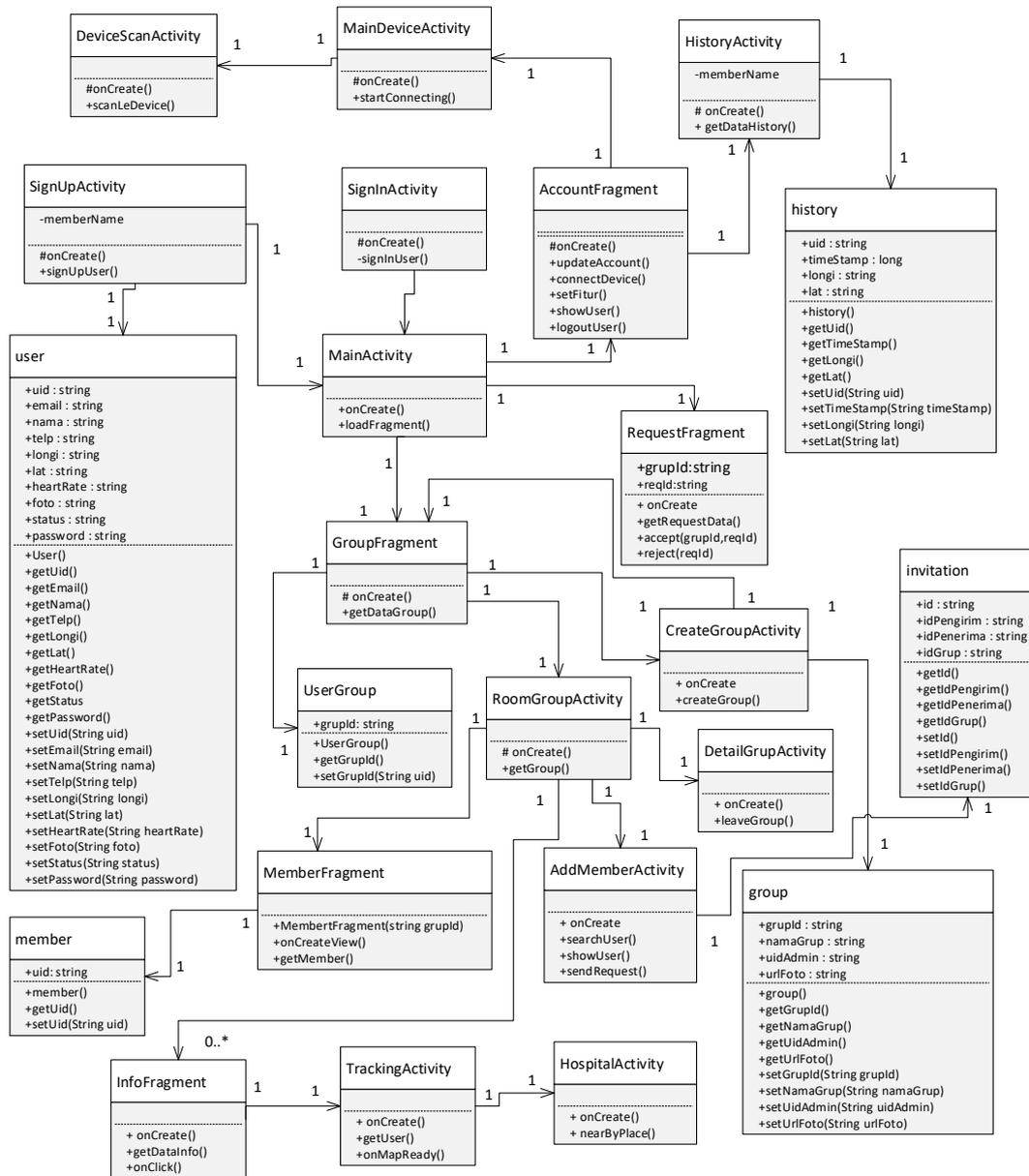
16. Activity diagram logout digunakan ketika pengguna ingin keluar dari aplikasi. Dapat dilihat pada Gambar 3.20 *Activity Diagram Logout*.



**Gambar 3.20 Activity Diagram Logout**

#### **3.1.8.4 Class Diagram**

Class diagram memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas-kelasnya dan hubungan mereka. Class diagram bersifat statis menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan. Class menggambarkan keadaan (atribut/properti). Class diagram menggambarkan struktur dan deskripsi class, package, atau objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain. Class diagram keseluruhan dapat dilihat pada Gambar 3.21 Class Diagram.

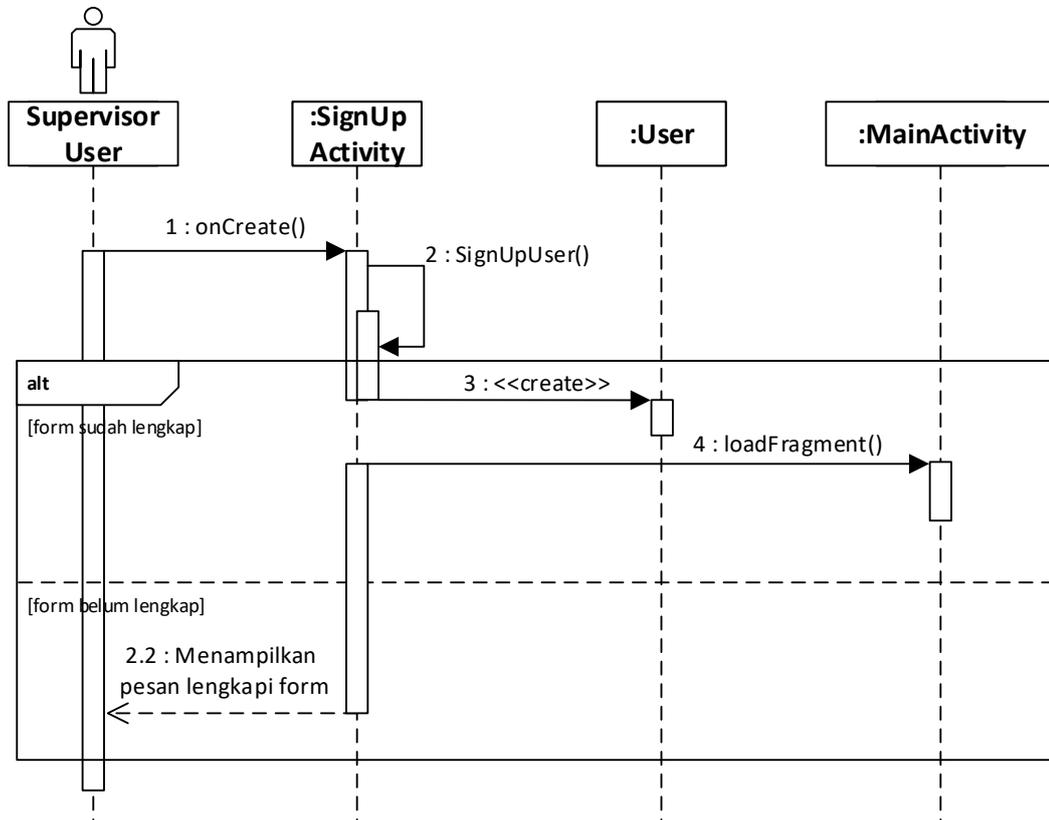


Gambar 3.21 Class Diagram

### 3.1.8.5 Sequence Diagram

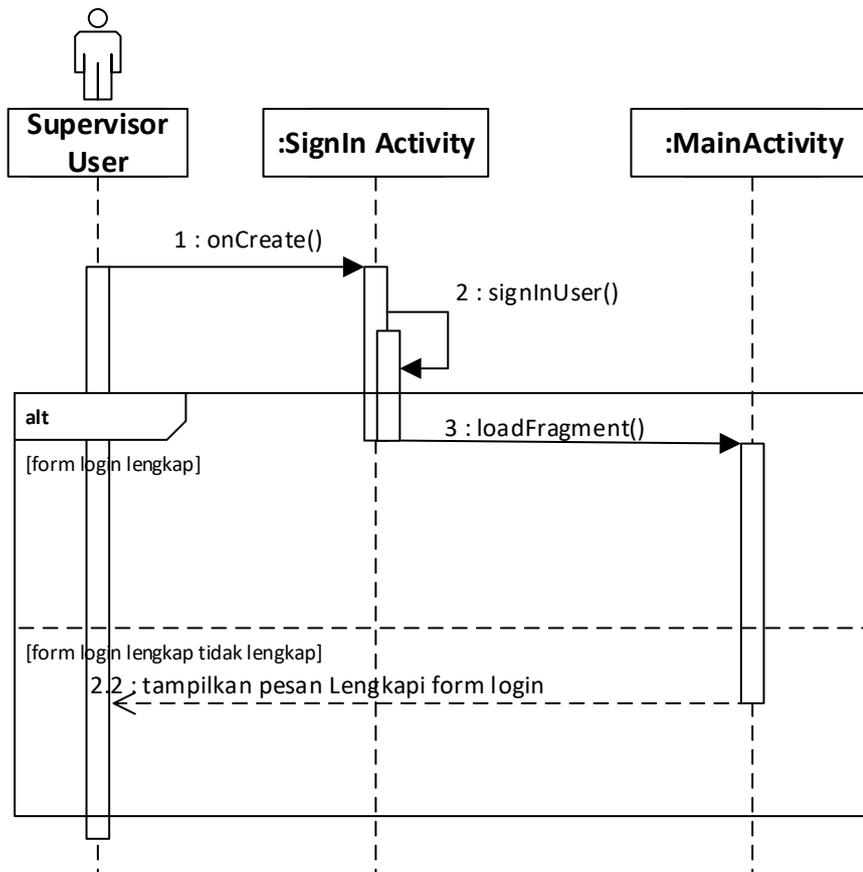
Sequence diagram merupakan salah satu diagram Interaction yang menjelaskan bagaimana suatu operasi itu dilakukan; message (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

1. *Sequence* diagram registrasi memberikan gambaran mengenai sejumlah objek dan pesan yang diletakkan pada *use case* registrasi. *Sequence* diagram registrasi dapat dilihat pada Gambar 3.22 *Sequence Diagram Registrasi*.



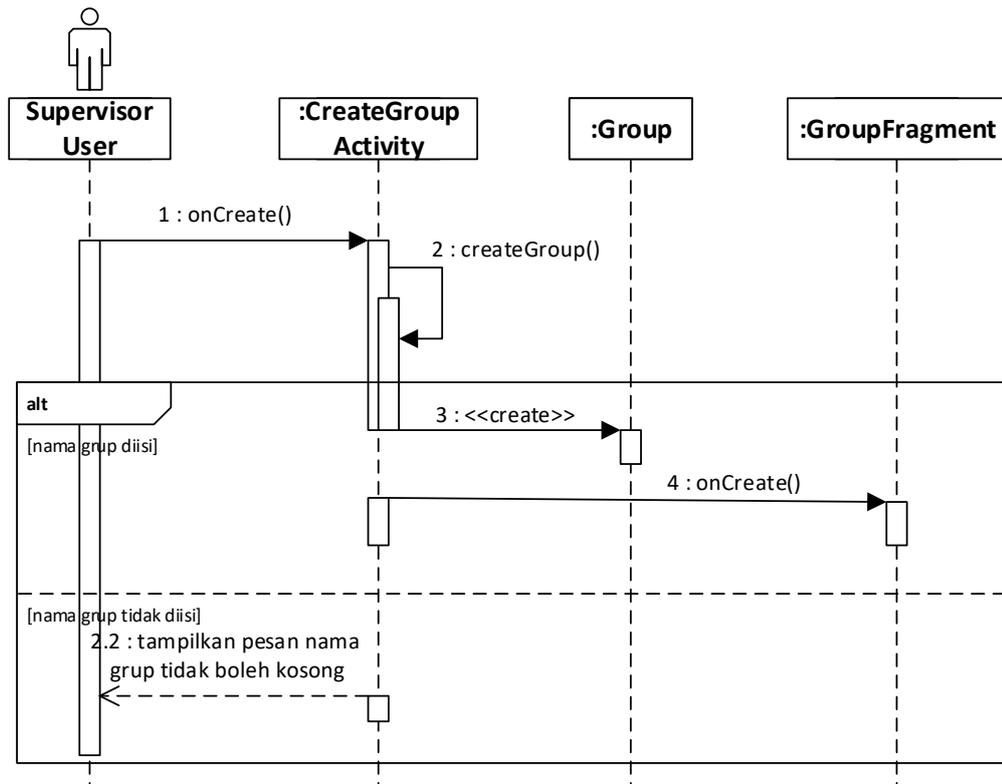
**Gambar 3.22** *Sequence Diagram Registrasi*

2. *Sequence* diagram *login* memberikan gambaran mengenai sejumlah objek dan pesan yang diletakkan pada *use case login*. *Sequence* diagram *login* dapat dilihat pada Gambar 3.23 *Sequence Diagram Login*.



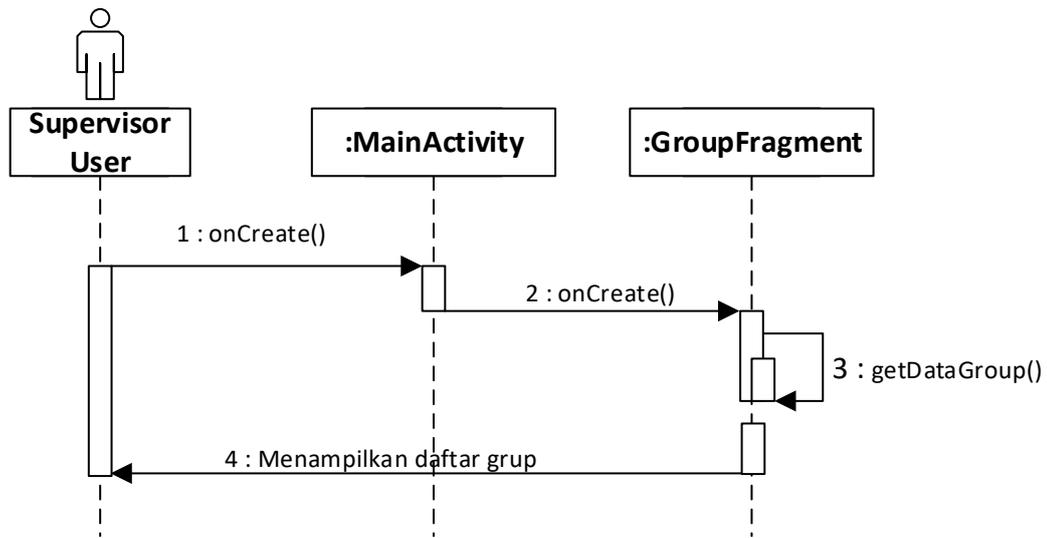
**Gambar 3.23** *Sequence Diagram Login*

3. *Sequence* diagram buat grup memberikan gambaran mengenai sejumlah objek dan pesan yang diletakkan pada *use case* buat grup. *Sequence* diagram buat grup dapat dilihat pada Gambar 3.24 *Sequence Diagram Buat Grup*.



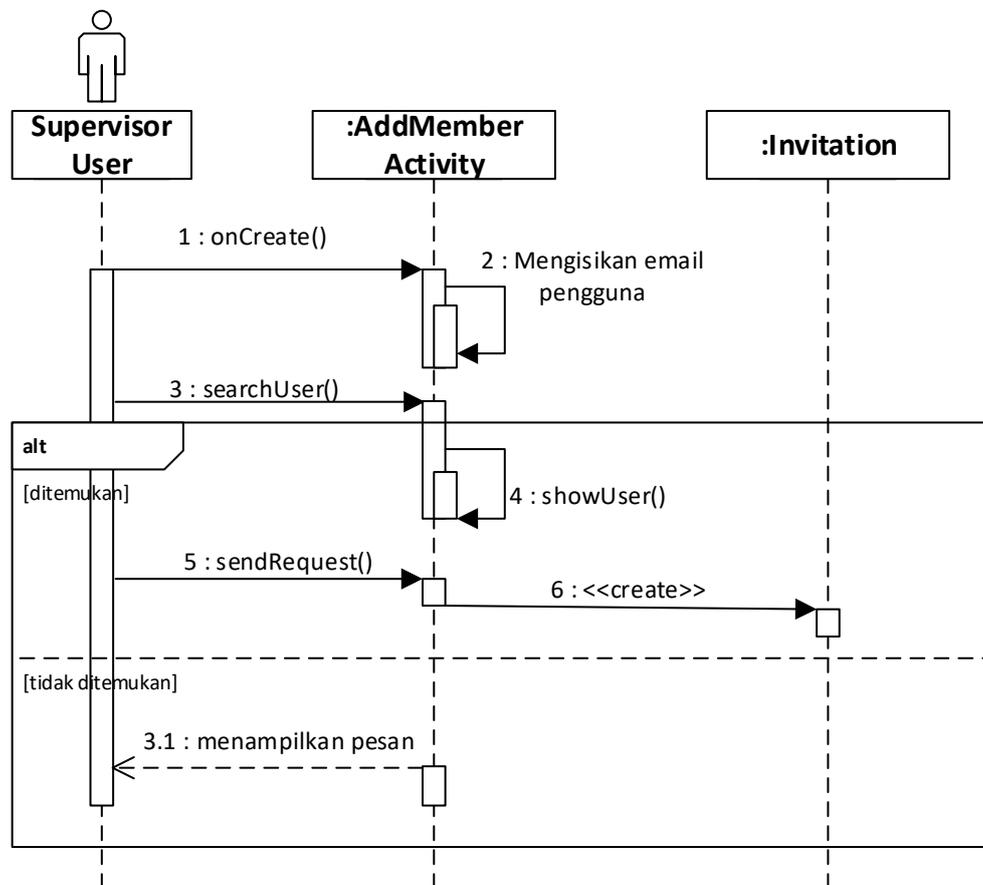
**Gambar 3.24 Sequence Diagram Buat grup**

4. *Sequence* diagram tampil grup memberikan gambaran mengenai sejumlah objek dan pesan yang diletakkan pada *use case* tampil grup. *Sequence* diagram tampil grup dapat dilihat pada Gambar 3.25 *Sequence* Diagram Tampil Grup.



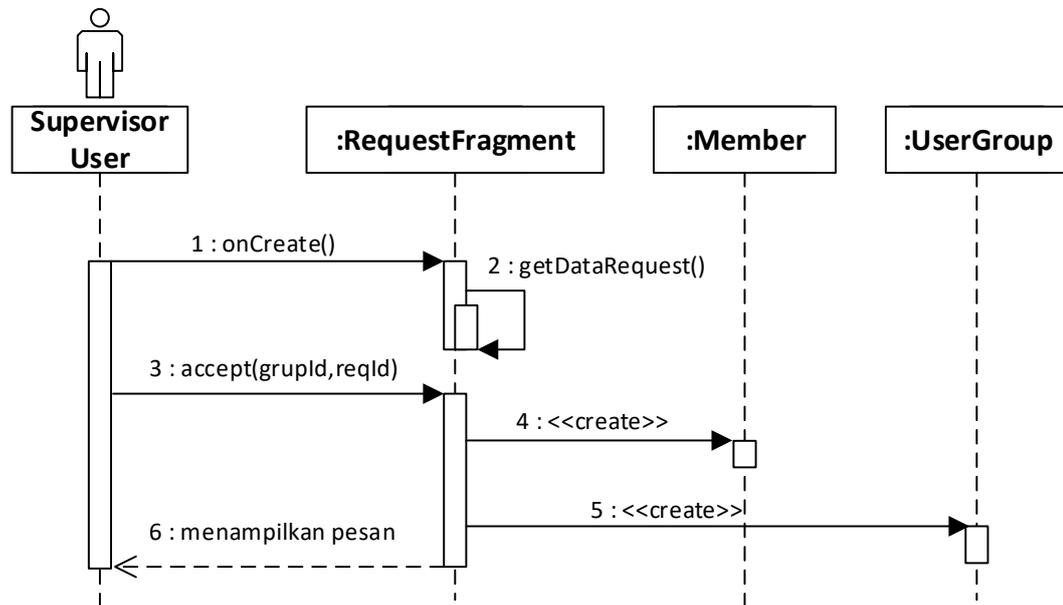
**Gambar 3.25** *Sequence Diagram Tampil Grup*

5. *Sequence* diagram tambah pengguna memberikan gambaran mengenai sejumlah objek dan pesan yang diletakkan pada *use case* tambah pengguna. *Sequence* diagram tambah pengguna dapat dilihat pada Gambar 3.26 *Sequence Diagram Tambah Pengguna*.



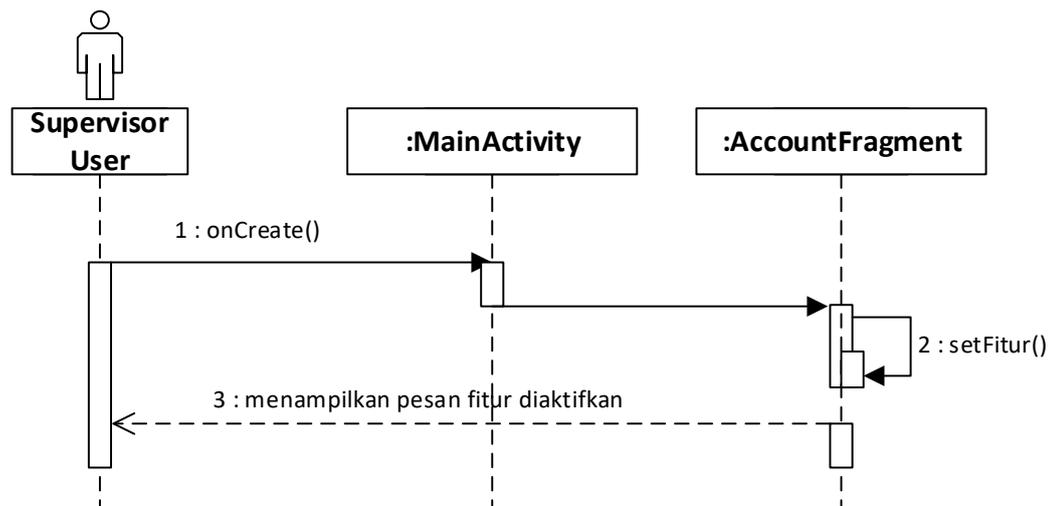
**Gambar 3.26 Sequence Diagram Tambah Pengguna**

6. *Sequence* diagram terima permintaan memberikan gambaran mengenai sejumlah objek dan pesan yang diletakkan pada *use case* terima permintaan. *Sequence* diagram terima permintaan dapat dilihat pada Gambar 3.27 *Sequence* Diagram Terima Permintaan.



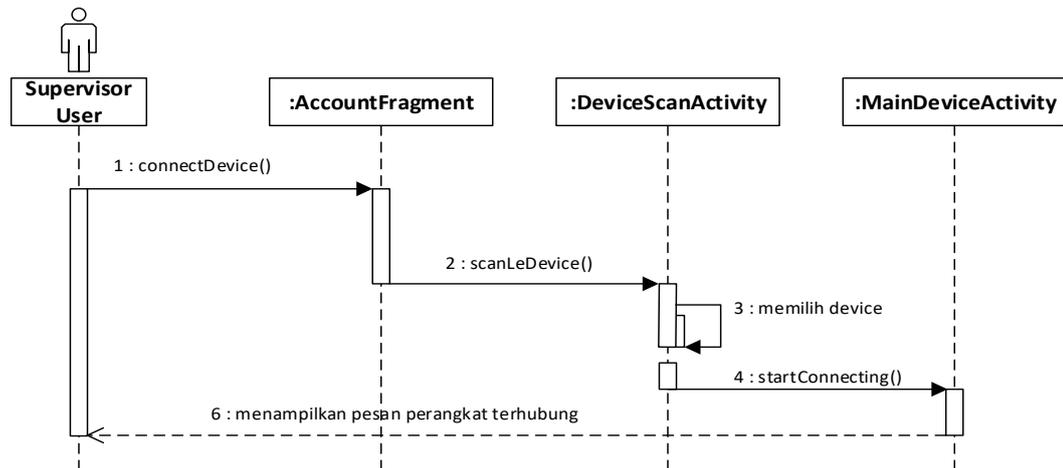
**Gambar 3.27** *Sequence Diagram Terima Permintaan*

7. *Sequence* diagram fitur deteksi memberikan gambaran mengenai sejumlah objek dan pesan yang diletakkan pada *use case* fitur deteksi. *Sequence* diagram fitur deteksi dapat dilihat pada Gambar 3.28 *Sequence Diagram Fitur Deteksi*.



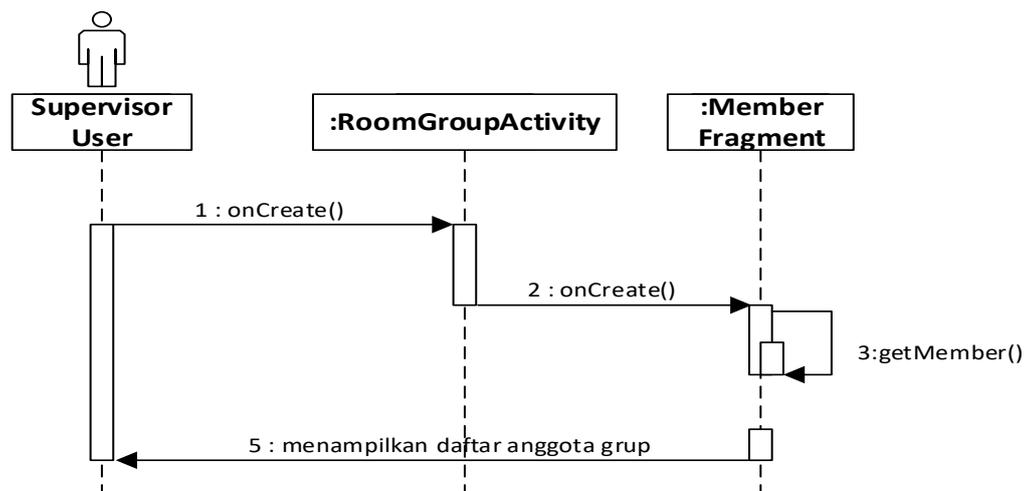
**Gambar 3.28** *Sequence Diagram Fitur Deteksi*

8. *Sequence* diagram hubungan perangkat memberikan gambaran mengenai sejumlah objek dan pesan yang diletakkan pada *use case* hubungan perangkat. *Sequence* diagram hubungan perangkat dapat dilihat pada Gambar 3.29 *Sequence* Diagram Hubungan Perangkat.



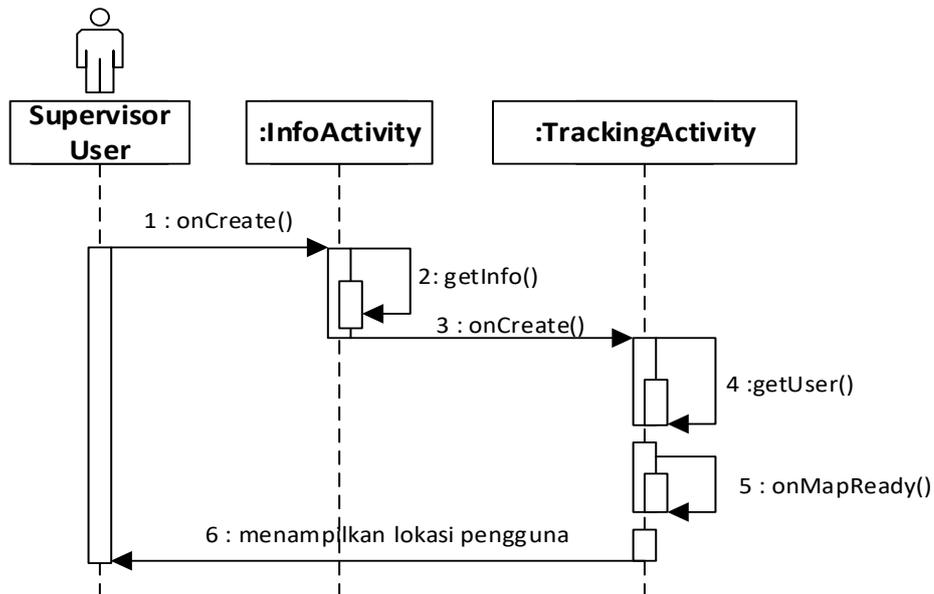
**Gambar 3.29** *Sequence* Diagram Hubungan Perangkat

9. *Sequence* diagram tampil daftar anggota memberikan gambaran mengenai sejumlah objek dan pesan yang diletakkan pada *use case* tampil daftar anggota. *Sequence* diagram tampil daftar anggota dapat dilihat pada Gambar 3.30 *Sequence* Diagram Tampil Daftar Anggota.



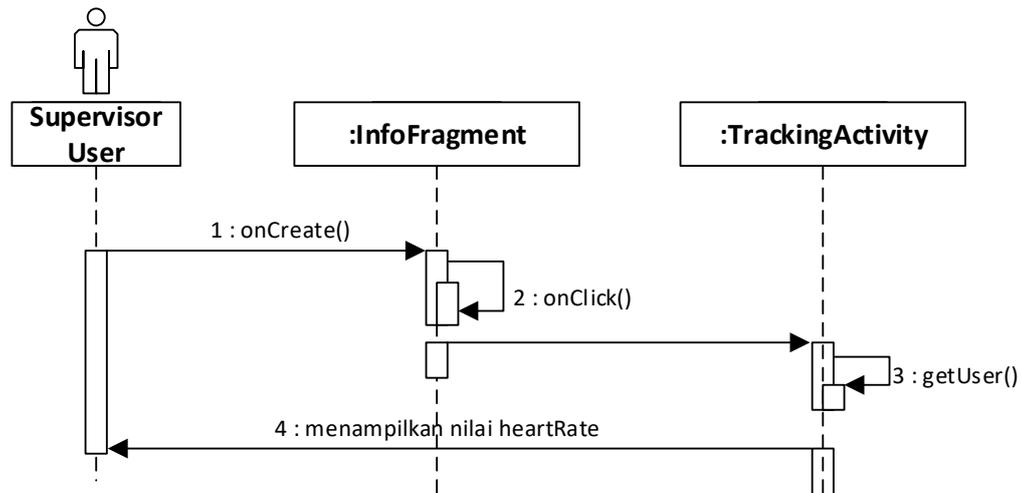
**Gambar 3.30** *Sequence* Diagram Tampil Daftar Anggota

10. *Sequence* diagram Tampil Lokasi memberikan gambaran mengenai sejumlah objek dan pesan yang diletakkan pada *use case* tampil lokasi. *Sequence* diagram tampil lokasi dapat dilihat pada Gambar 3.31 *Sequence* Diagram Tampil Lokasi.



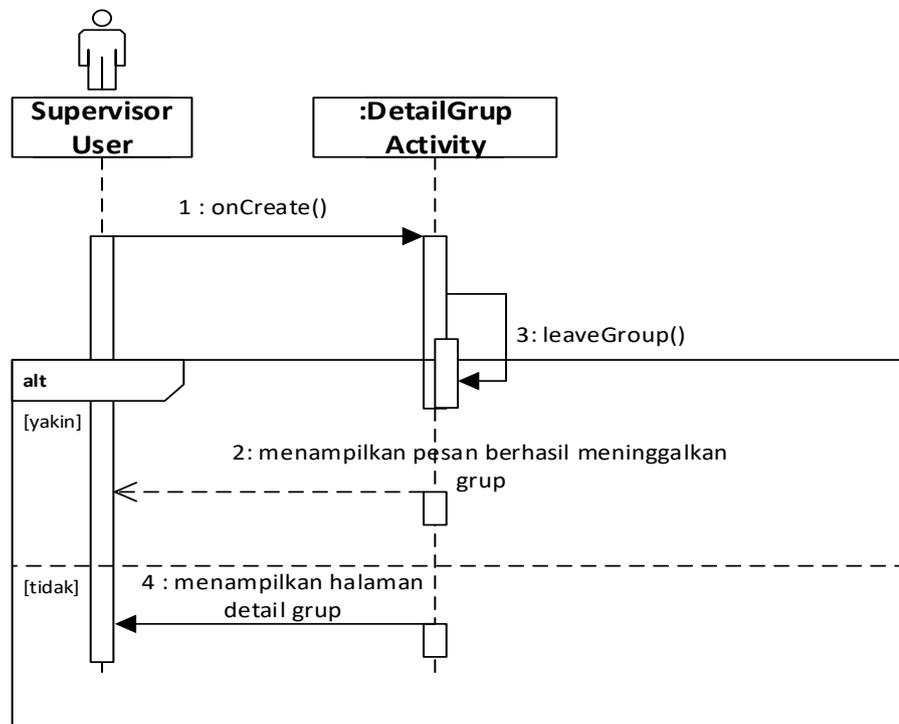
**Gambar 3.31** *Sequence* Diagram Tampil Lokasi

11. *Sequence* diagram tampil nilai *heart rate* memberikan gambaran mengenai sejumlah objek dan pesan yang diletakkan pada *use case* tampil nilai *heart rate*. *Sequence* diagram tampil nilai *heart rate* dapat dilihat pada Gambar 3.32 *Sequence* Diagram Tampil Nilai *Heart Rate*.



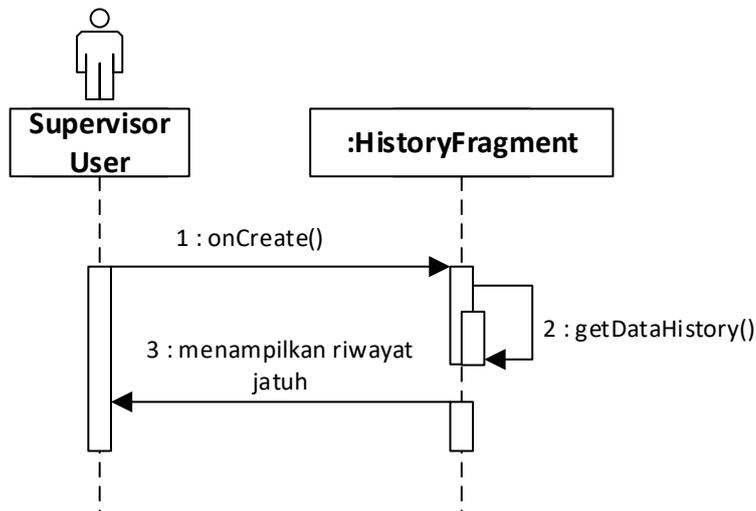
**Gambar 3.32 Sequence Diagram Tampil Nilai Heart Rate**

12. *Sequence* diagram keluar grup memberikan gambaran mengenai sejumlah objek dan pesan yang diletakkan pada *use case* keluar grup. *Sequence* diagram keluar grup dapat dilihat pada Gambar 3.33 *Sequence* Diagram Keluar Grup.



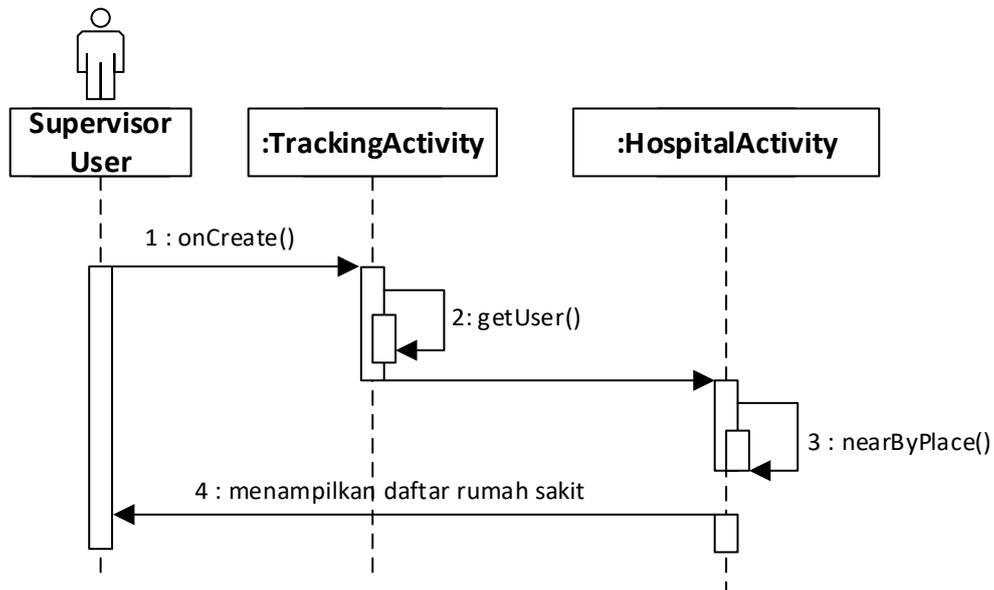
**Gambar 3.33 Sequence Diagram Keluar Grup**

13. *Sequence* diagram riwayat jatuh memberikan gambaran mengenai sejumlah objek dan pesan yang diletakkan pada *use case* riwayat jatuh. *Sequence* diagram riwayat jatuh dapat dilihat pada Gambar 3.34 *Sequence* Diagram Riwayat Jatuh.



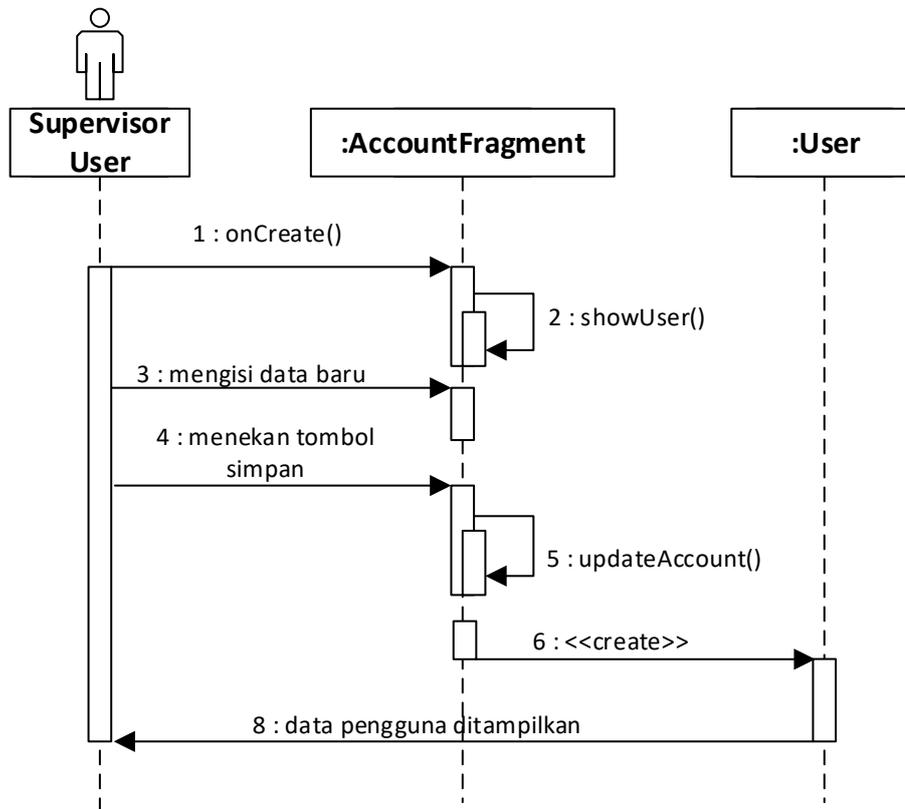
**Gambar 3.34** *Sequence* Diagram Riwayat Jatuh

14. *Sequence* diagram daftar rumah sakit memberikan gambaran mengenai sejumlah objek dan pesan yang diletakkan pada *use case* daftar rumah sakit. *Sequence* diagram daftar rumah sakit dapat dilihat pada Gambar 3.35 *Sequence* Diagram Daftar Rumah Sakit.



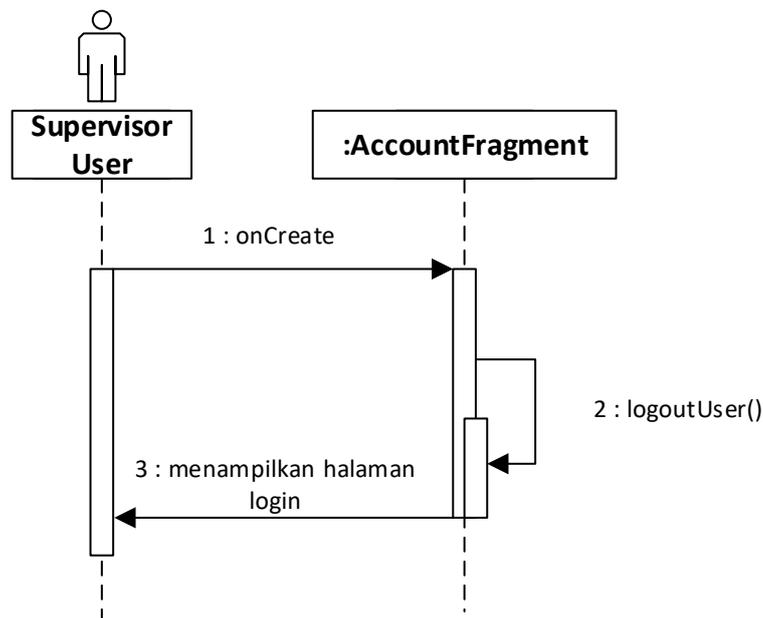
**Gambar 3.35 Sequence Diagram Daftar Rumah Sakit**

15. *Sequence* diagram ubah data akun memberikan gambaran mengenai sejumlah objek dan pesan yang diletakkan pada *use case* ubah data akun. *Sequence* diagram ubah data akun dapat dilihat pada Gambar 3.36 *Sequence* Diagram Ubah Data Akun.



**Gambar 3.36** *Sequence Diagram Ubah Data Akun*

16. *Sequence diagram logout* memberikan gambaran mengenai sejumlah objek dan pesan yang diletakkan pada *use case logout*. *Sequence diagram logout* dapat dilihat pada Gambar 3.37 *Sequence Diagram Logout*.



**Gambar 3.37 Sequence Diagram Logout**

### 3.2 Perancangan Sistem

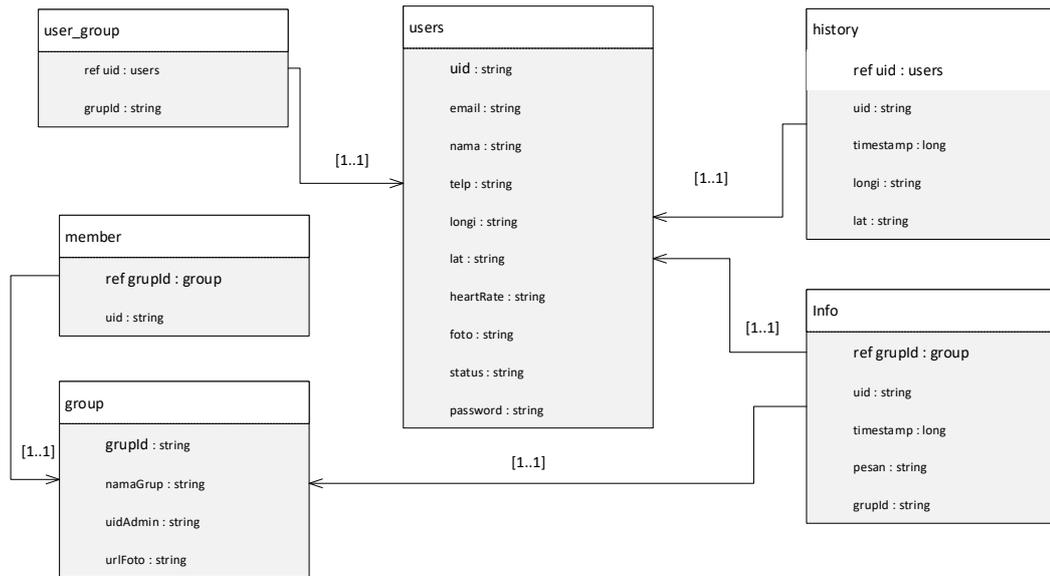
Perancangan sistem merupakan bagian dari metodologi penelitian pengembangan suatu perangkat lunak yang dilakukan setelah tahap analisis. Perancangan bertujuan untuk memberikan gambaran secara terperinci. Perancangan sistem diharapkan dapat memecahkan permasalahan yang ada. Perancangan sistem dimulai setelah tahap analisis terhadap sistem telah dilakukan. Perancangan dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi. Tahap ini menyangkut konfirmasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah instalasi dari sistem akan benar-banar memuaskan dari rancang bangun yang telah ditetapkan pada tahap akhir analisis sistem.

#### 3.2.1 Perancangan Basis Data

Pada tahap perancangan basis data, dilakukan perancangan tabel atau dalam lingkup NoSQL *database* lebih dikenal dengan collection, untuk setiap collection

akan memiliki struktur document yang berbeda-beda. Selain merancang tabel, pada tahap ini dilakukan juga perancangan struktur dokumen untuk setiap collection yang telah didefinisikan.

### 3.2.1.1 Model Database



Gambar 3.38 Model Database

### 3.2.1.2 Struktur Collection

Struktur collection merupakan rincian collection yang berisi document yang didalamnya terdapat field beserta keterangannya. Berikut struktur collection yang terdapat pada database aplikasi smart fall alert :

1. Struktur Collection Users

Struktur collection users dapat dilihat pada Tabel 3.25 Struktur Collection Users.

Tabel 3.25 Struktur Collection Users

Nama Field	Type Data	Keterangan
uid	String	key
email	String	
nama	String	

telp	String	
longi	String	
lat	String	
heartRate	String	
foto	String	
status	String	
password	String	

## 2. Struktur *Collection Group*

Struktur *collection group* dapat dilihat pada Tabel 3.26 Struktur *Collection Group*.

**Tabel 3.26 Struktur *Collection Group***

Nama Field	Tipe Data	Keterangan
grupId	String	key
namaGrup	String	
Admin	String	
urlFoto	String	

## 3. Struktur *Collection Member*

Struktur *collection member* dapat dilihat pada Tabel 3.27 Struktur *Collection Member*.

**Tabel 3.27 Struktur *Collection Member***

Nama Field	Tipe Data	Keterangan
grupId	Object	Ref : group
Uid	String	Ref : users

## 4. Struktur *Collection User Group*

Struktur *collection User Group* dapat dilihat pada Tabel 3.28 Struktur *Collection User Group*.

**Tabel 3.28 Struktur *Collection User Group***

Nama Field	Tipe Data	Keterangan
Uid	Object	Ref : users
grupId	String	Ref : group

5. Struktur *Collection History*

Struktur collection riwayat dapat dilihat pada Tabel 3.29 Struktur *Collection Riwayat*.

**Tabel 3.29 Struktur *Collection History***

Nama Field	Tipe Data	Keterangan
Uid	Object	Ref : users
timeStamp	Long	
Longi	String	
Lat	String	

6. Struktur *Collection Info*

Struktur collection message dapat dilihat pada Tabel 3.30 Struktur *Collection Info*.

**Tabel 3.30 Struktur *Collection Info***

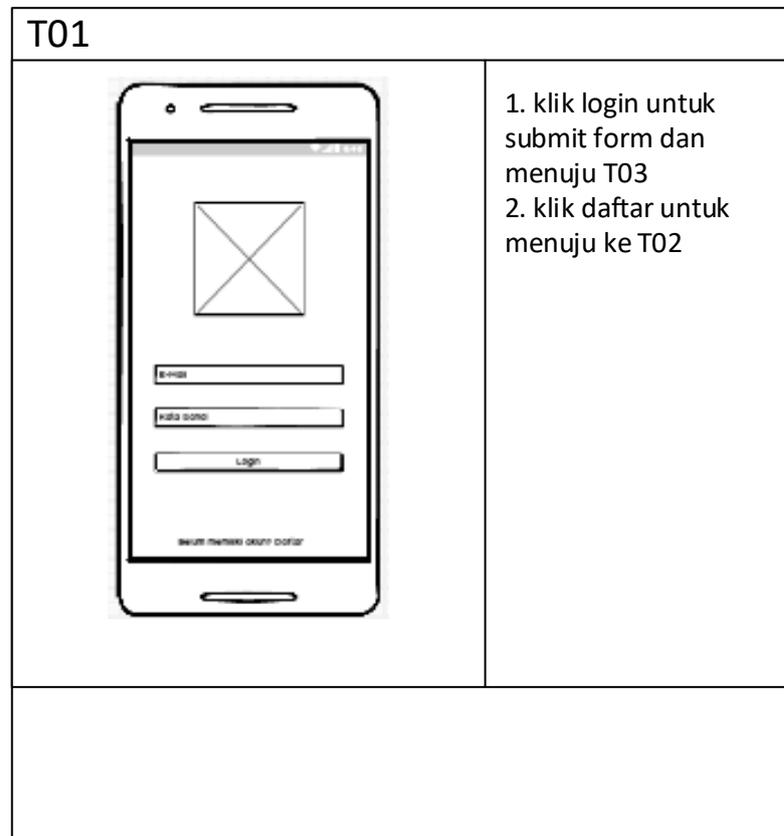
Nama Field	Tipe Data	Keterangan
Uid	Object	Ref : users
timeStamp	Long	
grupId	String	Ref : Group
Pesan	String	
Id	String	

### 3.2.2 Perancangan Antarmuka

Perancangan antarmuka adalah tampilan program yang masih berbentuk rancangan atau mockup yang akan diimplementasikan ke program atau aplikasi ketika program dan aplikasi tersebut dibuat. Berikut perancangan antarmuka yang terdapat pada aplikasi yang dibangun :

1. Tampilan *Login*

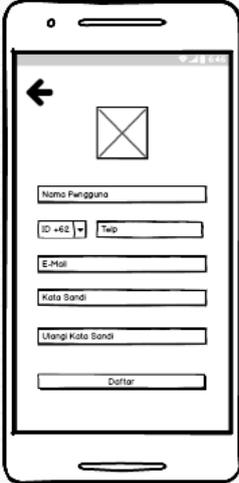
Tampilan *login* memberikan kejelasan halaman mengenai form untuk masuk dan mengakses semua fitur yang terdapat di dalam aplikasi. Tampilan *login* dapat dilihat pada Gambar 3.39 Tampilan *login*.



**Gambar 3.39 Tampilan Login**

## 2. Tampilan Registrasi

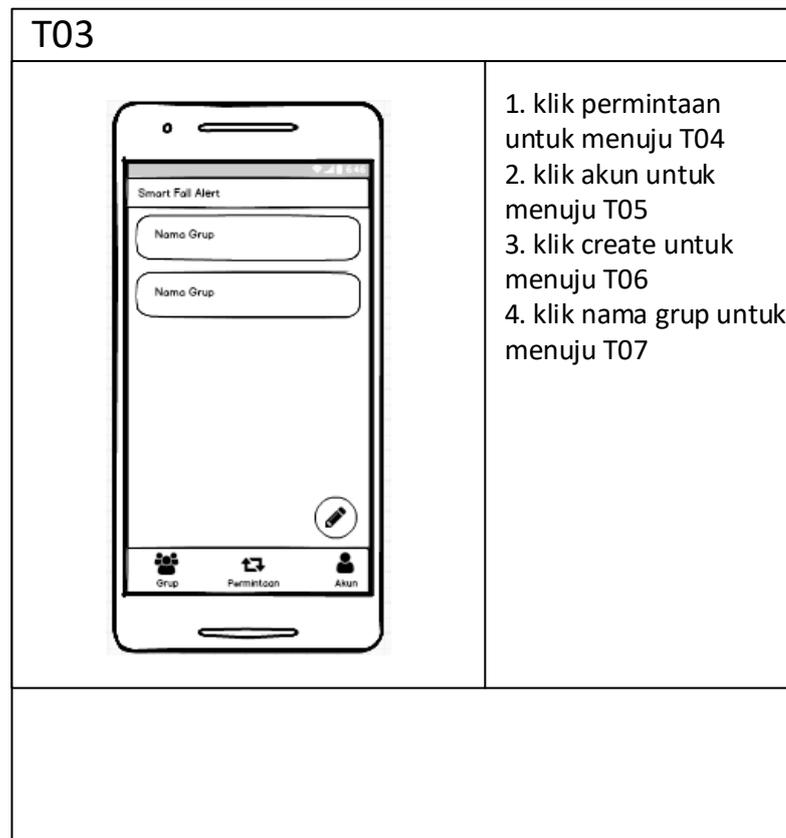
Tampilan registrasi memberikan kejelasan halaman mengenai form untuk registrasi. Tampilan registrasi dapat dilihat pada Gambar 3.40 Tampilan registrasi.

T02	
	1. klik daftar untuk mendaftar dan menuju T03

**Gambar 3.40 Tampilan Registrasi**

### 3. Tampilan Grup

Tampilan grup memberikan kejelasan halaman utama dari aplikasi dan untuk melihat daftar grup dan menampilkan menu. Tampilan grup dapat dilihat pada Gambar 3.41 Tampilan Grup.



**Gambar 3.41 Tampilan Grup**

#### 4. Tampilan Permintaan

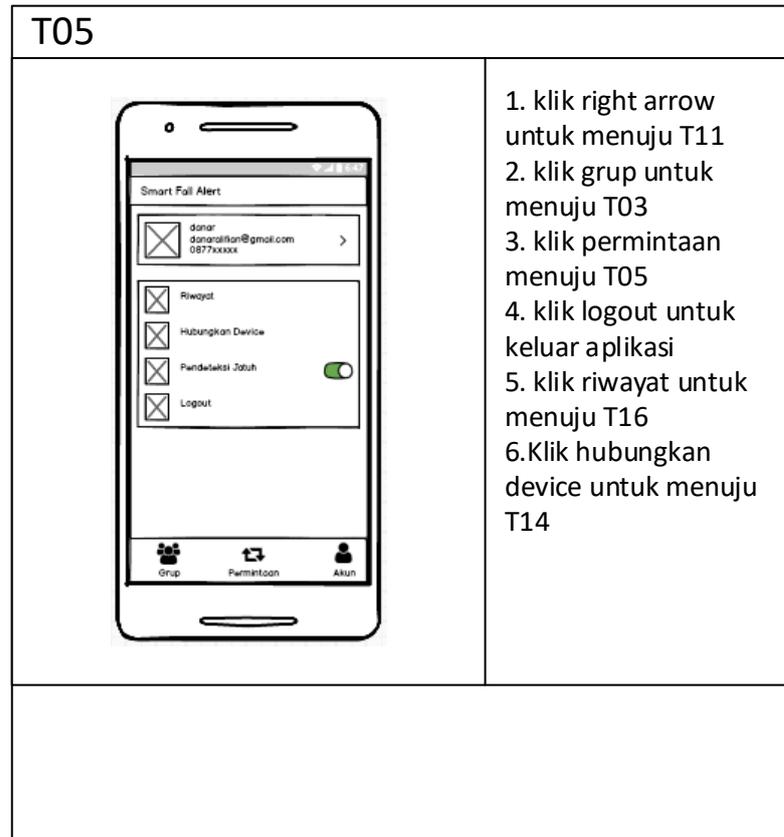
Tampilan permintaan memberikan kejelasan halaman permintaan dari aplikasi dan untuk melihat daftar permintaan gabung grup. Tampilan permintaan dapat dilihat pada Gambar 3.42 Tampilan Permintaan.



**Gambar 3.42 Tampilan Permintaan**

#### 5. Tampilan Akun

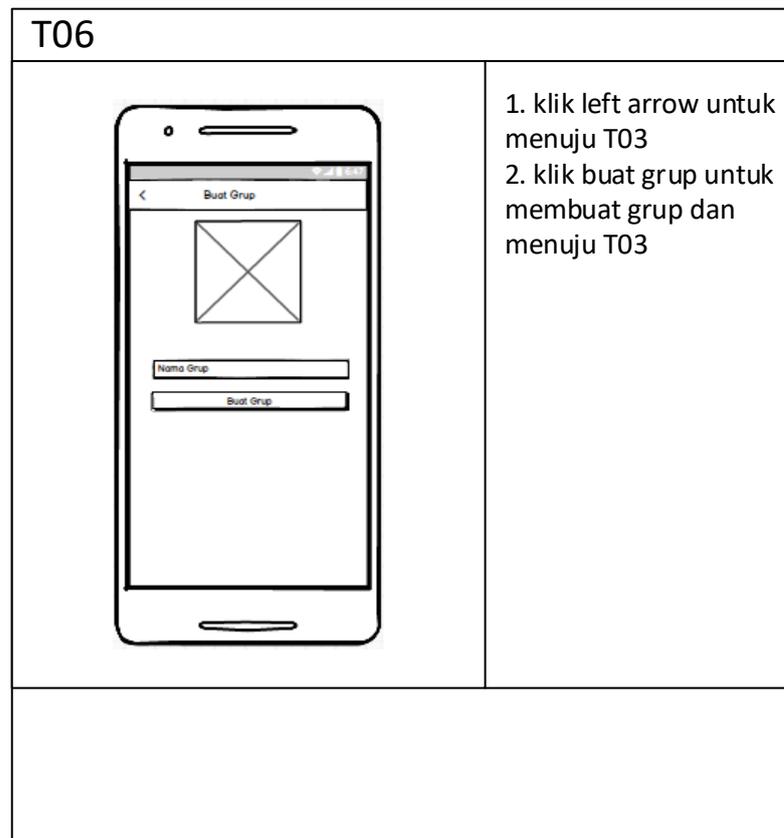
Tampilan Akun memberikan kejelasan halaman mengenai informasi dari pengguna aplikasi. Tampilan akun dapat dilihat pada Gambar 3.43 Tampilan Akun.



**Gambar 3.43 Tampilan Akun**

#### 6. Tampilan Buat Grup

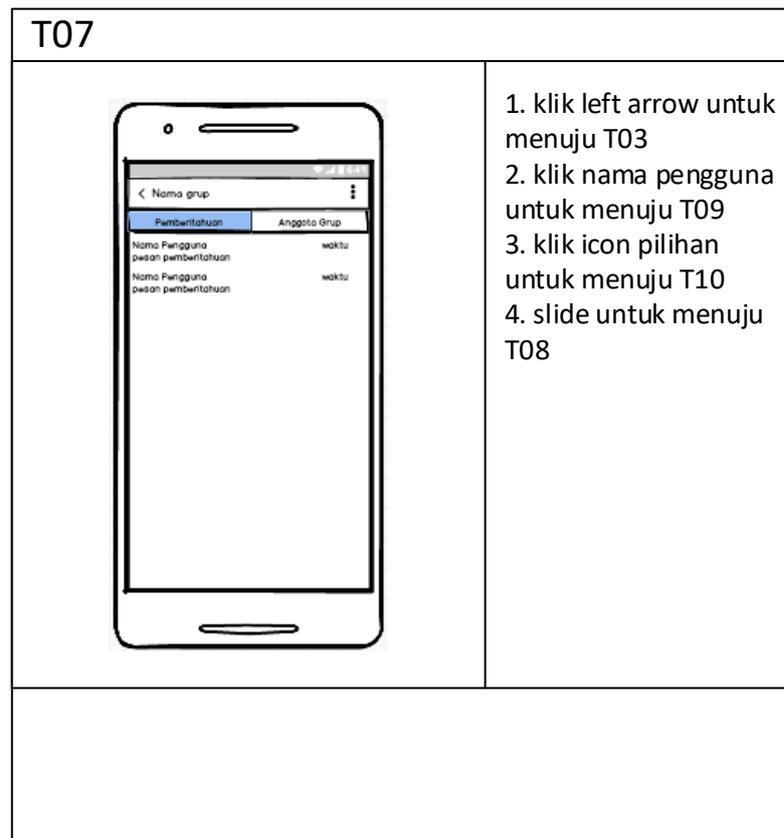
Tampilan buat grup memberikan kejelasan halaman mengenai form untuk membuat grup untuk melakukan monitoring. Tampilan buat grup dapat dilihat pada Gambar 3.44 Tampilan Buat Grup.



**Gambar 3.44 Tampilan Buat Grup**

#### 7. Tampilan Pemberitahuan

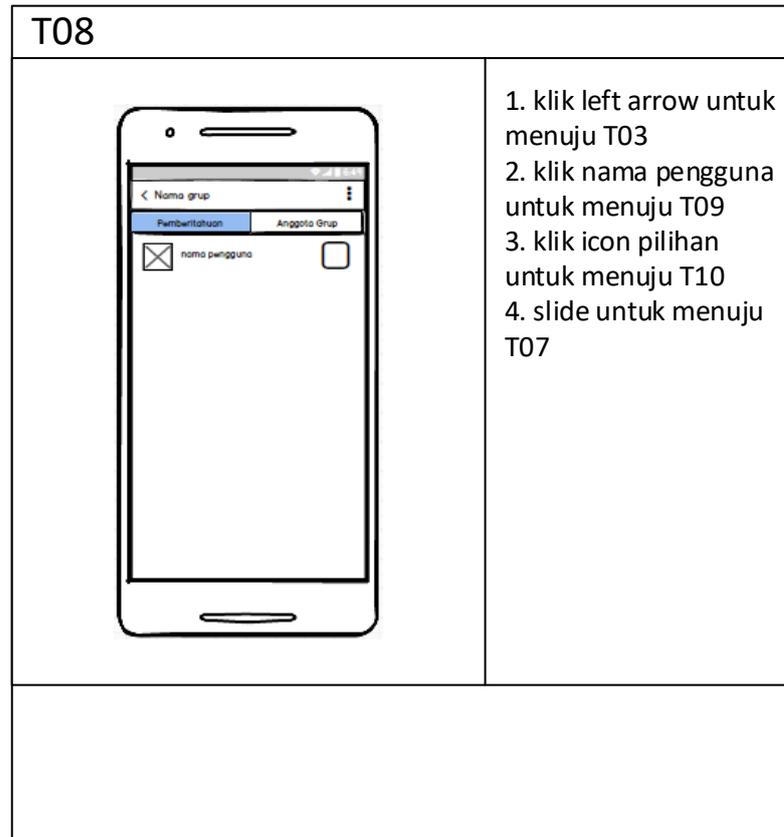
Tampilan pemberitahuan memberikan kejelasan untuk menampilkan pemberitahuan kejadian jatuh. Tampilan pemberitahuan dapat dilihat pada Gambar 3.45 Tampilan Pemberitahuan.



**Gambar 3.45 Tampilan Pemberitahuan**

#### 8. Tampilan Daftar Anggota

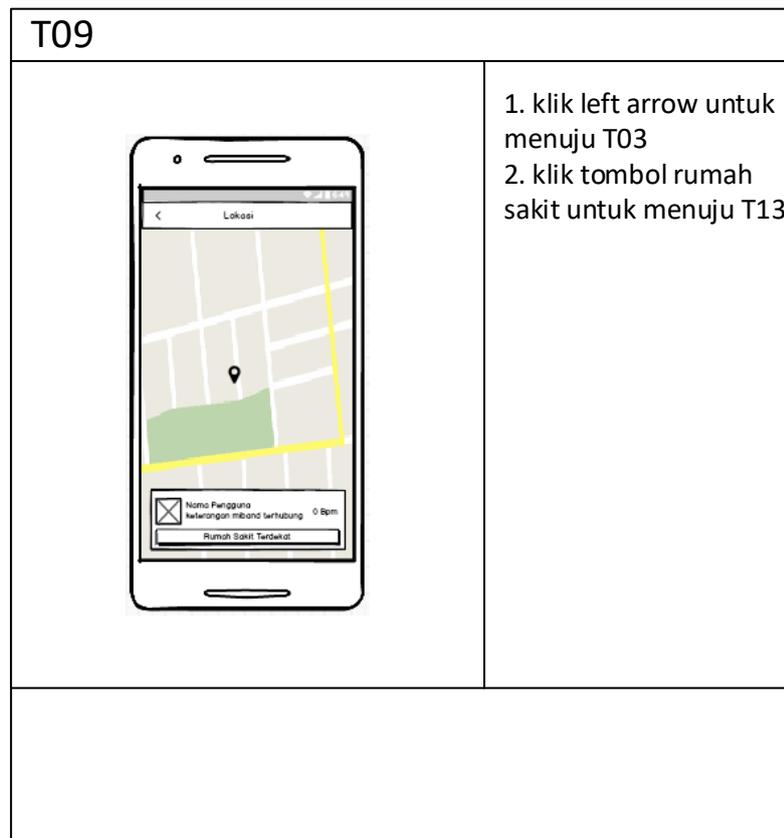
Tampilan tampil lokasi memberikan kejelasan untuk menampilkan daftar anggota dalam sebuah grup. Tampilan daftar anggota dapat dilihat pada Gambar 3.46 Tampilan Daftar Anggota.



**Gambar 3.46 Tampilan Tampil Lokasi**

9. Tampilan Tampil Lokasi

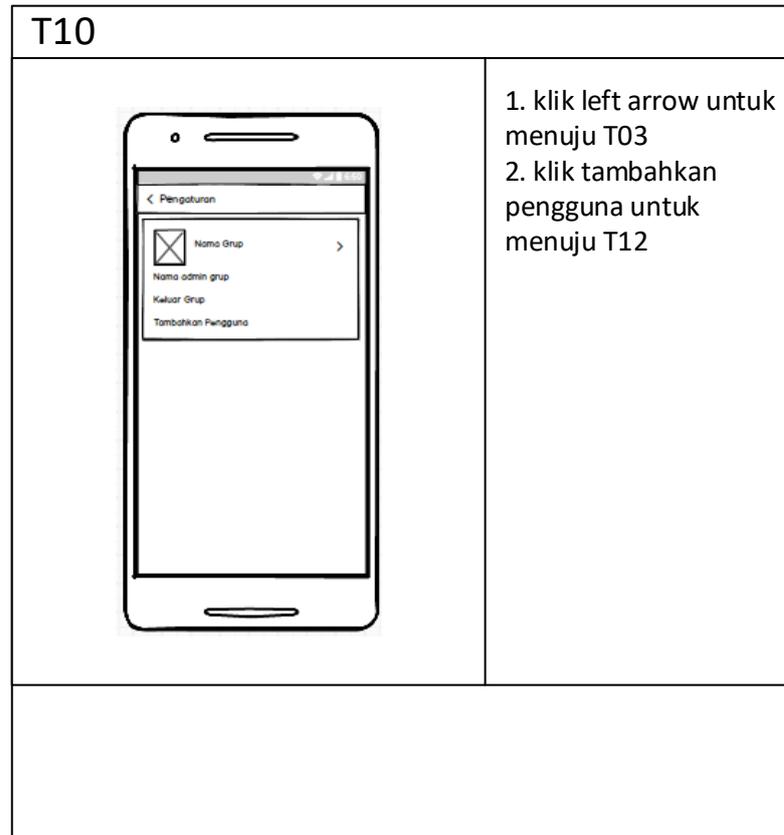
Tampilan tampil lokasi memberikan kejelasan halaman untuk menampilkan lokasi kejadian jatuh. Tampilan tampil lokasi dapat dilihat pada Gambar 3.47 Tampilan Lokasi.



**Gambar 3.47 Tampilan Tampil Lokasi**

#### 10. Tampilan Detail grup

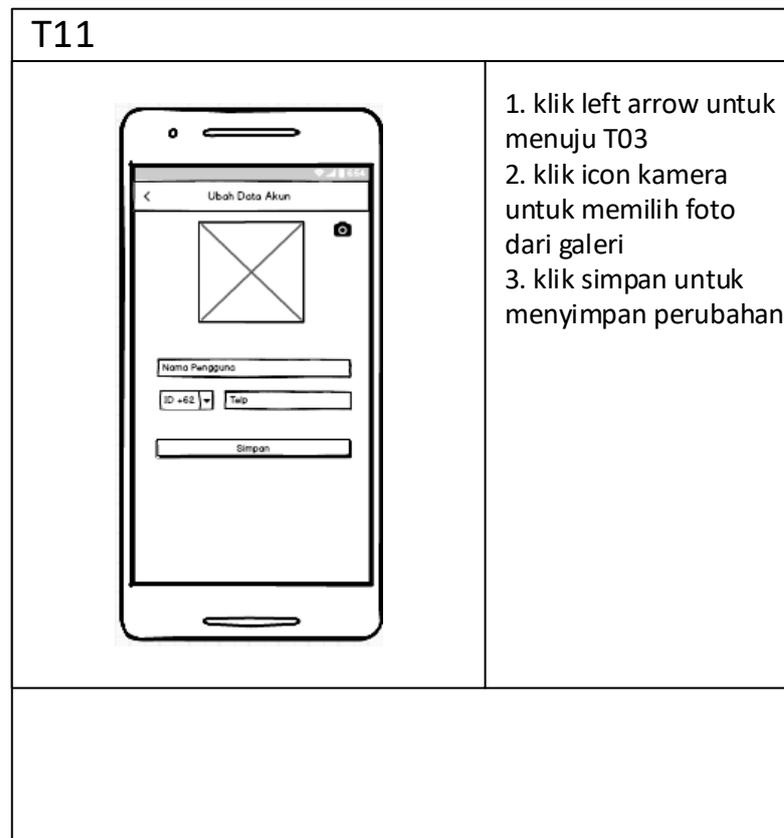
Tampilan detail grup memberikan kejelasan halaman untuk menampilkan detail dari sebuah grup. Tampilan detail grup dapat dilihat pada Gambar 3.48 Tampilan Detail grup.



**Gambar 3.48 Tampilan Detail Grup**

#### 11. Tampilan Ubah Data Akun

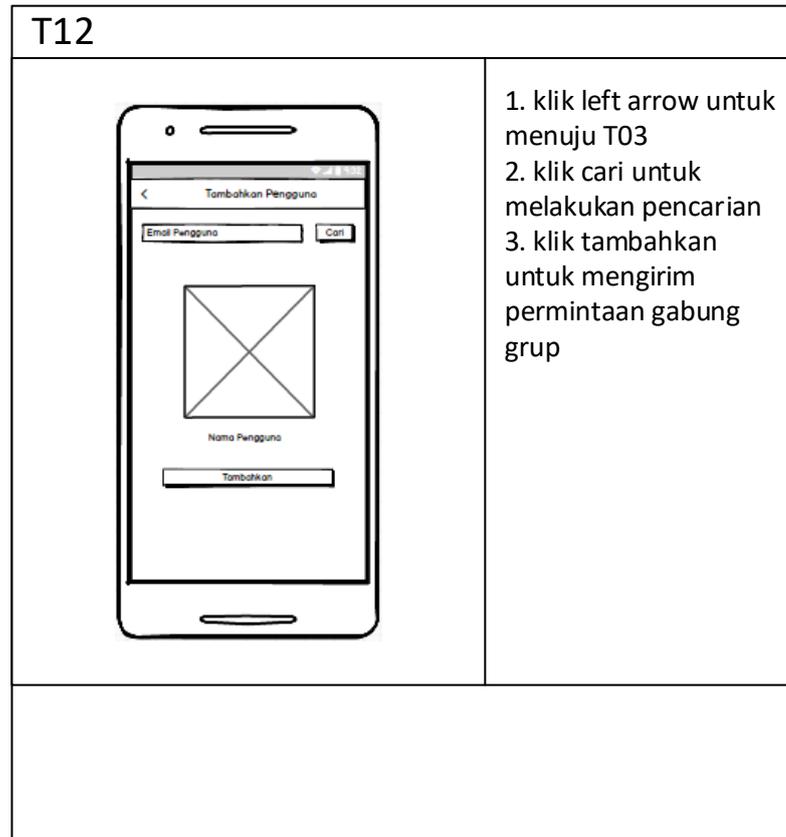
Tampilan ubah data akun memberikan kejelasan halaman mengenai form ubah data akun. Tampilan ubah data akun dapat dilihat pada Gambar 3.49 Tampilan Ubah Data Akun.



**Gambar 3.49 Tampilan Ubah Data Akun**

## 12. Tampilan Tambah Pengguna

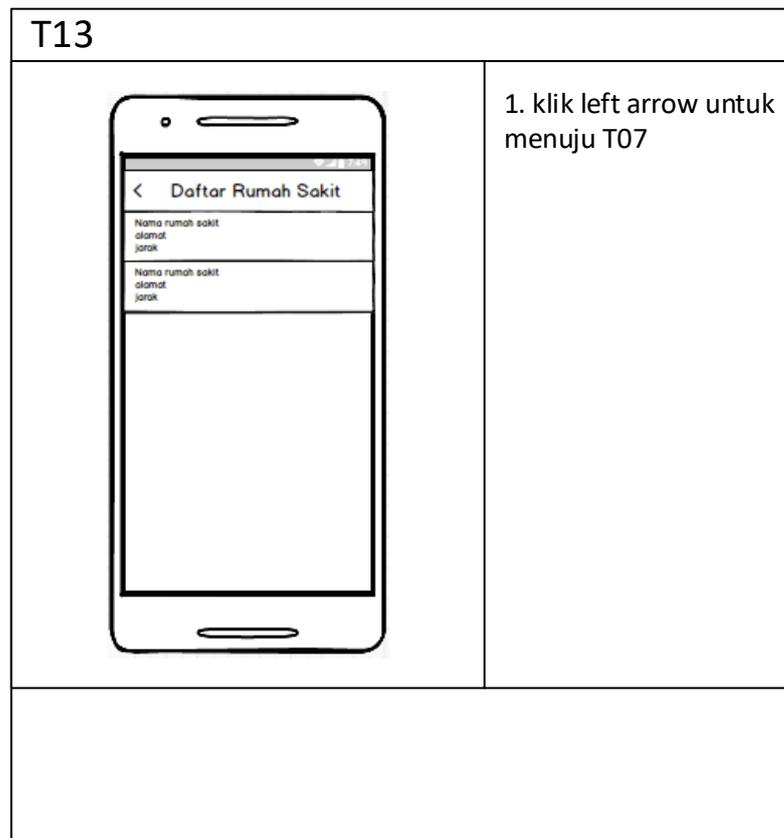
Tampilan tambah pengguna memberikan kejelasan halaman untuk menambah pengguna yang akan dimonitoring atau yang memonitoring. Tampilan tambah pengguna dapat dilihat pada Gambar 3.50 Tampilan Tambah Pengguna.



**Gambar 3.50 Tampilan Tambah Pengguna**

### 13. Tampilan Daftar Rumah Sakit

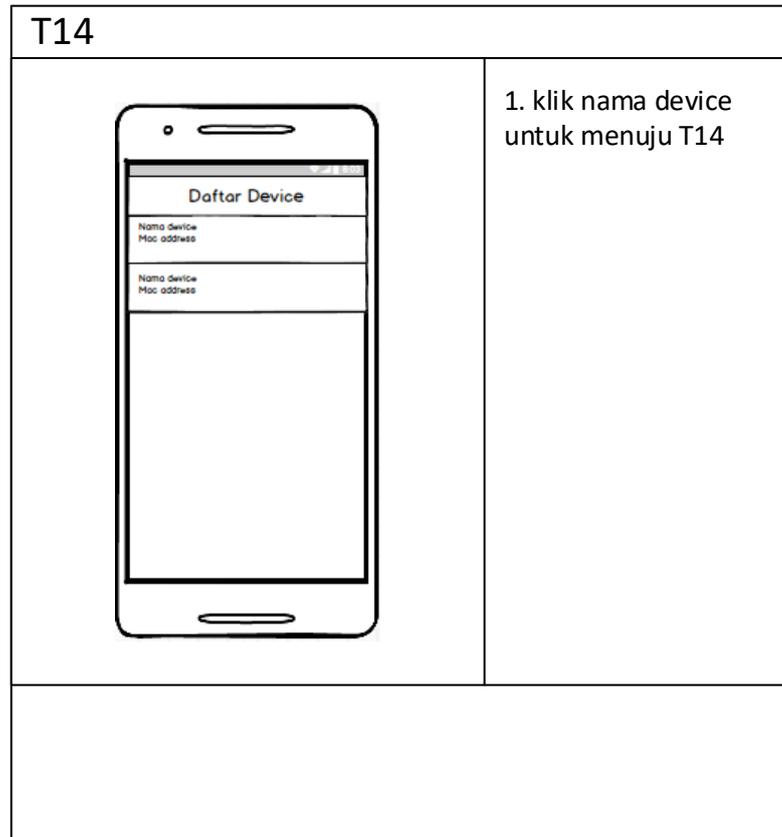
Tampilan daftar rumah sakit memberikan kejelasan halaman untuk menampilkan daftar rumah sakit terdekat. Tampilan daftar rumah sakit dapat dilihat pada Gambar 3.51 Tampilan Daftar Rumah Sakit.



**Gambar 3.51 Tampilan Daftar Rumah Sakit**

#### 14. Tampilan Pindai *Device*

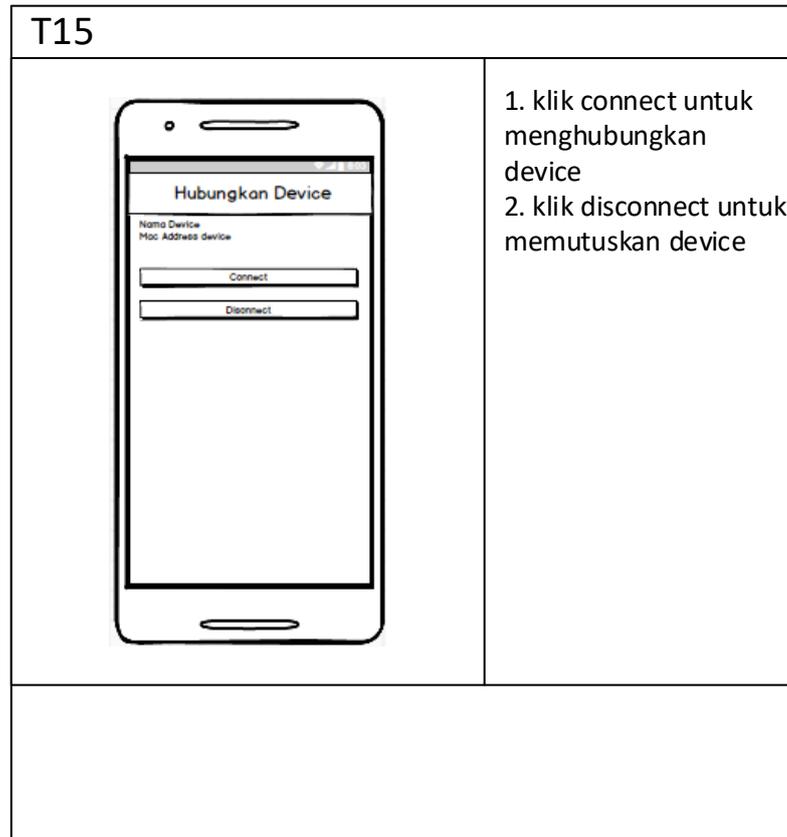
Tampilan pindai *device* memberikan kejelasan halaman untuk menampilkan *device* yang terpindai. Tampilan pindai *device* dapat dilihat pada Gambar 3.52 Tampilan Pindai *Device*.



**Gambar 3.52 Tampilan Pindai *Device***

#### 15. Tampilan Hubungkan *Device*

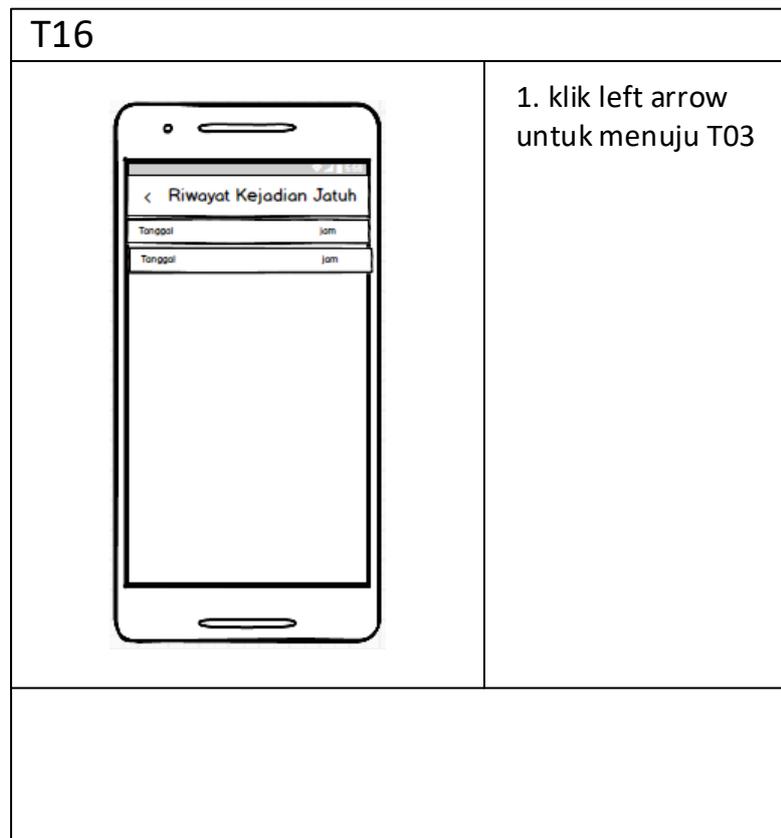
Tampilan hubungkan *device* memberikan kejelasan halaman untuk menghubungkan *device*. Tampilan hubungkan *device* dapat dilihat pada Gambar 3.53 Tampilan Hubungkan *Device*.



**Gambar 3.53** *Hubungkan Device*

#### 16. Tampilan Riwayat

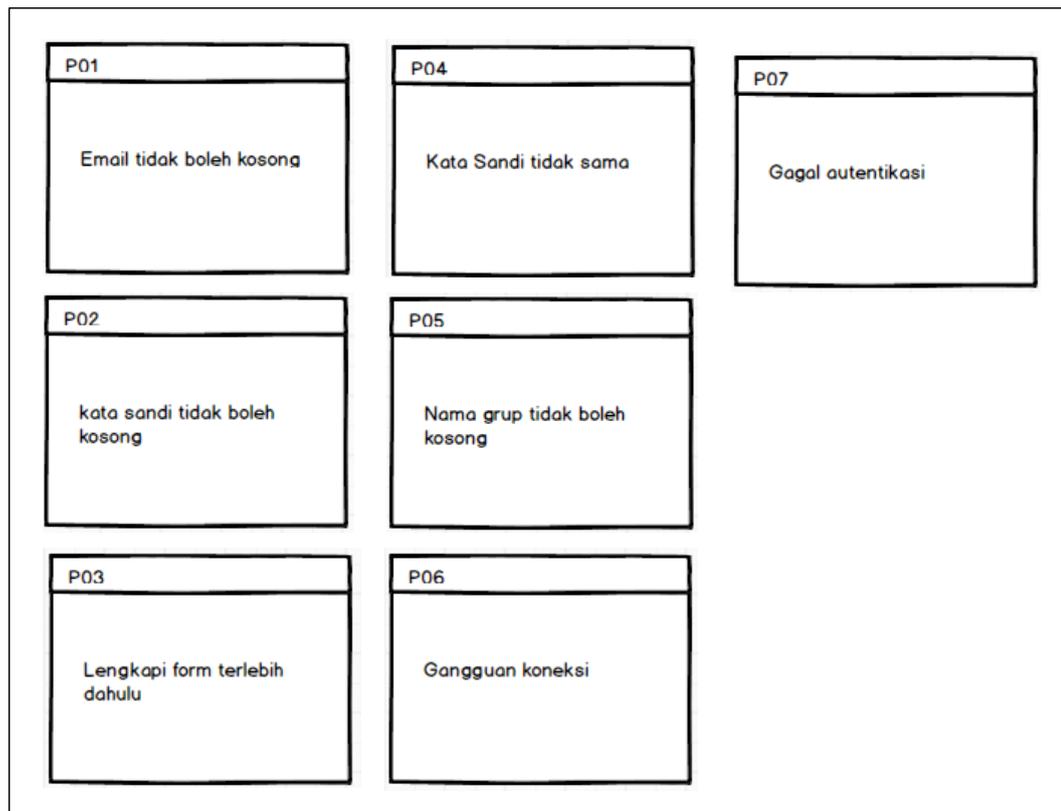
Tampilan riwayat memberikan kejelasan halaman untuk menampilkan riwayat kejadian jatuh yang dialami oleh seorang pengguna. Tampilan riwayat dapat dilihat pada Gambar 3.54 Tampilan Riwayat



**Gambar 3.54 Tampilan Riwayat**

### **3.2.3 Perancangan Pesan**

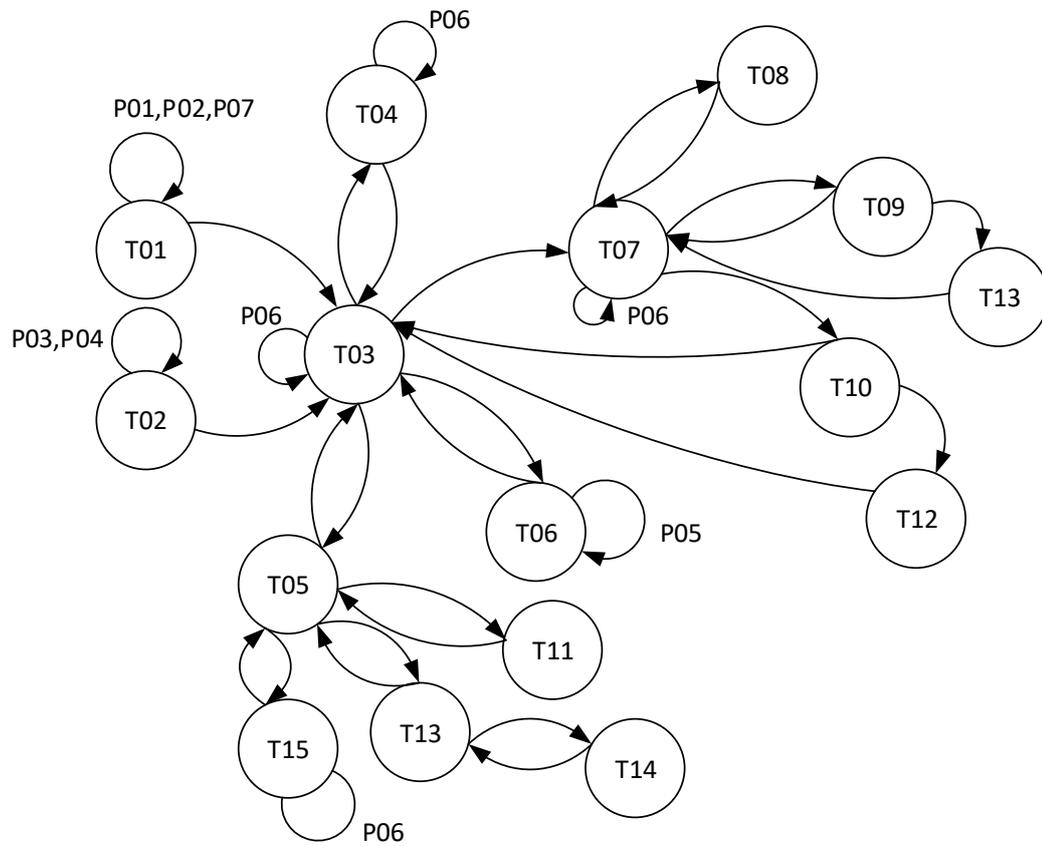
Perancangan pesan di dalam aplikasi memberikan kejelasan jika masukan dari pengguna sesuai atau tidak sesuai dengan keinginan sistem. Perancangan tampilan pesan pada aplikasi yang dibangun dapat dilihat pada gambar 3.55 Perancangan Tampilan Pesan.



**Gambar 3.55 Perancangan Tampilan Pesan**

### 3.2.4 Jaringan Semantik

Jaringan semantik merupakan gambar grafik yang menunjukkan relasi/hubungan antar perancangan antarmuka dan pesan. Jaringan semantik untuk antarmuka dapat dilihat pada gambar 3.56 Jaringan Semantik.



**Gambar 3.56 Jaringan Semantik**