BAB 3

ANALISIS DAN PERANCANGAN SISTEM

Analisis dan perancangan aplikasi memerlukan tahapan yang sistematis untuk mendapatkan aplikasi yang baik dan bersesuaian dengan tujuannya. Tahapan awal dari analisis adalah menganalisis masalah dan analisis arsitektur sistem. Sedangkan untuk tahap perancangan aplikasi yaitu perancangan antarmuka

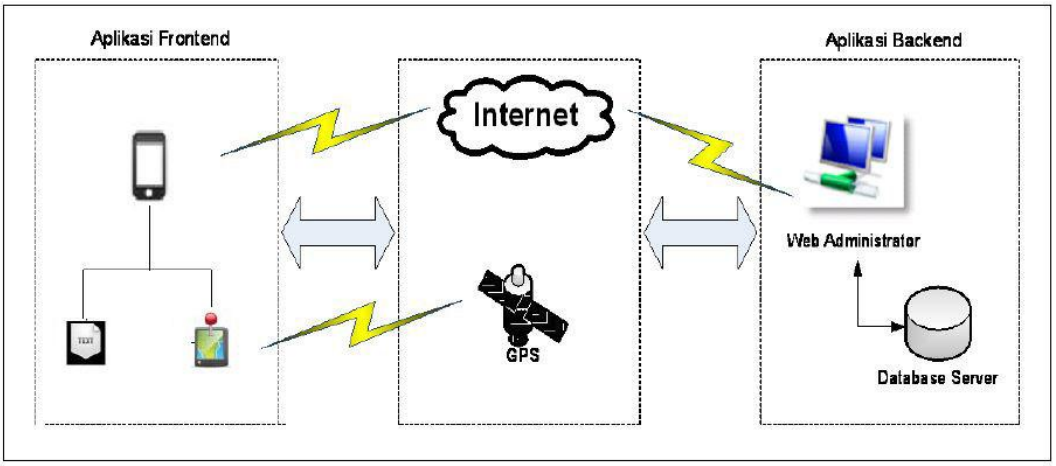
3.1 Analisis Masalah

Sulitnya menemukan informasi mengenai rumah sakit, puskesmas, klinik pemadam kebakaran, dan polisi menjadi kendala bagi semua orang yang sedang mencari informasi tersebut dalam keadaan darurat. Ini semua disebabkan karena kurang atau bahkan tidak adanya informasi mengenai lokasi *emergency service*. Cara yang dapat dilakukan untuk mendapatkan informasi tersebut adalah dengan bertanya kepada masyarakat lain atau berkunjung ke kantor dinas terkait untuk mendapatkan informasi yang dicari. Namun cara tersebut kurang efektif, mengingat bahwa orang di sekitar belum tentu mengetahui informasi yang dibutuhkan

Oleh karena itu dibutuhkan sebuah sistem yang efektif, cepat dan mudah digunakan dalam memberikan informasi tersebut yaitu dalam sebuah aplikasi *mobile* maka dibutuhkan sebuah sistem yang menjadi dua sisi, sisi *frontend* yang merupakan aplikasi *mobile*. yang dipakai oleh masyarakat untuk mengakses informasi dan juga bisa memberikan informasi yang berisi informasi seputar *emergency services* serta data GPS dan di sisi *backend* terdapat aplikasi web yang berguna untuk menampung informasi, serta dapat menampilkan dan meng-update informasi yang dikirimkan oleh masyarakat karena dengan perangkat *mobile* aplikasi ini dapat digunakan kapan saja dan dimana saja asalkan terintegrasi dengan jaringan internet.

3.2 Analisis Arsitektur Sistem

Arsitektur fisik sistem mengacu kepada model arsitektur aplikasi two tier. arsitektur fisik sistem terdiri dari dua komponen utama, yaitu aplikasi *frontend* yang berjalan pada pihak *client* dan aplikasi *backend*. Pada sistem ini, aplikasi *frontend* merupakan komponen yang aktif, sedangkan aplikasi *backend* bertindak sebagai penerima dan pengolah data. Arsitektur sistem dalam pembangunan aplikasi ini ditunjukan oleh gambar 3.1



Gambar 3. 1 Arsitektur Sistem

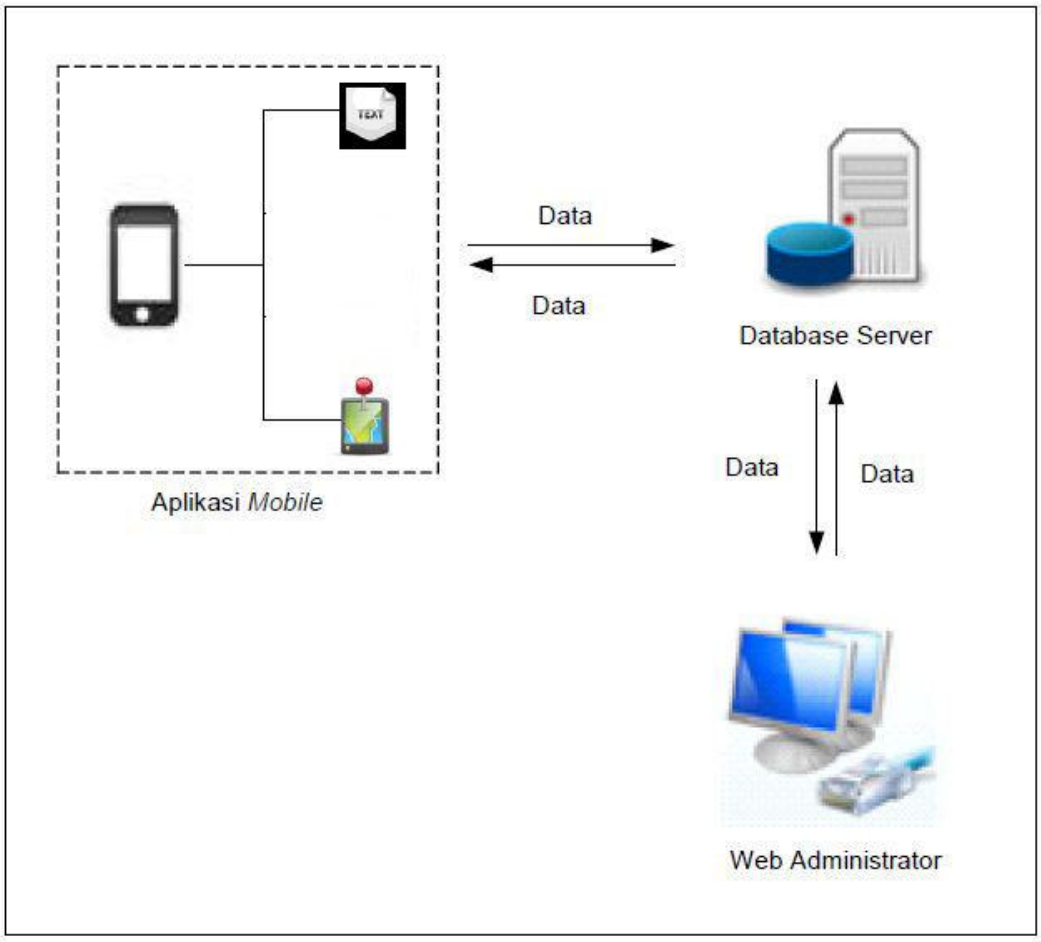
Aplikasi *frontend* adalah aplikasi yang berfungsi sebagai media untuk pengambilan data yang berhubungan dengan informasi *emergency services* melalui perangkat *mobile* terutama smartphone. GPS *receiver* yang terdapat pada perangkat *mobile* berfungsi untuk menentukan posisi pengguna dengan menggunakan koordinat longitude maupun latitude.

Aplikasi *backend* merupakan aplikasi yang berfungsi sebagai penghubung antara aplikasi *frontend* dengan database server. Aplikasi ini terdiri dari dua subbagian utama yaitu web administrator dan database server. Web administrator merupakan web yang berfungsi sebagai pengolah sumber basis data yang terdapat pada database server yang dioperasikan oleh admin. Sedangkan database server merupakan aplikasi yang berfungsi untuk menyimpan data-data yang dikirimkan oleh aplikasi *frontend*. Akses koneksi jaringan yang digunakan dari *frontend*

untuk berintegrasi dengan *backend* melalui perangkat mobile dapat menggunakan jaringan mobile GPRS/EDGE/UMTS.

3.2.1 Analisis Alur Data Sistem

Analisis alur data sistem merupakan analisis yang berfungsi untuk menggambarkan secara rinci bagaimana sistem ini dapat bekerja dapat di lihat pada gambar 3.2



Gambar 3. 2 Alur Data Sistem

Analisis ini meliputi:

1. Aplikasi *mobile*

Aplikasi *mobile* dalam penelitian ini adalah aplikasi untuk mengirim data ke database server. Data yang dikirim merupakan data multimedia dan data GPS berupa koordinat *longitude* dan *latitude*. Aplikasi ini dibangun pada platform Android dan komunikasi data antar aplikasi dengan database server menggunakan protokol HTTP.

2. Protokol HTTP

Merupakan protokol jaringan lapisan aplikasi yang digunakan untuk sistem informasi terdistribusi, kolaboratif, dan menggunakan hypermedia. Ada beberapa operasi HTTP yang salah satu contohnya adalah GET dan POST. Operasi GET digunakan untuk meminta sumber dari server. Operasi POST digunakan untuk mengirim informasi ke sumber yang ada di server. Server akan merespon dengan header HTTP yang diikuti dengan bagian data. Pada kasus operasi pemberian GET, respon dari server akan berisi sumber yang diminta klien. Jika server menerima operasi POST, server akan melewatkan Aplikasi *Mobile* Database Server Web Administrator data yang ada pada operasi POST ke sumber yang dapat dieksekusi. Setelah server menerima kembali hasil dari sumber tersebut, server akan memasukkan hasil tersebut ke HTTP respon yang dikirim kembali ke klien.

3. Database server

Merupakan aplikasi backend yang berfungsi untuk media penyimpanan data-data yang telah dikirmkan oleh aplikasi mobile. Pada penelitian ini, bagian pengelolaan database menggunakan MySQL.

4. Web Administrator

Web administrator merupakan web yang berfungsi sebagai pengolah sumber basis data yang terdapat pada database server yang dioperasikan oleh admin. Pada bagian ini, admin dapat melakukan pengolahan data

3.2.2 Analisis Komunikasi Aplikasi *Frontend* dan Aplikasi *Backend*

Pada sistem yang dikembangkan, client (*frontend*) bertindak sebagai komponen aktif yang mengirimkan data ke server (*backend*) tanpa menunggu perintah atau permintaan dari subsistem lainnya. Sedangkan backend bertindak sebagai pihak pasif yang hanya menerima data untuk kemudian menampilkannya dalam bentuk aplikasi web.

Pada sistem yang dibangun, data yang dikirimkan ke server adalah data multimedia dan data lokasi. Data yang membutuhkan pemrosesan lebih lanjut adalah data dari modul GPS. Data yang diperoleh dari modul GPS harus diparsing terlebih dahulu untuk dapat memperoleh data lokasi.

Karena *client* berkomunikasi secara langsung dengan server, maka protokol komunikasi yang digunakan adalah protokol HTTP. *Client* akan mengirimkan paket data yang berisi *request* ke server dengan protokol HTTP yang berisi semua data yang telah diperoleh.

3.3 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional menggambarkan kebutuhan sistem yang menitikberatkan pada properti prilaku yang dimiliki oleh sistem, diantaranya kebutuhan perangkat keras, perangkat lunak, serta *user* sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan.

3.3.1 Analisis Kebutuhan Data

Setelah melakukan proses analisis kebutuhan sistem, maka analisis selanjutnya adalah analisis kebutuhan data. Analisis ini bertujuan untuk memudahkan dalam perancangan informasi. Adapun analisis kebutuhan data yang diperlukan antara lain :

1. Data Puskesmas

   Data yang akan di tampilkan adalah nama puskesmas, alamat puskesmas, nomor telepon puskesmas, status puskesmas, fasilitas puskesmas, lokasi puskesmas

2. Data Rumah sakit

   Data yang akan di tampilkan adalah nama rumah sakit, alamat rumah sakit, nomor telepon rumah sakit, jenis rumah sakit, fasilitas rumah sakit, lokasi rumah sakit

3. Data Polres

   Data yang akan di tampilkan adalah nama polres, alamat polres, nomor telepon polres, lokasi polres

4. Data Polsek

   Data yang akan di tampilkan adalah nama polsek, nomor telepon polsek, alamat polsek, lokasi polsek

5. Data Pemadam kebakaran

   Data yang akan di tampilkan adalah nama pemadam kebakaran, nomor telepon, alamat, dan lokasi

6. Data lokasi *user*

   Menampilkan lokasi *user*

7. Data Klinik

   Data yang akan di tampilkan adalah nama pemadam kebakaran, nomor telepon, alamat, dan lokasi

3.3.2 Analisis Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan berdasarkan kebutuhan minimal yang harus terpenuhi antara lain :

1. Perangkat Keras Aplikasi *Backend*

Pada aplikasi *backend* yang berupa web, aplikasi dijalankan pada sebuah perangkat komputer dengan spesifikasi kebutuhan perangkat keras komputer sebagai berikut :

Tabel 3. 1 Spesifikasi Perangkat Keras *Backend*

No	Perangkat Keras	Spesifikasi
1	Monitor	14”
2	Processor	Intel Pentium Dual Core
3	Memori / RAM	1GB
4	Harddisk	320GB
5	VGA Card	VGA On-Board
6	Keyboard	Standar
7	Mouse	Standar
8	Printer	Standar

2. Perangkat Keras Aplikasi *Emergency services*

Aplikasi *Emergency services* dijalankan pada sebuah perangkat *SmartPhone* yaitu *SmartPhone* dengan sistem operasi Android. Kebutuhan perangkat keras untuk *SmartPhone* tersebut yaitu:

Tabel 3. 2 Spesifikasi Perangkat Keras *frontend*

No	Android	Spesifikasi
1	Sistem operasi	Versi 4.0 (Ice Cream Sandwich)
2	Display	240 x 320.
3	Jaringan	• Wi-Fi 802.11 b/g/n (2.4Ghz), HSPA, EDGE Rx (850/900/1800/1900 MHZ), Wi-Fi Direct available,gsm
4	Memori	4GB / 512MB Memory
5	Chipset	Single Core Application Processor Single Core Application Processor Cortex A9 850 MHz CPU Cortex A9 850 MHz CPU
6	Baterai	1,200 mAh Battery Capacity
7	Lokasi	Assisted GPS
8	Sensor	Accerlometer / Gyroscope / Proximity available

3.3.3 Analisis Kebutuhan Perangkat Lunak

Analisis perangkat lunak terdiri dari spesifikasi minimum perangkat yang dibutuhkan. Spesifikasi perangkat lunak untuk membangun aplikasi ini adalah sebagai berikut:

Tabel 3. 3 Spesifikasi perangkat lunak

No	Perangka lunak	Spesifikasi
1	Sistem Operasi	Windows.
2	Aplikasi	IDE Eclipse 3.8.2. JDK v 7 dan Android SDK windows, ADT versi 10, Dreamweaver, Wampserver 2.4
3	Bahasa Pemrograman	PHP (Hyper Text Preproccessor), Java
4	DBMS	MySQL
5	Browser	Mozila firefox

3.3.4 Analisis Pengguna Sistem (*User*)

Analisis dan kebutuhan pengguna dimaksudkan untuk mengetahui seberapa jauh pengguna sistem ini. Hal ini bertujuan untuk mengetahui karakteristik pengguna dalam mengoperasikan sistem yang telah ada, maupun sistem usulan

Untuk menggunakan sistem yang dibangun terdapat beberapa karakteristik yang harus dimiliki oleh pengguna, kebutuhan karakteristik dapat dilihat pada table dibawah ini.

Berdasarkan hasil pengamatan, pengguna yang terlibat dalam sistem yang sedang berjalan ialah:

1. Admin, sebagai pengelola aplikasi backend (web) untuk imigrasi server. Berikut karakteristik pengguna sistem yang ditunjukan oleh tabel 3.4

Tabel 3. 4 Karakterisitik Pengguna *backend*(web)

Katregori Pengguna	Hak akses	Tingkat pendidikan	Tingkat keterampilan
Administrator	Mengakses Dan mengelola sistem	S1 Teknik informatika	Mengerti pengelolaan database, koneksi data, SAS, dan database SQL, Pengetahuan tentang metodologi manajemen proyek teknologi informasi dan siklus hidup

Tabel 3. 5 Karakterisitik Pengguna aplikasi *frontend*

Kategori pengguna	Hak akses	Tingkat Pendidikan	Tingkat keterampilan
Masyarakat	Mengakses informasi melalui aplikasi	Minimal SMP	membuka aplikasi pada *mobile* android, mengerti koneksi gps, mengrti internet

3.4 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan. Analisis kebutuhan fungsional ini meliputi analisis kebutuhan sistem, analisis kebutuhan data, spesifikasi sistem dan pemodelan sistem.

Pemodelan sistem dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain *Use case diagram*, *Class Diagram*, *Activity Diagram*, *Sequence Diagram*, *Component Diagram* dan *Deployment Diagram*

3.4.1 Analisis kebutuhan sistem

Sebelum melakukan perancangan sistem, terlebih dahulu dilakukan analisis kebutuhan sistem yaitu. Hal ini dimaksudkan agar dapat mengatasi ketidak sesuaian antara aplikasi yang dirancang dengan kebutuhan pengguna. Adapun kebutuhan sistem yang diperlukan antara lain :

1. Menampilkan informasi mengenai data *emergency services* di Jawa Barat, meliputi data rumah sakit, data puskesmas data polres dan data polsek, data pemadam kebakaran dan klinik dengan lengkap dan rinci.
2. Menampilkan peta lokasi serta direction suatu *emergency services*
3. Menampilkan peta darurat *energency services* terdekat
4. Memberikan rekomendasi menuju *emergency services* yang tepat
5. Memberikan informasi data dan peta lokasi *user*
6. Memberikan detail informasi mengenai data *emergency services*
7. Memberikan fasilitas telepon darurat ke *emergency services*
8. Sistem dapat memberikan fitur pengambilan data secara *realtime*
9. Sistem dapat melakukan pengiriman data secara *realtime*

3.5 Perancangan fungsional sistem

Pada tahap ini dilakukan pemodelan sistem untuk menentukan fungsi-fungsi yang dapat dilakukan oleh sistem aplikasi serta menentukan kelas yang dibutuhkan untuk realisasi fungsi-fungsi sistem yang telah dianalisis sebelumnya dan mendeskripsikannya kedalam bentuk diagram

a. *Use Case* Diagram

Diagram *Use Case* merupakan bagian tertinggi dari fungsionalitas yang dimiliki sistem yang akan menggambarkan bagaimana seseorang atau aktor akan menggunakan dan memanfaatkan sistem. Diagram ini juga

mendeskripsikan apa yang akan dilakukan oleh sistem. *Use Case* terdiri dari tiga bagian yaitu identifikasi aktor, identifikasi *Use Case* dan skenario *Use Case*.

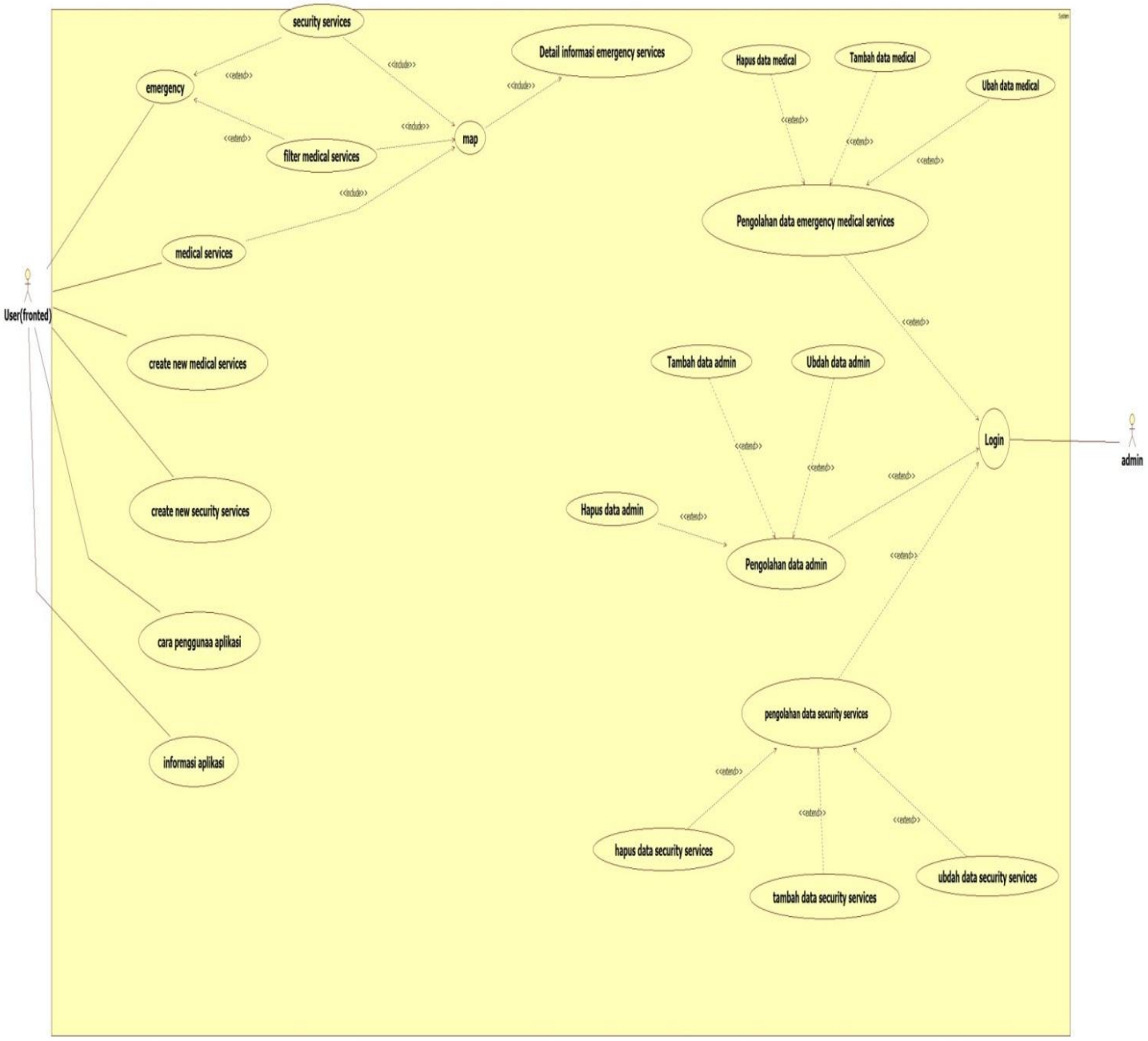
1. Identifikasi Aktor

Aktor yang berperan dalam menjalankan sistem dapat dilihat pada Tabel 3.6

Tabel 3. 6 Aktor Sistem

No	Aktor	Deskripsi
1	*User*	Merupakan aktor yang dapat mengakses dan menggunakan aplikasi frontend.
2	Administrator	Merupakan aktor yang akan melakukan pemeliharaan pada sistem web.

Pada Gambar 3.3 merupakan interaksi antara aktor dengan sistem. Dimana terdiri dari 2 aktor. *User* sebagai aktor utama yaitu pengguna berinteraksi dengan aplikasi frontend, pengguna dapat mengakses data dan mendapatkan informasi pada aplikasi, sedangkan aktor ke 2 yaitu admin berinteraksi dengan backend untuk mengolah data.



Gambar 3. 3 Usecase diagram frontend dan backend

Tabel 3. 7 Deskripsi *use case backend*

No	*Use case*	Deskripsi
UCB-01	Login Admin	Fungsionalitas untuk proses autentifikasi hak akses kepada admin
UCB-02	Pengolahan data admin	Fungsionalitas yang memungkinkan pengguna *backend* dapat mengelola data admin
UCB-3	Ubah data admin	Fungsionalitas untuk mengubah atau *update* data admin
UCB-4	Hapus data admin	Fungsionalitas untuk menghapus data admin
UCB-5	Tambah data admin	Fungsionalitas untuk menambah data admin
UCB-06	Pengolahan data *medical services*	Fungsionalitas yang memungkinkan pengguna *backend* dapat mengelola data *medical services*
UCB-7	Ubah data *medical services*	Fungsionalitas untuk mengubah atau *update* data *medical services*
UCB-8	Hapus data *medical services*	Fungsionalitas untuk menghapus data *medical services*
UCB-9	Tambah data *medical services*	Fungsionalitas untuk menambah data *medical services*
UCB-10	Pengolahan data *security services*	Fungsionalitas yang memungkinkan pengguna *backend* dapat mengelola data *security services*
UCB-11	Ubah data *security services*	Fungsionalitas untuk mengubah atau *update* data *security services*
UCB-12	Hapus data *security services*	Fungsionalitas untuk menghapus data *security services*
UCB-13	Tambah data *security services*	Fungsionalitas untuk menambah data *security services*

Tabel 3. 8 Deskripsi *usecase frontend*

No	*Use case*	Deskripsi
UCF-01	*Emergency*	Fcungsionalitas untuk memilih kategori *emergency services* yang tepat
UCF-02	S*ecurity services*	Fungionalitas untuk melihat lokasi *emergency security services* yang meliputi polsek, polres, pemadam kebakaran pada map di sekitar user
UCF-03	*Filter medical services*	Fungsionalitas untuk mengubah atau *update* data admin
UCF-04	Informasi cara penggunaan aplikasi	Fungsionalitas untuk menghapus data admin
UCF-05	Informasi aplikasi	Fungsionalitas untuk menampilkan tentang aplikasi *emergency services*
UCF-06	Detail *emergency services*	Fungsionalitas untuk melihat detail dari *emergency services* yang terdiri dari berbagai macam menu anatara lain menu *Emergency call, Get direction*
UCF-7	*Create new medical services*	Fungsionalitas untuk *user* menambah data *medical services* melalui *mobile*
UCF-8	*Create new security services*	Fungsionalitas untuk *user* menambah data *security services* melalui *mobile*
UCF-9	Map	Fungsionalitas untuk menampilkan lokasi *emergency services* yang dipilih oleh *user* pada map
UCF-10	*Medical services*	Fungionalitas untuk melihat lokasi *emergency medical services* yang meliputi puskesmas,rumah sakit,klinik pada map di sekitar user

3.6 Skenario Use Case *Frontend*

Skenario proses-proses yang terdapat didalam *Use Case frontend* dapat dilihat pada tabel-tabel dibawah ini.

Identifikasi		
Nomor	UCF-02	
Nama *Use Case*	S*ecurity services*	
Tujuan	untuk melihat lokasi *emergency security services* terdekat	
Aktor	*User*	
Skenario Utama		
Kondisi Awal	Menampilkan kategori *emergency services*.	
Aksi Aktor		Reaksi Sistem
1. Menekan *security services*		
		2. Menampilkan lokasi *security services* terdekat
Kondisi Akhir	Menampilkan lokasi *emergency security services* pada map	

Identifikasi

Tabel 3. 9 Lihat *emergency*

Identifikasi		
Nomor	UCF-01	
Nama *Use Case*	*Emergency*	
Tujuan	untuk melihat dan memilih kategori *emergency services* yang tepat	
Aktor	*User*	
Skenario Utama		
Kondisi Awal	Menampilkan tampilan utama aplikasi	
Aksi Aktor		**Reaksi Sistem**
1. Menekan *emergency*		
		2. Menampilkan kategori *emergency services*.
Kondisi Akhir	Sistem akan menampilkan kategori *emergency services*.	

Tabel 3. 10 Lihat *security services*

Nomor	UCF-03	
Nama *Use Case*	*Filter medical services*	
Tujuan	Untuk mencari fasilitas *medical emergency services* apa saja yang tersedia dan terdekat dengan user	
Aktor	*User*	
Skenario Utama		
Kondisi Awal	Menampilkan kategori *emergency services*.	
Aksi Aktor		**Reaksi Sistem**
1. Menekan *Filter medical services*		
		2. Menampilkan form pencarian fasilitas
3. Mengisi form pencarian lalu tekan ok		
		4. Menampilkan lokasi terdekat *emergency medical services* sesuai pencarian fasilitas
Kondisi Akhir	Menampilkan lokasi terdekat *emergency medical services* sesuai pencarian fasilitas	

Tabel 3. 11 lihat kategori emergency services

Tabel 3. 11 Informasi cara penggunaan aplikasi

Identifikasi		
Nomor	UCF-04	
Nama *Use Case*	Informasi cara penggunaan aplikasi	
Tujuan	untuk memberitahuakan cara penggunaan aplikasi *emergency services*	
Aktor	*User*	
Skenario Utama		
Kondisi Awal	Menampilkan tampilan utama aplikasi	
Aksi Aktor		**Reaksi Sistem**
1. Menekan informasi cara penggunaan aplikasi		
		2. Menampilkan cara penggunaan aplikasi
Kondisi Akhir	Menampilkan cara penggunaan aplikasi	

Tabel 3. 12 Informasi aplikasi

Identifikasi		
Nomor	UCF-05	
Nama *Use Case*	Informasi aplikasi	
Tujuan	untuk menampilkan informasi tentang aplikasi *emergency services*	
Aktor	*User*	
Skenario Utama		
Kondisi Awal	Menampilkan tampilan utama aplikasi	
Aksi Aktor		**Reaksi Sistem**
1. Menekan *Button* informasi aplikasi		
		2. Menampilkan informasi tentang aplikasi
Kondisi Akhir	Menampilkan informasi aplikasi *emergency services*	

Tabel 3. 13 Detail *emergency services*

Identifikasi		
Nomor	UCF-06	
Nama *Use Case*	Detail *emergency services*	
Tujuan	untuk melihat detail dari *emergency services* yang terdiri dari berbagai macam menu	
Aktor	*User*	
Skenario Utama		
Kondisi Awal	Menampilkan detail *emergency services*	
Aksi Aktor		**Reaksi Sistem**
1. Menekan button Call		
		2. Menghubungi nomer telepon *emergency services*
3. Menekan button get direction		
		4. Menunjukan rute dari posisi *user* ke posisi *emergency services* yang dipilih
Kondisi Akhir	Menampilkan detail *emergency services*	

Tabel 3. 14 *Create new medical services*

Identifikasi		
Nomor	UCF-7	
Nama *Use Case*	*Create new medical services*	
Tujuan	*user* menambah data *medical services* melalui *mobile*	
Aktor	*User*	
Skenario Utama		
Kondisi Awal	Menampilkan halaman utama aplikasi	
Aksi Aktor		**Reaksi Sistem**
1. Memilih *Create new medical services*		
		2. Menampilkan form pengisian data
3. Mengisi form dan tekan simpan		
		4. Menampilkan data data sudah disimpan
Kondisi Akhir	Menampilkan form pengisian data	

Tabel 3. 15 *Create new security services*

Identifikasi		
Nomor	UCF-8	
Nama *Use Case*	*Create new security services*	
Tujuan	*user* menambah data *security services* melalui *mobile*	
Aktor	*User*	
Skenario Utama		
Kondisi Awal	Menampilkan halaman utama aplikasi	
Aksi Aktor		**Reaksi Sistem**
1. Memilih *Create new security services*		
		2. Menampilkan form pengisian data
3. Mengisi form dan tekan simpan		
		4. Menampilkan data data sudah disimpan
Kondisi Akhir	Menampilkan form pengisian data	

Tabel 3. 16 Map

Identifikasi	
Nomor	UCF-9
Nama *Use Case*	Map
Tujuan	Menampilkan lokasi emergency services pada map
Aktor	*User*
Skenario Utama	
Kondisi Awal	Menampilkan kategori *emergency services*
Aksi Aktor	**Reaksi Sistem**
1. Memilih salah satu kategori *emergency services*	
	2. Menampilkan lokasi *emergency services* pada map
Kondisi Akhir	Menampilkan lokasi *emergency services* pada map

Tabel 3. 17 *Medical services*

Identifikasi	
Nomor	UCF-10
Nama *Use Case*	*Medical services*
Tujuan	untuk melihat semua lokasi *emergency medical services* terdekat
Aktor	*User*
Skenario Utama	
Kondisi Awal	Menampilkan kategori *emergency services*.
Aksi Aktor	**Reaksi Sistem**
3. Menekan *medical services*	
	4. Menampilkan lokasi *medical services* terdekat
Kondisi Akhir	Menampilkan lokasi *emergency medical services* pada map

4. Skenario *Use Case Backend*

Skenario proses-proses yang terdapat didalam *Use Case backend* dapat dilihat pada tabel-tabel dibawah ini.

Tabel 3. 18 *Login*

Identifikasi		
Nomor	UCB-01	
Nama *Use Case*	*Login*	
Tujuan	Untuk proses autentifikasi hak akses kepada admin	
Aktor	Admin	
Skenario Utama		
Kondisi Awal	Sistem menampilkan halaman login admin	
Aksi Aktor		**Reaksi Sistem**
1. Mengisi *username* dan *password* pada halaman *web login* admin		
		2. Mengautentikasi *username* dan *password* yang diinputkan 3. Jika cocok menampilkan halaman web menu utama admin. 4. Jika tidak akan menampilkan pesan kesalahan login
Kondisi Akhir	Menampilkan halaman utama admin	

Tabel 3. 19 Pengolahan data admin

Identifikasi		
Nomor	UCB-02	
Nama *Use Case*	Pengolahan data admin	
Tujuan	memungkinkan pengguna *backend* dapat mengelola data admin	
Aktor	admin	
Skenario Utama		
Kondisi Awal	Sistem menampilkan halaman utama pengolahan data admin	
Aksi Aktor		**Reaksi Sistem**
1. Memilih pengolahan data admin		
		2. Menampilkan halaman data admin
Kondisi Akhir	Menampilkan halaman data admin	

Tabel 3. 20 Ubah data admin

Identifikasi		
Nomor	UCB-03	
Nama *Use Case*	Ubah data admin	
Tujuan	untuk mengubah atau *update* data admin	
Aktor	admin	
Skenario Utama		
Kondisi Awal	Sistem menampilkan halaman utama pengolahan data admin	
Aksi Aktor		**Reaksi Sistem**
1. Memilih ubah data admin		
		2. Menampilkan halaman form ubah data admin
3. Mengisi *field- field* ubah data admin dan menekan tombol simpan		
		4. Melakukan proses ubah data pada basis data dan menyimpanya
Kondisi Akhir	Perubahan data admin sesuai dengan yang diubah	

Tabel 3. 21 Hapus data admin

Identifikasi		
Nomor	UCB-04	
Nama *Use Case*	Hapus data admin	
Tujuan	Untuk mengahapus data admin	
Aktor	admin	
Skenario Utama		
Kondisi Awal	Sistem menampilkan halaman utama pengolahan data admin	
Aksi Aktor		**Reaksi Sistem**
1. Memilih hapus data admin		
		2. Melakukan proses hapus data admin pada basis data
Kondisi Akhir	Penghapusan data admin sesuai dengan data yang dihapus.	

Tabel 3. 22 Tambah data admin

Identifikasi		
Nomor	UCB-05	
Nama *Use Case*	Tambah data admin	
Tujuan	Untuk Menambah data admin	
Aktor	admin	
Skenario Utama		
Kondisi Awal	Sistem menampilkan halaman utama pengolahan data admin	
Aksi Aktor		**Reaksi Sistem**
1. Memilih tambah data admin		
		2. Menampilkan halaman form tambah admin
3. Mengisi field-field pada halaman tambah admin		
		4. Melakukan proses tambah admin pada basis data dan menyimpannya.
Kondisi Akhir	Penambahan data admin sesuai dengan yang ditambahkan	

Tabel 3. 22 Pengolahan data *Medical services*

Identifikasi		
Nomor	UCB-06	
Nama *Use Case*	Pengolahan data *medical services*	
Tujuan	pengguna *backend* dapat mengelola data *medical services*	
Aktor	Admin	
Skenario Utama		
Kondisi Awal	Sistem menampilkan halaman pengolahan *medical services*	
Aksi Aktor		**Reaksi Sistem**
1. Memilih pengolahan data *medical services*		
		2. Menampilkan halaman *medical services*
Kondisi Akhir	Menampilkan data *medical services*	

Tabel 3. 23 Ubah data *medical services*

Identifikasi		
Nomor	UCB-07	
Nama *Use Case*	Ubah *medical services*	
Tujuan	untuk mengubah atau *update* data *medical services*	
Aktor	admin	
Skenario Utama		
Kondisi Awal	Sistem menampilkan halaman utama pengolahan data *medical services*	
Aksi Aktor		**Reaksi Sistem**
1. Memilih ubah data *medical services*		
		2. Menampilkan halaman form ubah data *medical services*
3. Mengisi *field- field* ubah data admin dan menekan tombol simpan		
		4. Melakukan proses ubah data pada basis data dan menyimpanya
Kondisi Akhir	Perubahan data *medical services* sesuai dengan yang diubah	

Tabel 3. 24 Hapus data *medical services*

Identifikasi		
Nomor	UCB-08	
Nama *Use Case*	Hapus *medical services*	
Tujuan	Untuk mengahapus data *medical services*	
Aktor	admin	
Skenario Utama		
Kondisi Awal	Sistem menampilkan halaman utama pengolahan data *medical services*	
Aksi Aktor		**Reaksi Sistem**
1. Memilih hapus data *medical services*		
		2. Melakukan proses hapus data admin pada basis data
Kondisi Akhir	Penghapusan data *medical services* sesuai dengan data yang dihapus.	

Tabel 3. 25 Tambah *medical services*

Identifikasi		
Nomor	UCB-09	
Nama *Use Case*	Tambah *medical services*	
Tujuan	Untuk Menambah data *medical services*	
Aktor	admin	
Skenario Utama		
Kondisi Awal	Sistem menampilkan halaman utama pengolahan data *medical services*	
Aksi Aktor		**Reaksi Sistem**
1. Memilih tambah data *medical services*		
		2. Menampilkan halaman form tambah *medical services*
3. Mengisi field-field pada halaman Tambah *medical services*		
		4. Melakukan proses tambah *medical services* pada basis data dan menyimpannya.
Kondisi Akhir	Penambahan kategori sesuai dengan yang ditambahkan	

Tabel 3. 26 Pengolahan data *Security services*

Identifikasi		
Nomor	UCB-10	
Nama *Use Case*	Pengolahan data s*ecurity services*	
Tujuan	pengguna *backend* dapat mengelola data *Security services*	
Aktor	Admin	
Skenario Utama		
Kondisi Awal	Sistem menampilkan halaman pengolahan data *Security services*	
Aksi Aktor		**Reaksi Sistem**
1. Memilih pengolahan data *Security services*		
		2. Menampilkan halaman data *Security services*
Kondisi Akhir	Menampilkan halaman utama pengolahan *Security services*	
Identifikasi		

Tabel 3. 27 Ubah data *Security services*

Identifikasi		
Nomor	UCB-11	
Nama *Use Case*	Ubah data *Security services*	
Tujuan	untuk mengubah atau *update* data *medical services*	
Aktor	admin	
Skenario Utama		
Kondisi Awal	Sistem menampilkan halaman utama pengolahan data data *Security services*	
Aksi Aktor		**Reaksi Sistem**
1. Memilih ubah data data *Security services*		
		2. Menampilkan halaman form ubah data *Security services*
3. Mengisi *field- field* ubah data admin dan menekan tombol simpan		
		4. Melakukan proses ubah data pada basis data dan menyimpanya
Kondisi Akhir	Perubahan data *Security services* sesuai dengan yang diubah	

Nomor	UCB-12
Nama *Use Case*	Hapus *security services*
Tujuan	Untuk mengahapus data *security services*
Aktor	admin
Skenario Utama	
Kondisi Awal	Sistem menampilkan halaman utama pengolahan data *security services*
Aksi Aktor	**Reaksi Sistem**
1. Memilih hapus data *security services*	
	2. Melakukan proses hapus data admin pada basis data
Kondisi Akhir	Penghapusan data *medical services* sesuai dengan data yang dihapus.

Tabel 3. 28 Hapus data *Security services*

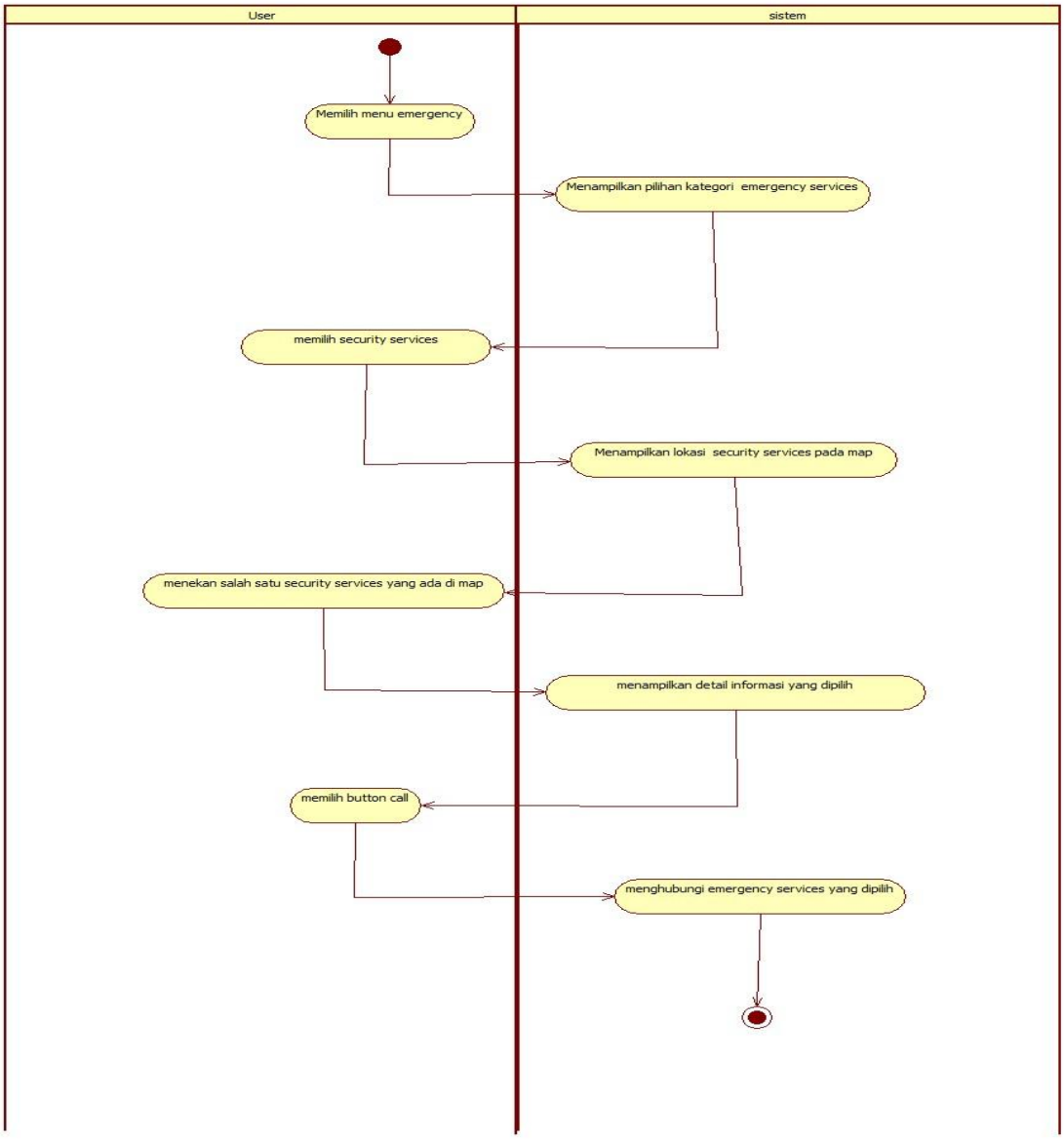
Tabel 3. 29 Tambah data *Security services*

b. *Activity* diagram

Identifikasi		
Nomor	UCB-13	
Nama *Use Case*	Tambah *security services*	
Tujuan	Untuk Menambah data *security services*	
Aktor	admin	
Skenario Utama		
Kondisi Awal	Sistem menampilkan halaman utama pengolahan data *Security services*	
Aksi Aktor		**Reaksi Sistem**
1. Memilih tambah data *security services*		
		2. Menampilkan halaman form tambah *security services*
3. Mengisi field-field pada halaman tambah *security services*		
		4. Melakukan proses tambah kategori pada basis data dan menyimpannya.
Kondisi Akhir	Penambahan kategori sesuai dengan yang ditambahkan	

Diagram activity merupakan bagian dari penggambaran sistem secara fungsional menjelaskan proses-proses logika atau fungsi yang terimplementasi oleh kode program. *Activity* Diagram memodelkan *event-event* yang terjadi didalam suatu Use Case dan digunakan untuk pemodelan aspek dinamis dari sistem.

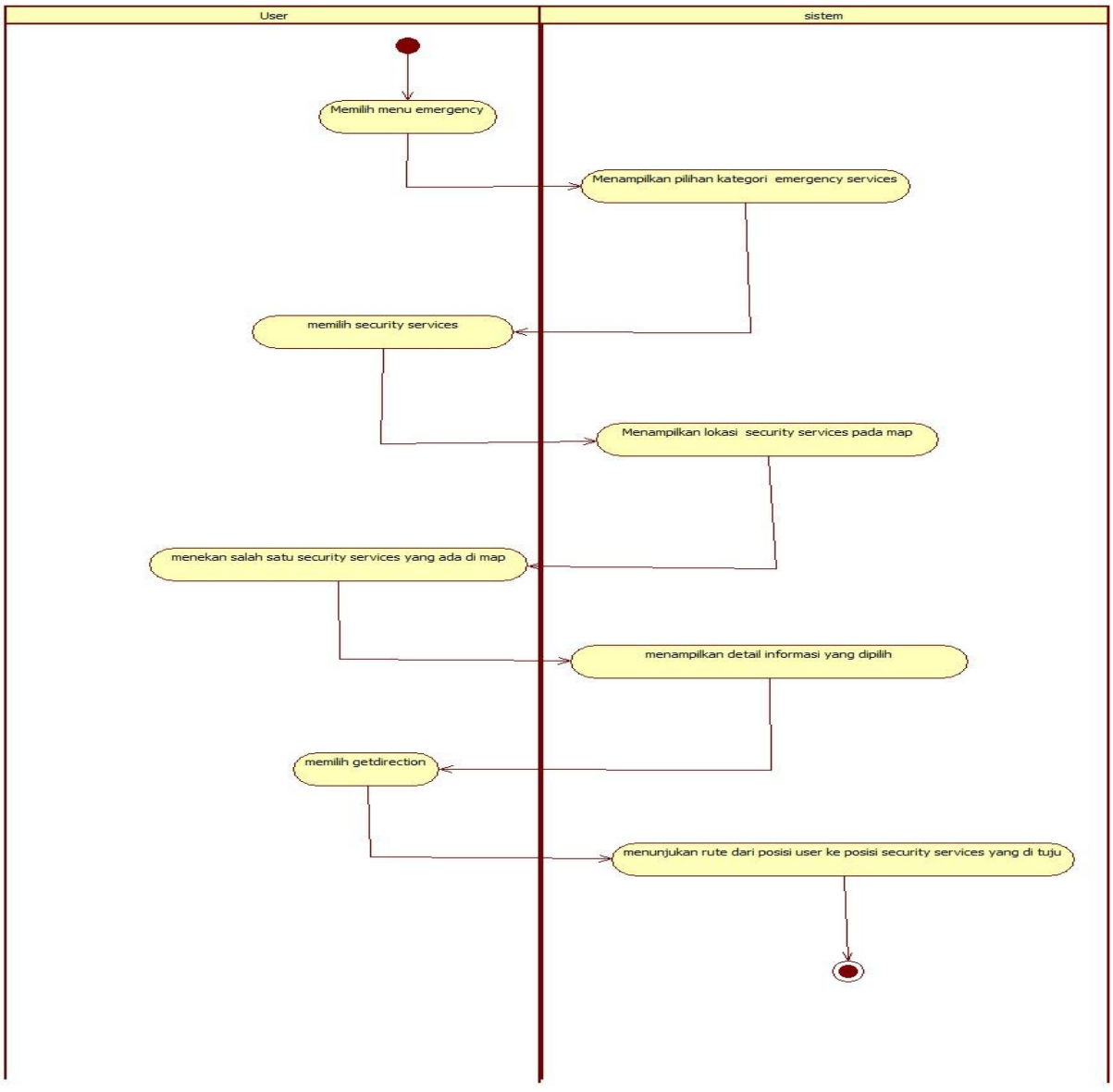
a. *Activity diagram emergency security services call*(*Frontend*)



Gambar 3. 4 *Activity Diagram emergency*

Pada *activity* ini *User* memilih menu *emergency* pada aplikasi, lalu aplikasi menampilkan pilihan kategori *emergency services* lalu user memilih *security services*, aplikasi menampilkan lokasi *security services* pada map dan user menekan pin lokasi,aplikasi menampilkan detail informasi lokasi *security services* lalu user menekan tombol *call*, aplikasi otomatis langsung menghubungi *security services* yang dipilih *user*

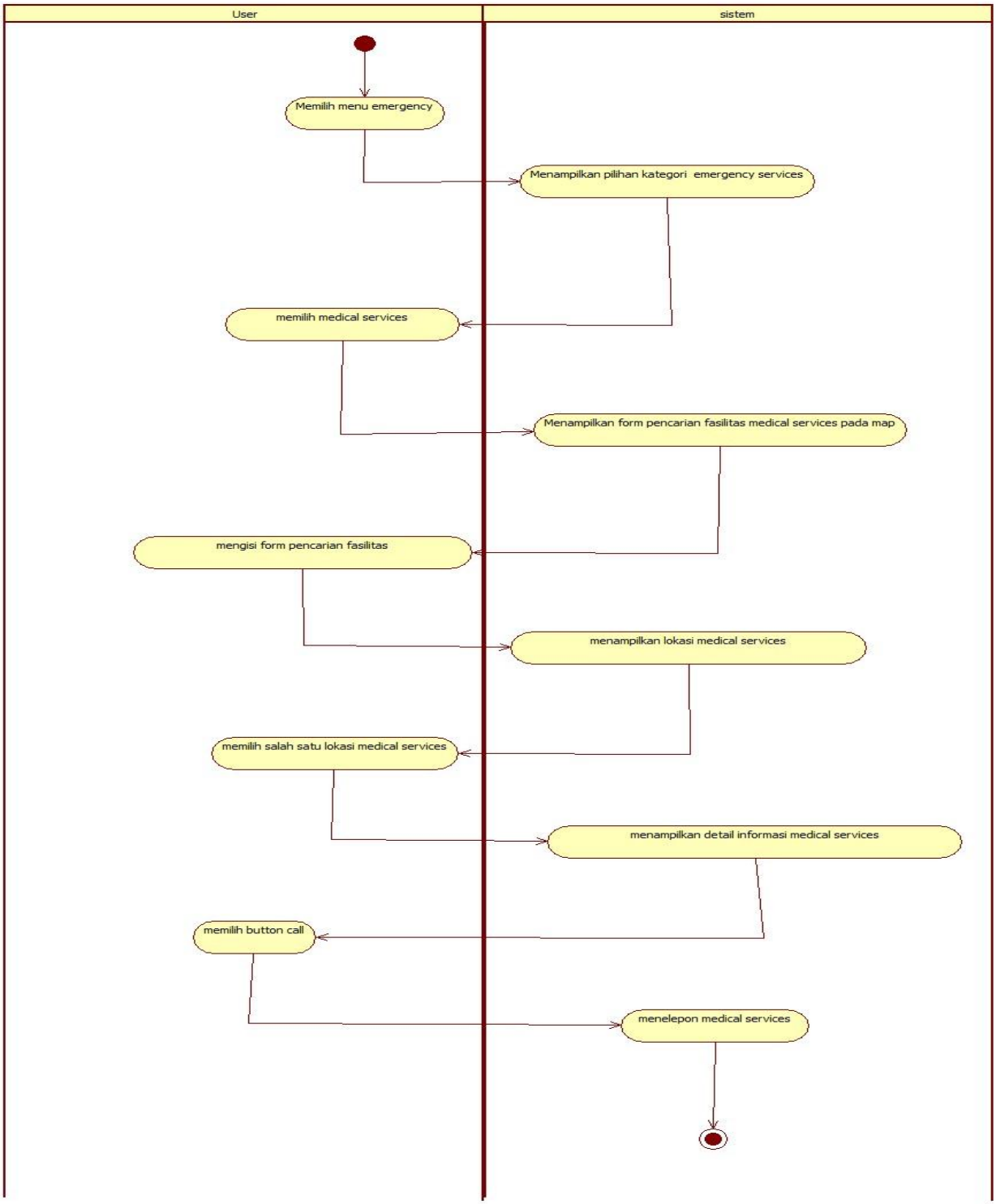
b. *Activity diagram emergency security services getdirection*



Gambar 3.5 *Activity diagram* darurat

Pada *activity* ini *User* memilih menu *emergency* pada aplikasi, lalu aplikasi menampilkan pilihan kategori *emergency services* lalu user memilih *security services,* aplikasi menampilkan lokasi *security services* pada map dan user menekan pin lokasi,aplikasi menampilkan detail informasi lokasi *security services* lalu user menekan tombol *get direction*, aplikasi otomatis langsung menentukan rute perjalanan pada gmap dari posisi user ke posisi *security services* yang dipilih *user*

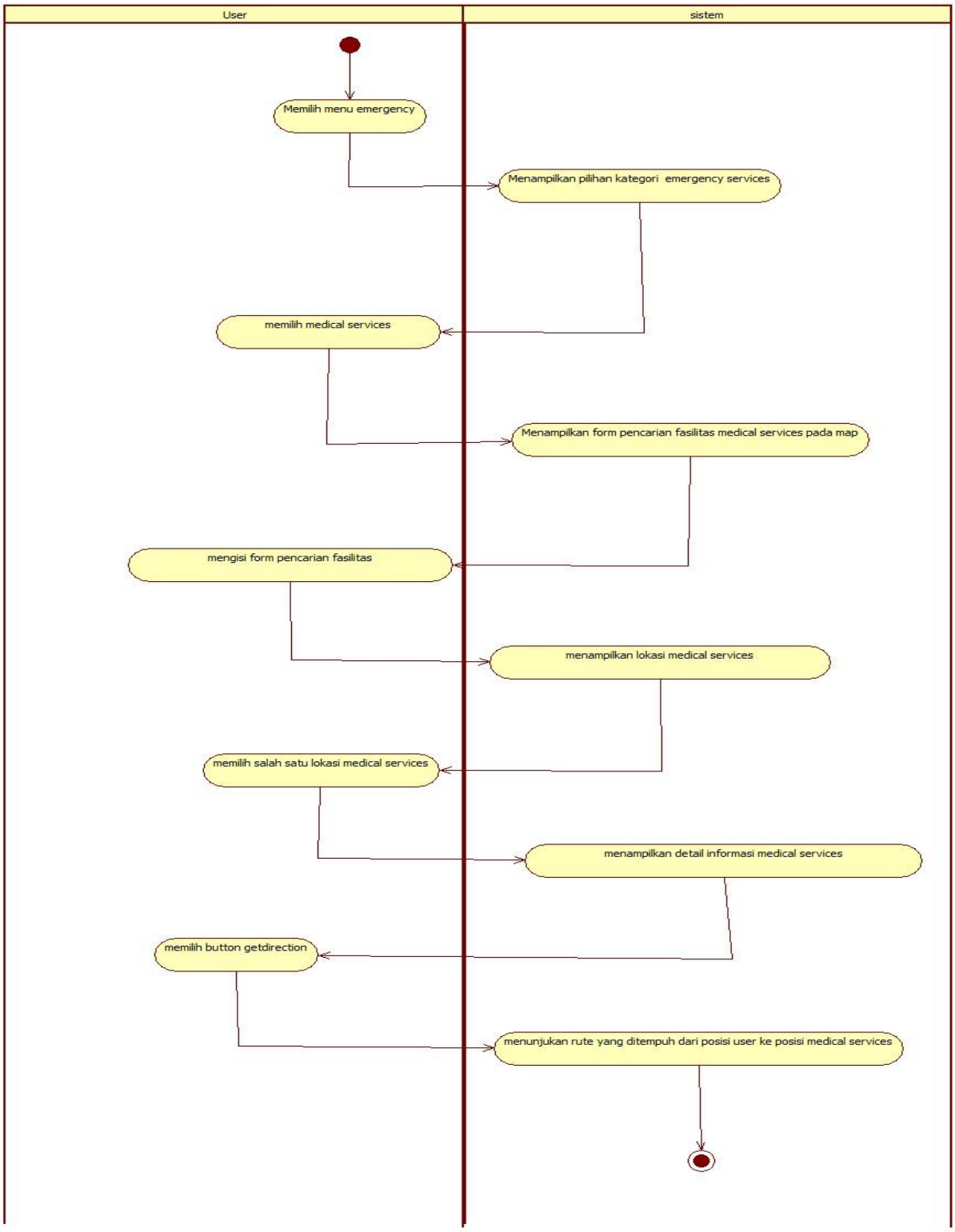
c. *Activity diagram emergency medical services call*



Gambar 3. 6 ***Activity diagram emergency medical services call***

Pada activity ini User memilih menu emergency pada aplikasi, lalu aplikasi menampilkan pilihan kategori emergency services lalu user memilih *medical services*, aplikasi menampilkan form pencarian fasilitas, kemudain usermemasukan fasilitas yang ingin di cari di sekitar keberadaan user dan tekan ok, kemudian aplikasi menampilkan lokasi medical services yang di cari pada gmap dan user menekan pin lokasi,aplikasi menampilkan detail informasi lokasi medical services lalu user menekan tombol call, aplikasi otomatis langsung menghubungi medical services yang dipilih user

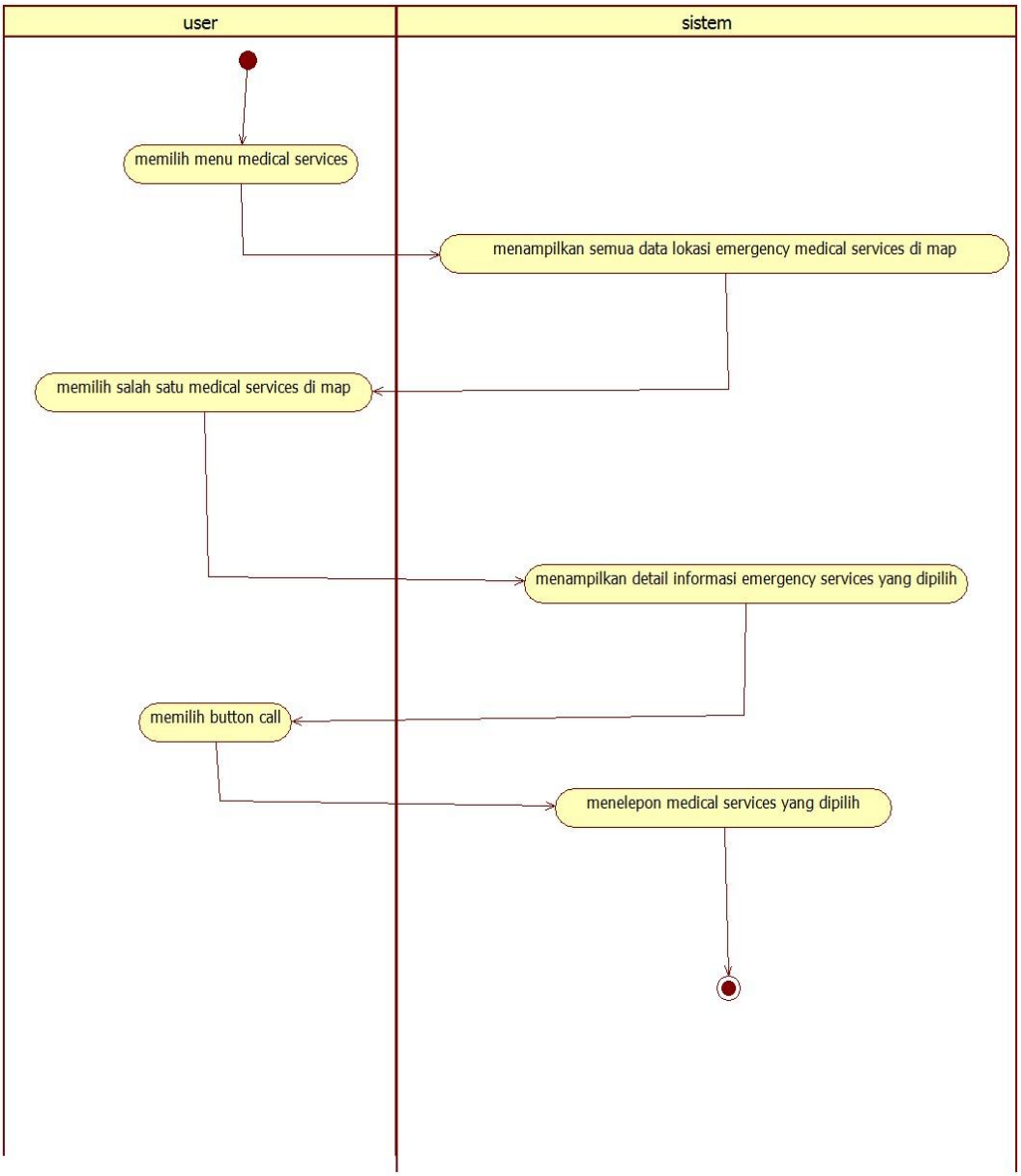
d. Activity diagram emergency *medical services getdirection*



Gambar 3. 7 ***Activity diagram emergency medical services getdirectionl***

Pada activity ini *user* memilih menu emergency pada aplikasi, lalu aplikasi menampilkan pilihan kategori *emergency services* lalu user memilih *medical services*, aplikasi menampilkan form pencarian fasilitas, kemudain *user* memasukan fasilitas yang ingin di cari di sekitar keberadaan user dan tekan ok, kemudian aplikasi menampilkan lokasi medical services yang di cari pada gmap dan user menekan pin lokasi,aplikasi menampilkan detail informasi lokasi *medical services* lalu user menekan tombol *getdirection*, aplikasi otomatis langsung menentukan rute perjalanan pada gmap dari posisi user ke posisi security services yang dipilih user

e. *Activity diagram medical services call*

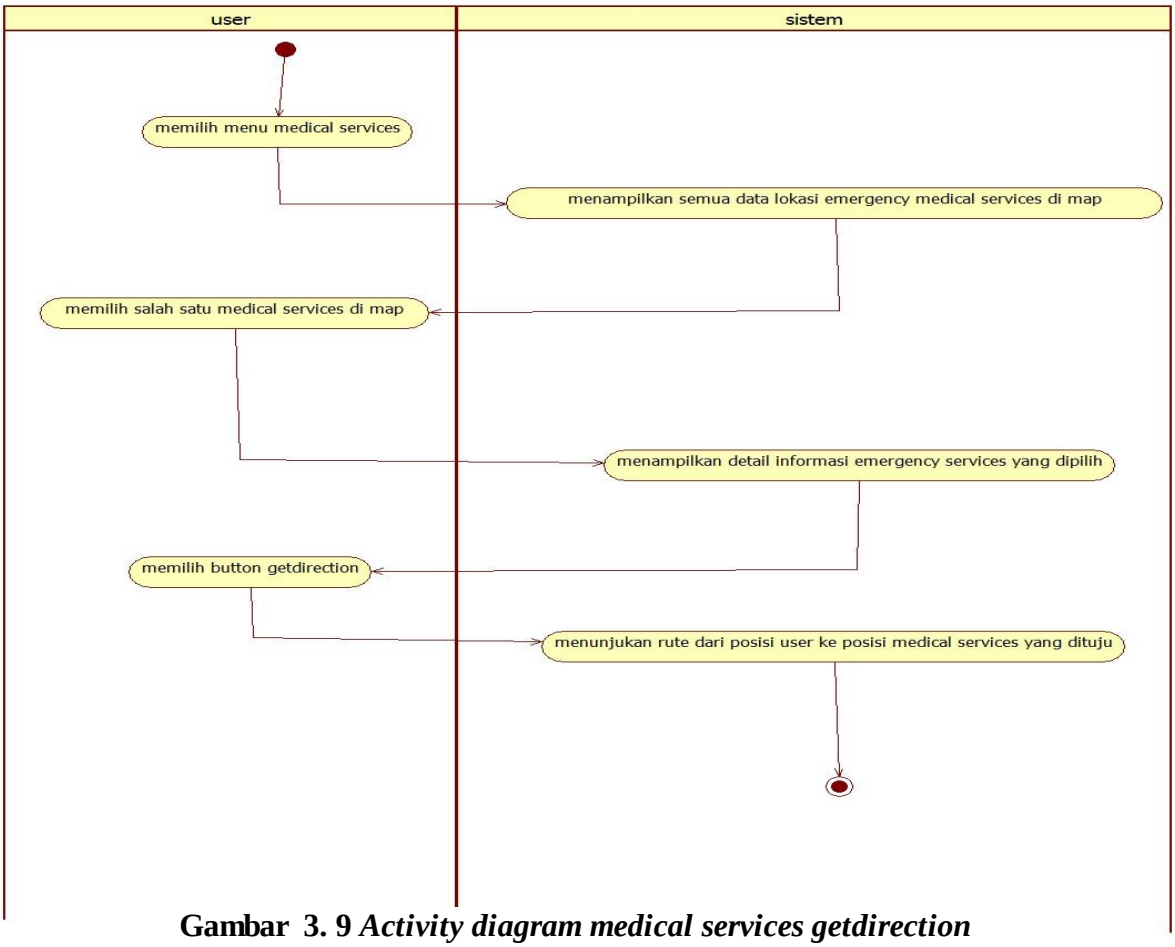


Gambar 3. 8 *Activity diagram medical services call*

Pada activity ini user memilih menu *medical services* pada aplikasi, lalu aplikasi menampilkan lokasi medical services di gmap kemudain user menekan pin lokasi, aplikasi menampilkan detail informasi lokasi *medical*

services lalu user menekan tombol *call*, aplikasi otomatis langsung menghubungi *medical services* yang dipilih user

f. *Activity diagram medical services getdirection*



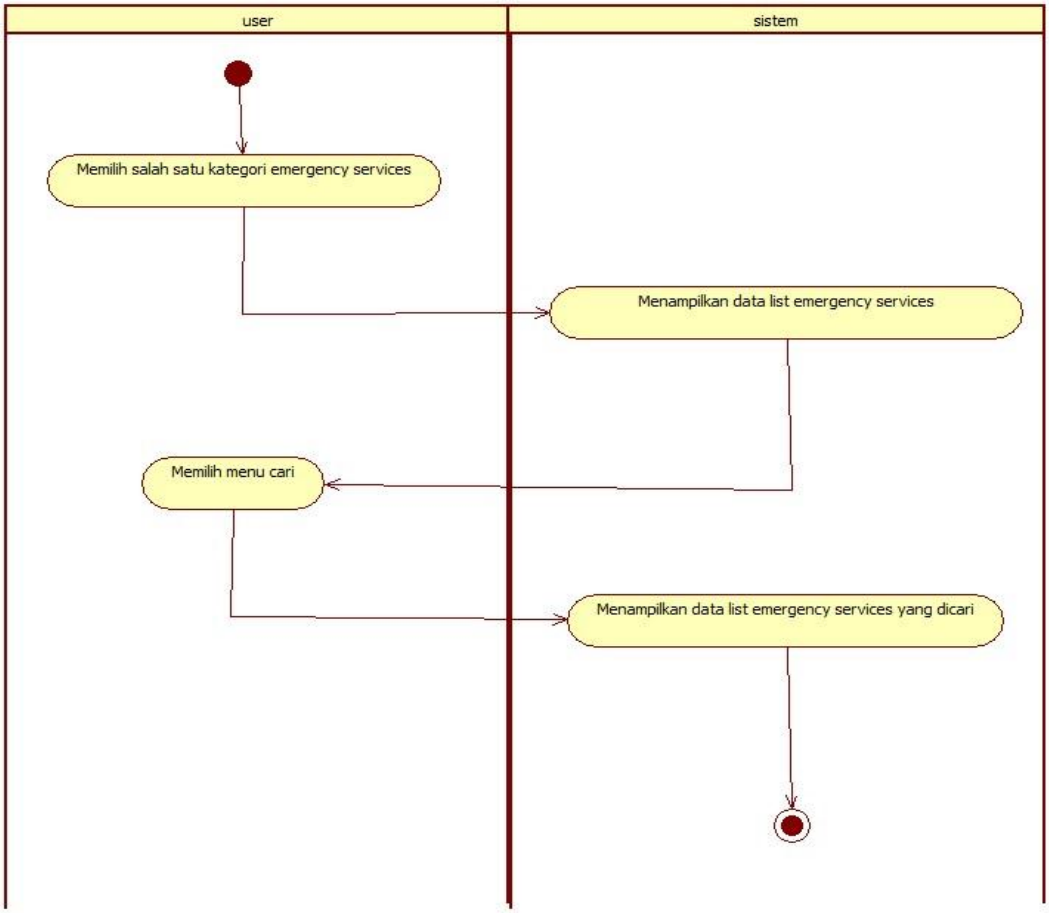
Gambar 3. 9 *Activity diagram medical services getdirection*

Pada activity ini user memilih menu *medical services* pada aplikasi, lalu aplikasi menampilkan lokasi *medical services* di gmap kemudain user menekan pin lokasi, aplikasi menampilkan detail informasi lokasi *medical services* lalu user menekan tombol *getdirection*, aplikasi otomatis langsung menujukan rute perjalanan pada gmap dari lokasi *user* ke lokasi *medical services* yang dituju

Gambar 3. 10 Activity diagram detail emergency services

Pada *activity* ini *user* memilih menu data list *emergency services* , yang mana aplikasi akan menampilkan daftar kategori *emergency services* dan detail dari kategori tersebut, dan *user* dapat memilih menu lainya seperti menu call untuk menghubungi *emergency services* yang dipilih, menu *get direction* untuk memberikan petunjuk rute dari posisi *user* ke posisi *emergency services* yang di tuju, menu sara untuk memberikan saran kepada *user* untuk pemilihan *emergency services* yang tepat sesuai keriteria

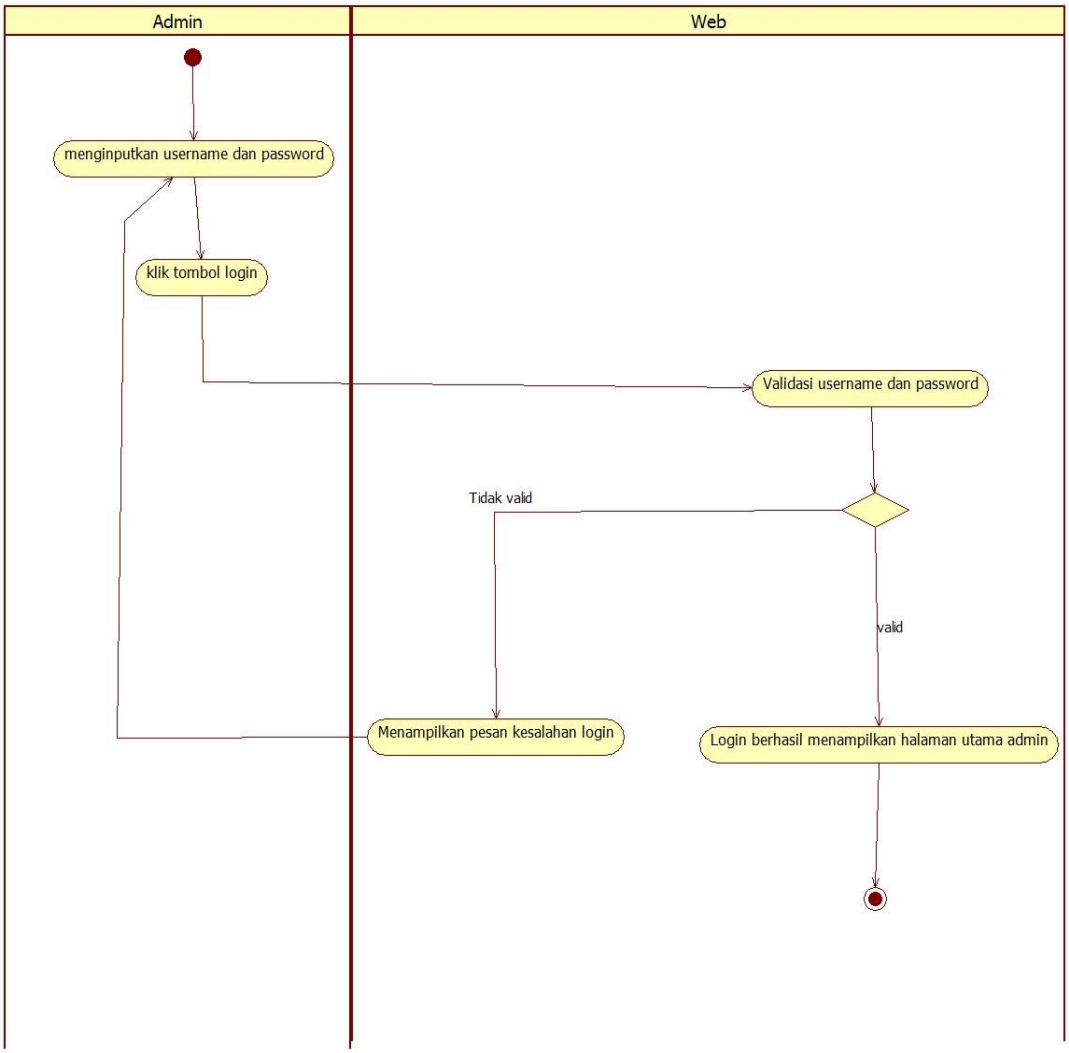
g. *Activity* diagram data list *emergency services* (*frontend*)



Gambar 3. 11 Activity diagram data list

Pada *activity* ini *user* memilih menu data list *emergency services*,kemudian memilih salah satu kategori,dan aplikasi menampilkan data list *emergency services*, *user* untuk mencari daftar nama *emergency services* dapat melakukan dengan memilih menu cari dan aplikasi akan menampilkan data *emergency services* yang di cari

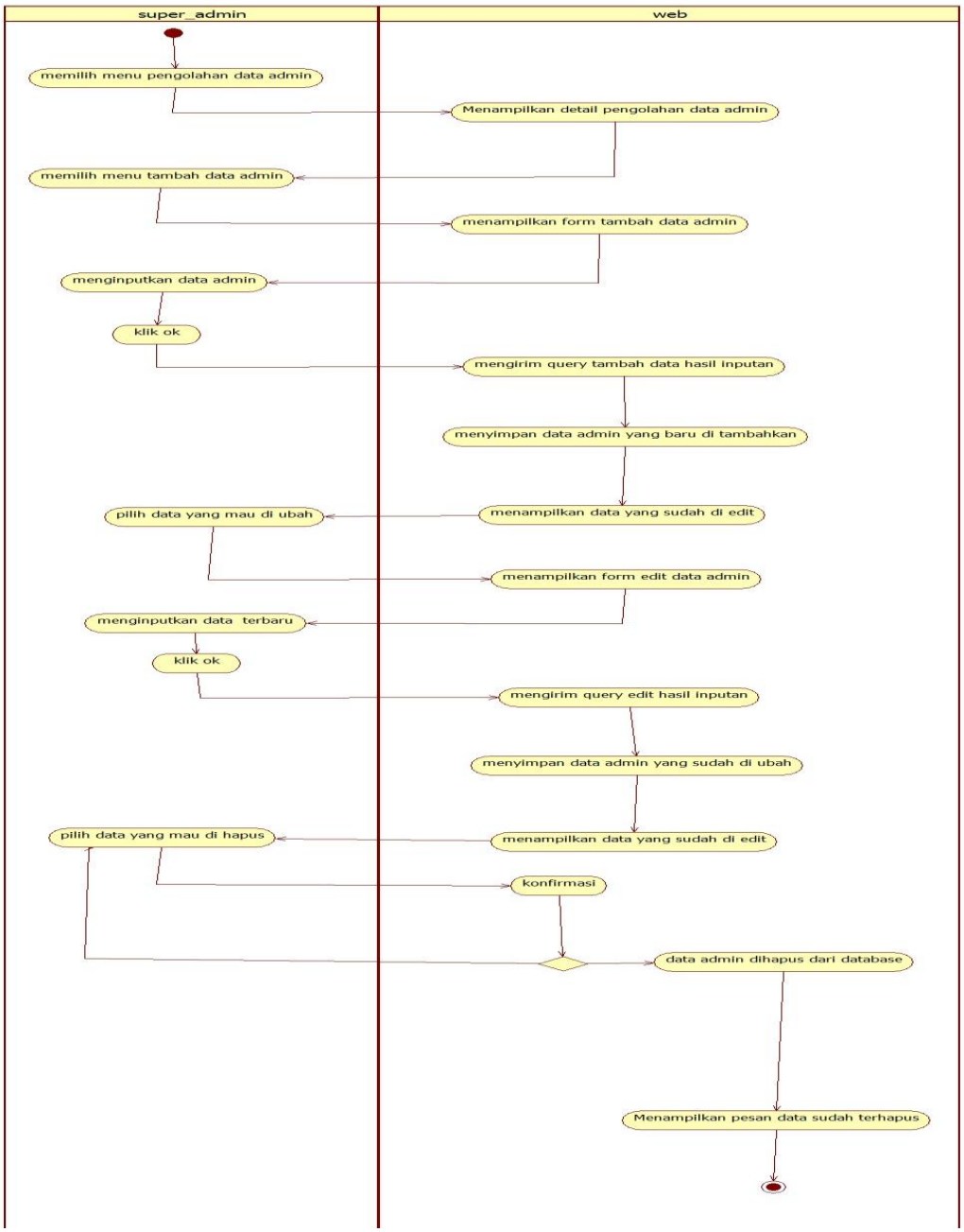
h. *Activity diagram* log in admin (*backend*)



Gambar 3. 12 Activity diagram log in

Pada *activity* ini *user* melakukan login dan memasukan *user*name dan password, kemudian akan di cek validasi oleh sistem, jika berhasil login maka *user* masuk ke menu utama, dan jika tidak *user* memasukan ulang *user*name dan passwordnya

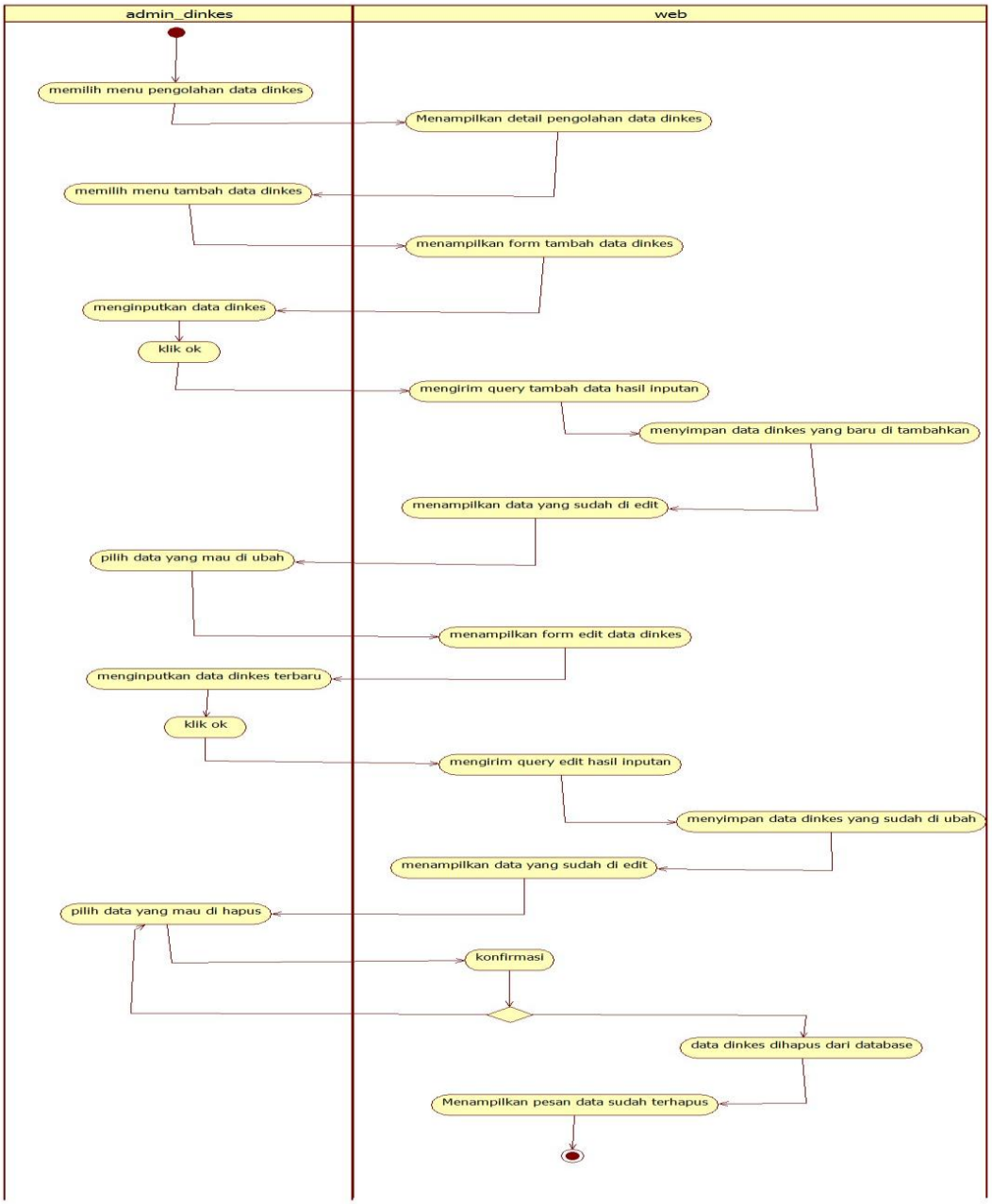
i. *Activity diagram* Pengolahan data admin (*Backend*)



Gambar 3. 13 Activity pengolahan data admin

Pada *activity* diatas *user* / *super* admin memilih menu pengolahan data admin, yang kemudian sitem menampilkan tampilan pengolahan admin, *user* dapat melakukan pengolahan admin seperti tambah data admin, ubah data admin, dan hapus data admin.

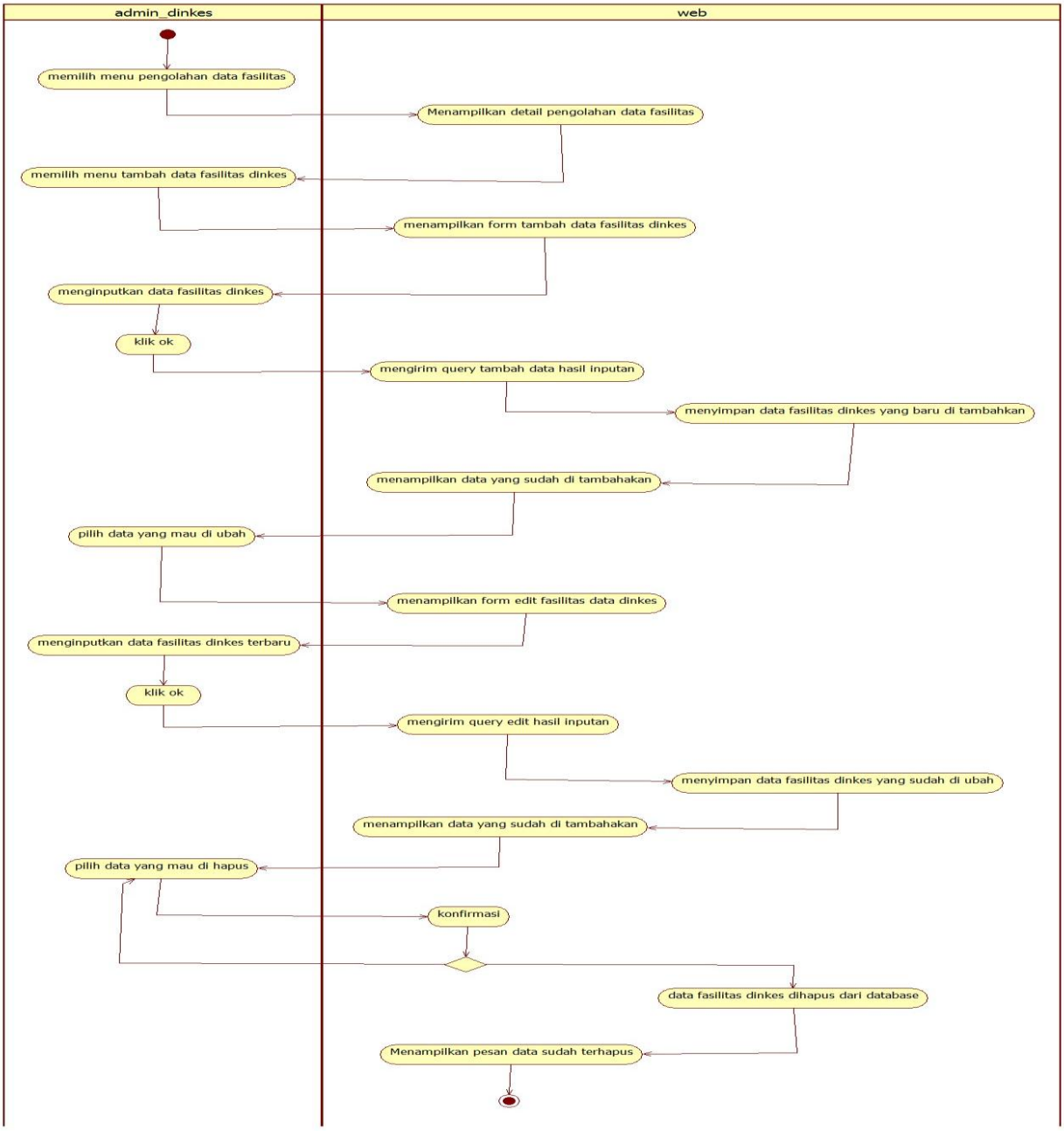
j. *Activity diagram* pengolahan data *medical services*



Gambar 3. 14 Activity diagram pengolahan data *medical services*

Pada *activity* diatas *user* / admin dinkes memilih menu pengolahan data dinkes, yang kemudian sitem menampilkan tampilan pengolahan dinkes, *user* dapat melakukan pengolahan dinkes seperti tambah data dinkes, ubah data dinkes, dan hapus data dinkes

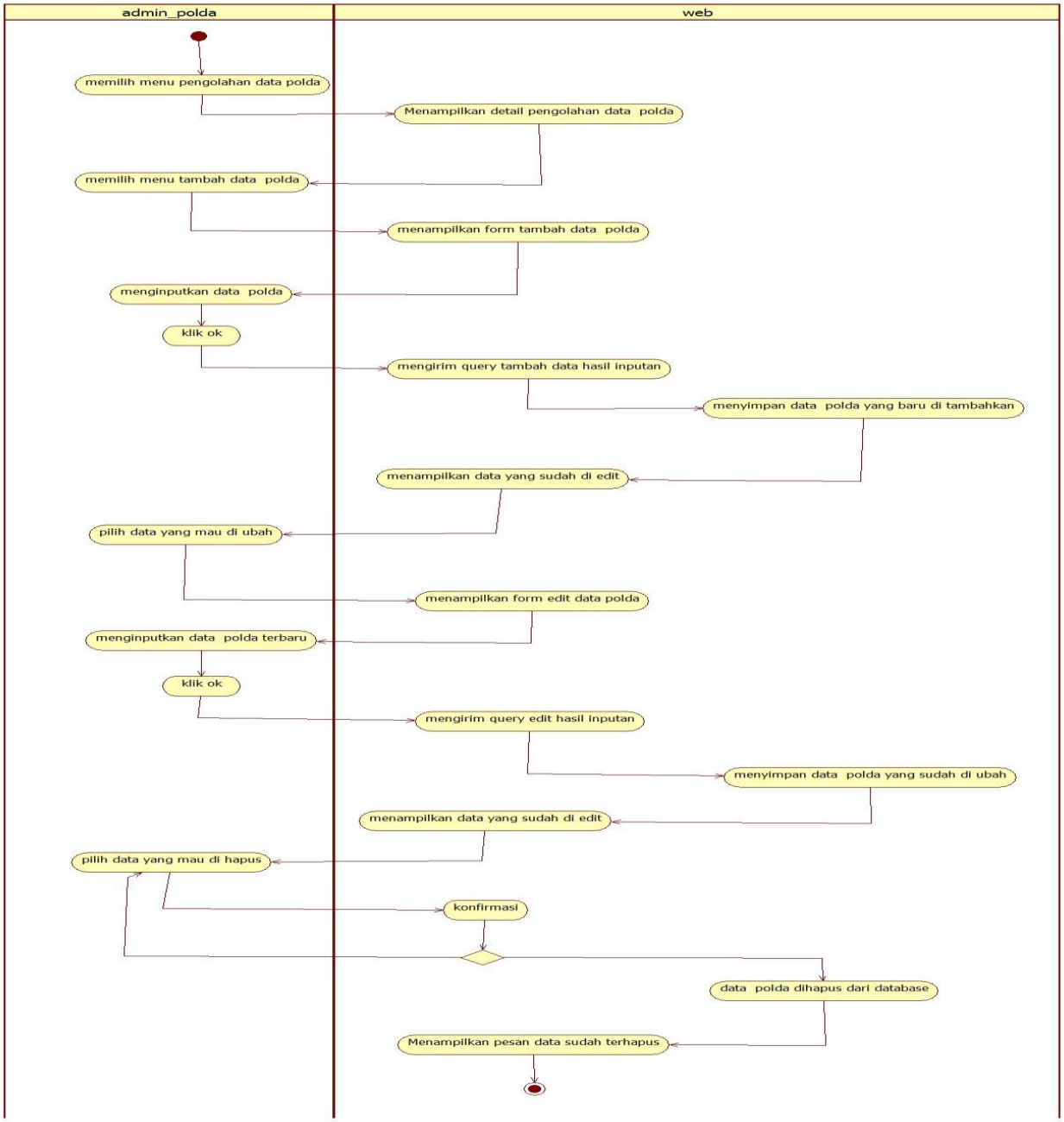
k. *Activity diagram* pengolahan data fasilitas (*Backend*)



Gambar 3. 15 Activity diagram pengolahan fasilitas

Pada *activity* diatas *user* / admin dinkes memilih menu pengolahan data fasilitas, yang kemudian sitem menampilkan tampilan pengolahan fasilitas, *user* dapat melakukan pengolahan fasilitas seperti tambah data fasilitas , ubah data fasilitas , dan hapus data fasilitas.

l. *Activity diagram* pengolahan data s*ecurity services*



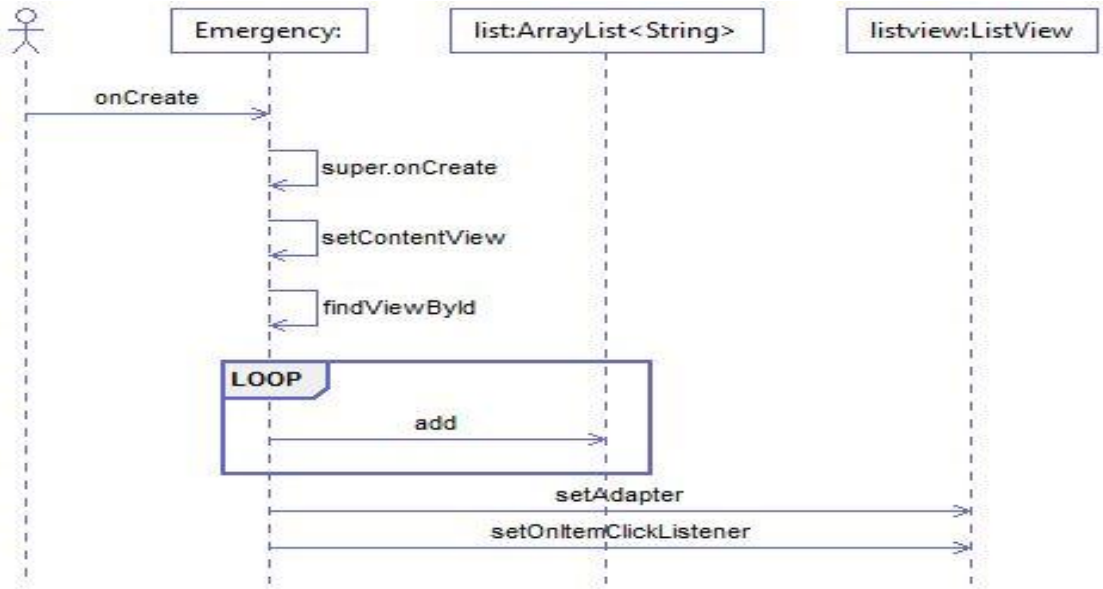
Gambar 3. 16 Activity diagram pengolahan data *security services*

Pada *activity* diatas *user* / admin polda memilih menu pengolahan data polda, yang kemudian sitem menampilkan tampilan pengolahan polda, *user* dapat melakukan pengolahan data polda seperti tambah data polda , ubah data polda , dan hapus data polda

c. Squence diagram

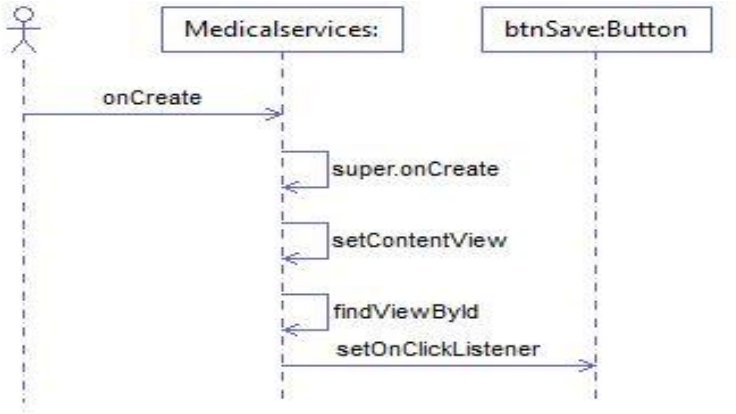
Pada tahap ini *Sequence Diagram* menjelaskan secara detail urutan proses yang dilakukan sistem untuk mencapai tujuan dari *Use Case*. interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.

a. ***Squence diagram emergency*(*frontend*)**



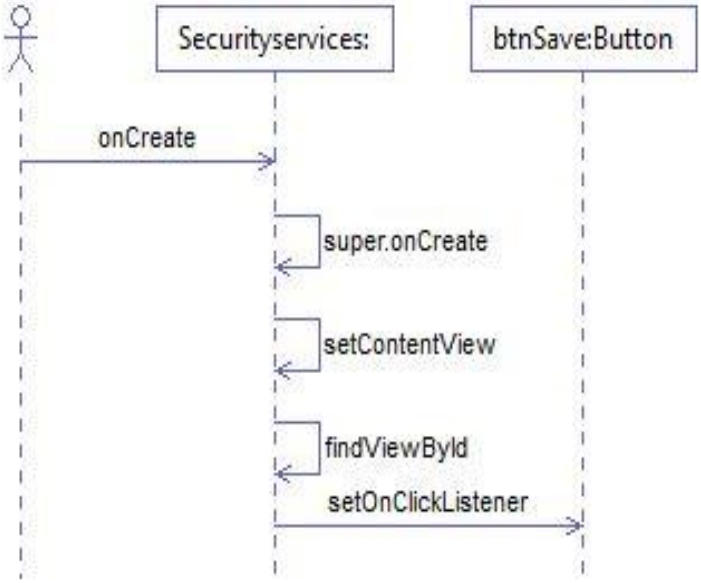
Gambar 3. 17 ***Squence diagram emergency***

b. *Squence diagram medical services*(*frontend*)



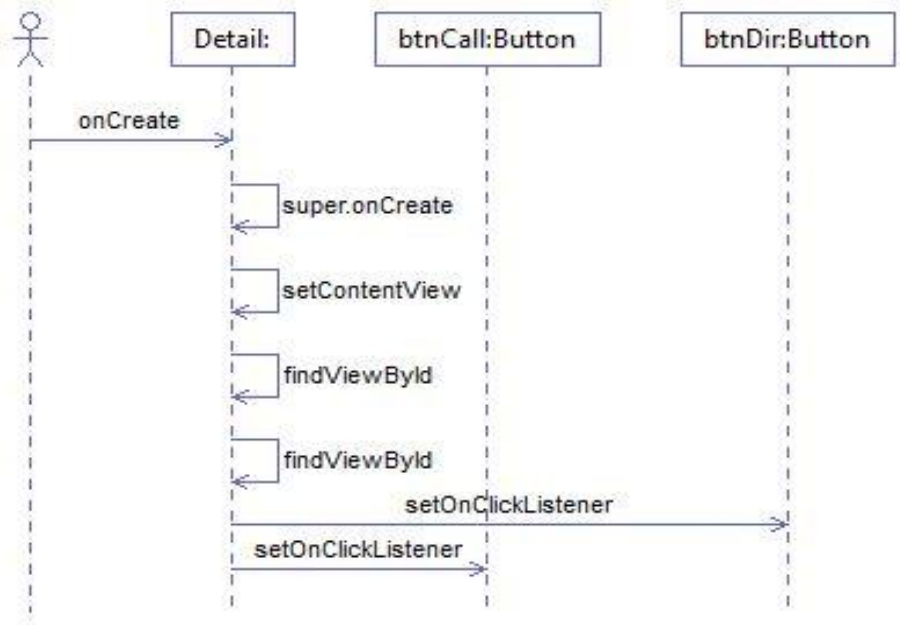
Gambar 3. 18 *squence diagram medical services*

c. *squence diagram security services*



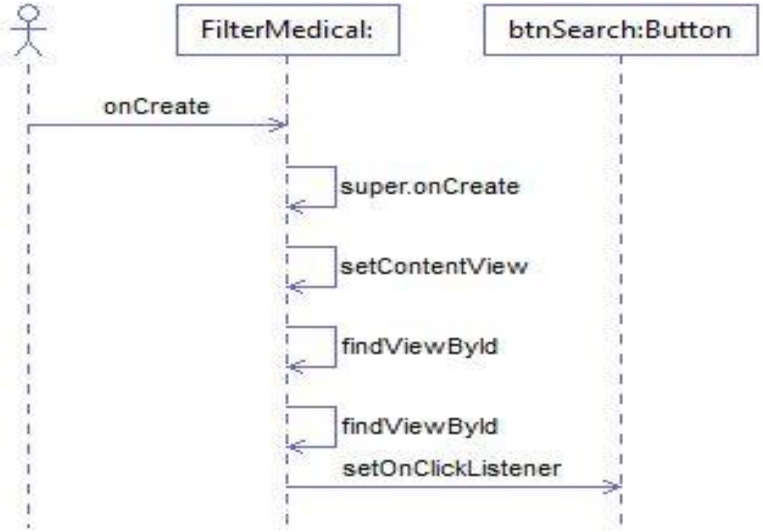
Gambar 3. 19 *squence diagram security services*

d. *Squence diagram* menu detail(*frontend*)



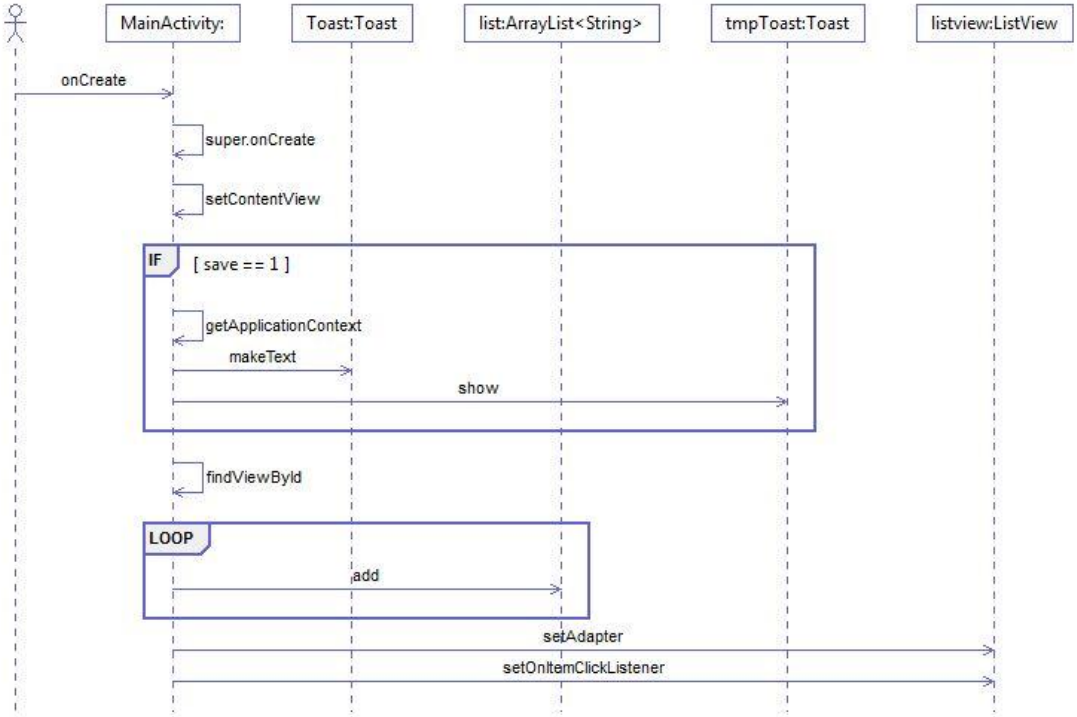
Gambar 3. 20 *Squence diagram* menu detail

***e. Squence diagram filter medical*(*frontend*)**



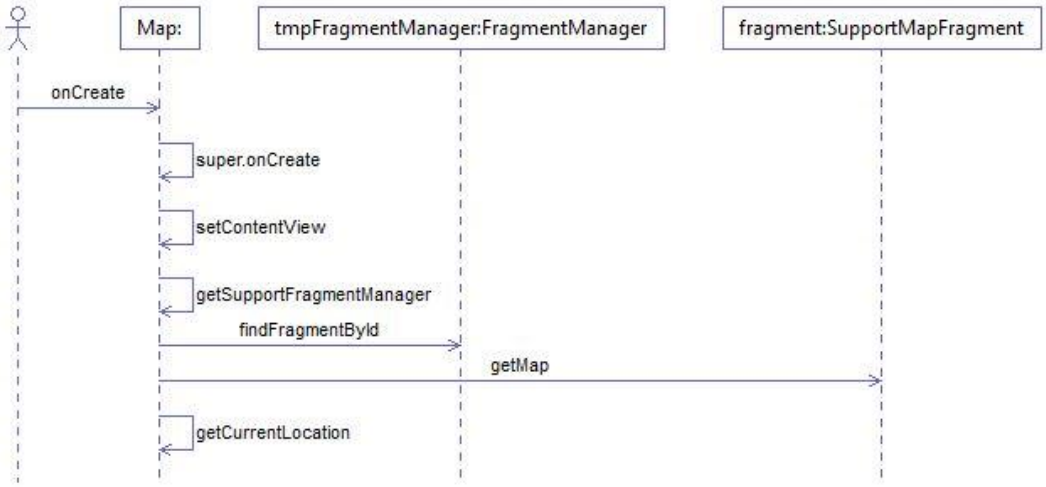
Gambar 3. 21 *squence diagram filter medical*

f. Squence diagram main(frontend)



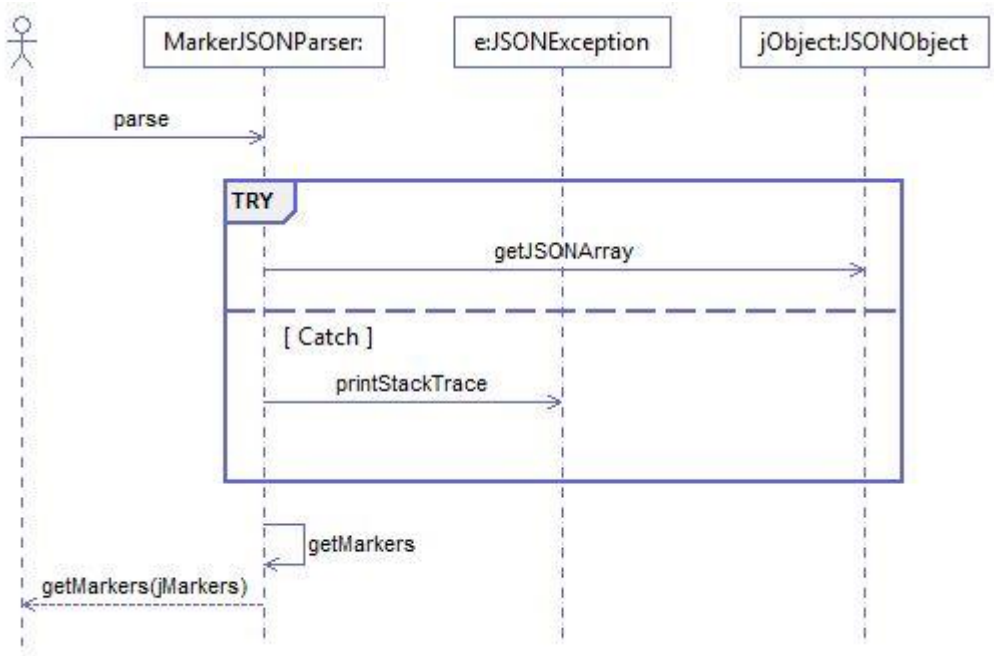
Gambar 3. 22 ***Squence diagram main***

g. Squence diagram map (frontend)



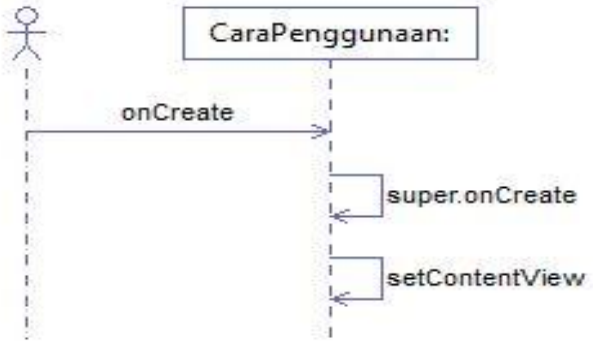
Gambar 3. 23 ***squence diagram map***

h. Squence diagram marker(frontend)



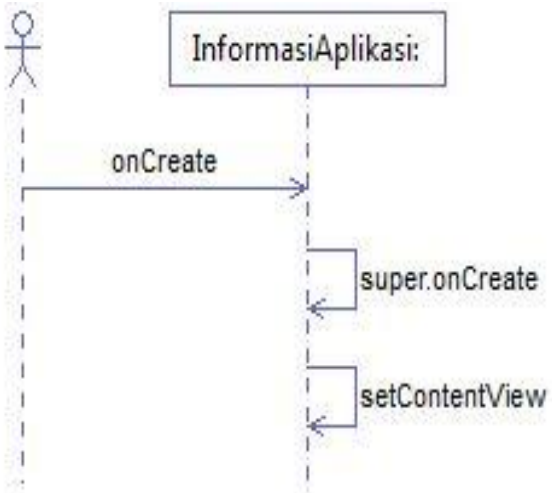
Gambar 3. 24 *squence diagram marker*

i. Squence diagram cara penggunaan aplikasi



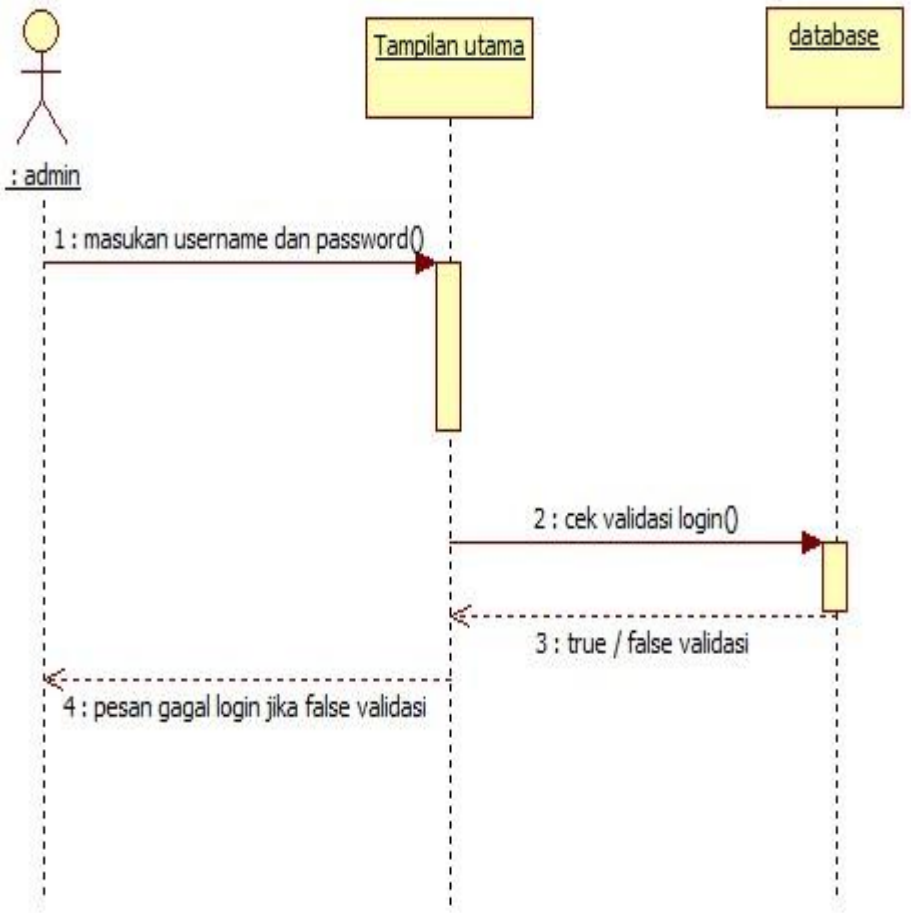
Gambar 3. 25 *squence diagram* cara penggunaan aplikasi

j. *Squence diagram* informasi aplikasi



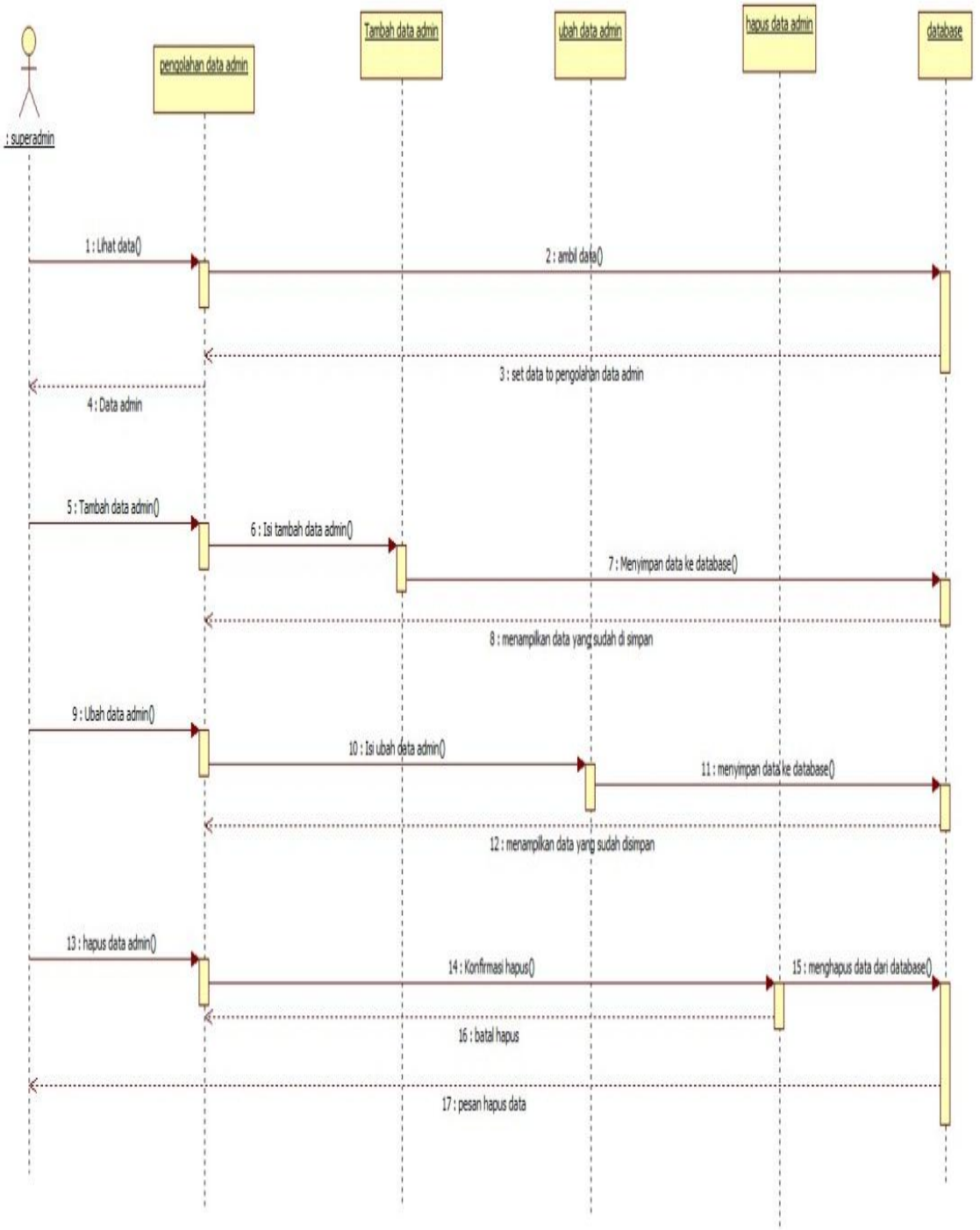
Gambar 3. 26 squence diagram informasi aplikasi

k. *Squence diagram login* admin (*Backend*)



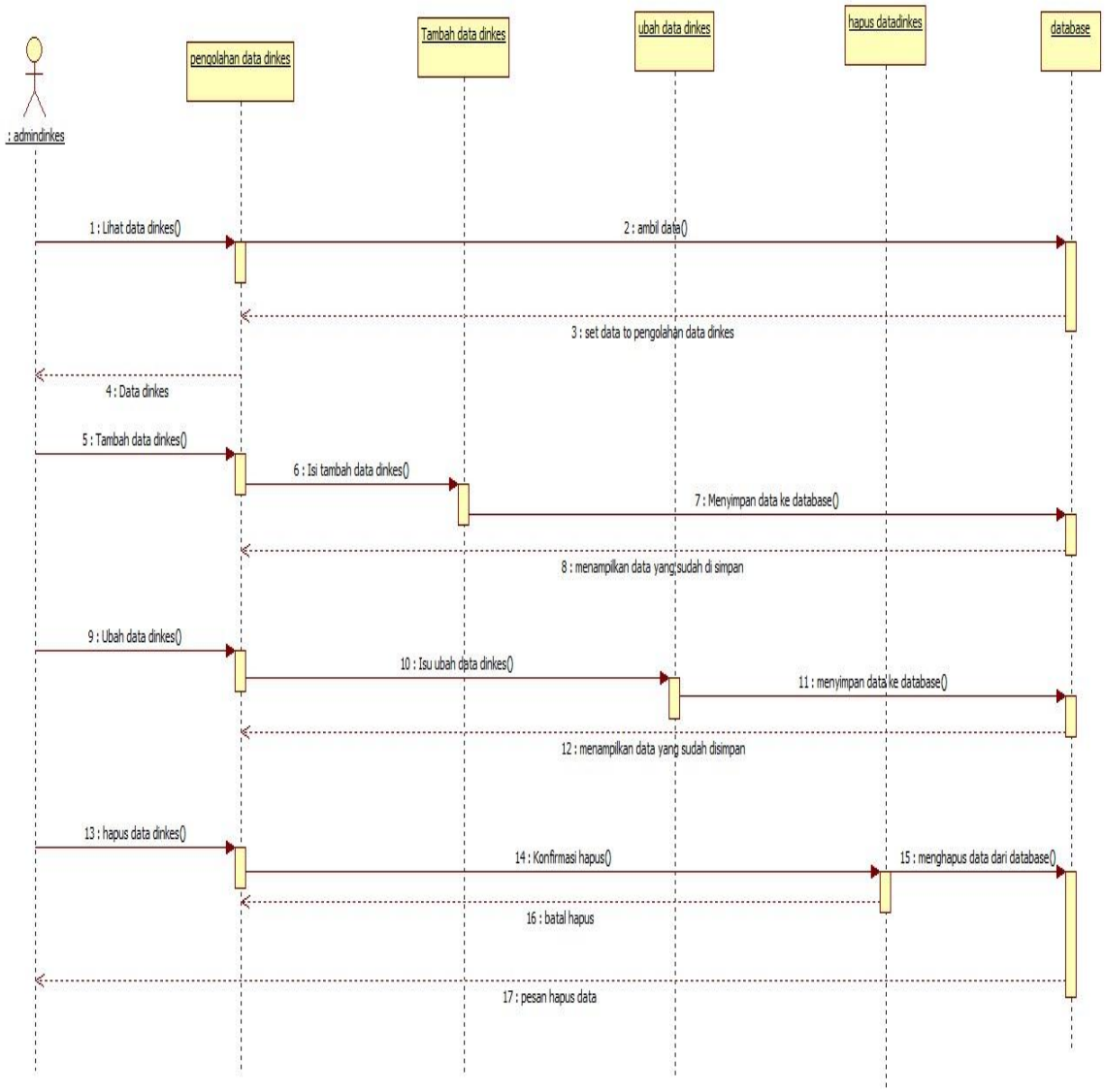
Gambar 3. 27 *Squence diagram* login admin

l. *Squence diagram* pengolahan data admin (*Backend*)



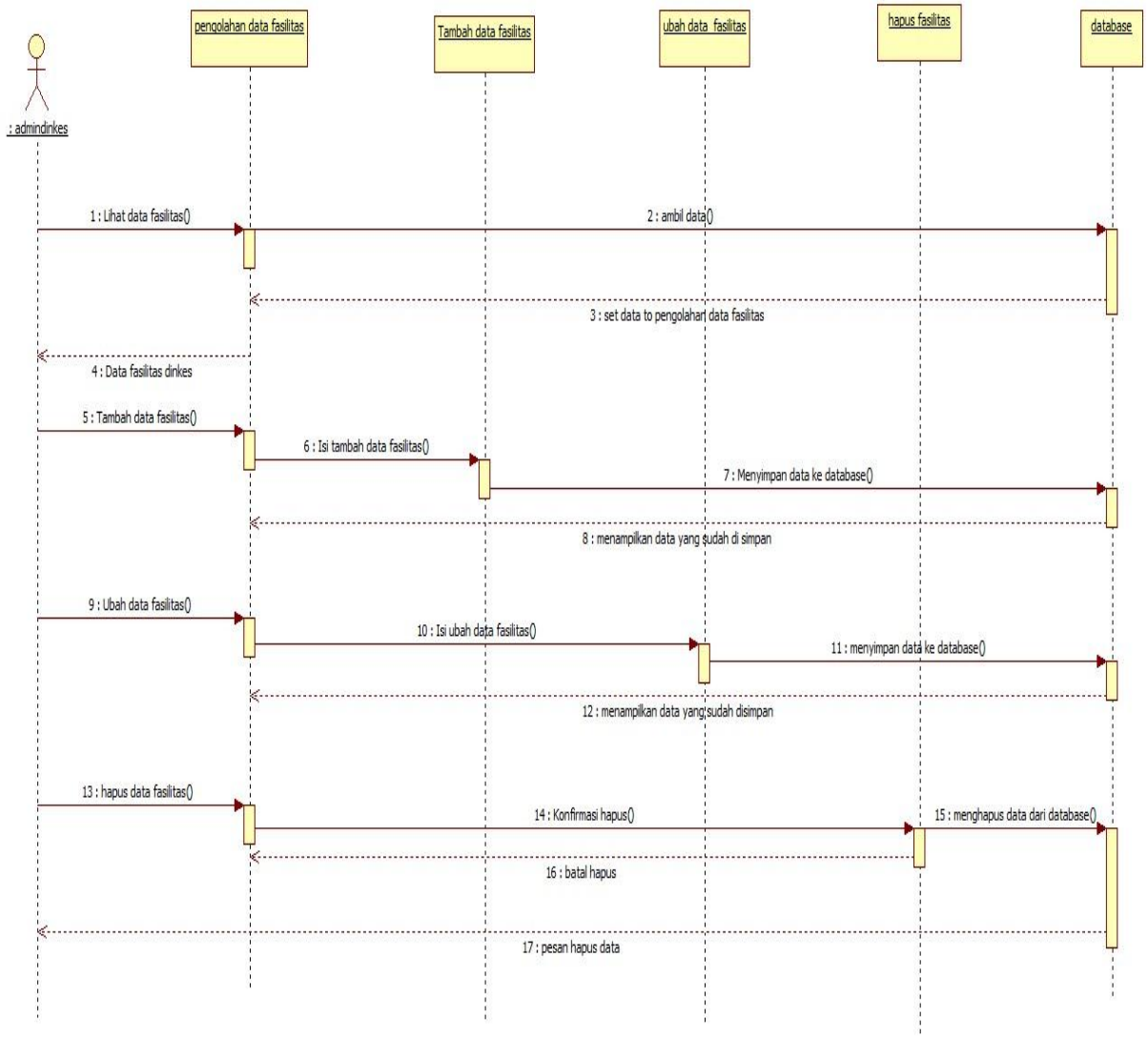
Gambar 3. 28 *Squence diagram* pengolahan data admin

m. *Squence diagram* pengolahan data *medical services* (*Backend*)



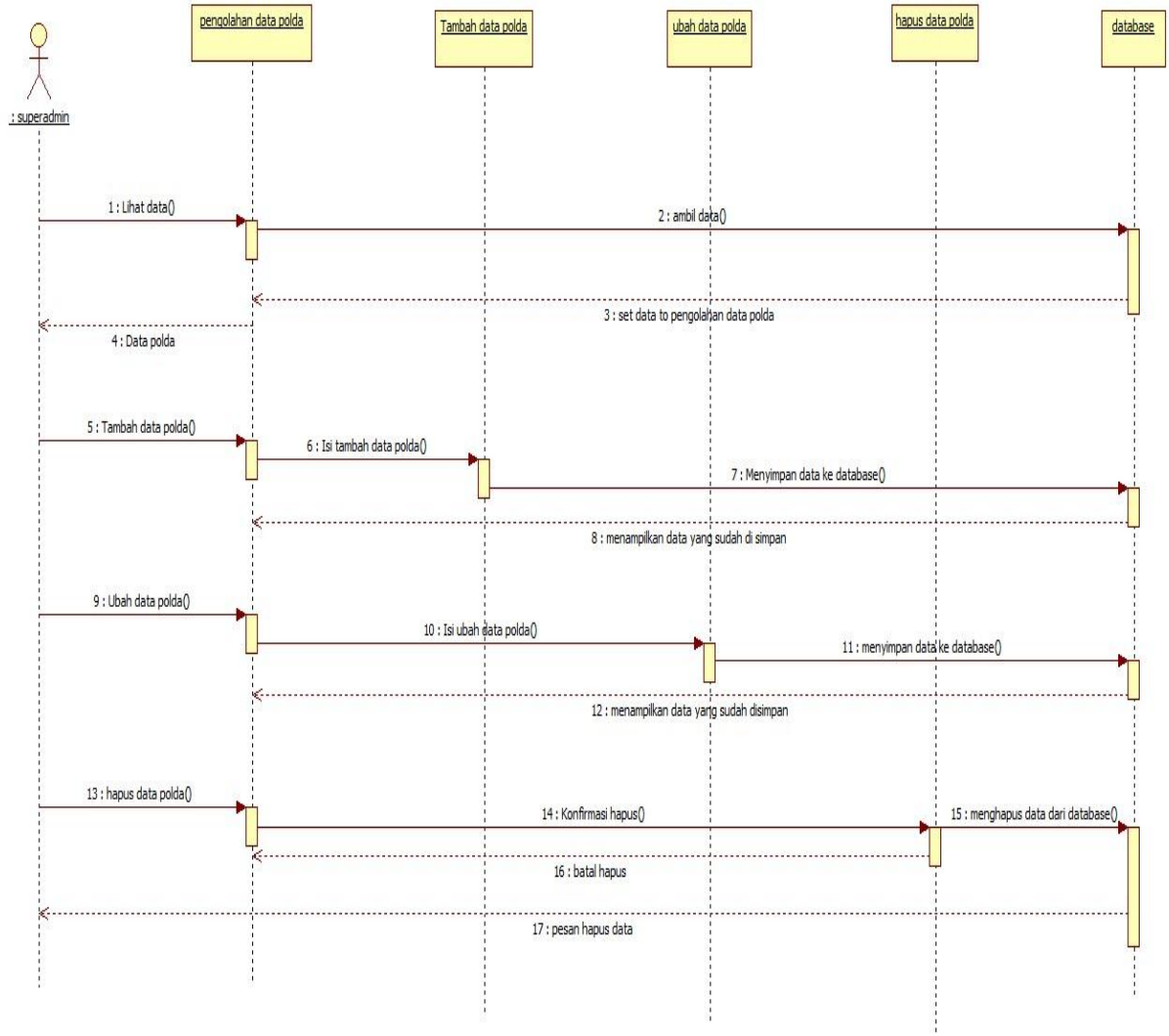
Gambar 3. 29 *Squence diagram* pengolahan data *medical services*

n. *Squence diagram* pengolahan data fasilitas (*Backend*)



Gambar 3. 30 *squence diagram* pengolahan data dinkes

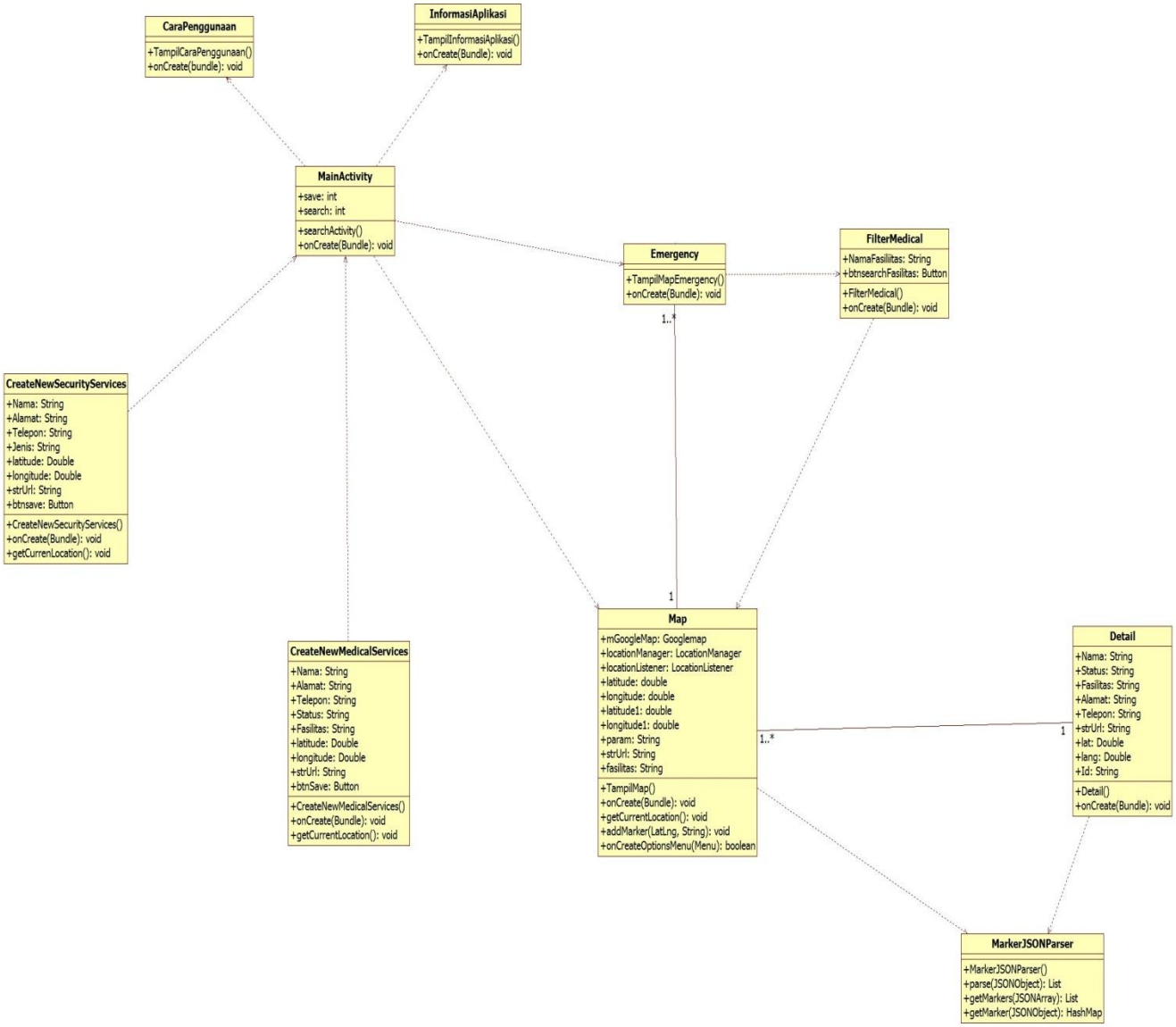
o. Squence diagram pengolahan data *security services*



Gambar 3. 31. *Squence diagram* pengolahan data *security services*

c. Classdiagram

class Diagram adalah sebuah spesifikasi dari fungsionalitas yang menghasilkan objek dan merupakan inti dari pengembangan aplikasi ini. Diagram ini juga merupakan gambaran keadaan atribut atau properti dari sistem yangelakukan manipulasi fungsi atau metode.

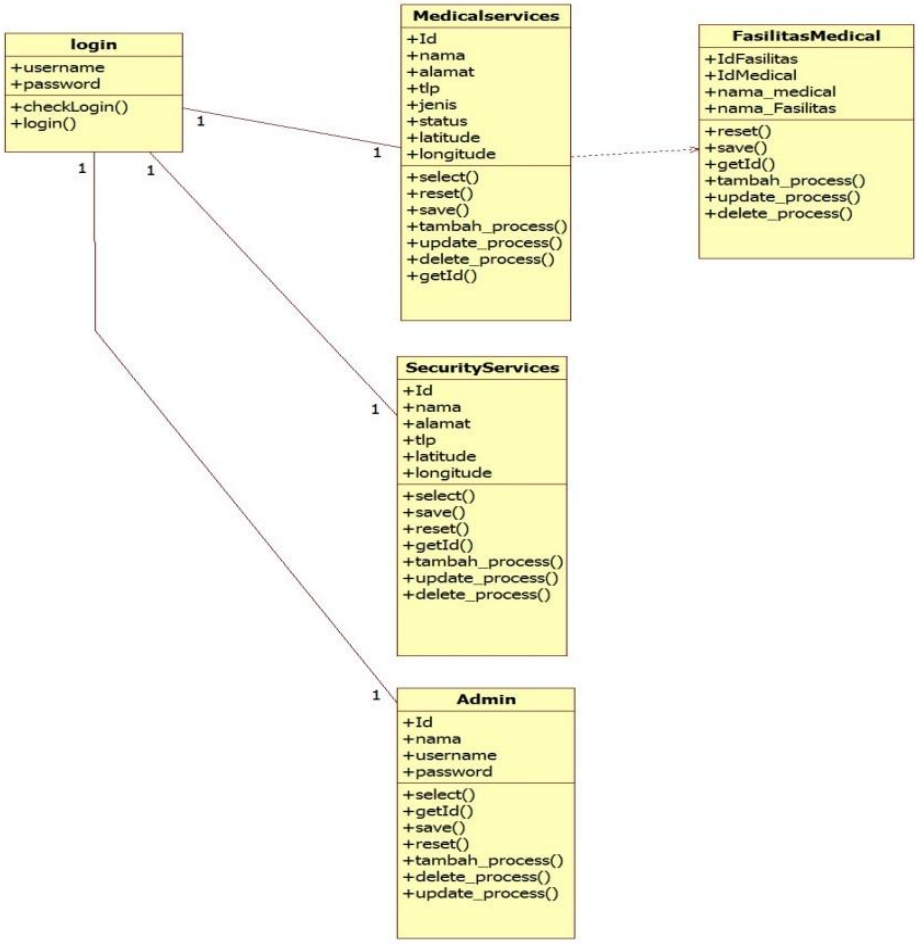


Gambar 3. 32 *Class diagram* aplikasi *emergency services*

Tabel 3. 23 Deskripsi *Class Diagram mobile*

Class	Jenis class	Deskripsi
Main	Control	Class utama yang berisikan method-method yang berhubungan dengan class lain dan berfungsi sebagai controller
Map	Control	Class yang berisikan method-method untuk menampilkan lokasi *user* dan *emergency services* serta detail *emergency services*
emergency	Control	Class yang berisikan method-method untuk menampilkan kategori *emergency services*
Marker	Control	Class berisikan method-method untuk memparsing data berformat JSON
Filter medical	Control	Class yang berisikan method-method untuk menampilkan *medical services* sesuai keriteria
detail	Control	Class yang berisikan method-method untuk menampilkan rincian data *emergency services*
Create new security services	Control	Class yang berisikan method tambah data *security services*
Create new medical services	Control	Class yang berisikan method tambah data *medical services*
Help	View	Class yang menampilkan cara penggunaan aplikasi
Security services	Control	Class yang berisikan method untuk menampilkan security services
Informasi aplikasi	View	Clas yang menampilkan informasi pembuatan aplikasi

d. ***Class diagram* aplikasi *web server (Backend*)**



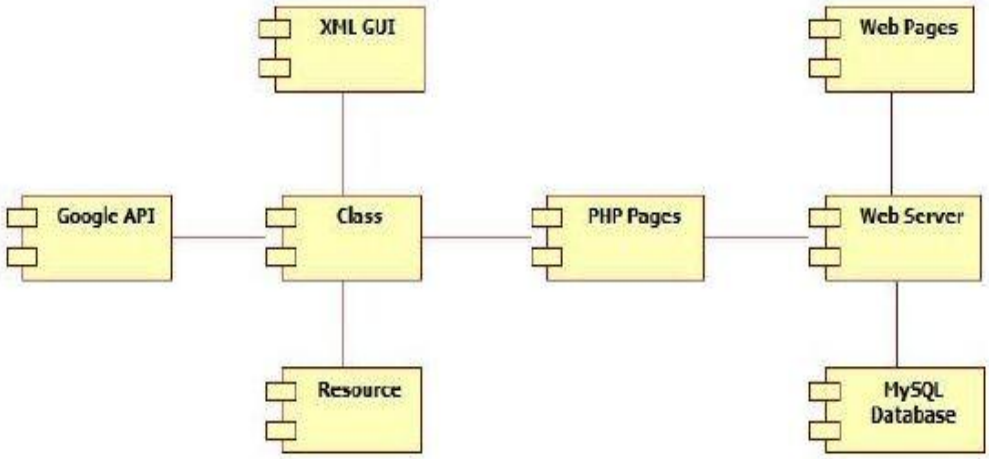
Gambar 3. 33 *Class diagram web server*

Tabel 3. 24 Deskripsi *Class Diagram web server*

Class	Jenis class	Deskripsi
Login	Control	Class yang berisikan method-method untuk login
Admin	Control	Class untuk mengontrol data admin
Fasilitas Medical	Control	Class untuk mengontrol data *Fasilitas Medical*
Security services	Control	Class untuk mengontrol data *Security services*
Medical services	Control	Class untuk mengontrol data *Medical services*

e. ***Component Diagram***

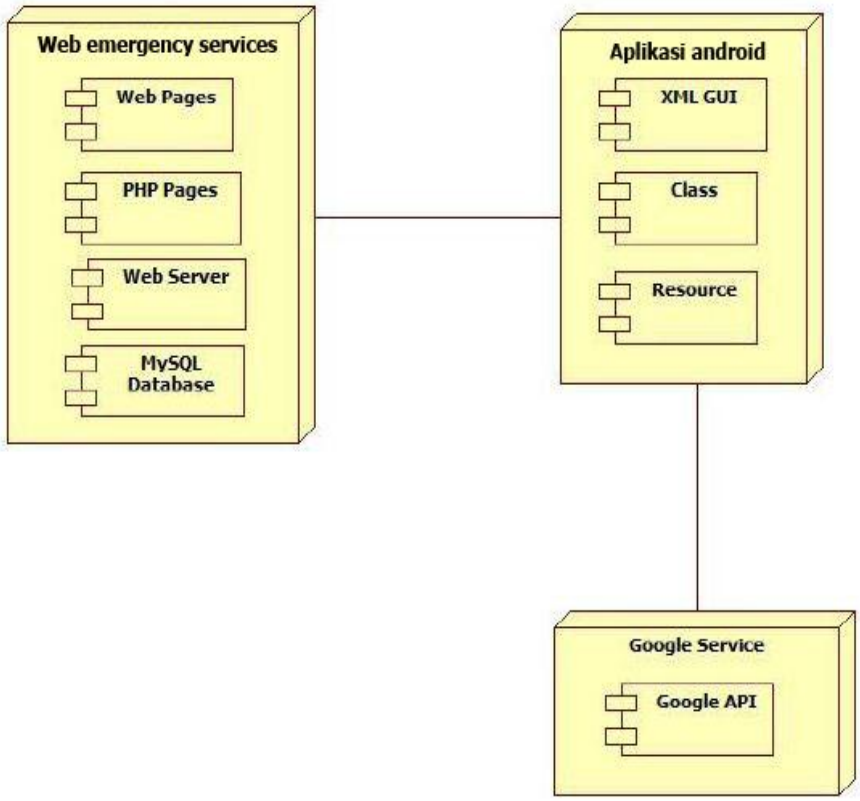
Pada tahap ini, Component Diagram menggambarkan struktur dan hubungan antar komponen piranti lunak termasuk ketergantungan (*dependency*). Komponen piranti lunak yang yang dimaksud adalah modul yang berisi *source code* atau binary code, yang ada library atau excutetable yang muncul pada *compile time*, *link time* ataupun pada r*untime*. *Component Diagram* ditunjukan pada gambar dibawah ini



Gambar 3. 34 *Component* Diagram

f. ***Deployment Diagram***

Diagram Deployment menggambarkan rinci bagaimana komponen dideploy dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin Server atau perangkat keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server* dan hal-hal yang lain bersifat fisikal. Gambaran arsitektur fisik dari perangkat keras dan perangkat lunak dari aplikasi *emergency services* seperti ditunjuk dalam gambar dibawah ini



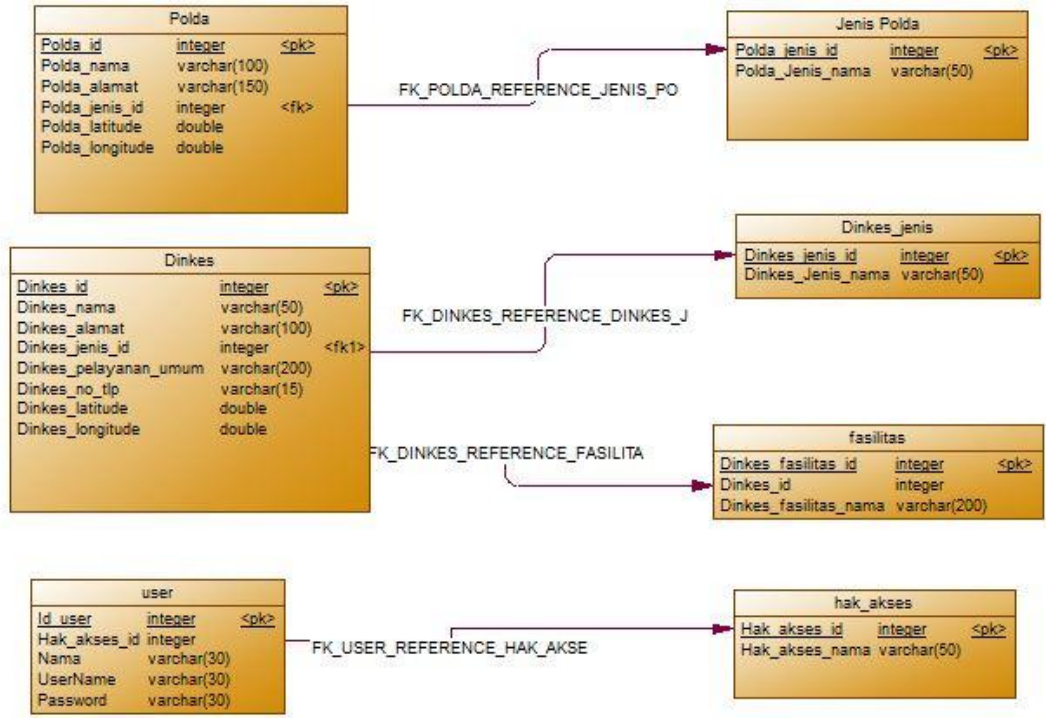
Gambar 3. ***35 Deplyoyment diagram***

3.6.1 Perancangan sistem

Perancangan sistem bertujuan untuk menspesifikasikan aspek-aspek teknik yang menjadi solusi dalam perencanaan. Pada tahap ini perancangan akan didefinisikan secara detail untuk mengatasi masalah-masalah yang lebih teknis, berkaitan dengan kegiatan implementasi seperti perancangan *database*, dan perancangan antarmuka.

3.6.1.1 Skema relasi

Relasi antar tabel merupakan gabungan antar file yang mempunyai kunci utama yang sama, sehingga file-file tersebut menjadi satu kesatuan yang dihubungkan oleh field kunci (*Primary Key*). Pada proses ini elemen-elemen data dikelompokkan menjadi satu file *database* beserta entitas dan hubungannya. Skema relasi ini dapat dilihat pada gambar 3.36



Gambar 3. 36 skema relasi

3.6.1.2 Struktur Tabel

Perancangan struktur tabel adalah perancangan tabel-tabel yang akan digunakan pada *database*. Tabel-tabel yang terdapat dalam database yang digunakan dalam aplikasi mobile android ini adalah **:**

Tabel 3. 25 Struktur tabel *user*

No	Nama field	Tipe Data	Panjang	Keterangan
1	Id_*user*	Integer	11	PK
2	Hak_akses_id	Integer	11	
3	Nama	Varchar	50	
4	*User*Name	Varchar	50	
5	Password	Varchar	50	

Tabel 3. 26 Struktur tabel hak akses

No	Nama field	Tipe Data	Panjang	Keterangan
1	Hak_akses_id	Integer	11	PK
2	Hak_akses_nama	Varchar	100	

Tabel 3. 27 Struktur tabel fasilitas

No	Nama field	Tipe Data	Panjang	Keterangan
1	Dinkes_fasilitas_id	Integer	11	PK
2	Dinkes_id	Integer	11	
3	Dinkes_fasilitas_nama	Varchar	50	

Tabel 3. 28 struktur tabel dinkes jenis

No	Nama field	Tipe Data	Panjang	Keterangan
1	Dinkes_jenis_id	Integer	11	PK
2	Dinkes_Jenis_nama	Varchar	50	

Tabel 3. 29 Struktur tabel dinkes

No	Nama field	Tipe Data	Panjang	Keterangan
1	Dinkes_id	Integer	11	PK
2.	Id_*user*	Integer	11	FK
3	Dinkes_nama	Varchar	100	
4	Dinkes_alamat	Varchar	150	
5	Dinkes_jenis_id	Integer	11	
6	Dinkes_pelayanan_umum	Varchar	150	
7	Dinkes_no_tlp	Varchar	50	
8	Dinkes_latitude	Double		
9	Dinkes_longitude	Double		

Tabel 3. 30 Tabel jenis polda

No	Nama field	Tipe Data	Panjang	Keterangan
1	Polda_jenis_id	Integer	11	PK
2	Polda_Jenis_nama	Varchar	50	

Tabel 3. 31 Struktur tabel polda

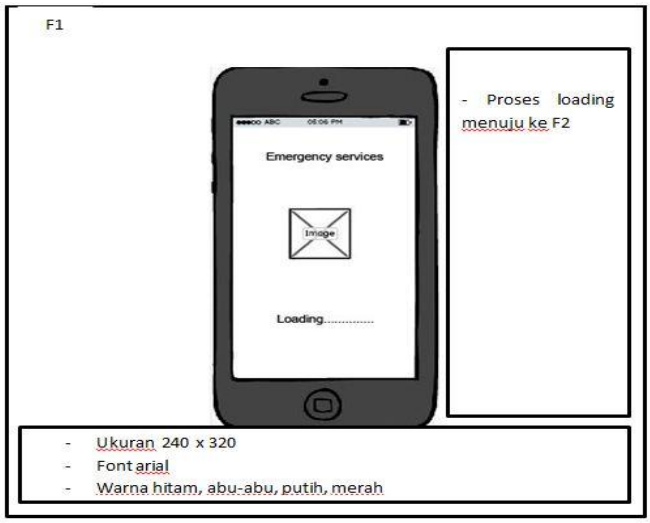
No	Nama field	Tipe Data	Panjang	Keterangan
1	Polda_id	Integer	11	PK
2	Id_*user*	Integer	11	FK
3	Polda_nama	Varchar	100	
4	Polda_alamat	Varchar	150	
5	Polda_jenis_id	Integer	11	
6	Polda_latitude	Double		
7	Polda_longitude	Double		

3.6.1.3 Perancangan Antar Muka

Perancangan antar muka merupakan penggambaran tampilan (*interface*) sebuah sistem yang akan digunakan secara langsung oleh pengguna, dan juga merupakan penggambaran interaksi yang dapat dilakukan oleh pengguna dalam sistem. Adapun dalam perancangan antar muka ini terdapat beberapa bagian yang harus dilakukan, yaitu dari menentukan rancangan input dan output pada setiap fungsi yang telah ditentukan

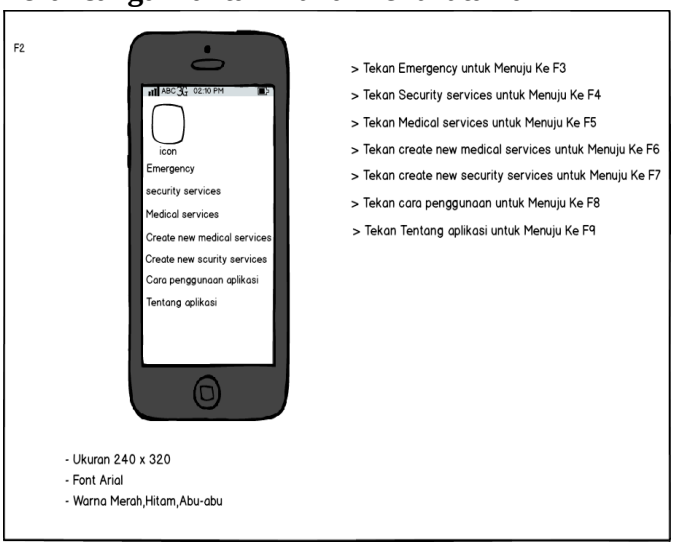
1. Perancangan Antarmuka *Frontend*

a. Perancangan antar muka loading screen



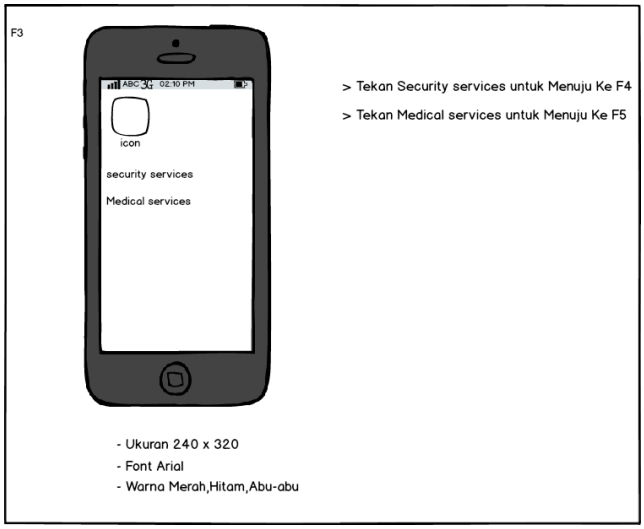
Gambar 3. 37 Perancangan *Loading Screen*

b. Perancangan antar muka menu utama



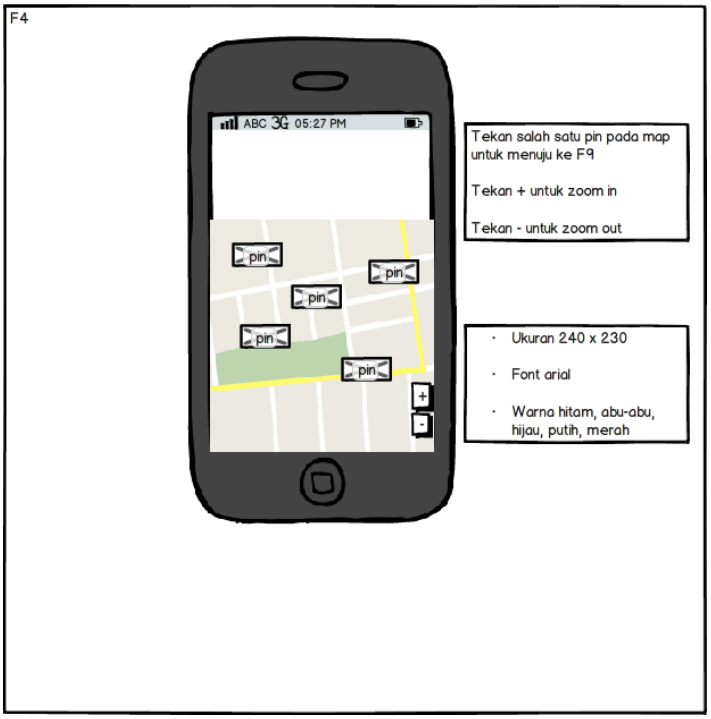
Gambar 3. 38 Perancangan antar muka menu utama

c. Perancangan antar muka menu darurat



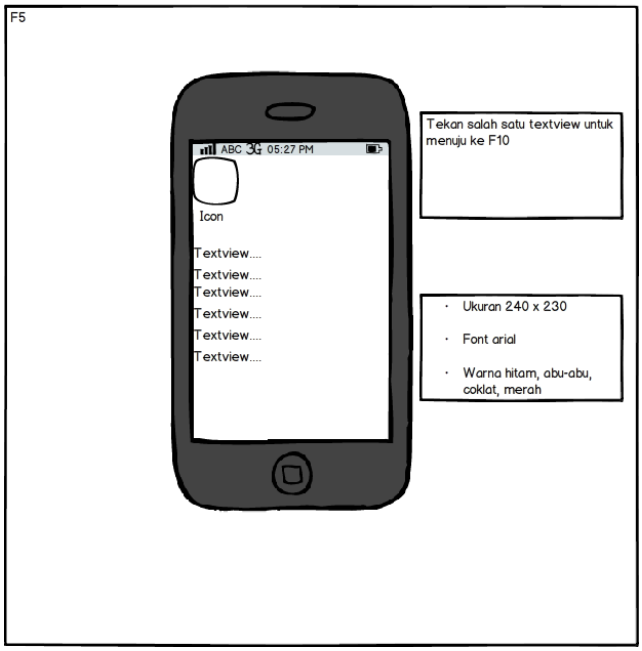
Gambar 3. 39 Perancangan antar muka menu darurat

d. Perancangan antar muka menu *security services*



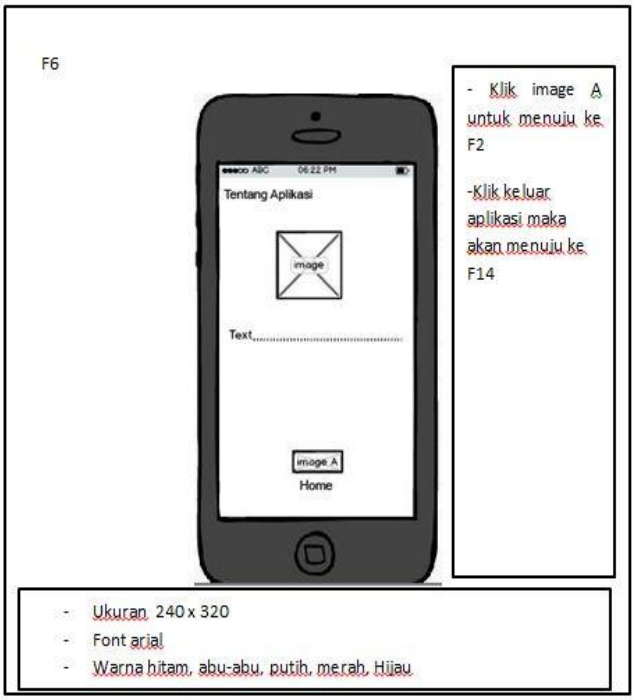
Gambar 3. 40 Perancangan antar muka menu security services peta

e. Perancangan antar muka kategori



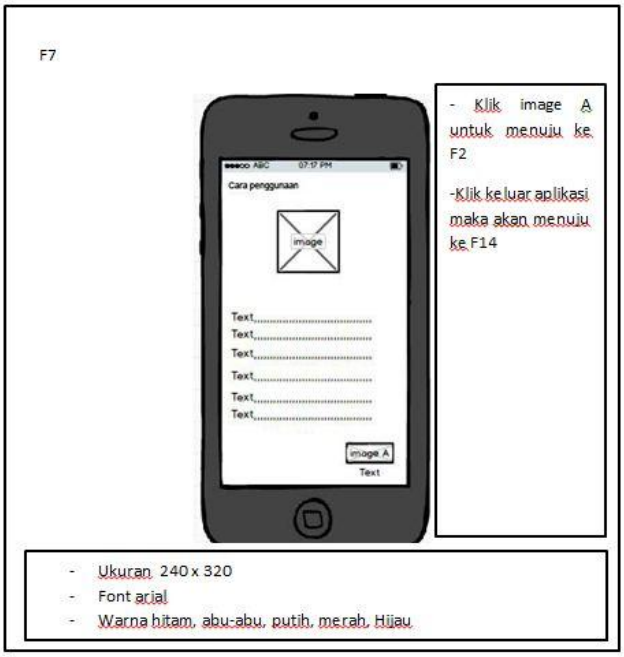
Gambar 3. 41 Perancangan antar muka menu kategori

f. Perancangan antar muka menu informasi aplikasi



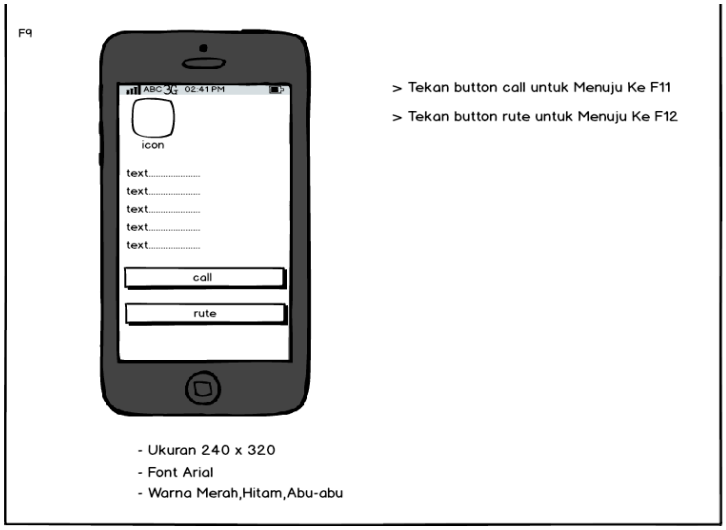
Gambar 3. 42 Perancangan antar muka menu informasi aplikasi

g. Perancangan antar muka menu cara penggunaan aplikasi



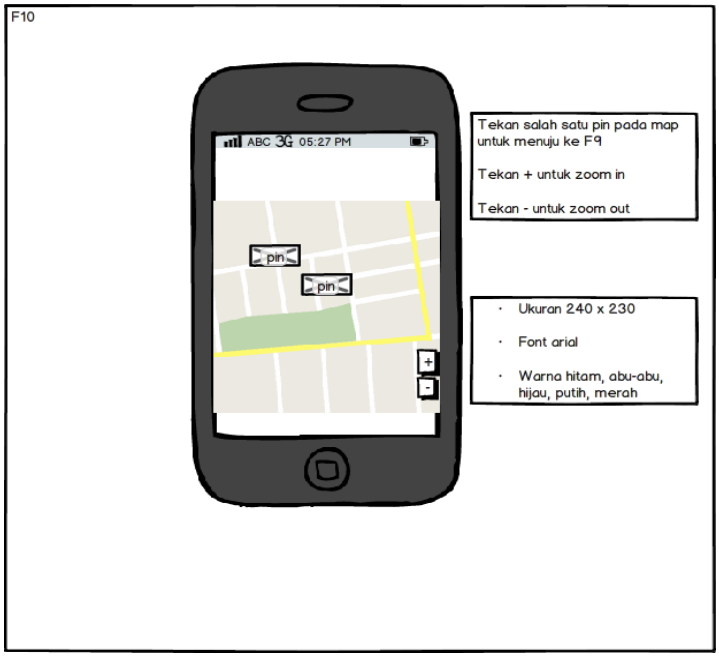
Gambar 3. 43 perancangan antar muka menu cara penggunaan aplikasi

h. Perancangan antar muka detail *emergency services*



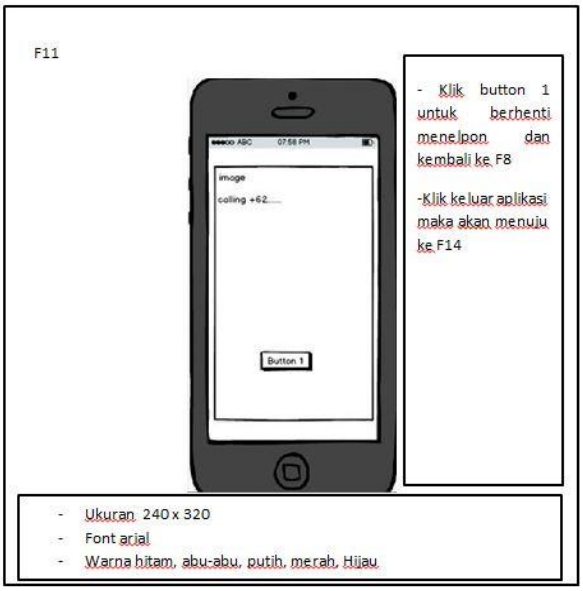
Gambar 3. 44 Perancangan antar muka detail emergency service

i. Perancangan antar muka menu detail kategori pada map



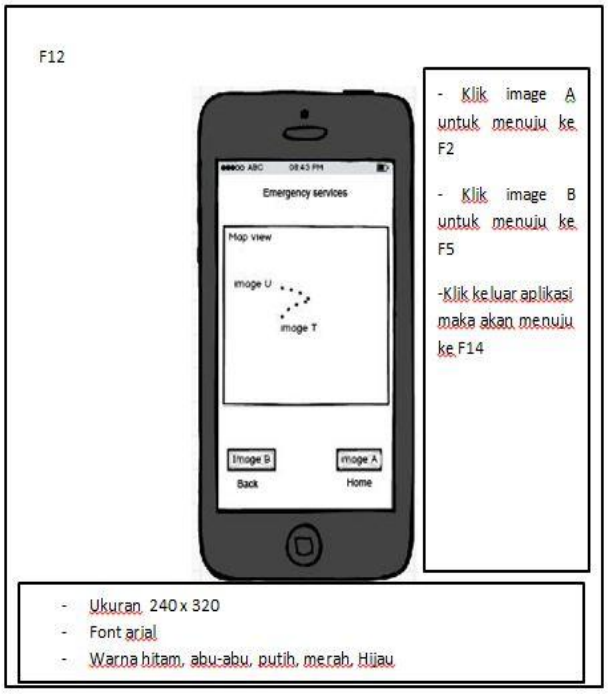
Gambar 3. 45 perancangan antar muka detail kategori

j. Perancangan antar muka menu *call*



Gambar 3. 46 perancangan antar muka menu *call*

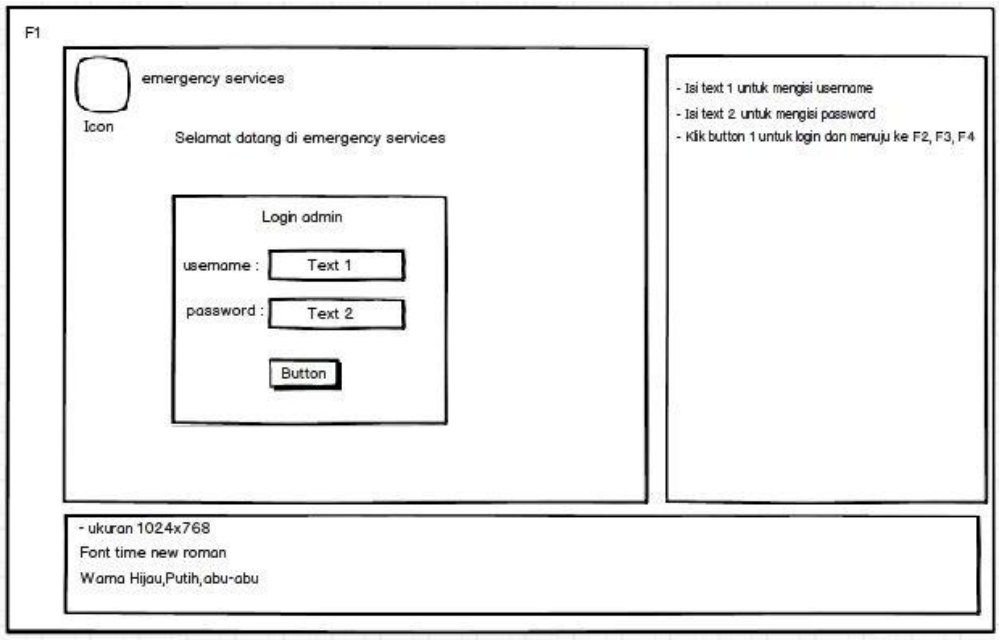
k. Perancangan antar muka menu *get direction*



Gambar 3. 47 Perancangan antar muka *get direction*

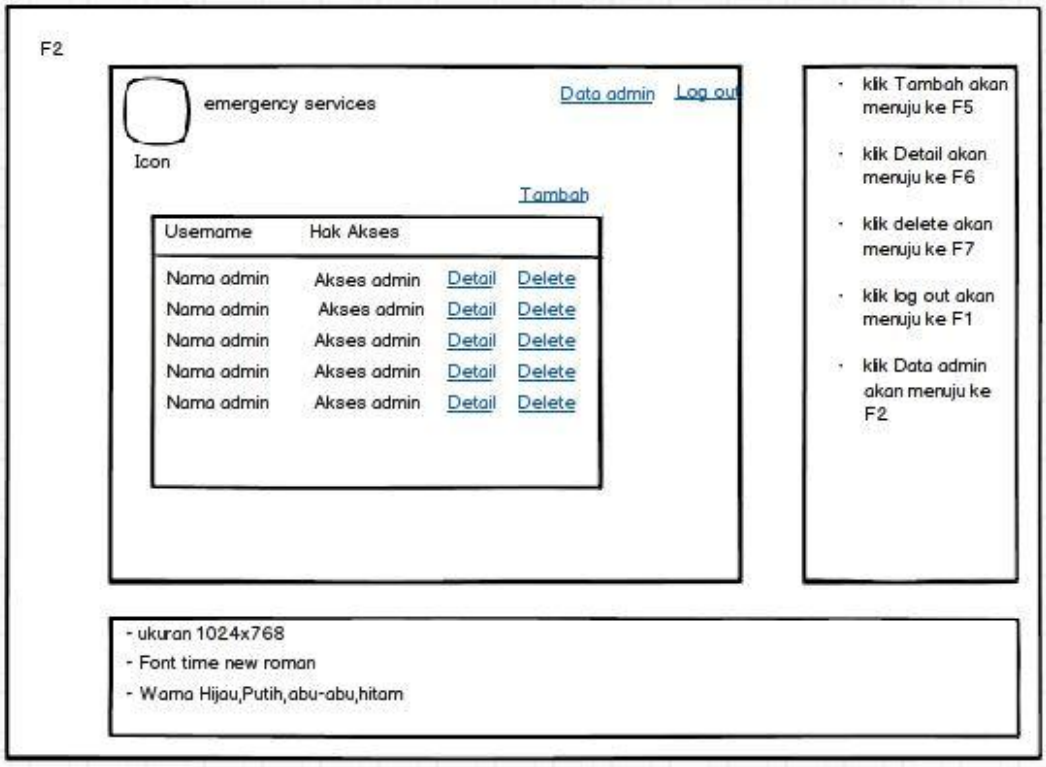
2. Perancangan antar muka *backend*

a. Perancangan antar muka login admin



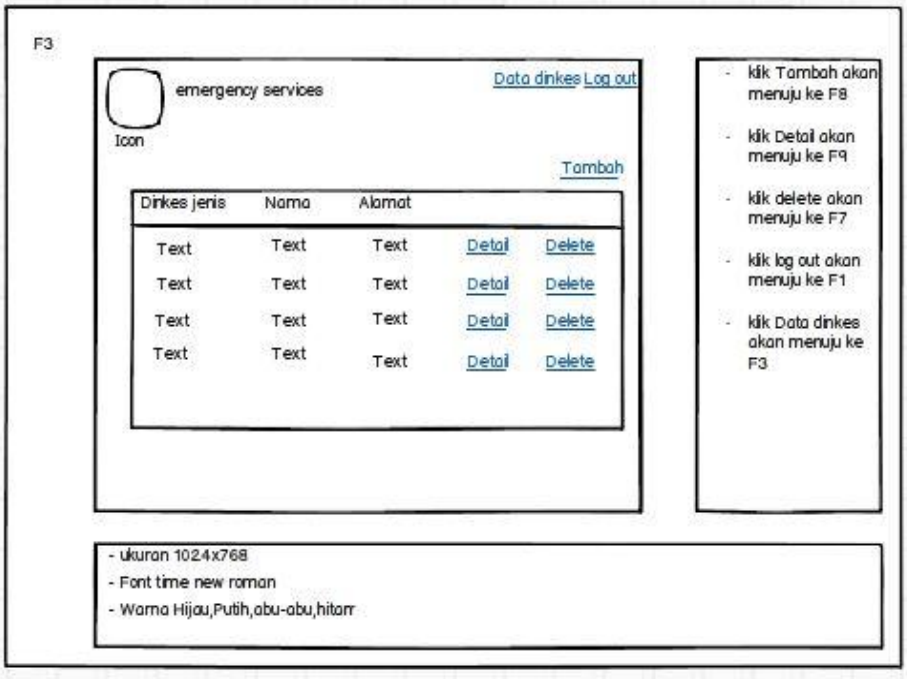
Gambar 3. 48 Perancangan antar muka login admin

b. Perancangan antar muka pengolahan data admin



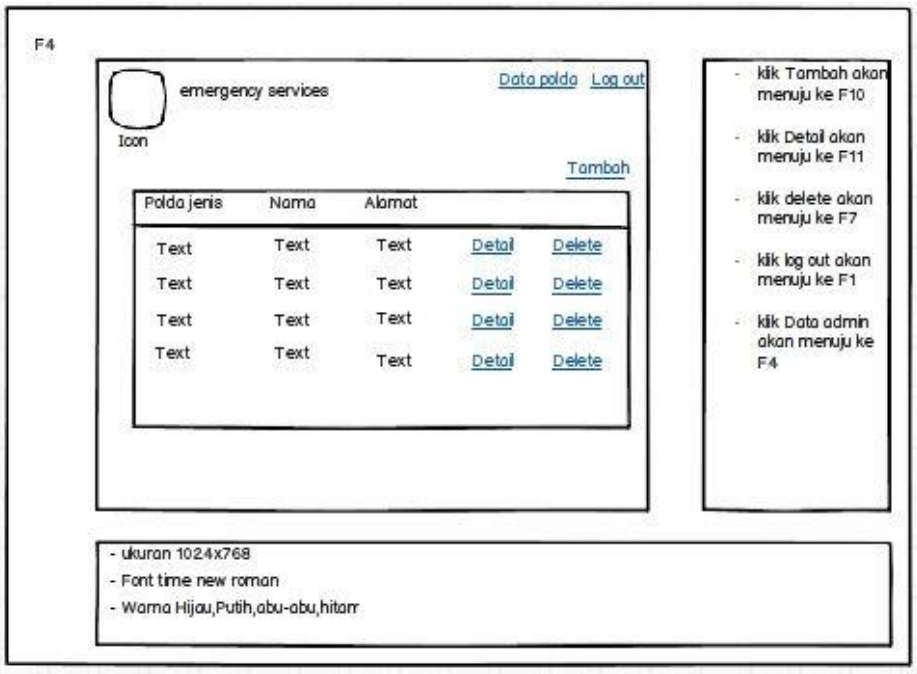
Gambar 3. 49 perancangan antar muka pengolahan data admin

c. Perancangan antar muka pengolahan data *medical services*



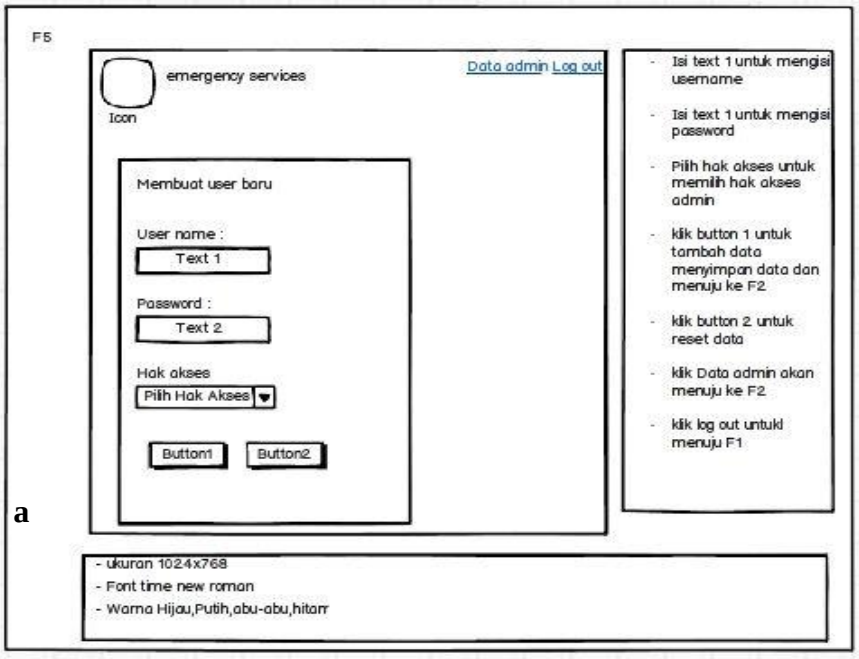
Gambar 3. 50 Perancangan antar muka pengolahan data *medical services*

d. perancanagn antar muka pengolahan data *security services*



Gambar 3. 51 Perancangan antar muka pengolahan *security services*

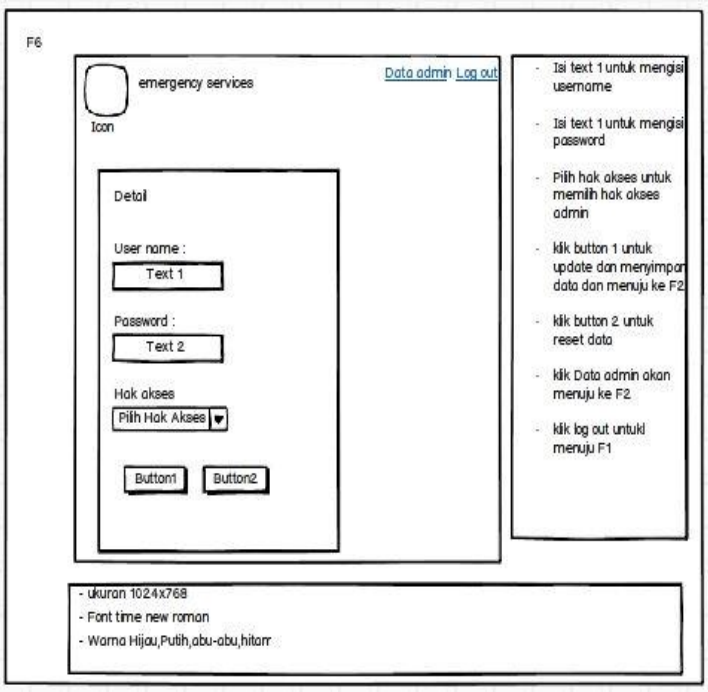
e. Perancangan antar muka menu tambah data admin



a

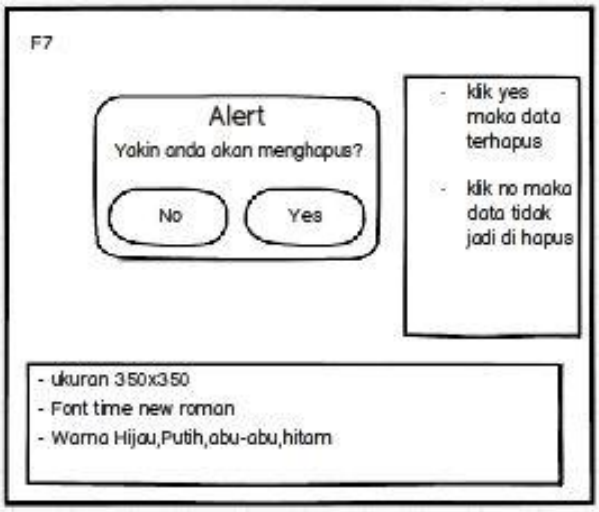
Gambar 3. 52 Perancangan antar muka menu tambah data admin

f. Perancangan antar muka detail data *security services*



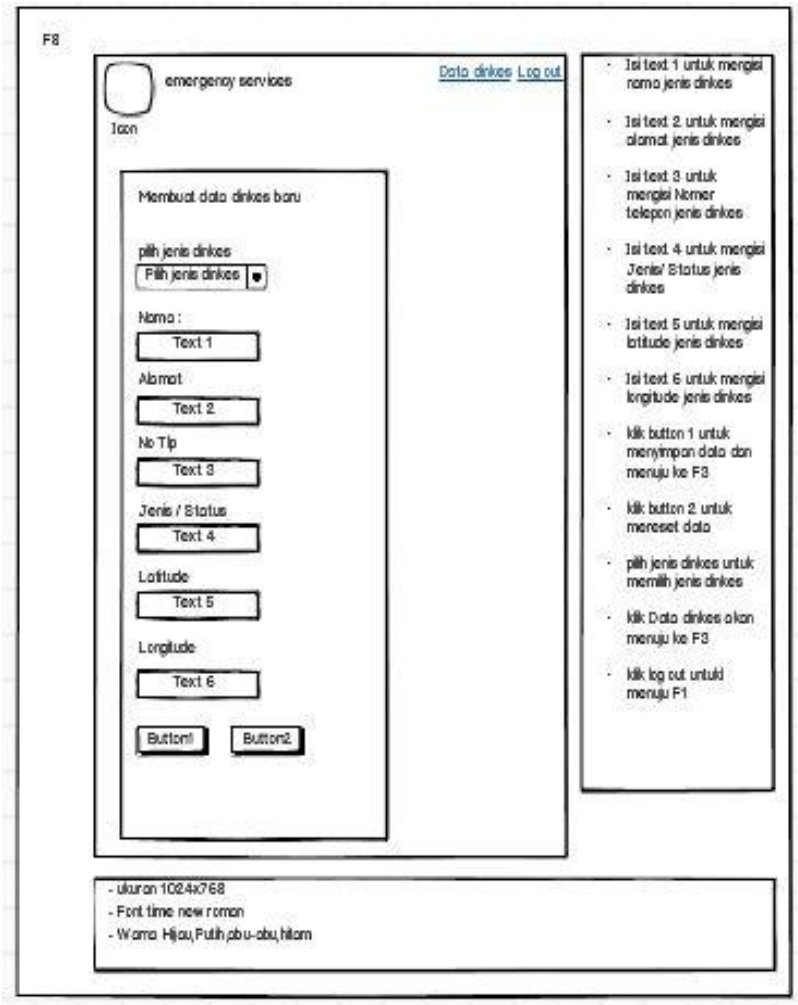
Gambar 3. 53 Perancangan antar muka detail data *security services*

g. Perancangan antar muka alert



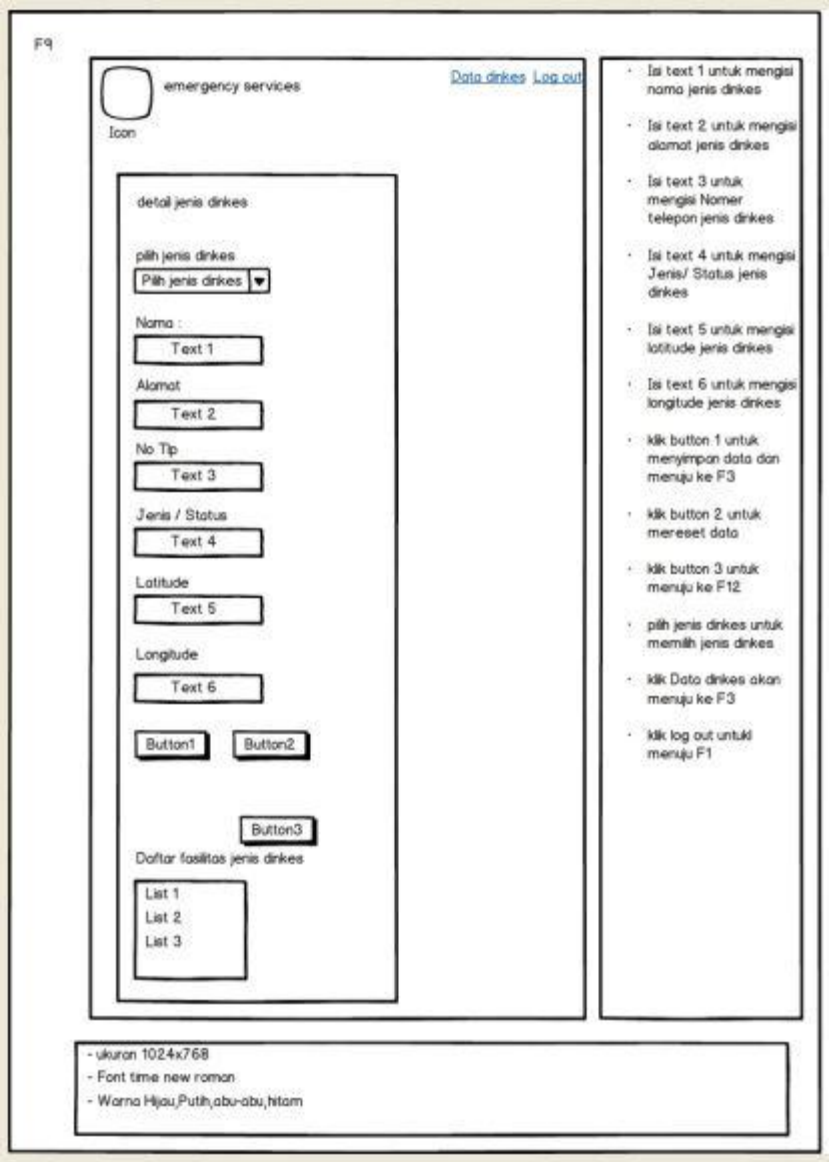
Gambar 3. 54 Perancangan antar muka alert

h. Perancangan antar muka menu tambah data *medical services*



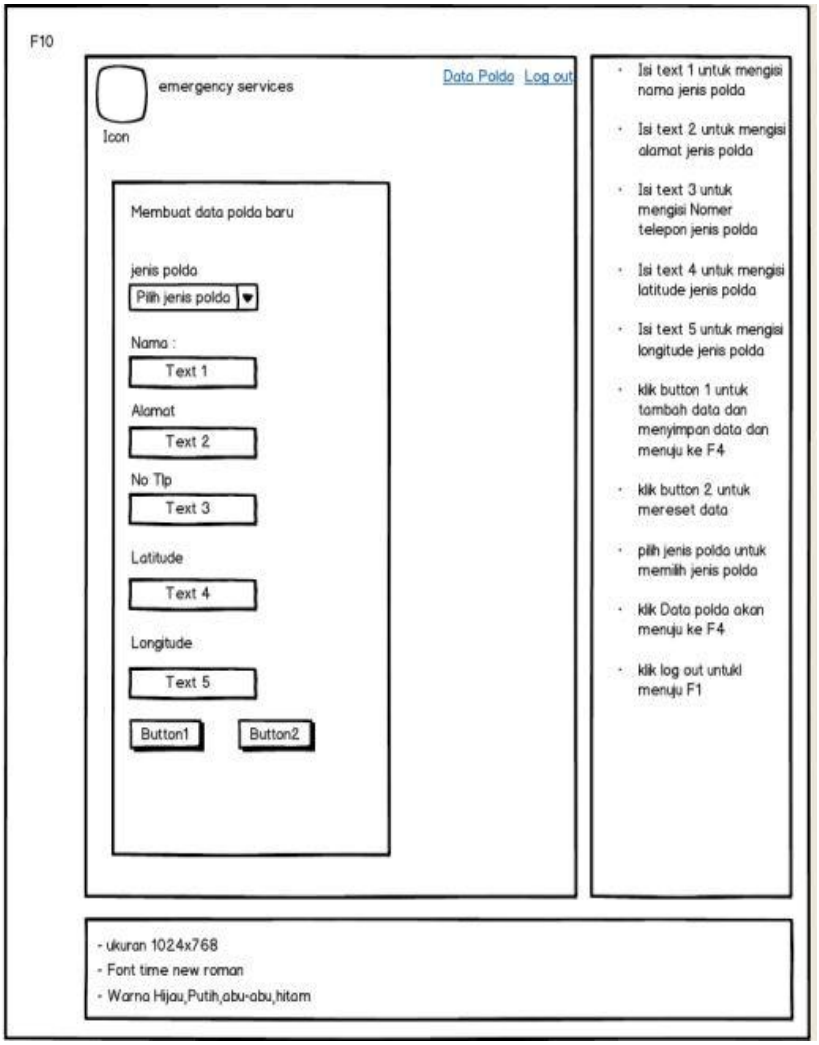
Gambar 3. 55 perancangan antar muka menu tambah data *medical services*

i. Perancangan antar muka menu detail jenis *medical services*



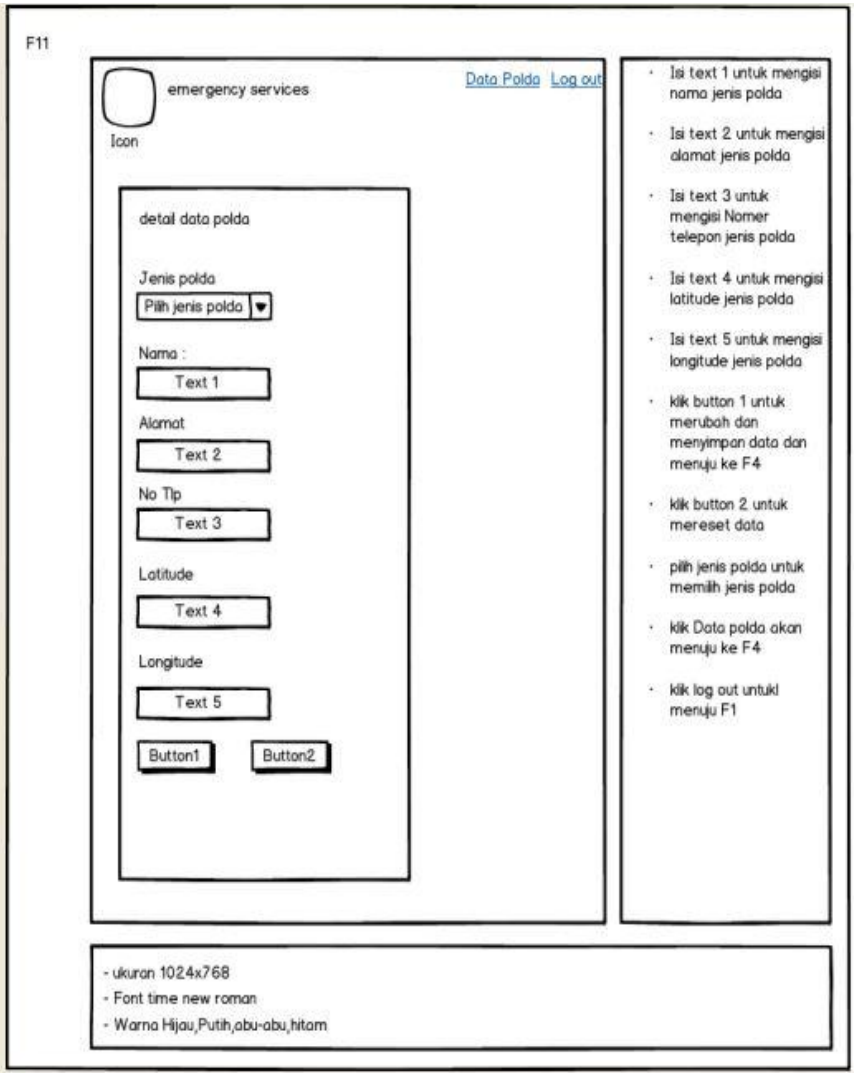
Gambar 3. 56 Perancangan antar muka detail jenis *medical services*

j. Perancangan antar muka menu tambah data *security services*



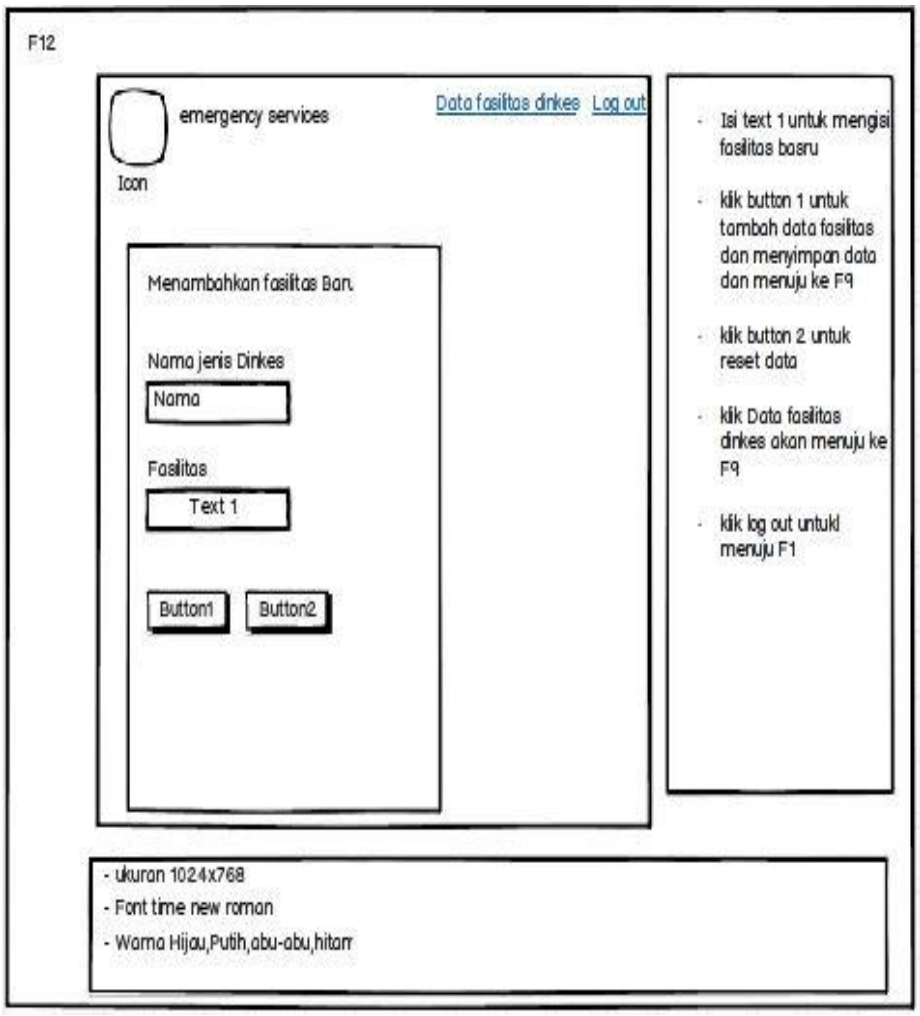
Gambar 3. 57 Perancangan antar muka menu tambah data *security services*

k. Perancangan antar muka menu detail data *security services*



Gambar 3. 58 Perancangan antar muka menu detail *security services*

l. **Perancangan antar muka menu tambah fasilitas *medical services***



Gambar 3. 59 Perancangan antar muka menu tambah data *medical serrvices*