BAB III
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem merupakan penguraian dari suatu sistem yang utuh ke dalam beberapa komponen dengan maksud untuk mengidentifikasikan dan mengevaluasikan permasalahan-permasalahan yang terdapat dalam suatu sistem.

Dalam pembangunan perangkat lunak ini dilakukan beberapa tahap analisis, yaitu :

a. Menentukan masalah yang akan dibangun untuk perangkat lunak deteksi gerak.
b. Mengumpulkan data yang diperlukan untuk membangun perangkat lunak melalui studi literatur.
c. Analisis kebutuhna Non Fungsional yang merupakan batasan-batasan dari layanan-layanan dan fungsi-fungsi dari sebuat perangkat lunak.
d. Analisis Kebutuhan Fungsinal berupa fungsionalitas dan layanan yang terdapat dalam perangkat lunak.

3.1.1 Analisis Masalah

Pada sistem yang telah dibangun sebelumnya pengambilan gambar pada ruangan menggunakan periodik watu tertentu dan hasil monitoring tersebut tersimpan pada server, untuk melihat hasil monitoring tersebut pengguna mengirimkan request berupa sms ke sistem dan sistem akan mengirimkan hasil monitoring tersebut ke *handphone* pengguna melalui *email*. Terdapat beberapa masalah yang terdapat dalam sistem yang telah dibuat sebelumnya

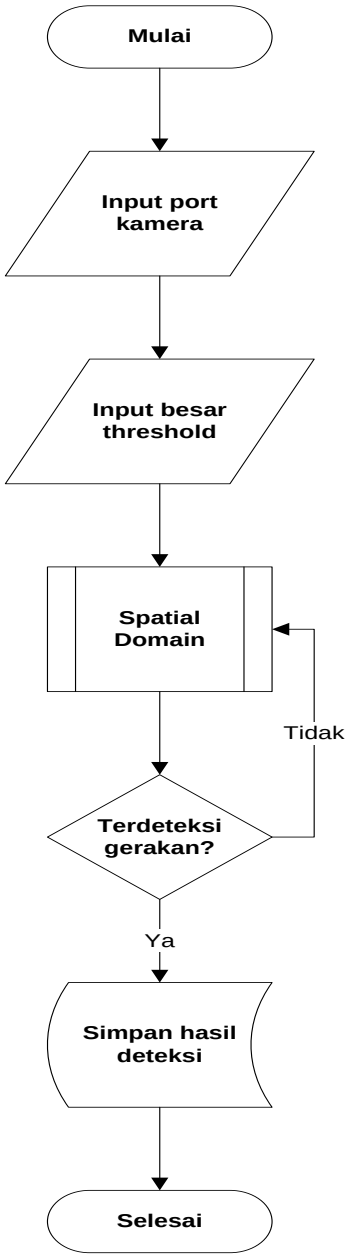
1. Hasil monitoring masih belum akurat karena metode pengambilan gambar menggunakan periodik waktu.
2. Hasil monitoring tidak langsung terkirim ke pengguna karena sistem menunggu request dari pengguna.

3.1.2 Analisis Sistem Yang Ditawarkan

Untuk mengatasi permasalahan tersebut maka perlu dibuat suatu sistem yang hanya akan meng-*capture* ruangan yang dipantau bila terjadi perubahan kondisi atau adanya gerakan pada rungan yang dipantau dengan menggunakan motode deteksi gerak, hasil monitoring bisa diakses melalu *web* dan *mobile*, pada aplikasi *mobile* pengguna akan menerima peringatan berupa notifikasi jika terjadi gerakan pada area yang dipantau.

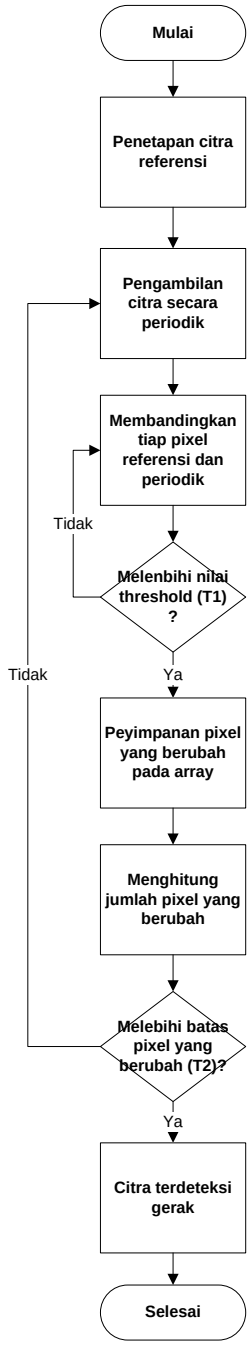
3.1.3 Analisis Metode

Penelitian ini menggunakan metode *spatial domain* untuk pendekteksian gerak. Dalam sistem ini tahap analisis dikhususkan dalam proses pendeteksian gerak, bagan pengenalan secara utuh dapat dilihat pada Gambar 3.1.



Gambar 3.1 Gambaran umum sistem pendeteksi gerak

Metode *Spatial Domain* mempuyai tahapan-tahapan dalam proses pendeteksian gerak, seperti yang dapat dilihat pada Gambar 3.2.

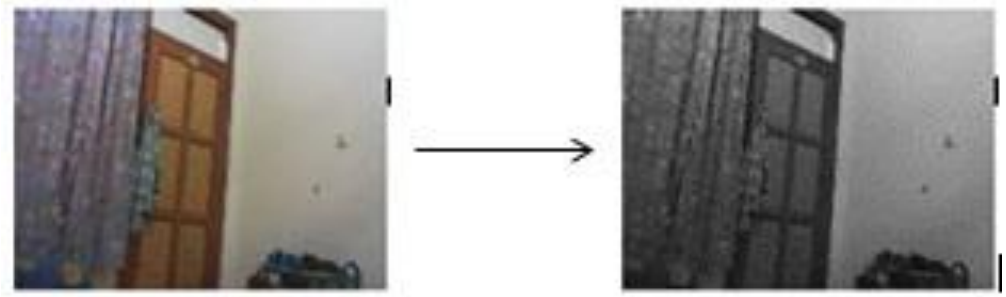


Gambar 3.2 ***Flowchart Spatial Domain***

Berikut ini merupakan penjelasan dari tahap-tahap yang ada dalam proses pendeteksian gerak pada metode *spatial domain* :

a. Penetapan sebuah citra referensi

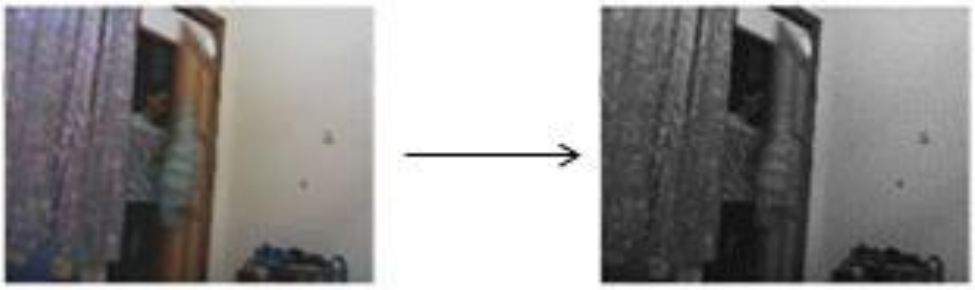
Sebuah citra yang diambil oleh kamera pada saat situasi lingkungan yang stabil ditetapkan sebagai citra referensi, Cintra ini kemudian diubah menjadi citra *grayscale* dan disimpan sebagai acuan dalam membandingkan.



Gambar 3.3 Tahap Citra Referensi

b. Pengambilan citra secara periodik

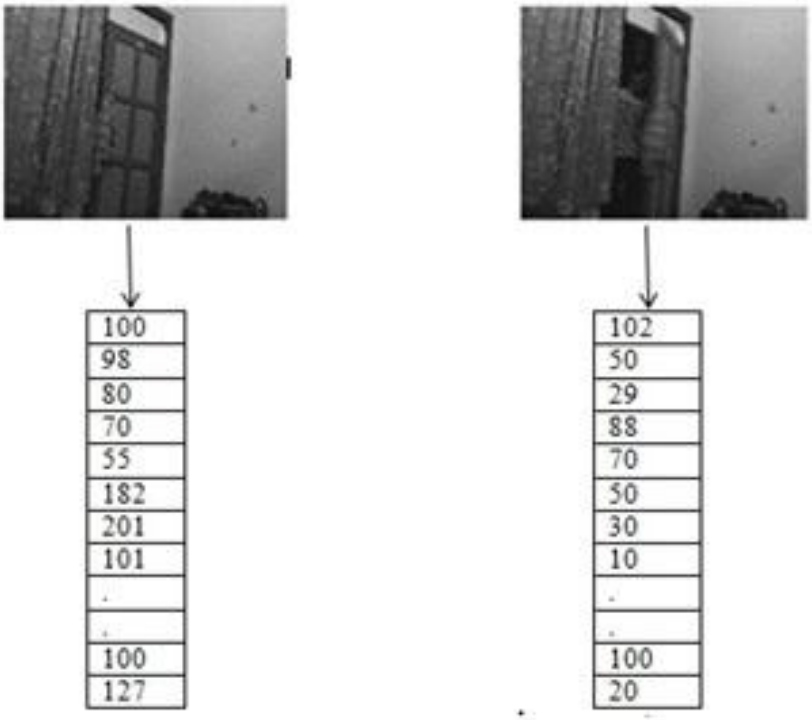
Setiap jangaka waktu tertentu, sebuah citra diambil oleh kamera. Citra ini pun kemudian diubah menjadi citra *grayscale* dan disimpan untuk perhitungan selanjutnya.



Gambar 3.4 Tahap Citra Periodik

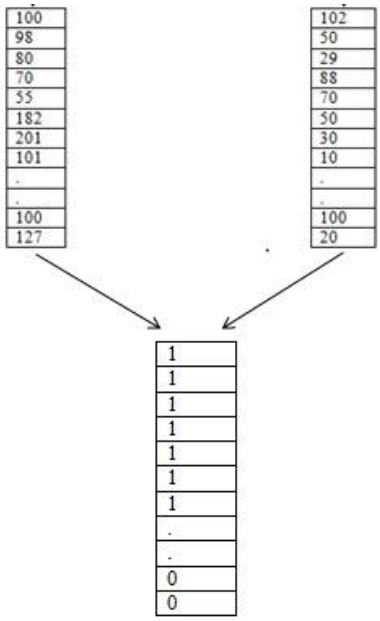
c. Pendeteksi Gerakan

Dari citra referensi dan citra periodik yang sudah disimpan, maka proses selanjutnya adalah membandingkan kedua citra tersebuh dengan mengambil nilai piksel dari tiap titik yang sama, untuk mempermudah proses perbandingan nilai piksel, nilai piksel dari kedua citra tersebut masing-masing dipindahkan pada *array*.



Gambar 3.5 Proses Pemindahan Nilai Piksel ke *Array*

Isi dari dua array tersebut akan dibandingkan dengan pehitungan pengurangan absolut pada tiap indeknya, jika hasil pengeurangan melebihi dari batas nilai yang sudah ditentukan maka hasilnya adalah 1 jika tidak maka hasilnya adalah 0.

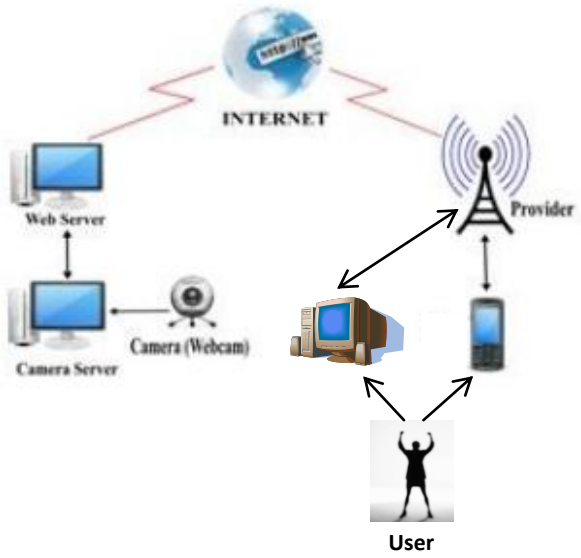


Gambar 3.6 Proses Proses Perbandingan Citra

Dasil dari pengurangan tersebut disimpan pada *array* baru. Jika jumlah nilai 1 pada hasil pengurangan tersebut melebihi dari 12% dari besar ukuran array, maka citra periodik tersebut dianggap sebagai citra yang bergerak.

3.1.4 Deskripsi Sistem

Arsitektur sistem ini terdiri dari beberapa komponen yaitu *frontend* berbasis Android *mobile, frontend* berbasis *web* dan *backend* berupa aplikasi *desktop*, *web server* dan *web service.*



Gambar 3.7 Deskripsi Sistem

Pada gambar 3.3 menjelaskan alur sistem yang akan dibangun. Pada sistem tersebut terdapat tiga aplikasi yang akan dibangun yaitu aplikasi *frontend* pada *mobile*, *frontent* pada *web* dan aplikasi *backend* sebagai aplikasi pendeteksi gerak.

3.1.4.1 *Aplikasi Backend*

Pada bagian ini terdapat dua bagian applikasi *desktop* sebagai aplikasi yang memantau ruangan dan menangkap gerakan yang terjadi pada ruangan yang dipantau menggunakan metode *spatial domain*, dan sebuah *web service* sebagai penghubung berbagai *platform* aplikasi yang dibangun dengan berbagai *platform* yang berbeda (contohnya Java, C#, VB, PHP, Android) yang memungkinkan seluruh aplikasi untuk dapat saling berkomunikasi. *Web service* yang digunakan untuk membangun sistem ini adalah JSON (*javascript Object Notation*).

3.1.4.2 *Aplikasi Frontend Mobile*

Aplikasi ini berfungsi untuk menerima notifikasi jika terjadi gerakan pada area yang dipantau, menampilkan hasil deteksi dan memberikan aksi alarm ke sistem *backend.*

3.1.4.3 *Aplikasi Frontend Web*

Aplikasi ini berfungsi untuk memantau hasil deteksi melalui web, pada sistem web ini fungsionalitasnya sama seperti pada *Mobile*.

3.1.5 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional menjelaskan pendukung sistem yang akan dijalankan. Adapun kebutuhan-kebutuhan yang diperlukan untuk mendukung kinerja dari perangkat lunak yaitu :

1. Analisis Kebutuhan Perangkat Keras

Komputer merupakan gabungan dari perangkat keras dan perangkat lunak yang satu sama lain saling mendukung kinerjanya. Perangkat lunak memerintahkan perangkat keras atau instruksi untuk melakukan tugas-tugas tertentu.

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

a. Perangkat Keras Aplikasi *Backend*

Pada aplikasi *backend* yang berupa aplikasi desktop, aplikasi ini berjalan pada sebuah perangkat komputer dengan spesifikasi kebutuhan perangkat keras komputer sebagai berikut :

Tabel 3.1 Spesifikasi Kebutuhan Perangkat Keras backend

No	Perangkat Keras	Spesifikasi
1	Processor	Dual Core @ 2.2GHz
2	VGA	512 MB
3	Memori	2GB
4	Kamera	2.0 Megapixel
5	Keyboard dan Mouse	

b. Perangkat Keras Aplikasi *Frontend Mobile*

Aplikasi *frontend mobile* dijalankan pada sebuah perangkat *smartphone* yaitu *smartphone* dengan dengan sistem Android, kebutuhan perangkat keras untuk *smartphone* tersebut yaitu :

Tabel 3.2 Spesifikasi Perangkat Keras frontend mobile

No	Perangkat Keras	Spesifikasi
1	Resolusi	240 x 230
2	Memori	100 MB
3	CPU	300 Mhz
4	WLAN	Wi-fi 802.11 b/g/n
5	Jaringan 3G UMTS	

c. Perangkat Keras Aplikasi *Frontend Web*

Spesifikasi kebutuhan perangkat keras untuk mendukung sistem monitoring pada *web* yaitu sebagai berikut

Tabel 3.3 Spesifikasi Perangkat Keras frontend web

No	Perangkat Keras	Spesifikasi
1	Processor	Kecepatan minimum 2.0 Ghz
2	Monitor	Minimum Monitor 14,1", resolusi 1280 x 800
3	VGA	Kecepatan minimum 128 MB
4	Memori	Minimum DDR2 512 MB
5	Keyboar	Standard, Port USB
6	Mouse	Optical Mouse USB
7	Koneksi	Kecepatan minimum 1 Mbp/s

2. Analisis Kebutuhan Perangkat Lunak

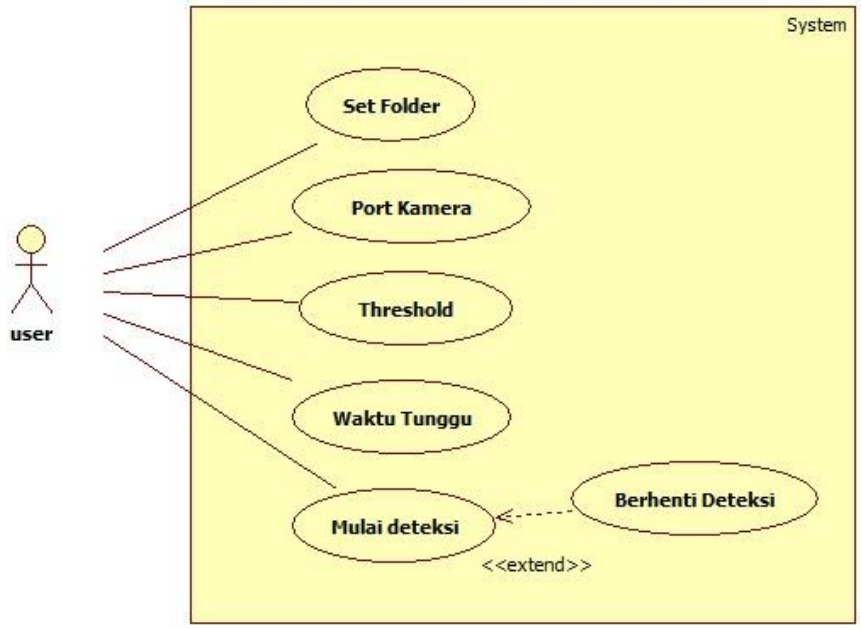
Perangkat lunak digunakan untuk memberikan perintah-perintah kepada perangkat keras untuk melakukan instruksi-instruksi. Berikut kriteria dari perangkat lunak yang dibutuhkan untuk mendukung perangakat lunak monitoring ini :

a. Sistem operasi berbasis Linux Ubuntu.
b. Platform Android versi 2.2 keatas.
c. Qt
d. Aplikasi IDE Eclipse Juno.
e. JDK v 1.6 dan Android SDK Linux

3.1.6 Analisis Kebutuhan Fungsional

Analisis kebutuhan Fungsional merupakan alur dan pekerjaan dari perangkat lunak yang akan dibangun. Dalam perangkat lunak monitoring ini digunakan metode pendekatan berorientasi objek.

1. *Use Case Backend*



Gambar 3.8 *Use Case Backend*

Pada gambar di atas terdapat aktor user yang berhubungan langsung dengan perangkat lunak pada sisi *backend*. Sedangkan perangkat lunak sendiri mempuyai beberapa *use case* yang merupakan kegiatan atau pekerjaan yang dilakukan oleh perangkat lunak. Setiap *use case* dijelaskan dalam *use case* scenario.

Nama *use case* : Port Kamera

Aktor : Admin

Skenario :

Tabel 3. 4 *Use Case Skenario Port* Kamera

No	Aksi Aktor	No	Reaksi Sistem
Skenario normal			
1	Pengguna memilih port kamera yang akan digunakan		
		2	Menyimpan port kamera

Nama *use case* : Threshold

Actor : Admin

Skenario :

Tabel 3.5 *Use Case* Skenario *Threshold*

No	Aksi Aktor	No	Reaksi Sistem
Skenario normal			
1	Pengguna menentukan nilai threshold		
		2	Menyimpan nilai threshold

Nama *use case* : Waktu tunggu

Actor : Admin

Skenario :

Tabel 3.6 Use Case Skenario Waktu Tunggu

No	Aksi Aktor	No	Reaksi Sistem
Skenario normal			
1	Pengguna menentukan waktu tunggu		
		2	Menyimpan nilai threshold

Nama *use case* : Mulai deteksi

Aktor : User

Skenario :

Tabel 3.7 Use Case Skenario Mulai Deteksi

No	Aksi Aktor	No	Reaksi Sistem
Skenario normal			
1	User menekan tombol mulai deteksi		
		2	Memeriksa ketersedian port kamera
		4	Menyalakan kamera
		5	Kamera menangkap citra referensi dan citra periodik

		6	Deteksi gerak menggunakan metode spatial domain
		7	Menyimpan hasil deteksi gerak
		8	Menampilkan hasil deteksi gerak
Skenario ekstension			
1	Pengguna menekan tombol mulai deteksi		
		2	Memeriksa ketersedian port kamera
3	Port kamera tidak tersedia		
		4	Menampilkan pesan kesalahan

Nama *use case* : Berhenti deteksi

Aktor : User

Skenario :

Tabel 3. 8 Use Case Skenario Berhenti Deteksi

No	Aksi Aktor	No	Reaksi Sistem
Skenario normal			
1	Pengguna menekan tombol berhenti deteksi		

		2	Mematikan kamera
		3	Berhenti mendeteksi gerak

Nama *use case* : Set Folder

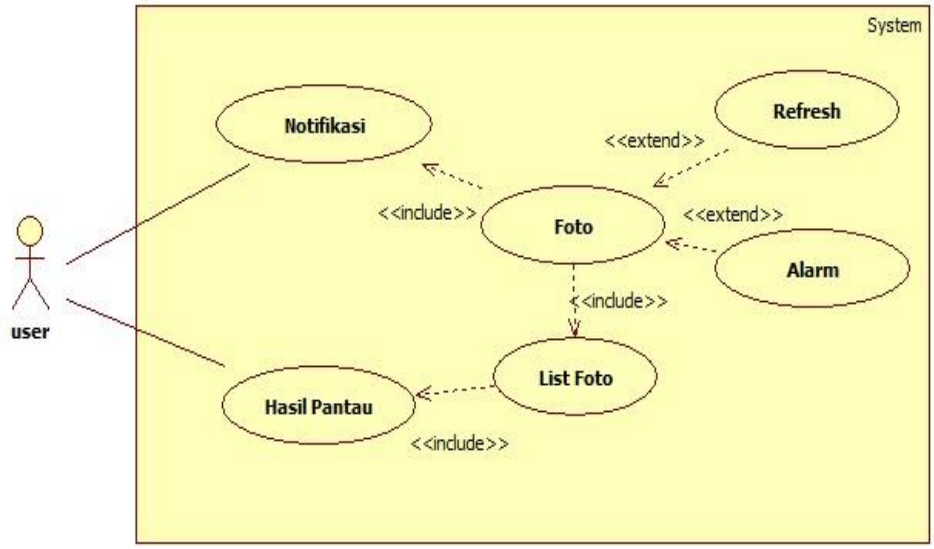
Aktor : User

Skenario :

Tabel 3.9 Use Case Skenario Set Folder

No	Aksi Aktor	No	Reaksi Sistem
Skenario normal			
1	User menekan tombol set folder		
		2	Menampilkan window *explorer*
3	Menentukan folder tempat disimpannya hasil monitoring		
		4	Meyimpan alamat folder

2. *Use Case Frontend mobile*



Gambar 3.9 Use Case Frontend Mobile

Pada gambar di atas terdapat aktor user yang berhubungan langsung dengan perangkat lunak pada sisi *frontend mobile*. Sedangkan perangkat lunak sendiri mempuyai beberapa *use case* yang merupakan kegiatan atau pekerjaan yang dilakukan oleh perangkat lunak. Setiap *use case* dijelaskan dalam *use case* scenario.

Nama *use case* : Hasil pantau

Aktor : User

Skenario :

Tabel 3.10 Use Case Skenario Hasil Pantau

No	Aksi Aktor	No	Reaksi Sistem
Skenario normal			
1	User memilih tombol hasil pantau		
		2	Ambil data (JSON)
		3	Meyimpan data dalam list
		4	Menampilkan activity list foto

Nama *use case* : List Foto

Aktor : User

Skenario :

Tabel 3.11 Use Case Skenario List Foto

No	Aksi Aktor	No	Reaksi Sistem
Skenario normal			
1	User memilih salah satu isi list		
		3	Menampilkan activity foto

Nama *use case* : Refresh

Aktor : User

Skenario :

Tabel 3.12 Use Case Skenario Refresh

No	Aksi Aktor	No	Reaksi Sistem
Skenario normal			
1	User menekan menu		
2	User menekan refresh		
		3	Ambil data (JSON)
		4	Menampilkan hasil foto

Nama *use case* : Notifikasi

Aktor : User

Skenario :

Tabel 3.13 Use Case Skenario Notifikasi

No	Aksi Aktor	No	Reaksi Sistem
Skenario normal			
		1	Background service
		2	Ambil data (JSON)
		3	Ada data baru
		4	Memberikan Notifikasi
5	Membuka notifikasi		

		6	Menampilkan activity foto

Nama *use case* : Alarm

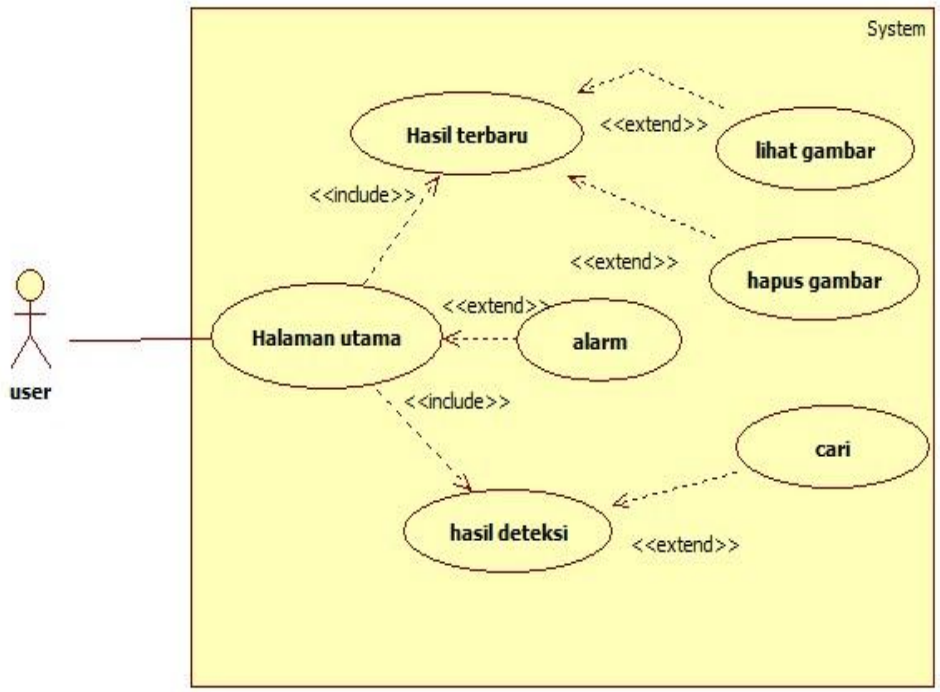
Aktor : User

Skenario :

Tabel 3.14 Use Case Skenario Alarm

No	Aksi Aktor	No	Reaksi Sistem
Skenario normal			
1	User menekan alarm		
		2	Mengirim data alarm aktif
		3	Membaca database
		4	Status alarm ON

3. *Use Case Frontend web*



Gambar 3.10 Use Case Frontend Web

Pada gambar di atas terdapat aktor *user* yang berhubungan langsung dengan perangkat lunak pada sisi *frontend web*. Sedangkan perangkat lunak sendiri mempuyai beberapa *use case* yang merupakan kegiatan atau pekerjaan yang dilakukan oleh perangkat lunak. Setiap *use case* dijelaskan dalam *use case* skenario.

Nama *use case* : Hasil terbaru

Aktor : User

Skenario :

Tabel 3.15 Use Case Skenario Hasil Terbaru

No	Aksi Aktor	No	Reaksi Sistem
Skenario normal			
		2	Ambil data (GET)
		3	Ada data baru
		4	Menampilkan list hasil terbaru

Nama *use case* : Alarm

Aktor : User

Skenario :

Tabel 3.16 Use Case Skenario Alarm

No	Aksi Aktor	No	Reaksi Sistem
Skenario normal			
1	User menekan alarm		
		2	Mengirim data alarm aktif
		3	Membaca database
		4	Status alarm ON

Nama *use case* : Hasil Deteksi

Aktor : User

Skenario :

Tabel 3.17 Use Case Skenario Hasil Deteksi

No	Aksi Aktor	No	Reaksi Sistem
Skenario normal			
		2	Ambil data (GET)
		3	Menampilkan list hasil deteksi

Nama *use case* : Cari

Aktor : User

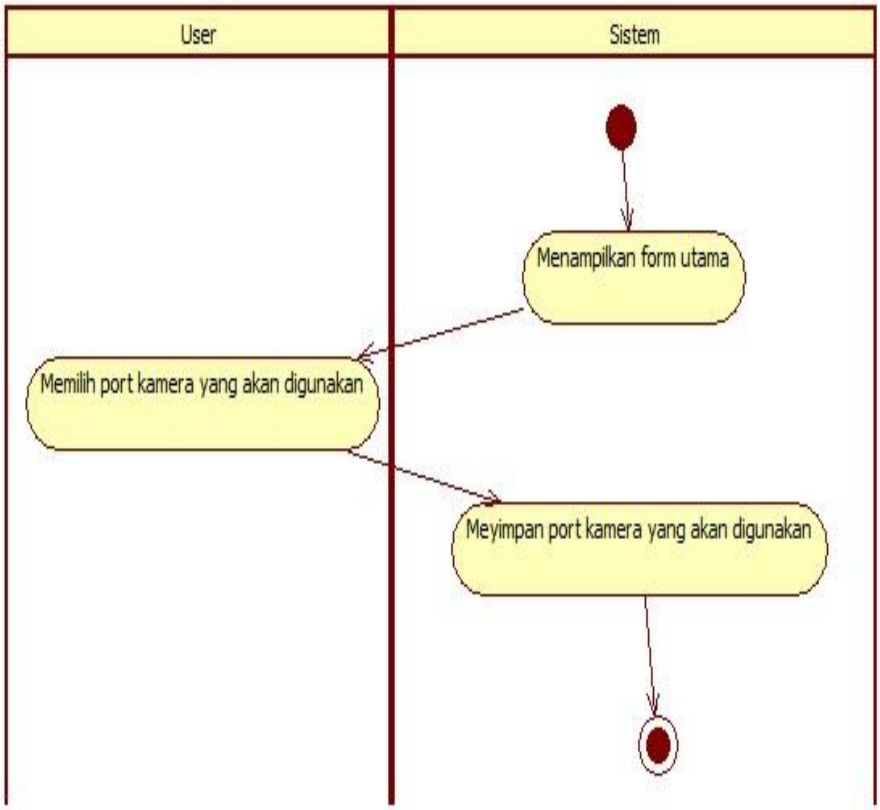
Skenario :

Tabel 3.18 Use Case Skenario Cari

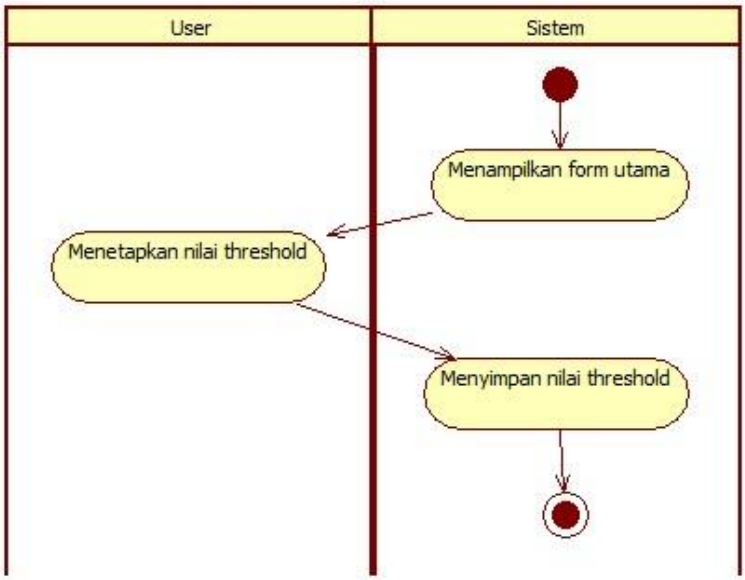
No	Aksi Aktor	No	Reaksi Sistem
Skenario normal			
1	Memilih tanggal		
		2	Mengambil data sesuai tanggal yang dipilih
		3	Menampilkan list data sesuai tanggal yang dipilih

4. *Activity Diagram*

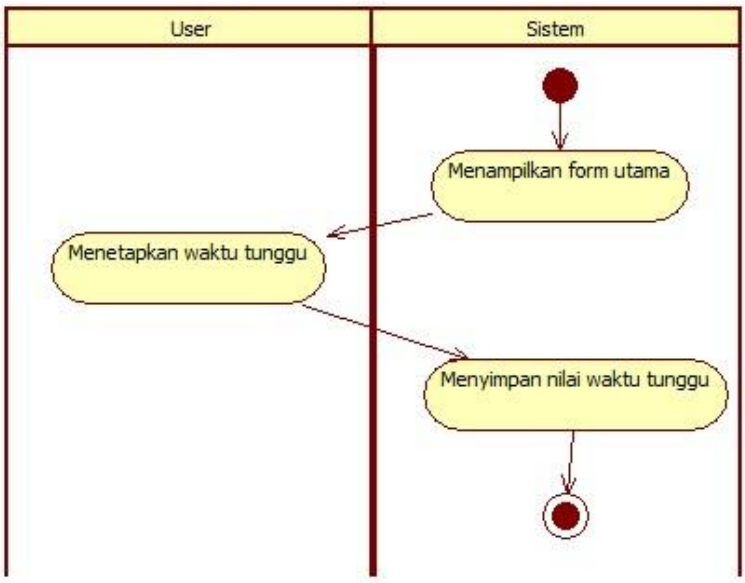
Untuk menjelaskan aktifitas setiap user digunakan *tools* yang bernama *activity diagram*. Berikut ini merupakan gambaran aktifitas pada setiap *use case fronend* dan *backend*:



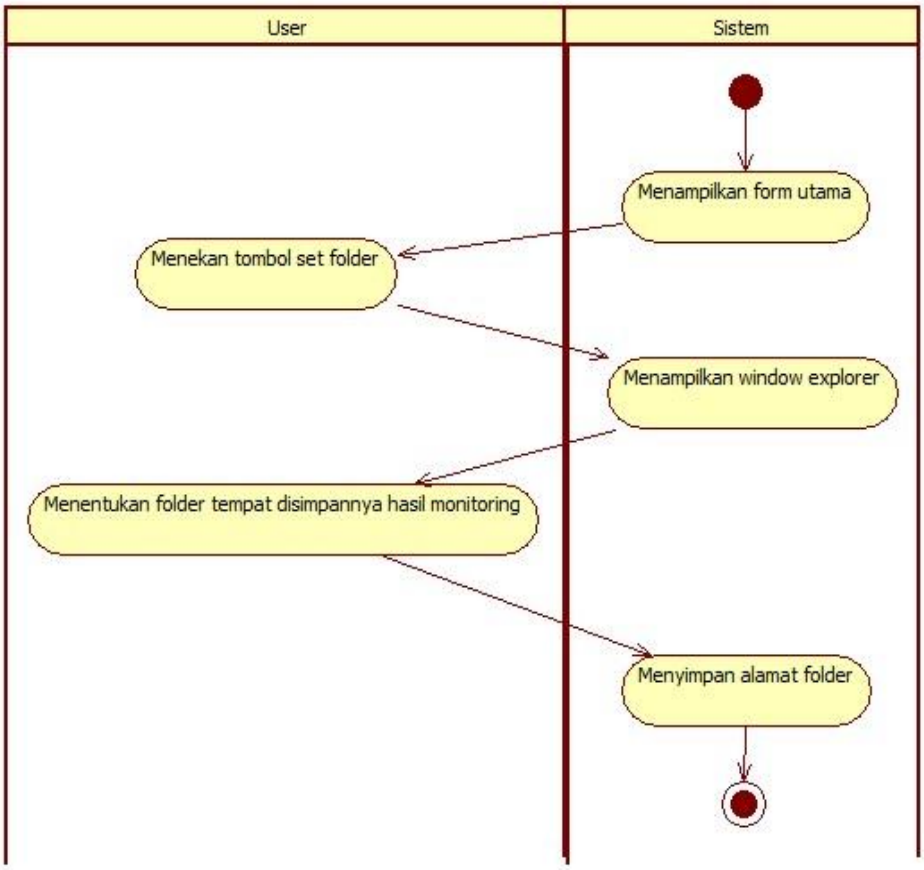
Gambar 3.11 *Acitvity* Diagram *Port* Kamera



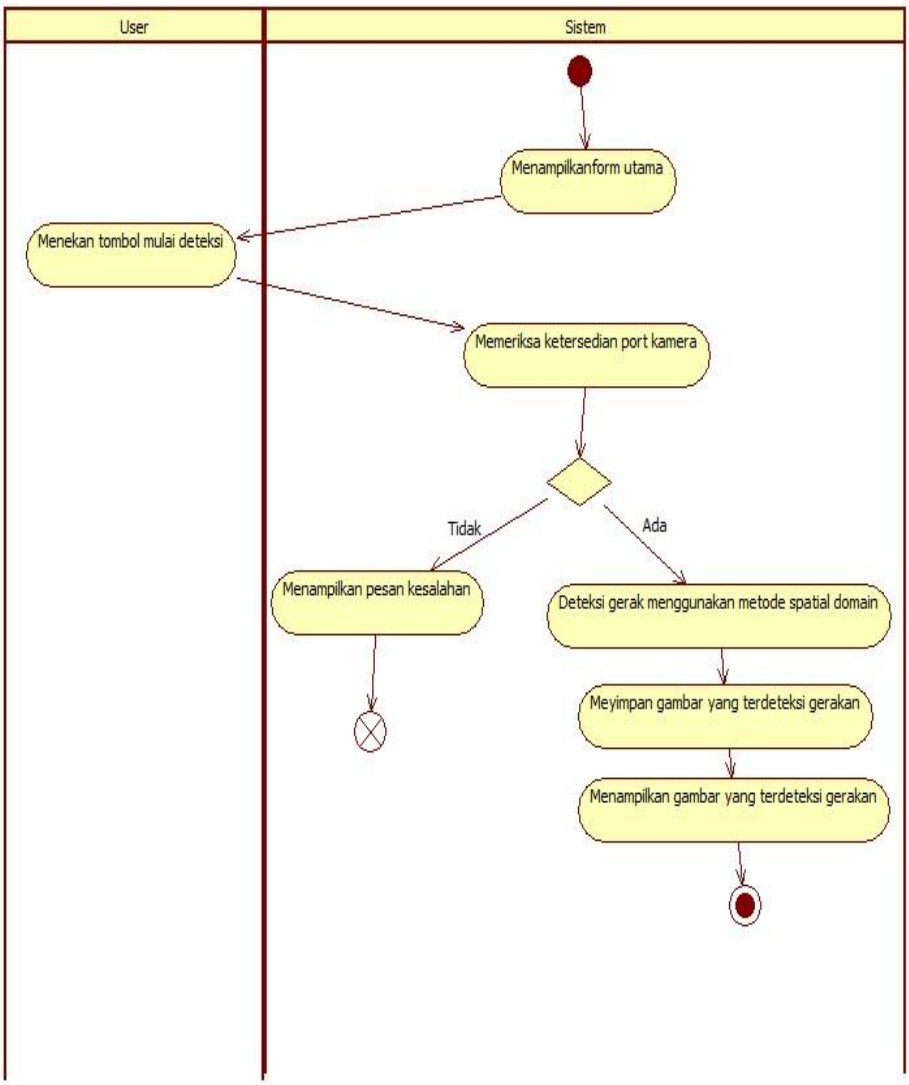
Gambar 3.12 *Activity* Diagram *Threshold*



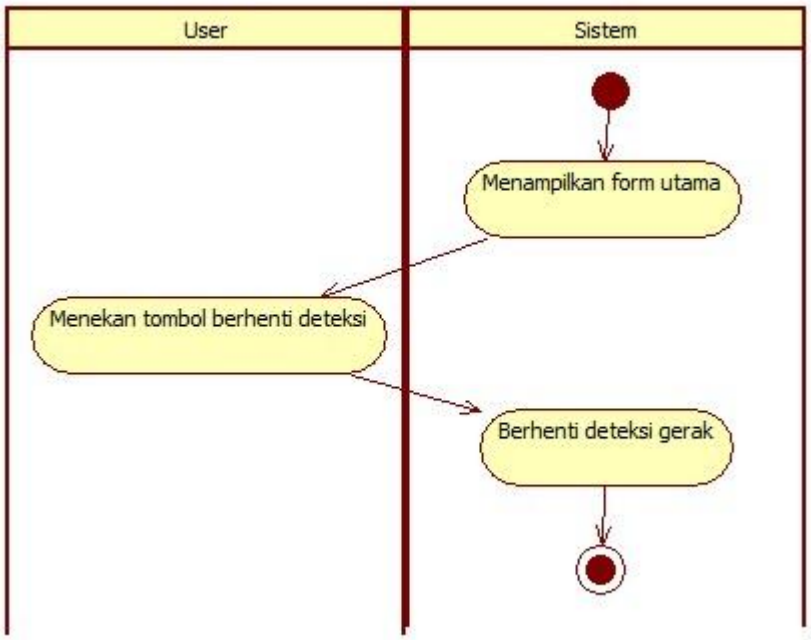
Gambar 3.13 *Activity* Diagram Waktu Tunggu



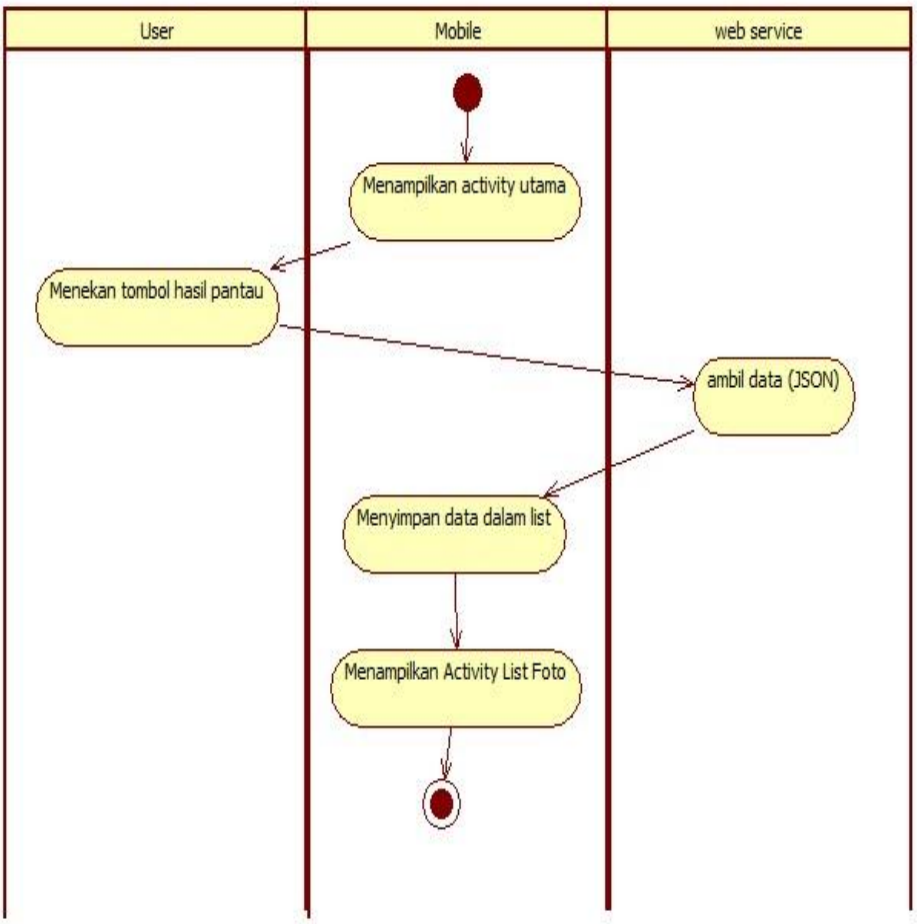
Gambar 3.14 *Activity* Diagram *Set Folder*



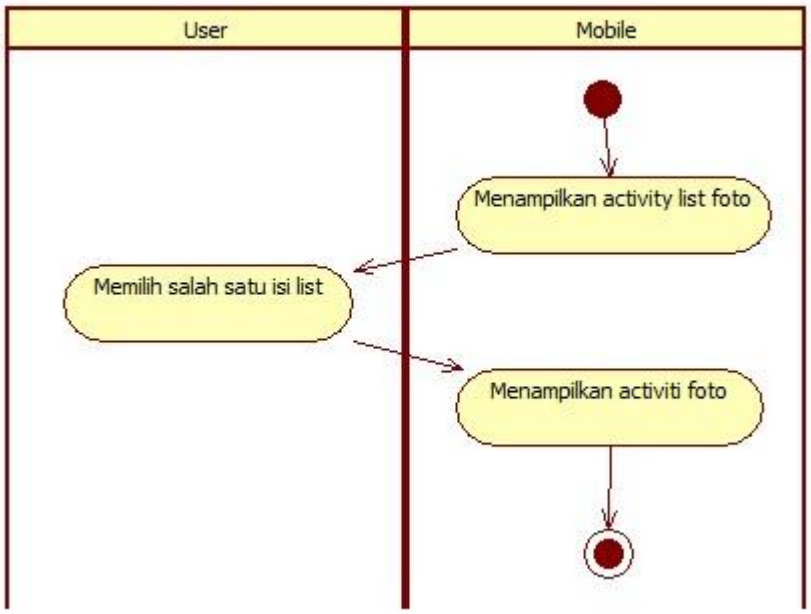
Gambar 3.15 *Activity* Diagram Mulai Deteksi



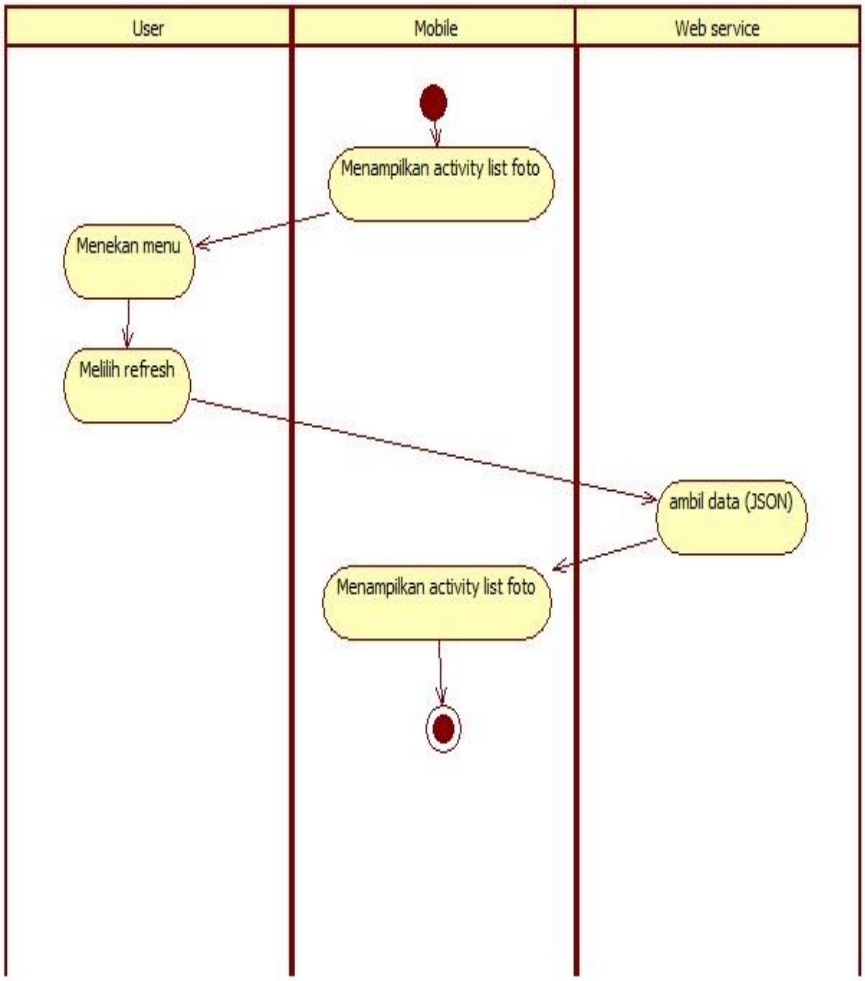
Gambar 3.16 *Activity* Diagram Berhenti Deteksi



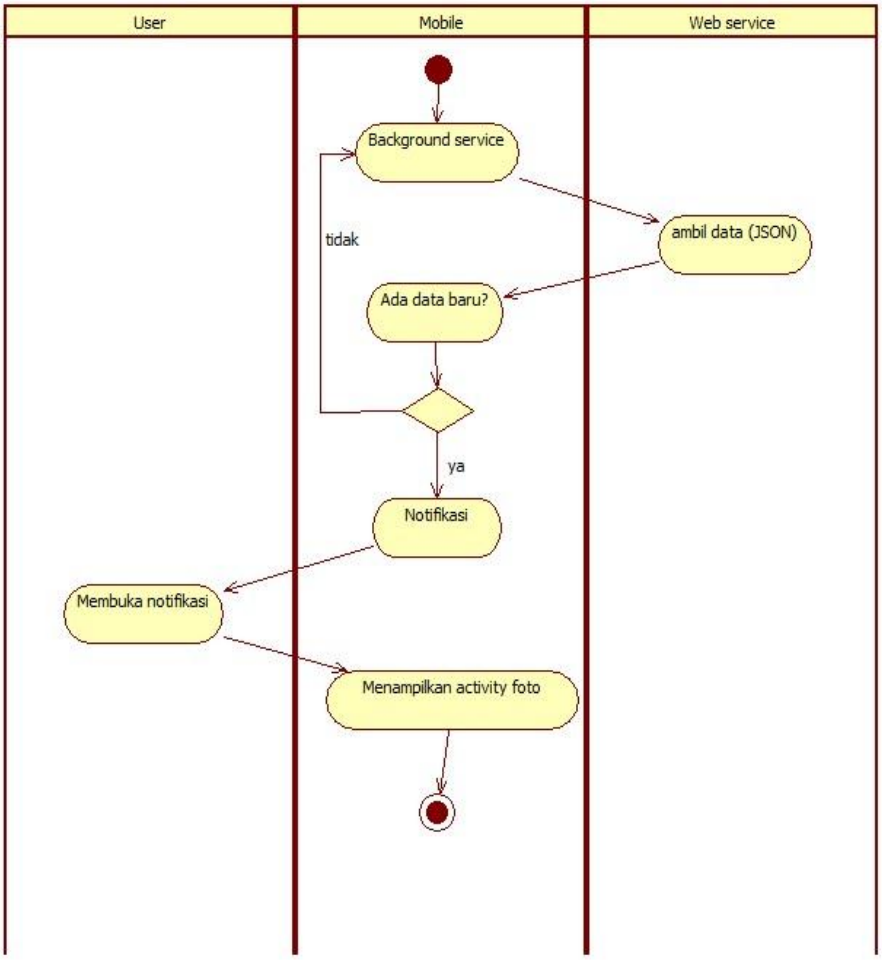
Gambar 3.17 *Activity* Diagram Hasil Pantau



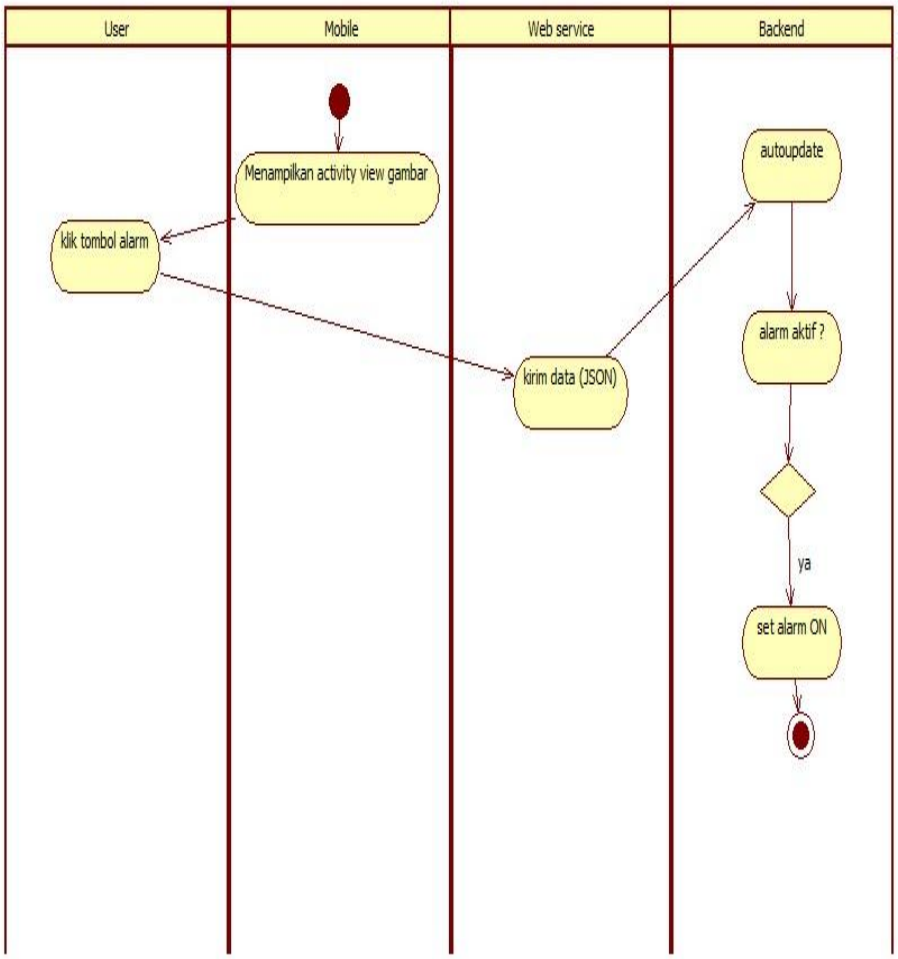
Gambar 3.18 *Activity* Diagram *List* Foto



Gambar 3.19 *Avtivity* Diagram Refresh



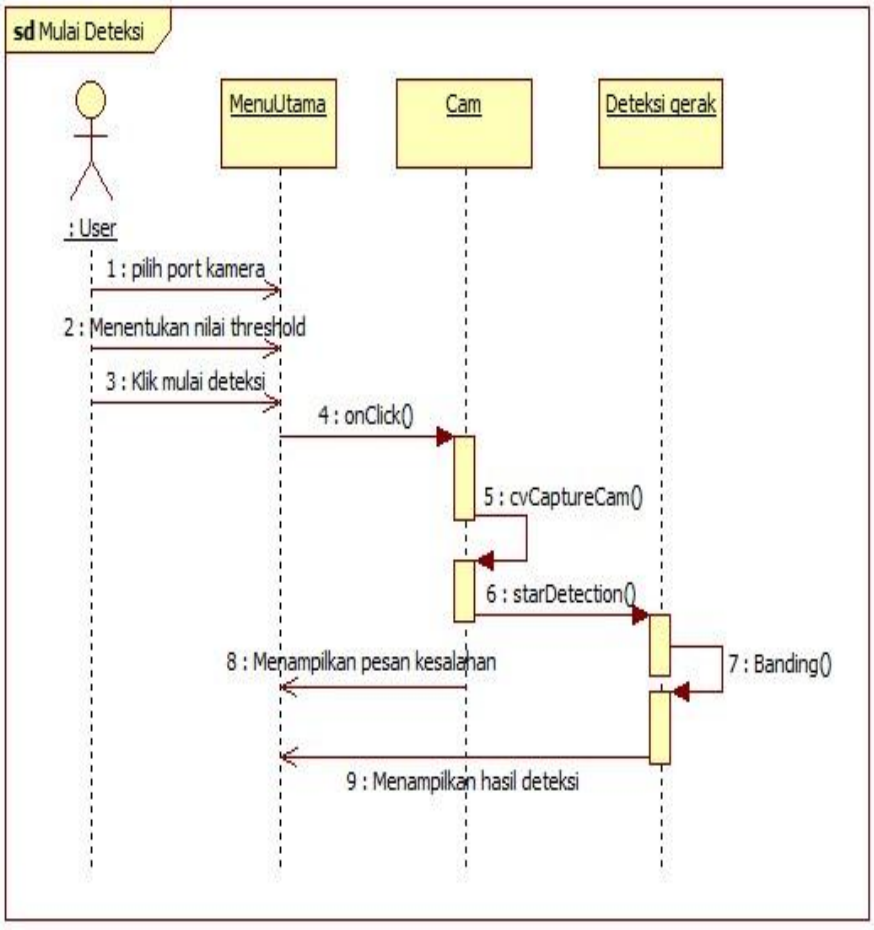
Gambar 3.20 *Activity* Diagram Notifikasi



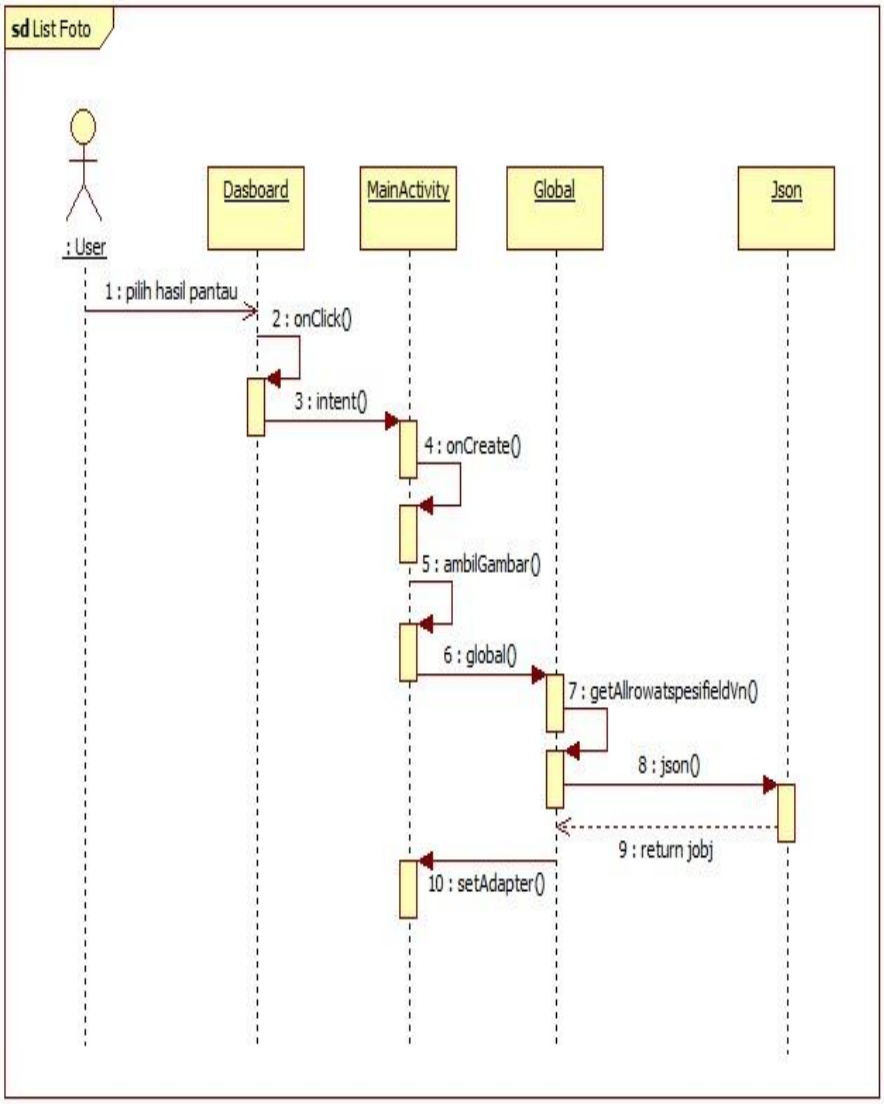
Gambar 3.21 *Activity* Diagram Alarm

5. *Sequence* Diagram

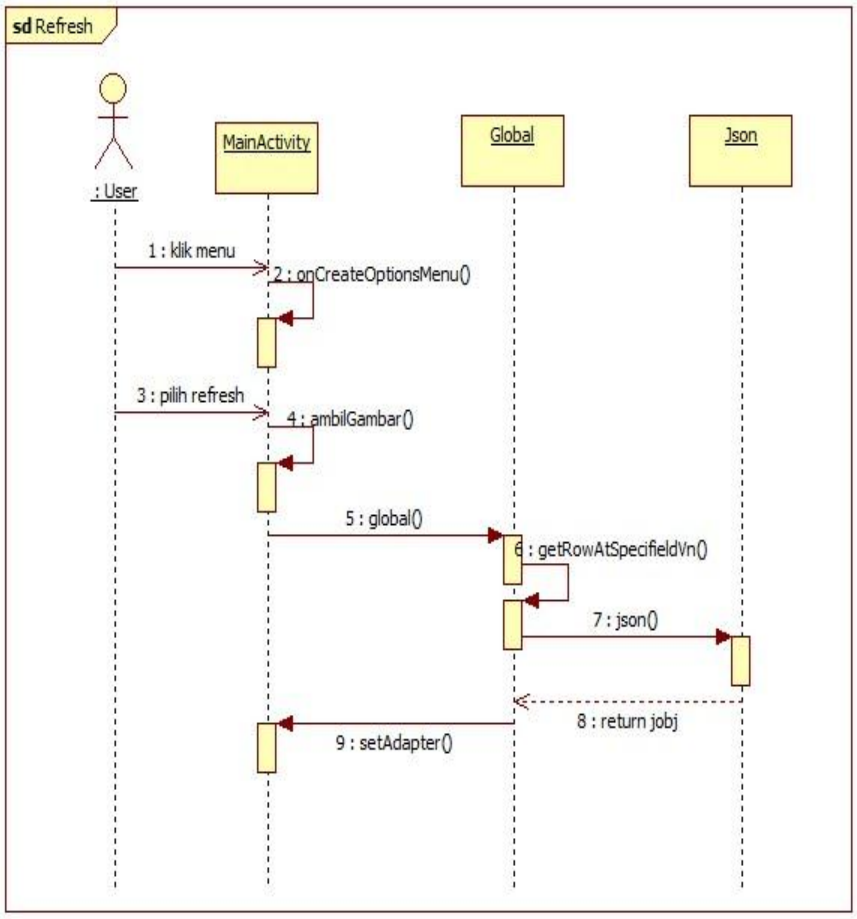
Sequence Diagram menjelaskan secara detail urutan proses yang dilakukan sistem untuk mencapai tujuan dari *Use Case*. Interaksi yang terjadi antara class, operasi apa saja yang terlibat, urutan antar operasi dan informasi yang diperlukan oleh masing – masing operasi.



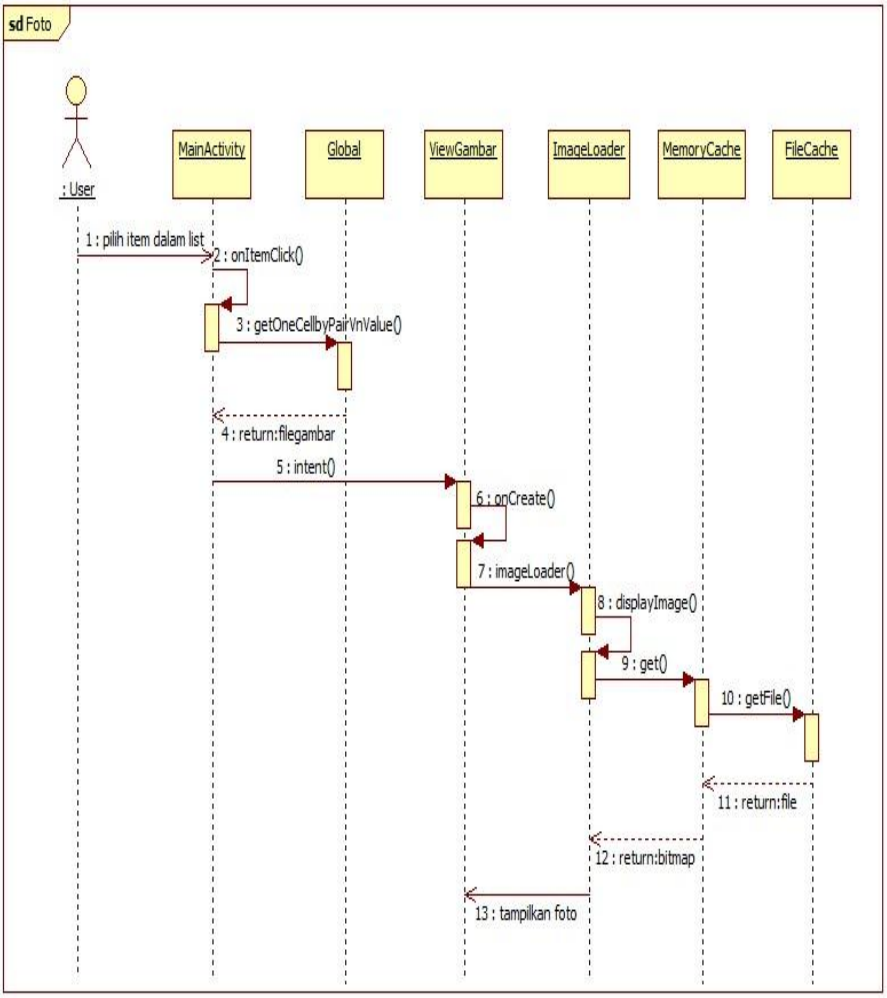
Gambar 3.22 *Suquence* Diagram Mulai Deteksi



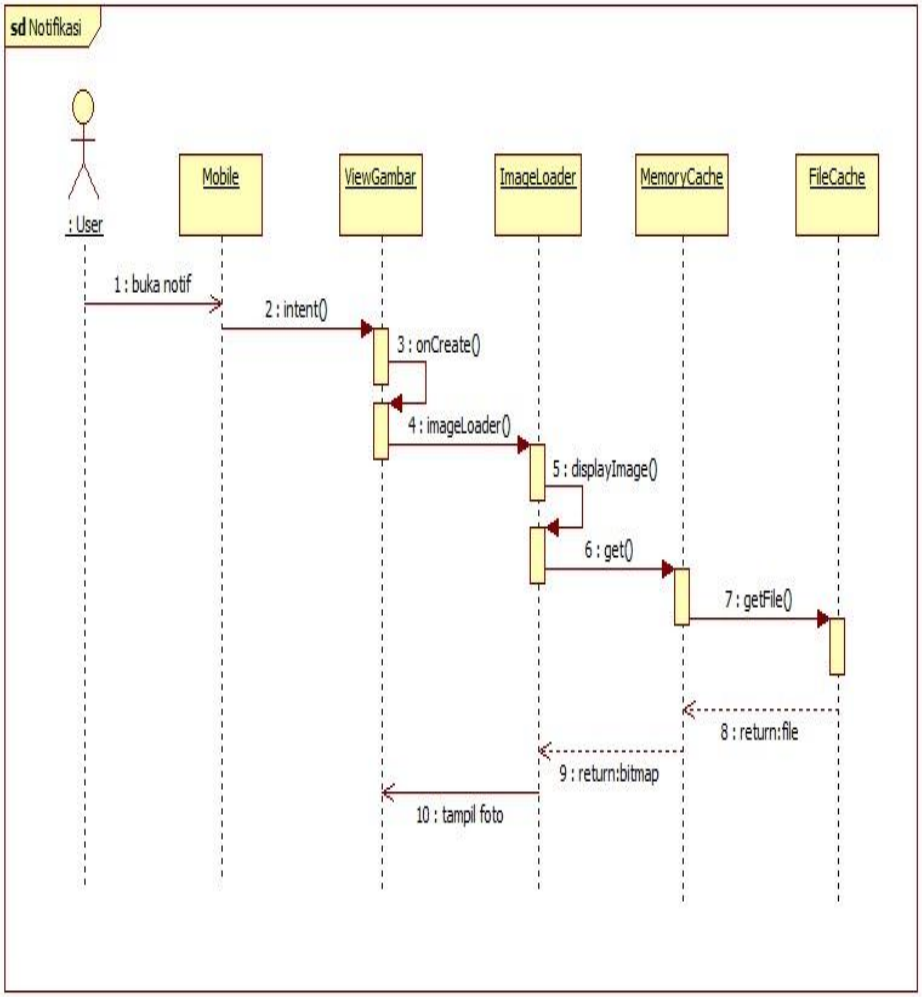
Gambar 3.23 *Sequence* Diagram *List* Foto



Gambar 3.24 *Sequence* Diagram Refresh

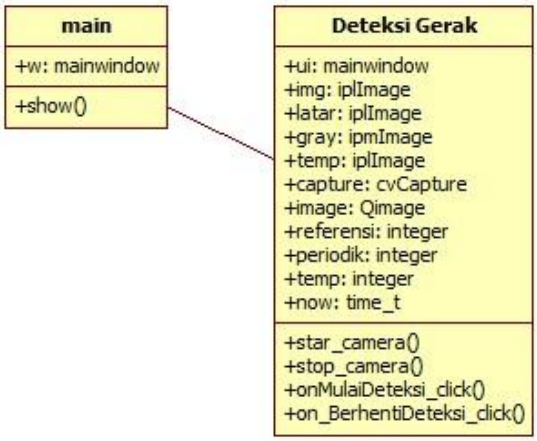


Gambar 3.25 *Sequence* Diagram Foto

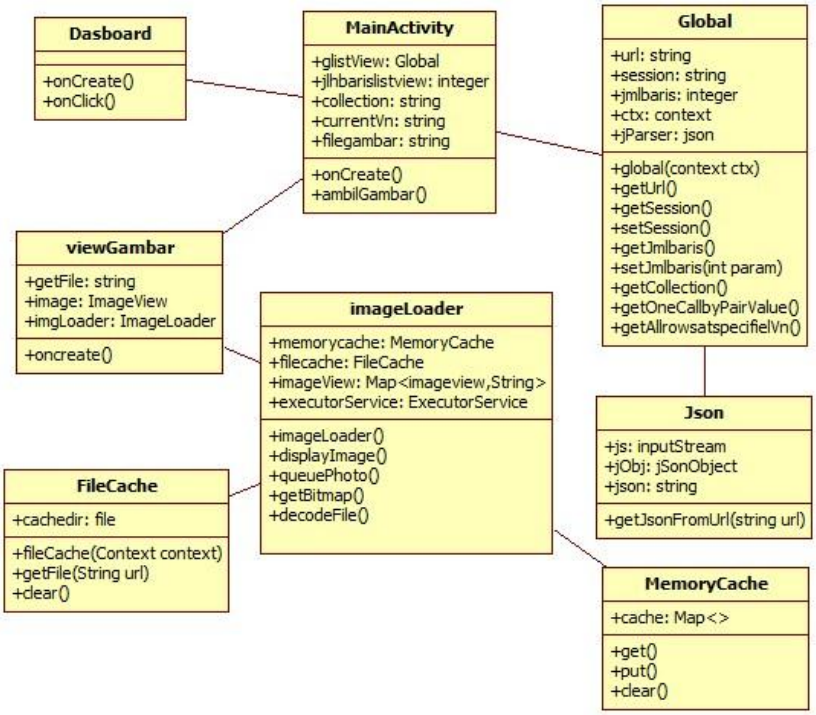


Gambar 3.26 *Sequence* Diagram Notifikasi

6. *Class Diagram*



Gambar 3.27 *Class* Diagram *Backend*



Gambar 3.28 *Class* Diagram *Frontend Mobile*

3.2 Perancangan Sistem

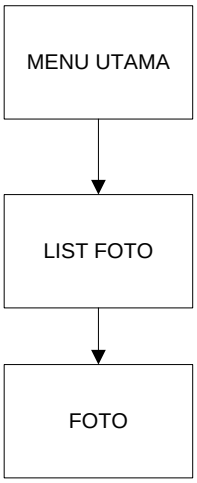
Perancangan sistem merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh. Tahapan ini meliputi mengkonfigurasi komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem. Adapun perancangan sistem dari perangkat lunak yang akan dibuat dijelaskan sebagai berikut.

3.2.1 Perancangan Arsitektur Perangkat Lunak

Perancangan arsitektur merupakan perancangan yang dibuat sebelum perangkat lunak dibuat. Perancangan arsitektur yang digunakan dalam perangkat lunak ini pada bagi *backend* adalah berbasis desktop, sedangankan pada bagian *frontend* terdapat 2 aplikasi yaitu berbasis web dan *mobile* android. Perancangan arsitektur menu dalam perangkat lunak ini adalah *sequential linear.*

3.2.2 Perancangan Struktur Menu

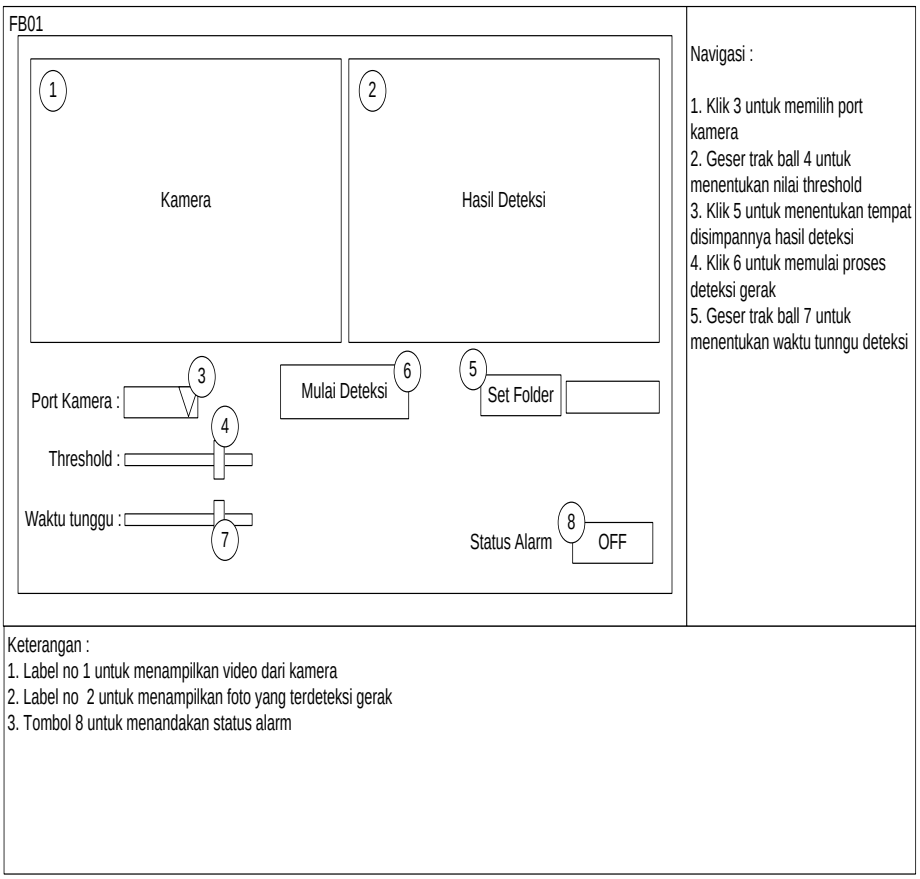
1. Perancangan Struktur Menu *Frontend mobile*



Gambar 3.29 Struktur Menu *Frontend Mobile*

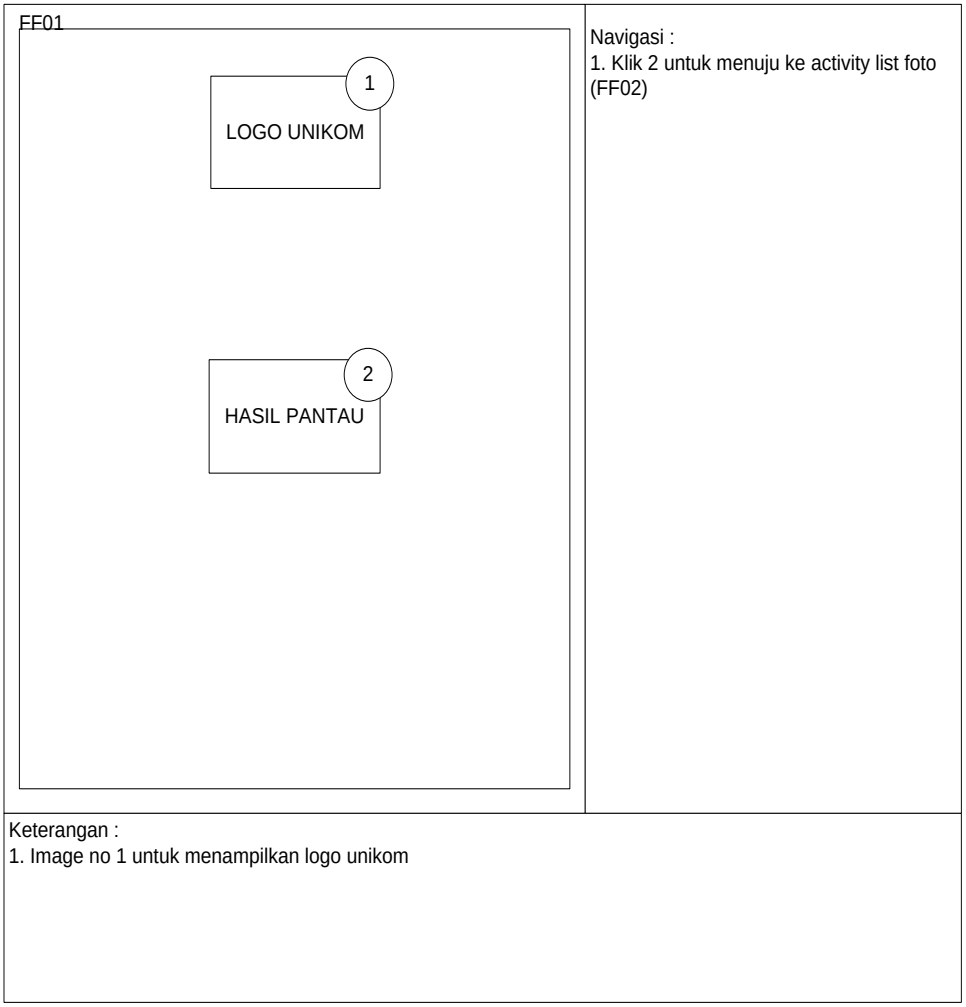
3.2.3 Perancangan Antarmuka Perangkat Lunak

1. Antarmuka menu utama *backend* (FB01)



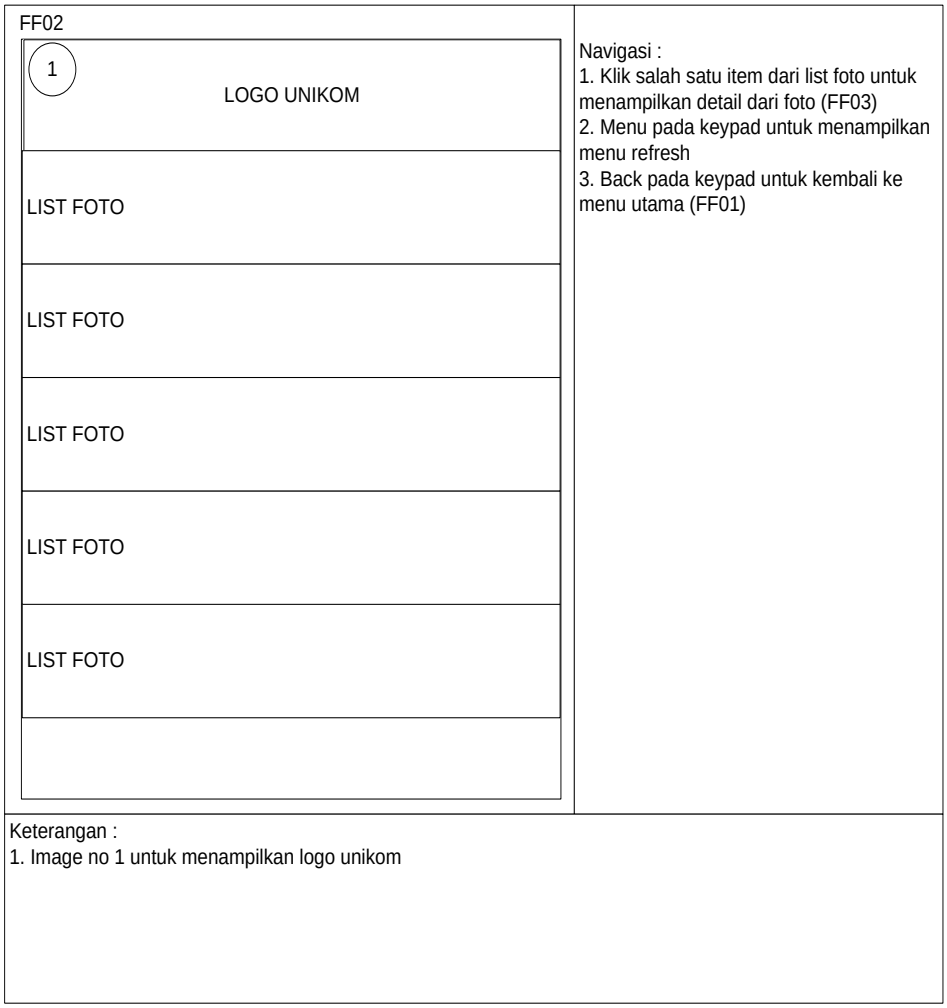
Gambar 3.30 *Form* Menu Utama *Backend* (FB01)

2. Antarmuka menu utama *frontend mobile* (FF01)



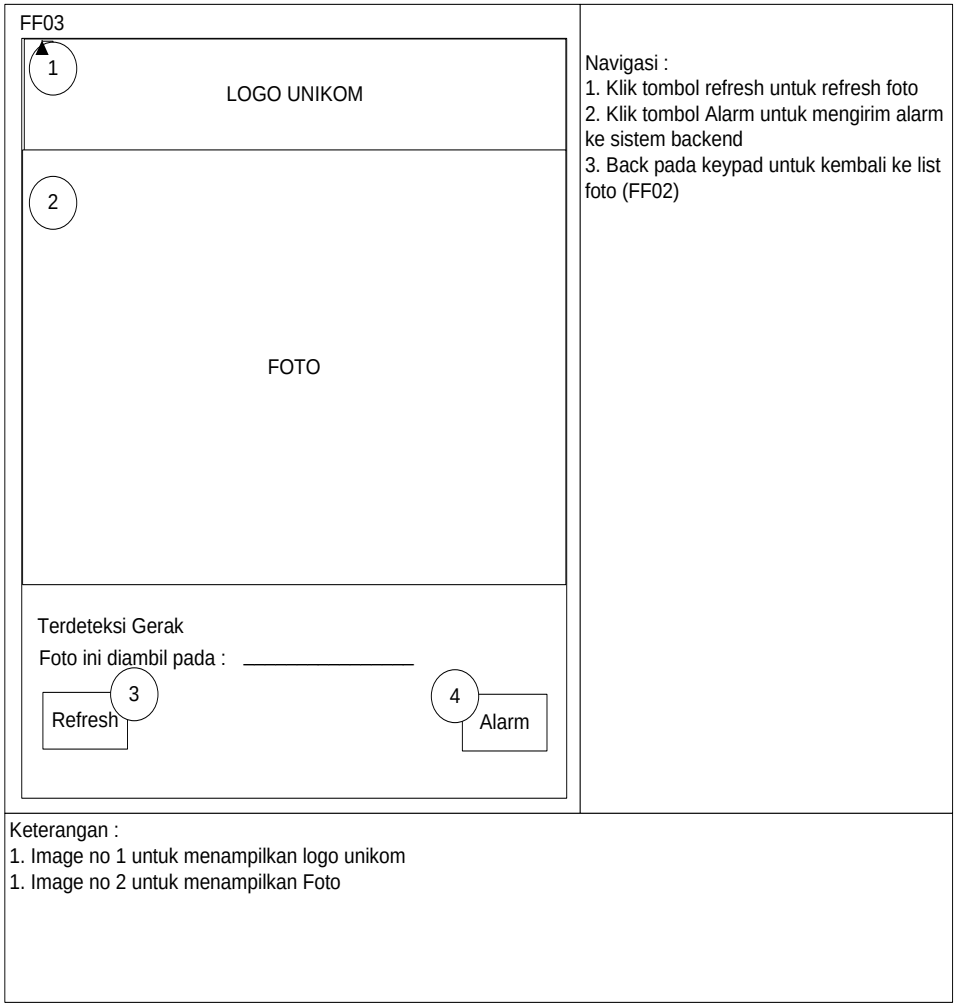
Gambar 3.31 *Form* Utama *Frontend Mobile* (FF01)

3. Antarmuka list foto *frontend mobile* (FF02)



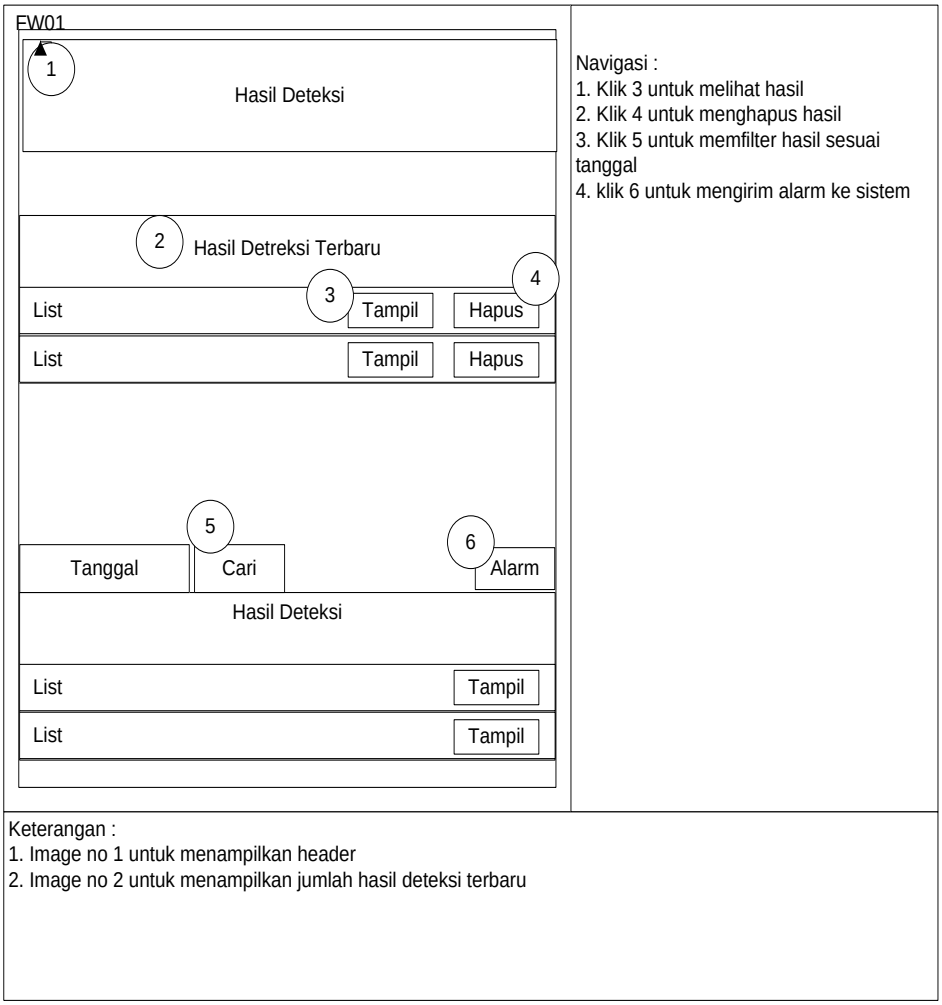
Gambar 3.32 *Form List* Foto *Frontend Mobile* (FF02)

4. Antarmuka foto *frontend mobile* (FF03)



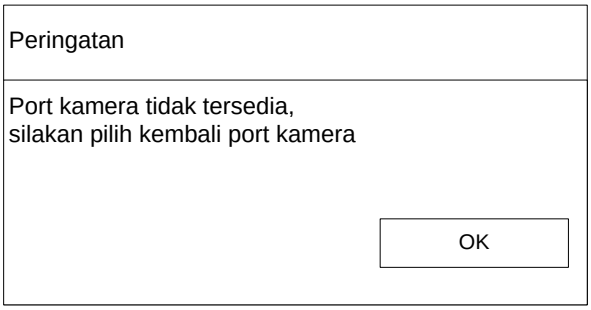
Gambar 3.33 *Form* Foto *Frontend Mobile* (FF03)

5. Antarmuka foto *frontend Web* (FW01)

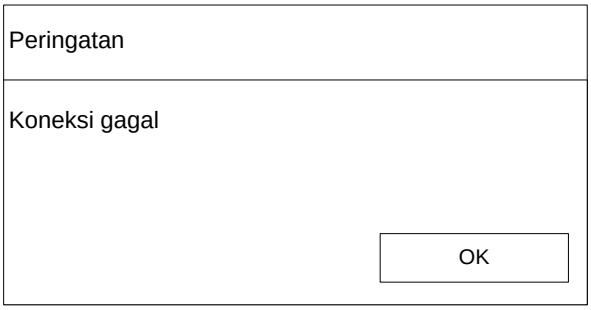


Gambar 3.34 *Form* Foto *Frontend Web* (FW01)

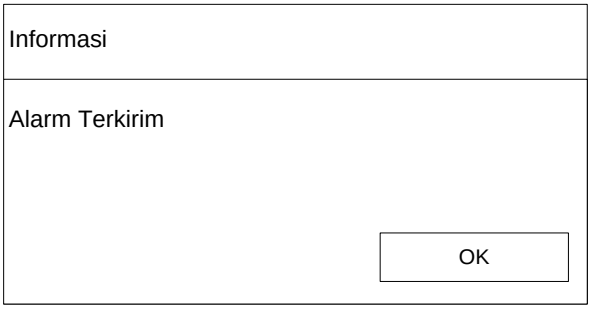
3.2.4 Perancangan Pesan



Gambar 3.35 Pesan MB01



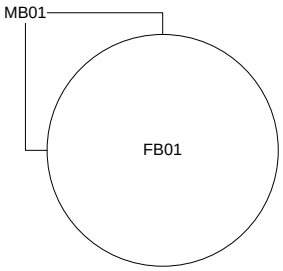
Gambar 3.36 Pesan MF01



Gambar 3.37 Pesan MFW01

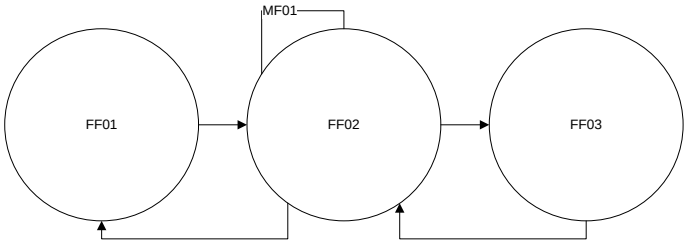
3.2.5 Jaringan Semantik

1. Jaringan Semantik *Backend*



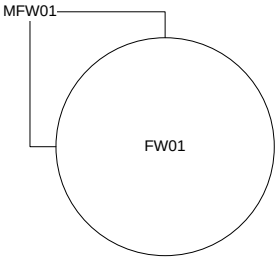
Gambar 3.38 Jaringan Semantik *Backend*

2. Jaringan Semantik *Frontend Mobile*



Gambar 3.39 Jaringan Semantik *Frontend Mobile*

3. Jaringan Semantik *Frontend Web*



Gambar 3.40 Jaringan Semantik *Frontend Web*