BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis sistem bertujuan untuk mengidentifikasi permasalahan-permasalahan yang terdapat pada sistem serta menentukan kebutuhan-kebutuhan dari sistem yang akan dibangun. Analisis tersebut meliputi analisis masalah, analisis arsitektur sistem, spesifikasi kebutuhan non fungsional, analisis kebutuhan non fungsional, analisis data, dan analisis kebutuhan fungsional.

3.1.1 Analisis Masalah

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

1. Hasil pengawasan masih belum akurat karena metode pengambilan gambar menggunakan waktu periodik.
2. Hasil pengawasan 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 dapat mengambil gambar dan merekam keadaan ruangan yang dipantau bila terjadi perubahan kondisi atau adanya gerakan pada rungan yang dipantau dengan menggunakan motode deteksi gerak, hasil pengawasan bisa diakses melalu 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. Deteksi gerakan yang digunakan diaplikasi pengawas pendeteksi gerakan ini adalah pemrosesan citra spatial domain dengan point processing. Motede ini berkerja dengan cara sebagai berikut :

a. Penetapan sebuah citra referensi

   Sebuah citra yang diambil oleh kamera pada saat situasi lingkungan yang stabil ditetapkan sebagai citra referensi. Citra ini kemudian disimpan sebagai acuan dalam membandingkan citra.

b. Pengambilan citra secara periodik

   Setiap jangka waktu tertentu, sebuah citra diambil oleh kamera. Citra ini kemudian disimpan untuk perhitungan selanjutnya.

c. Pendeteksi Gerak

   Citra yang diambil secara periodik tadi dibandingkan dengan referensi dengan cara membandingkan setiap pixel yang berada dilokasi yang sama. Perhitungan perbandingan dilakukan dengan cara mengambil nilai absolute dari hasil pengurangan nilai RGB dari 2 pixel yang berbeda dilokasi yang sama. Jika nilai dari perhitungan tersebut melewati suatu batas nilai tertentu maka pixel tersebut dinyatakan sebagai pixel yang terdeteksi gerakan.

$$f(x,y) = \begin{cases} 1 \; if \; T1 < ABS[p(x,y) - q(x,y)] \\ 0 \; if \; T1 \geq ABS[p(x,y) - q(x,y)] \end{cases}$$

Persamaan pendeteksi gerak pada metode spatial domain dimana :

f = *Array 2* dimensi yang digunakan untuk menyampaikan hasil dari perhitungan

x = Posisi *pixel* terhadap sumbu x.

y = Posisi *pixel* terhadap sumbu y.

T1 = Batas nilai untuk membatasi banyaknya perubahan nilai *pixel* tersebut dinyatakan terhadap gerakan (*Thresold).*

ABS = Fungsi absolute.

p = Citra referensi

q = Citra yang diambil secara periodik (citra yang dibandingkan).

Variabel T1 merupakan sebuah nilai batas ambang perubahan 2 buah pixel diletak yang sama. Penggunaan variabel T1 ini memiliki dua tujuan. Tujuan yang pertama adalah untuk mengeliminasi noise yang terjadi, sedangkan tujuan yang kedua adalah untuk memberikan batas ambang perubahan intensitas warna antara dua buah pixel. Noise terjadi karna suatu lingkungan yang pencahayaannya labil. Hal ini dapat terjadi karena intensitas cahaya yang datang tidak selalu sama dari waktu ke waktu.

Sebuah array boolean f(x,y) digunakan untuk menyimpan nilai dari perhitungan. Jika nilai f(x,y) bernilai true berarti dua buat pixel dari 2 buah citra yang berbeda dengan letak yang sama terjadi perubahan warna yang melewati batas ambang T1. Sebaliknya, jika f(x,y) bernilai false berarti kedua buah pixel tersebut tidak terjadi perubahan warna atau perubahan yang terjadi tidak melewati batas ambang T1. Array ini digunakan untuk menghitung jumlah pixel yang berubah. Jika jumlah pixel yang berubah melebihi nilai batas ambang T2 maka citra tersebut dikatakan terdapat gerakan.

$$res = f(x) = \begin{cases} 1 \; if \; T2 < \sum_x \sum_y f(x,y) \\ 1 \; if \; T2 \geq \sum_x \sum_y f(x,y) \end{cases}$$

Persamaan perhitungan jumlah pixel pada metode spatial domain dimana :

res = Hasil deteksi gerakan, bernilai 1 jika terdapat gerakan, dan bernilai 0 jika tidak terdapat gerakan.

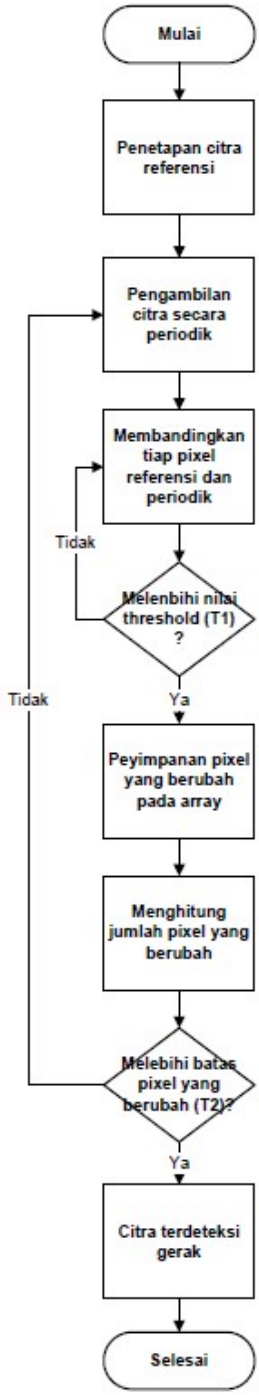
T2 = Batas nilai untuk membatasi banyak jumlah pixel yang berubah sebelum sebuah citra dinyatakan terdapat gerakan (*Thresold).*

x = Posisi *pixel* terhadap sumbu x.

y = Posisi *pixel* terhadap sumbu y

f = *Array* hasil perhitungan pendeteksi citra.

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



Gambar 3. 1 Proses Pendeteksian Gerak

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

a. Penetapan sebuah citra referensi

Citra referensi adalah sebuah gambar yang diambil oleh kamera pada saat situasi lingkungan yang stabil dan ditetapkan sebagai gambar awal, Citra ini kemudian diubah menjadi citra biner dan disimpan sebagai acuan dalam membandingkan.



Gambar 3. 2 Tahap Citra Referensi

b. Pengambilan citra secara periodik

Citra periodik adalah gambar yang diambil dalam setiap jangka waktu tertentu oleh kamera secara terus menerus. Citra ini pun kemudian diubah menjadi citra biner dan disimpan untuk perhitungan selanjutnya.

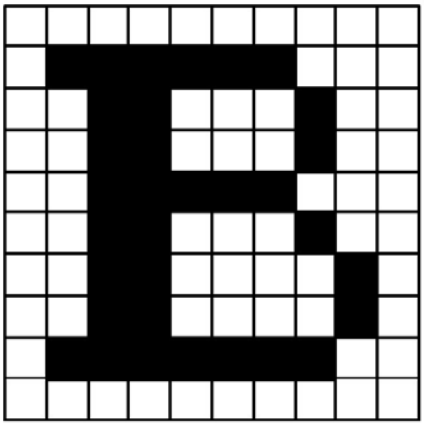


Gambar 3. 3 Tahap Citra Periodik

c. Pendeteksi Gerakan

Dari citra referensi dan citra periodik yang sudah disimpan, maka proses selanjutnya adalah membandingkan kedua citra biner tersebut dengan

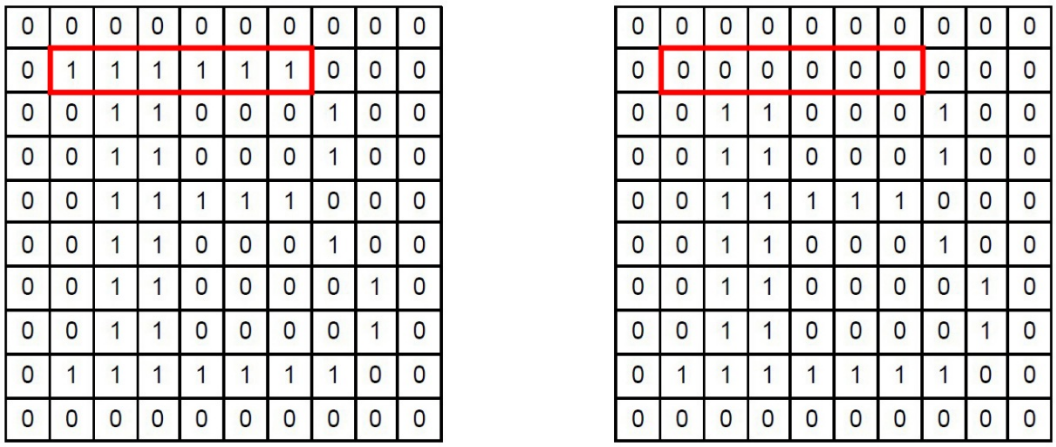
mengambil nilai piksel dari tiap titik yang sama, untuk mempermudah proses perbandingan nilai piksel, nilai piksel dari kedua citra biner tersebut direpresentasi derajat keabuannya seperti contoh dibawah :



0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0	0	0
0	0	1	1	0	0	0	1	0	0
0	0	1	1	0	0	0	1	0	0
0	0	1	1	1	1	1	0	0	0
0	0	1	1	0	0	0	1	0	0
0	0	1	1	0	0	0	0	1	0
0	0	1	1	0	0	0	0	1	0
0	1	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0

Gambar 3. 4 Representasi Citra Biner Dari Derajat Keabuannya

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



0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0	0	0
0	0	1	1	0	0	0	1	0	0
0	0	1	1	0	0	0	1	0	0
0	0	1	1	1	1	1	0	0	0
0	0	1	1	0	0	0	1	0	0
0	0	1	1	0	0	0	0	1	0
0	0	1	1	0	0	0	0	1	0
0	1	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0

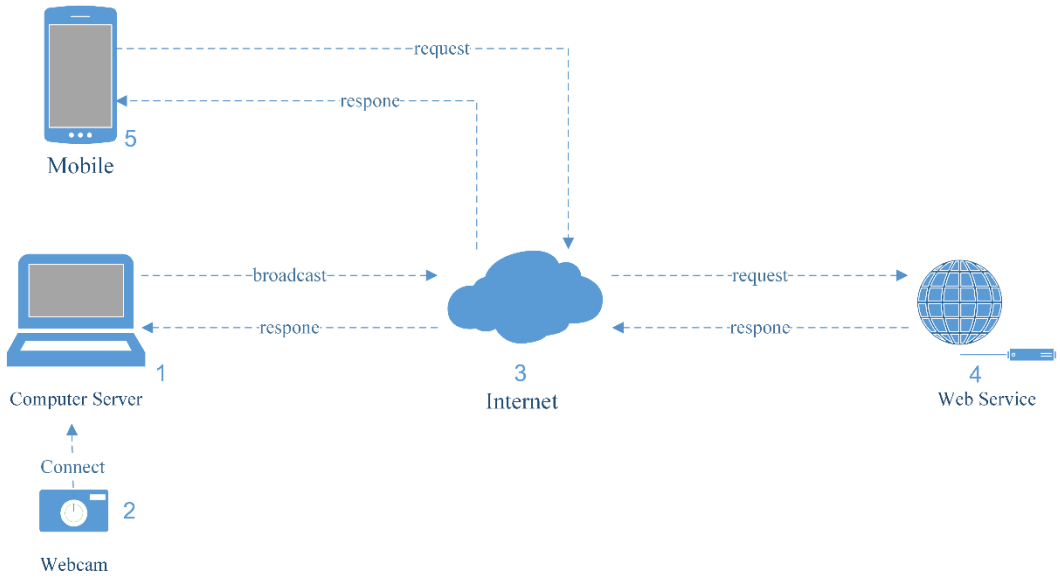
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0
0	0	1	1	0	0	0	1	0	0
0	0	1	1	1	1	1	0	0	0
0	0	1	1	0	0	0	1	0	0
0	0	1	1	0	0	0	0	1	0
0	0	1	1	0	0	0	0	1	0
0	1	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0

Gambar 3. 5 Proses Perbandingan Citra

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

3.1.4 Analisis Arsitektur Sistem

Analisis arsitektur sistem bertujuan untuk digunakan untuk menganalisis perangkat lunak yang dibangun dalam bentuk gambaran sistem secara umum. Arsitektur sistem dapat dilihat pada Gambar 3.1.



Gambar 3. 6 Arsitektur Sistem

Gambaran arisitektur sistem yang akan dibangun memiliki dua buah subsistem. Pertama adalah *desktop* dimana pada bagian ini terdapat laptop yang mempunyai *webcam* dan terhubung dengan internet sehingga dapat merekam dan mengirimkan hasil rekamannya ke *server*. Sedangkan subsistem kedua adalah subsistem *mobile* dimana pada bagian ini aplikasi akan mengirim dan menerima hasil rekaman dari *server*. Subsistem mobile melakukan pertukaran data ke *server* melalui internet kemudian masuk ke *web service*.

3.1.5 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak terbagi kedalam dua kebutuhan yaitu kebutuhan non fungsional dan fungsional. Kebutuhan fungsional dapat dilihat pada tabel 3.1 dan tabel 3.2

Tabel 3. 1 Spesifikasi Kebutuhan Perangkat Lunak *Desktop*

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-001	Subsistem *desktop* menyediakan fasilitas kepada pengguna untuk mengelola pengaturan *webcam.*
SKPL-F-002	Subsistem *desktop* menyediakan fasilitas kepada pengguna untuk melihat *video* yang sedang ditampilkan oleh *webcam.*
SKPL-F-003	Subsistem *desktop* menyediakan fasilitas kepada pengguna untuk mengelola *alarm.*
SKPL-F-004	Subsistem *desktop* menyediakan fasilitas kepada pengguna untuk mengelola *motion detection.*
SKPL-F-005	Subsistem *desktop* menyediakan fasilitas kepada pengguna untuk mengelola *motion sensitivity.*

Tabel 3. 2 Spesifikasi Kebutuhan Perangkat Lunak *Mobile*

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-007	Subsistem *mobile* menyediakan fasilitas bagi pengguna untuk mamantau kosan secara *streaming*
SKPL-F-008	Subsistem *mobile* menyediakan fasilitas bagi pengguna untuk mengambil gambar yang sedang ditampilkan secara *streaming.*
SKPL-F-009	Subsistem *mobile* menyediakan fasilitas bagi pengguna untuk melihat foto yang telah diambil.
SKPL-F-010	Subsistem *mobile* menyediakan fasilitas bagi pengguna untuk mengatur notifikasi.

Kebutuhan non fungsional pada perangkat lunak yang dibangun dapat dilihat pada tabel 3.3.

Tabel 3. 3 Spesifikasi Kebutuhan Non Fungsional

Kode SKPL	Spesifikasi Kebutuhan Non Fungsional
SKPL-NF-001	Sistem yang dibangun memiliki dua subsistem yaitu *desktop* dan *mobile*
SKPL-NF-002	Sistem yang dibangun minimal menggunakan sistem operasi android versi 4.1 *Jelly Bean.*
SKPL-NF-003	Sistem yang dibangun membutuhkan *hardware* laptop dan *webcam*
SKPL-NF-004	Sistem dibangun dengan spesifikasi *hardware* yang memenuhi standar minimum kebutuhan
SKPL-NF-005	Desain *user interface* pada perangkat lunak yang akan dibangun mengacu kepada desain dari aplikasi pemantau CCTV android saat ini.
SKPL-NF-006	Sistem tidak mengizinkan konten yang mengandung unsur sara, norma-norma kesusilaan dan tidak melanggar unsur hak kekayaan atas intelektual.

3.1.6 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional menggambarkan kebutuhan sistem yang akan mempengaruhi perilaku pada sistem, diantaranya kebutuhan perangkat keras dan perangkat lunak sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan ditetapkan.

3.1.6.1 Analisis kebutuhan perangkat keras

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

a. Perangkat keras subsistem *desktop*

   Pada subsistem *desktop*, aplikasi dijalankan pada sebuah perangkat komputer atau laptop dengan spesifikasi kebutuhan perangkat keras komputer pada Tabel 3.4.

Tabel 3. 4 Spesifikasi Kebutuhan Perangkat Keras Subsistem *Desktop*

Jenis Perangkat Keras	Kebutuhan Minimal
Processor	1.0 GHz
Memory	1 GB
Harddisk	5 GB
Webcam	1.3 Megapixel
Layar	1366 x 768 pixels
Perangkat Lain	Mouse, keyboard, koneksi Internet

b. Perangkat keras subsistem *Mobile*

   Subsistem *mobile* dijalankan pada sebuah perangkat *smartphone* dengan sistem operasi Android. Kebutuhan perangkat keras untuk *smartphone* ditunjukan pada tabel 3.5

Tabel 3. 5 Spesifikasi Kebutuhan Perangkat Keras Subsistem *Mobile*

Jenis Perangkat Keras	Kebutuhan Minimal
Layar	540 x 960 pixels
Memory	384 MB
Processor	800 MHz
Perangkat Lain	Koneksi Internet

3.1.6.2 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak merupakan proses analisis yang lebih menekankan kepada aspek pemanfaatan software. Kebutuhan minimal spesifikasi perangkat lunak pada laptop atau komputer dan perangkat *smartphone* yang akan menggunakan perangkat lunak ini dapat dilihat pada tabel 3.6.

Tabel 3. 6 Spesifikasi Minimum *Software*

Subsistem *Desktop*	Subsistem *Mobile*
Sistem Operasi Windows 7 Professional.	Sistem Operasi Android 4.1 Jelly Bean

Sedangkan dari sisi *developer*, perangkat lunak yang digunakan untuk membangun sistem ini adalah sebagai berikut:

1. Subsistem *Desktop*
   a. Sistem operasi Microsoft Windows 7 Professional.
   b. Microsoft Visual Studio C# 2010
2. Subsistem *Mobile*
   a. Sistem operasi Microsoft Windows 7 Professional.
   b. JDK (*Java Development Kit*) versi 7.
   c. Android SDK untuk Android versi 4.2 (API 17).
   d. ADT (*Android Developer Tools*) versi 21.0.1.

3.1.6.3 Analisis Pengguna Sistem (*User)*

Analisis pengguna sistem dimaksudkan untuk mengetahui siapa aktor yang terlibat dalam menjalankan sistem. Hal ini bertujuan untuk mengetahui karakteristik pengguna dalam mengoprasikan sistem yang telah ada, maupun sistem usulan.

Berdasarkan hasil pengamatan, pengguna yang terlibat dalam sistem yang sedang berjalan adalah sebagai berikut:

1. Pengguna subsistem *desktop*. Pengguna ini adalah individu
2. Pengguna subsistem *mobile*. Pengguna ini adalah individu yang sama dengan pengguna subsistem *desktop*

Untuk menggunakan sistem yang dibangun terdapat beberapa karakteristik yang harus dimiliki oleh pengguna. Karakteristik pengguna subsistem *desktop* ditunjukkan pada Tabel 3.7, sedangkan karakteristik sub sistem *mobile* ditunjukkan pada Tabel 3.8.

Tabel 3. 7 Karakteristik Pengguna Subsistem *Desktop*

Kategori Pengguna	Hak Akses	Tingkat Pendidikan	Tingkat Keterampilan
Individu	Mengelola pengaturan webcam, pengaturan alarm, pengaturan motion dan pengaturan web service	Minimal SMA	Minimal pernah menjalankan aplikasi pada komputer atau laptop

Tabel 3. 8 Karakteristik Pengguna Subsistem *Mobile*

Kategori Pengguna	Hak Akses	Tingkat Pendidikan	Tingkat Keterampilan
Individu	Mengambil gambar, melihat gambar, mengelola alarm dan mengelola notifikasi	Minimal SMA	Minimal pernah menjalankan aplikasi pada *mobile* Android.

3.1.7 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 dilakukan menggunakan UML (Unified Modelling Language) dimana tahap-tahap pemodelan dalam analisis tersebut antara lain use case diagram, identifikasi use case, scenario use case, activity diagram, sequence diagram dan class diagram.

3.1.7.1 Analisis Kebutuhan Fungsional Subsistem *Desktop*

1. *Use Case Diagram*

Use case diagram 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*.

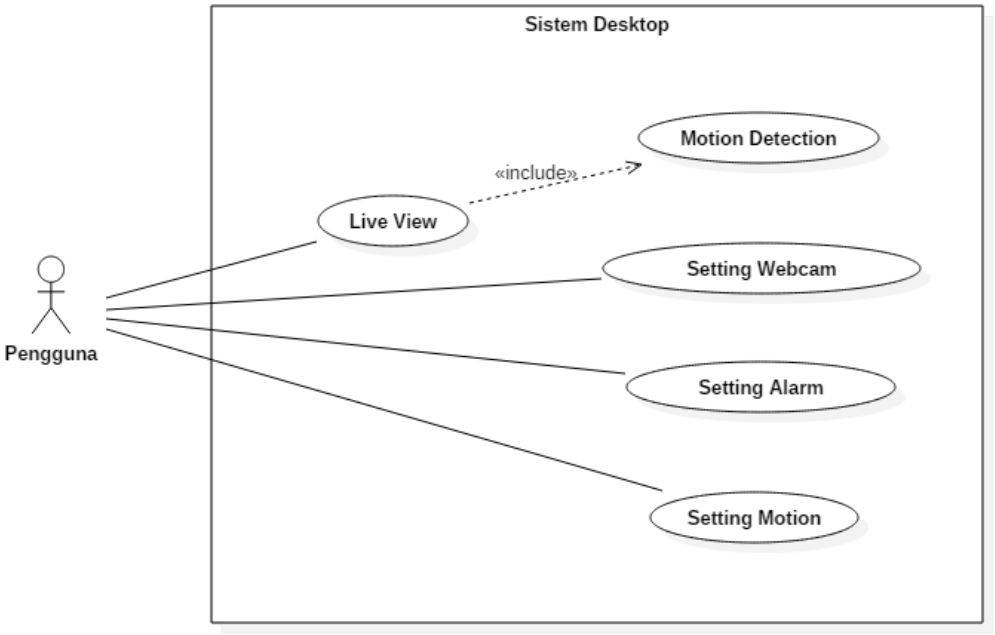
a. Identifikasi Aktor

Aktor adalah abstraksi dari orang dan sistem yang mengaktifkan fungsi dari target sistem. Pada Tabel 3.7 dijelaskan aktor-aktor yang berperan dalam menjalankan sistem yang dibangun.

Tabel 3. 9 Identifikasi Aktor

No.	Aktor	Deskripsi
1.	Pengguna	Merupakan aktor yang dapat berperan sebagai orang yang memakai aplikasi.

Gambar 3.2 merupakan penggambaran *use case diagram* untuk aplikasi yang akan dibangun. *Use case diagram* tersebut menunjukkan bagaimana interaksi antara aktor dan sistem, dimana terdapat satu aktor yang berperan, yaitu pengguna.



Gambar 3. 7 *Use Case Diagram Desktop*

b. Identifikasi *Use Case*

Identifikasi *use case* di dalam sistem dapat dilihat pada Tabel 3.8.

Tabel 3. 10 Identifikasi *Use Case*

No.	*Use Case*	Deskripsi
1.	*Live View*	Fungsionalitas ini digunakan oleh pengguna untuk melihat video yang sedang ditampilkan oleh *webcam* pada sistem.
2.	*Motion Detection*	Fungsionalitas ini digunakan oleh pengguna untuk melihat pergerakan yang sedang aktif.
4.	*Setting Webcam*	Fungsionalitas ini digunakan oleh pengguna untuk mengelola webcam yang terhubung.
5.	*Setting Alarm*	Fungsionalitas ini digunakan oleh pengguna untuk mengatur suara *alarm.*
6.	*Setting Motion*	Fungsionalitas ini digunakan oleh pengguna untuk mengatur sensitifitas pergerakan yang aktif.

c. Skenario *Use Case*

Skenario proses-proses yang terdapat didalam use case dapat dilihat pada tabel-tabel berikut.

Tabel 3. 11 Skenario *Use Case Live View*

Use Case Name	Live view	
Goal In Context	Menampilkan tayangan video secara langsung	
Precoditions	*Pengguna telah membuka halaman utama*	
Succesful End Condition	Berhasil menayangkan video secara langsung	
Failed End Condition	Gagal menayangkan video secara langsung	
Primary Actors	Pengguna	
Secondary Actors	-	
Trigger	Pengguna mengeklik tombol Live View	
Main Flow	***Step***	***Action***
	1	Pengguna mengeklik tombol Live View
	2	Sistem mengambil video yang diambil dari webcam
	3	Sistem menampilkan video secara langsung

Tabel 3. 12 Skenario *Use Case Motion Detection*

Use Case Name	Motion detection	
Goal In Context	Mendeteksi pergerakan yang ditangkap oleh kamera	
Precoditions	*Pengguna telah melakukan konfigurasi webcam*	
Succesful End Condition	Sistem menangkap pergerakan	
Failed End Condition	Sistem tidak menangkap pergerakan	
Primary Actors	Pengguna	
Secondary Actors	-	
Trigger	-	
Main Flow	***Step***	***Action***
	1	Pengguna mengaktifkan fitur motion detection
	2	Sistem mendeteksi pergerakan yang ditangkap kamera
	3	Sistem menangkap pergerakan
		Sistem memberikan pemberitahuan

Tabel 3. 13 Skenario *Use Case Setting Webcam*

Use Case Name	Setting Webcam	
Goal In Context	Melakukan konfigurasi untuk webcam yang akan dipakai	
Precoditions	*Pengguna telah membuka menu setting*	
Succesful End Condition	Berhasil terhubung dengan webcam	
Failed End Condition	Gagal terhubung dengan webcam	
Primary Actors	Pengguna	
Secondary Actors	-	
Trigger	Pengguna menyentuh menu Webcam Configuration	
Main Flow	***Step***	***Action***
	1	Pengguna melakukan konfigurasi untuk webcam
	2	Sistem berhasil terhubung dengan webcam
Extensions	***Step***	***Branching Action***
	2.1	Sistem gagal terhubung dengan webcam

Tabel 3. 14 Skenario *Use Case Setting Alarm*

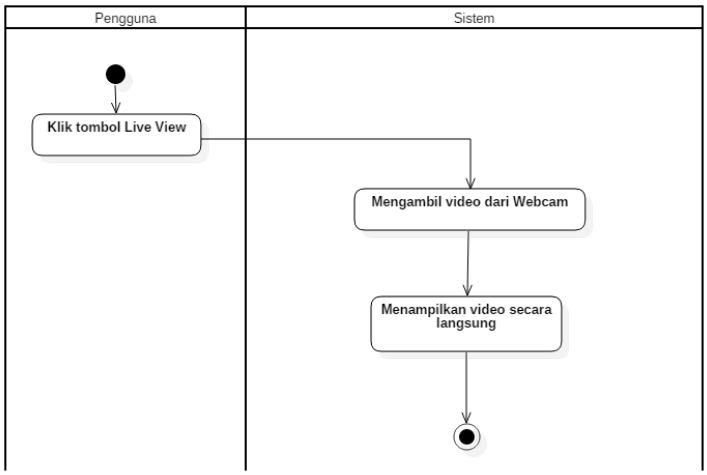
Use Case Name	Setting alarm	
Goal In Context	Mengubah pengaturan alarm	
Precoditions	*Pengguna telah membuka menu setting*	
Succesful End Condition	Berhasil mengubah pengaturan alarm	
Failed End Condition	Gagal mengubah pengaturan alarm	
Primary Actors	Pengguna	
Secondary Actors	-	
Trigger	Pengguna mengeklik menu Alarm	
Main Flow	***Step***	***Action***
	1	Pengguna mengatur pengaturan alarm
	2	Sistem menyimpan pengaturan yang diubah

Tabel 3. 15 Skenario *Use Case Setting Motion*

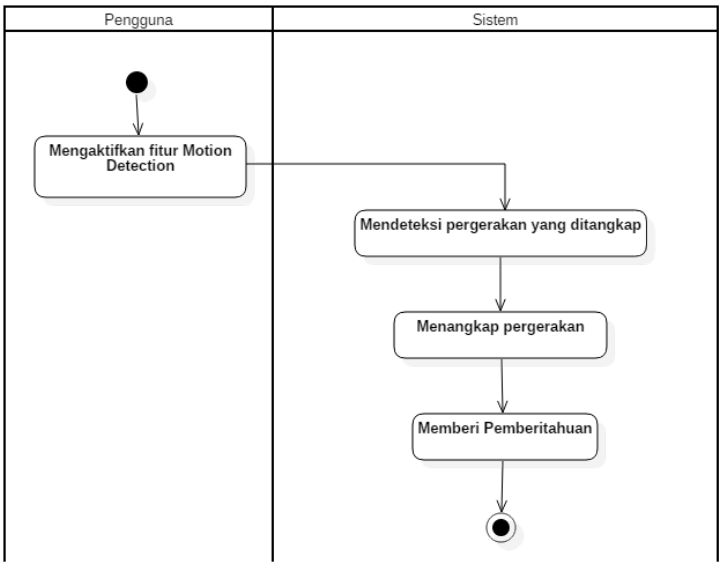
Use Case Name	Setting motion	
Goal In Context	Mengubah pengaturan motion	
Precoditions	*Pengguna telah membuka menu setting*	
Succesful End Condition	Berhasil mengubah pengaturan motion	
Failed End Condition	Gagal mengubah pengaturan motion	
Primary Actors	Pengguna	
Secondary Actors	-	
Trigger	Pengguna mengeklik menu Motion Detection	
Main Flow	***Step***	***Action***
	3	Pengguna mengatur motion detection untuk sistem
	4	Sistem menyimpan pengaturan yang diubah

2. *Activity Diagram*

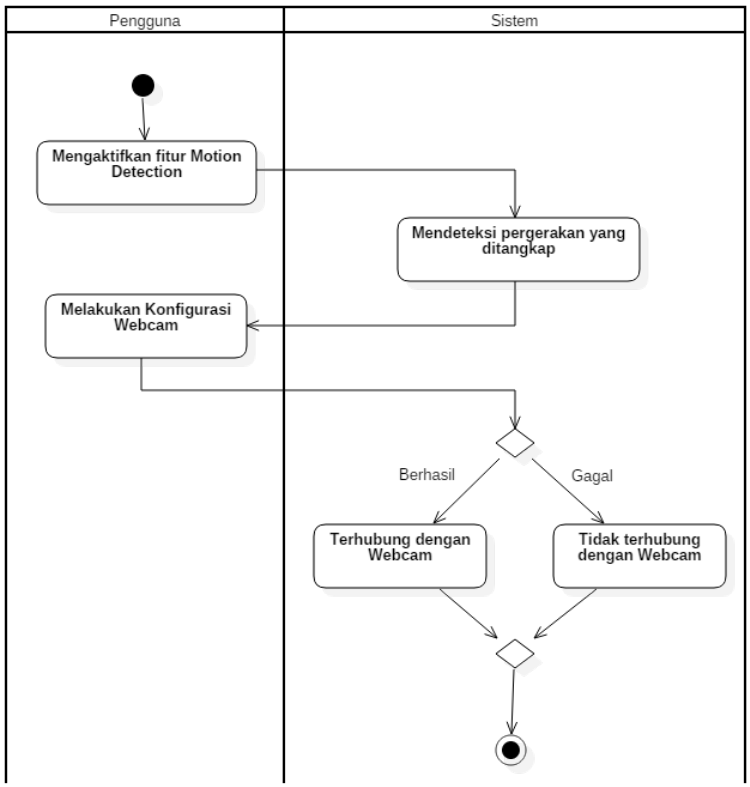
Activity diagram merupakan bagian dari penggambaran sistem secara fungsional menjelaskan proses-proses logika atau fungsi yang terimplementasi oleh kode program. *Activity diagram* memodelkan *event-event* yang terjadi didalam suatu *use case* dan digunakan untuk pemodelan aspek dinamis dari sistem. Berikut adalah masing-masing *activity diagram* untuk subsistem *desktop.*



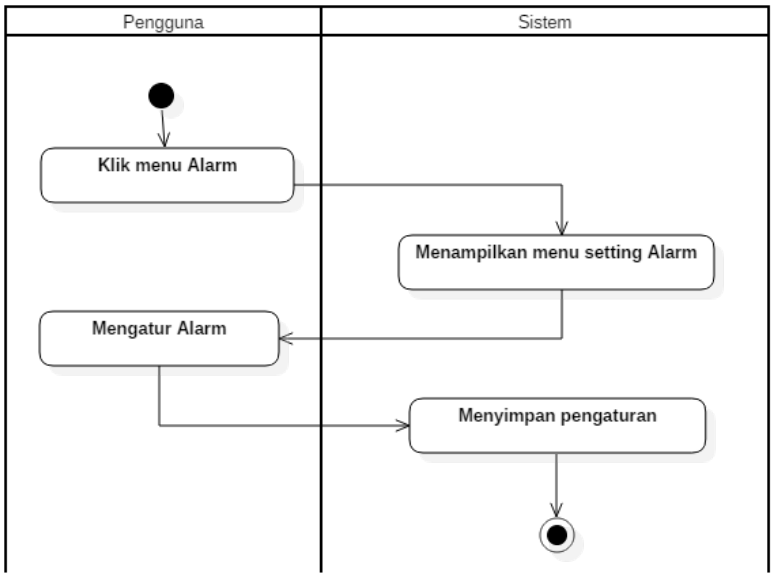
Gambar 3. 8 ***Activity Diagram Live View***



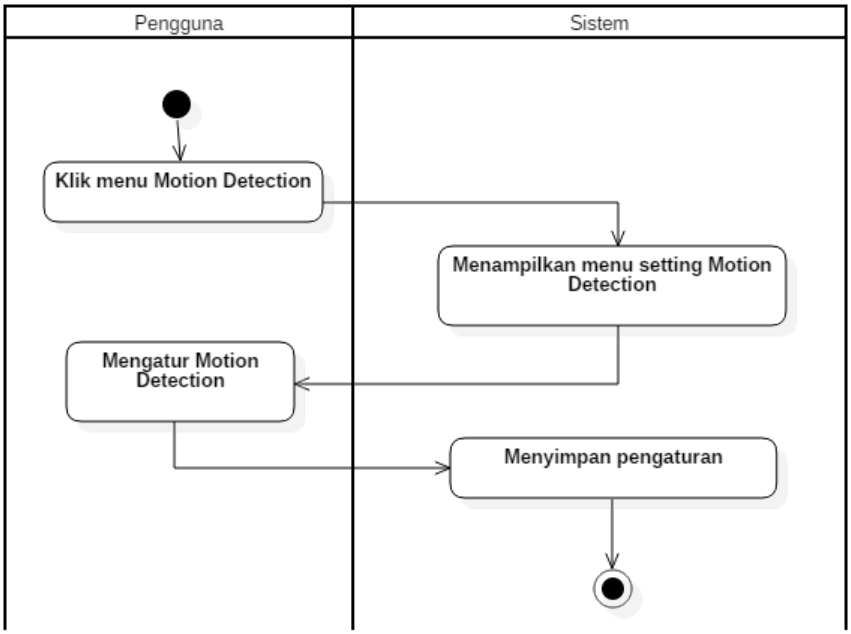
Gambar 3. 9 ***Activity Diagram Motion Detection***



Gambar 3. 10 ***Activity Diagram Setting Webcam***



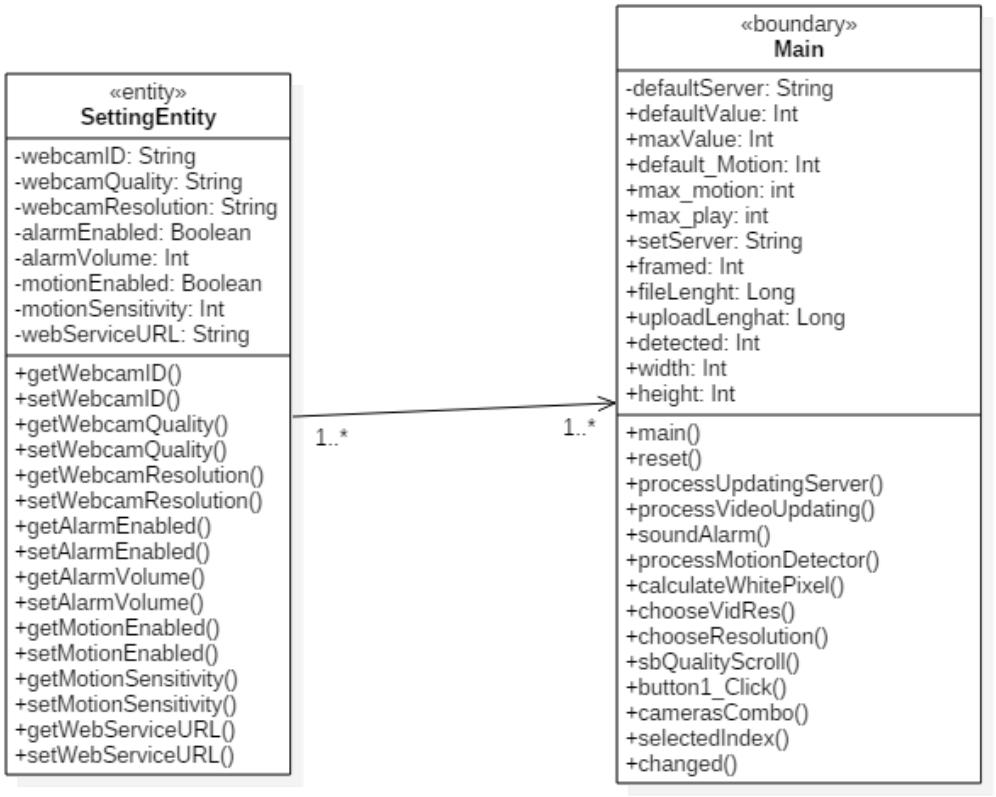
Gambar 3. 11 ***Activity Diagram Setting Alarm***



Gambar 3. 12 ***Activity Diagram Setting Motion***

3. *Class Diagram*

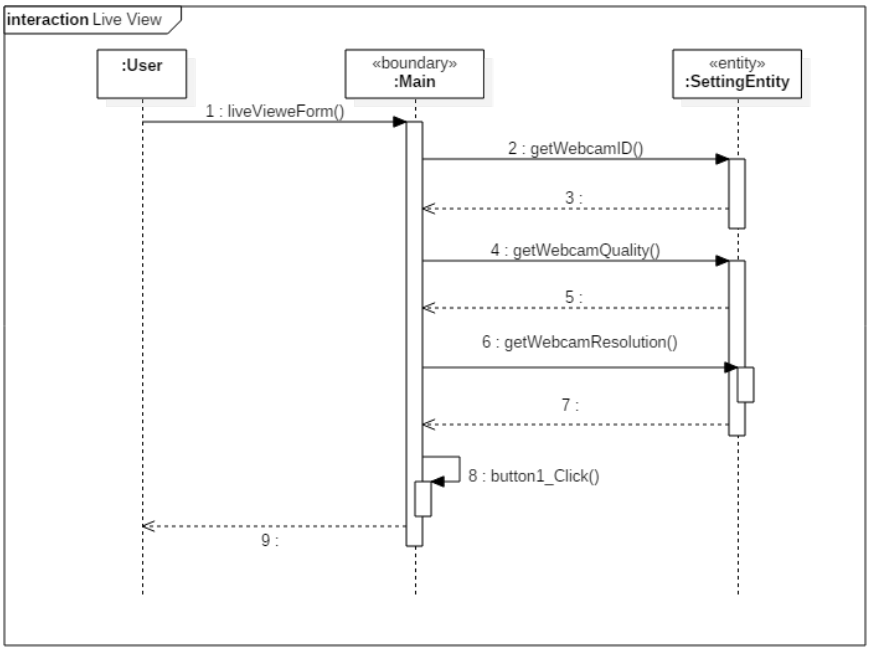
Class Diagram adalah sebuah spesifikasi dari fungionalitas yang menghasilkan objek dan merupakan inti dari pengembangan aplikasi ini. Gambar 3.10 menjelaskan class diagram dari sistem yang akan dibangun.



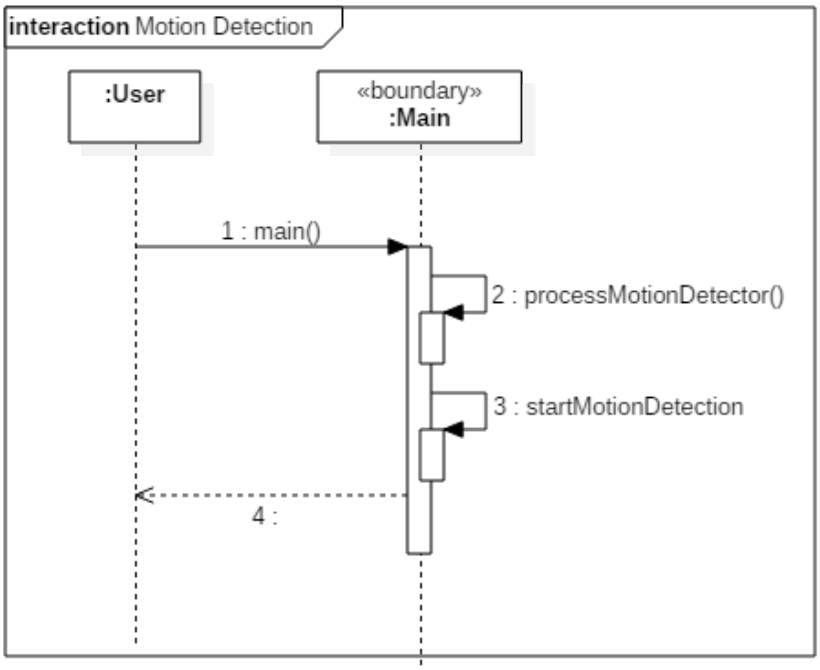
Gambar 3. 13 *Class Diagram Desktop*

4. *Sequance Diagram*

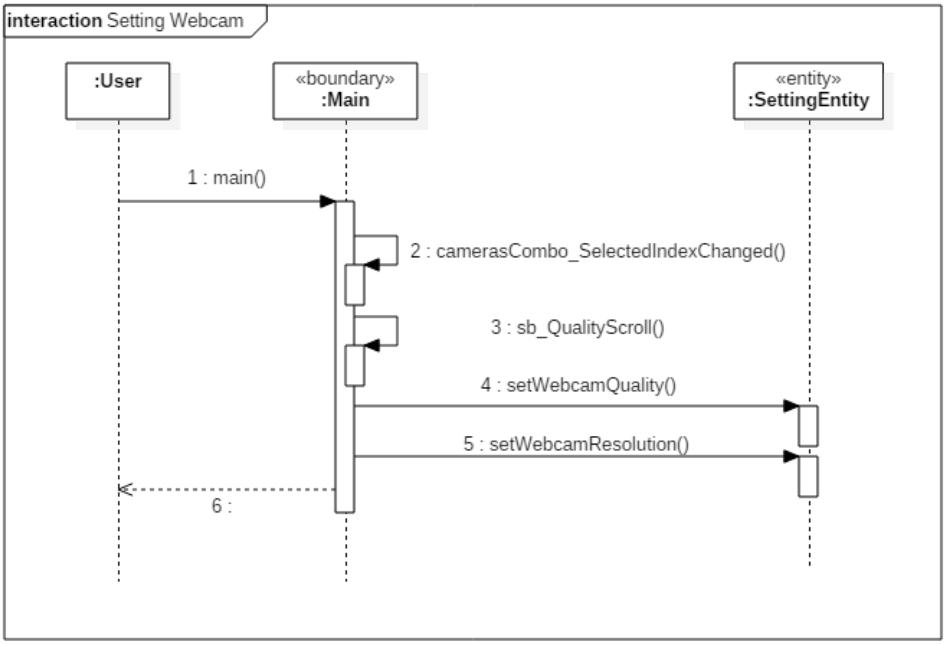
   Sequance diagram digunakan untuk menggambarkan arus pekerjaan, pesan yang disampaikan, dan bagaimana elemen-elemen di dalamnya bekerja sama dari waktu ke waktu untuk mencapai suatu hasil. *Sequence diagram* dari sistem yang akan dibangun dapat dilihat pada gambar-gambar berikut.



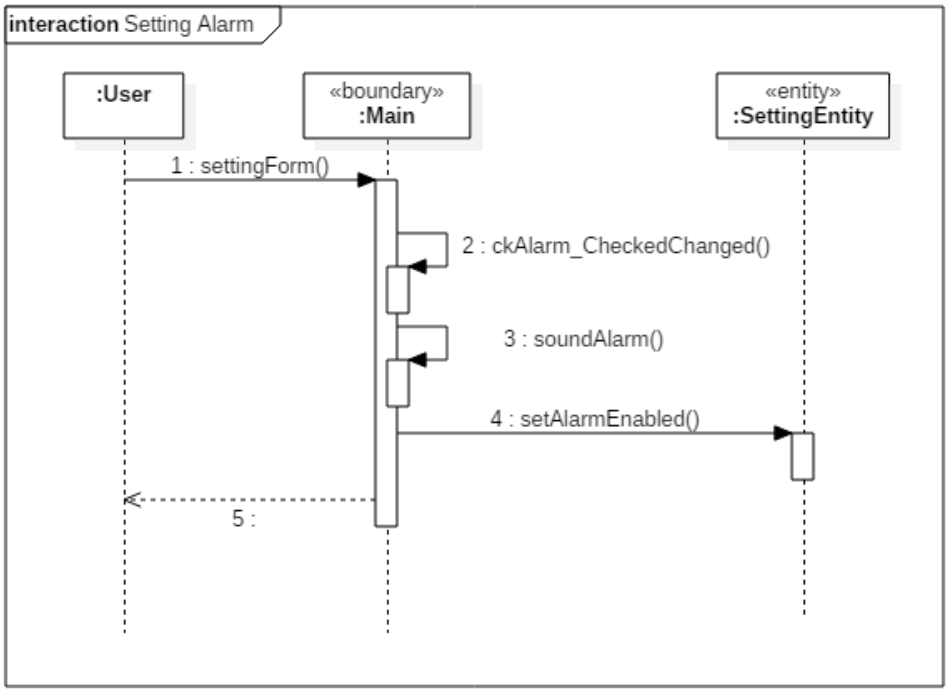
Gambar 3. 14 ***Sequence Diagram Live View***



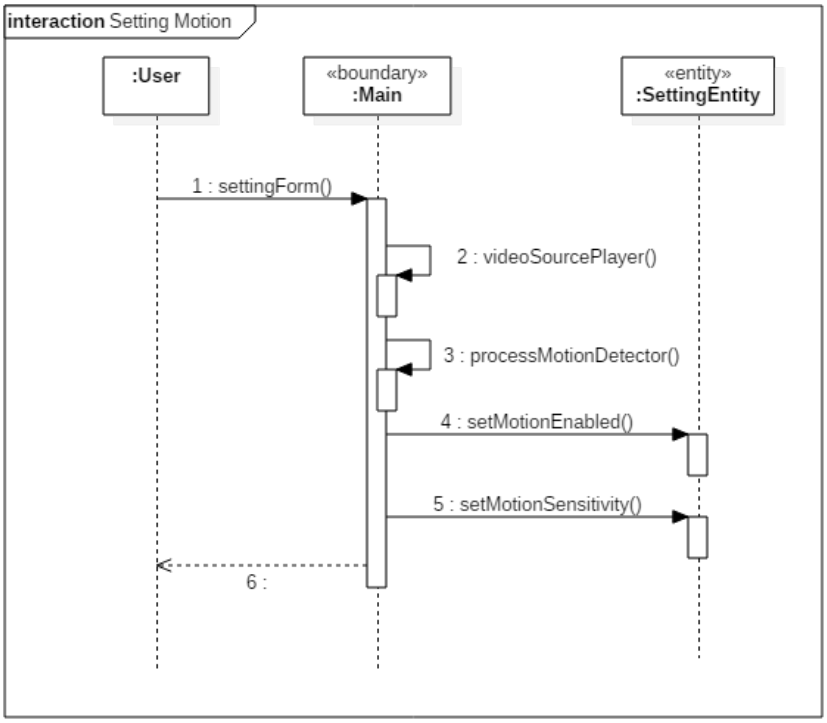
Gambar 3. 15 ***Sequence Diagram Motion Detection***



Gambar 3. 16 ***Sequence Diagram Setting Webcam***



Gambar 3. 17 ***Sequence Diagram Setting Alarm***



Gambar 3. 18 *Sequence Diagram Setting Motion*

3.1.7.2 Analisis Kebutuhan Fungsional Subsistem *Mobile*

1. *Use Case Diagram*

Use case diagram 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*.

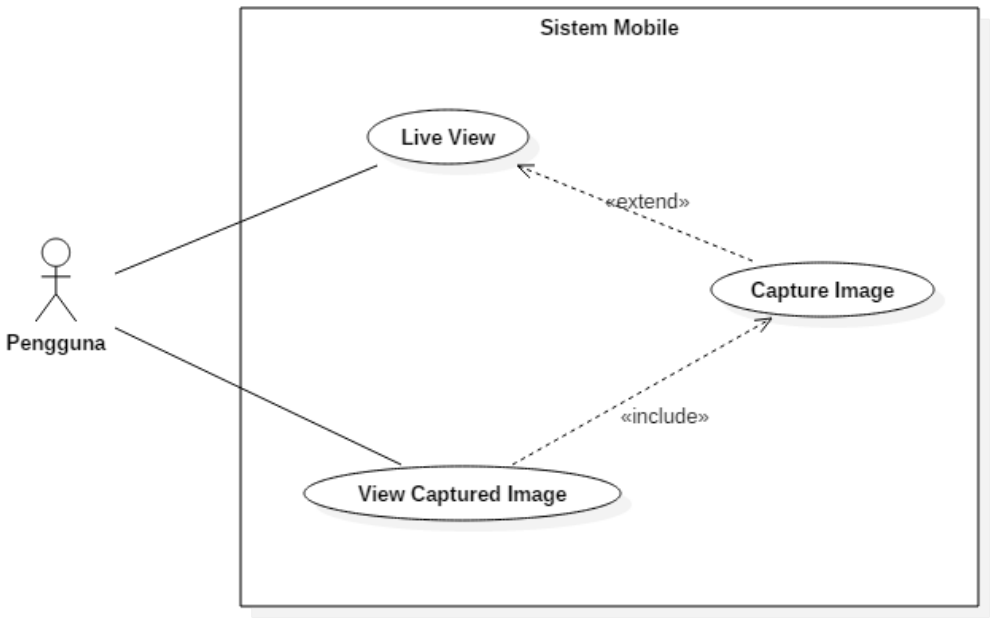
1. Identifikasi Aktor

Aktor adalah abstraksi dari orang dan sistem yang mengaktifkan fungsi dari target sistem. Pada Tabel 3.16 dijelaskan aktor-aktor yang berperan dalam menjalankan sistem yang dibangun.

Tabel 3. 16 Identifikasi Aktor

No.	Aktor	Deskripsi
1.	Pengguna	Merupakan aktor yang dapat berperan sebagai orang yang memakai aplikasi.

Gambar 3.18 merupakan penggambaran *use case diagram* untuk aplikasi yang akan dibangun. *Use case diagram* tersebut menunjukkan bagaimana interaksi antara aktor dan sistem, dimana terdapat satu aktor yang berperan, yaitu pengguna.



Gambar 3. 19 *Use Case Diagram Mobile*

2. Identifikasi *Use Case*

   Identifikasi *use case* di dalam sistem dapat dilihat pada Tabel 3.18.

Tabel 3. 17 Identifikasi *Use Case*

No.	*Use Case*	Deskripsi
1.	*Live View*	Fungsionalitas ini digunakan oleh pengguna untuk melihat video yang sedang ditampilkan oleh *webcam* secara *streaming* pada sistem.
2.	*Capture Image*	Fungsionalitas ini digunakan oleh pengguna untuk mengambil gambar yang sedang ditampilkan secara streaming.
3.	*View Captured Image*	Fungsionalitas ini digunakan oleh pengguna untuk melihat gambar yang telah diambil.

3. Skenario *Use Case*

Skenario proses-proses yang terdapat didalam use case dapat dilihat pada tabel-tabel berikut.

Tabel 3. 18 Skenario *Use Case Live View*

Use Case Name	Live view	
Goal In Context	Menampilkan tayangan video secara langsung	
Precoditions	*Pengguna telah membuka halaman utama*	
Succesful End Condition	Berhasil menayangkan video secara langsung	
Failed End Condition	Gagal menayangkan video secara langsung	
Primary Actors	Pengguna	
Secondary Actors	-	
Trigger	Pengguna menyentuh tombol Live View	
Main Flow	***Step***	***Action***
	1	Pengguna menyentuh tombol Live View
	2	Sistem mengambil video dari web service
	3	Sistem menampilkan video secara langsung
Extensions	***Step***	***Branching Action***
	2.1	Sistem menampilkan pesan bahwa video tidak bisa diputar

Tabel 3. 19 Skenario *Use Case Capture Image*

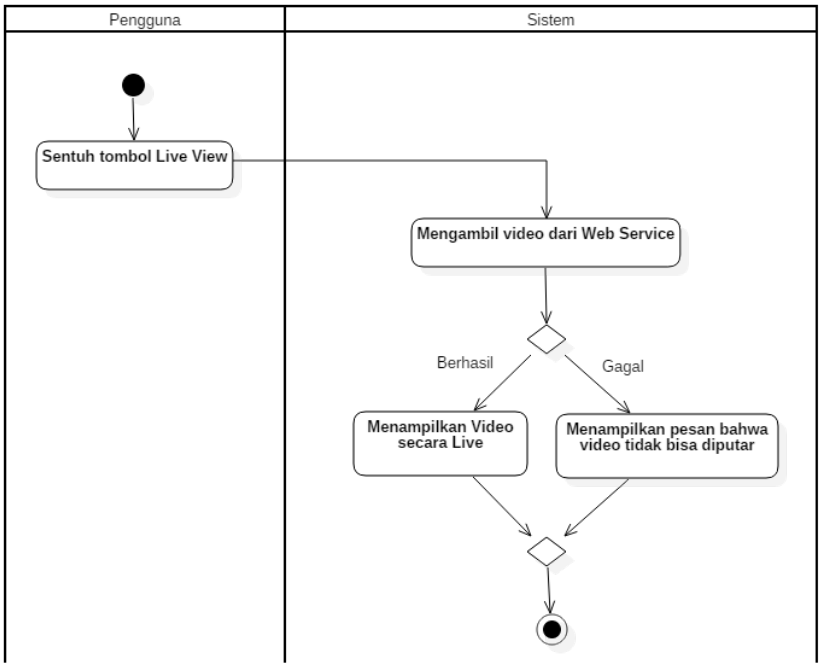
Use Case Name	Capture image	
Goal In Context	Mengambil gambar yang ditayangkan pada video	
Precoditions	*Pengguna telah membuka menu live view*	
Succesful End Condition	Berhasil menyimpan gambar	
Failed End Condition	Gagal menyimpan gambar	
Primary Actors	Pengguna	
Secondary Actors	-	
Trigger	Pengguna menyentuh tombol Capture	
Main Flow	***Step***	***Action***
	1	Pengguna menyentuh tombol Capture
	2	Sistem mengambil gambar dari video
	3	Sistem menyimpan gambar ke dalam memory card
	4	Sistem menampilkan pesan bahwa video berhasil di capture
Extensions	***Step***	***Branching Action***
	3.1	Sistem menampilkan pesan bahwa video gagal di capture

Tabel 3. 20 Skenario *Use Case View Captured Image*

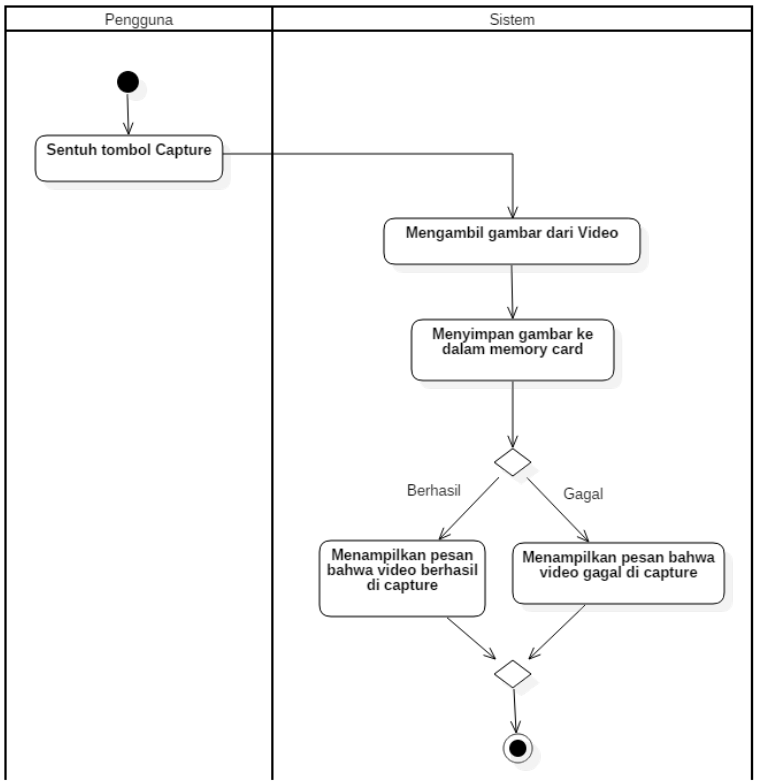
Use Case Name	View captured image	
Goal In Context	Menampilkan gambar hasil capture dari video	
Precoditions	*Pengguna telah membuka menu utama*	
Succesful End Condition	Berhasil menampilkan gambar	
Failed End Condition	Gagal menampilkan gambar	
Primary Actors	Pengguna	
Secondary Actors	-	
Trigger	Pengguna menyentuh tombol View Captured	
Main Flow	***Step***	***Action***
	1	Pengguna menyentuh tombol View Captured
	2	Sistem menampilkan daftar gambar hasil capture
	3	Pengguna memilih gambar
	4	Sistem menampilkan gambar yang dipilih

2. *Activity Diagram*

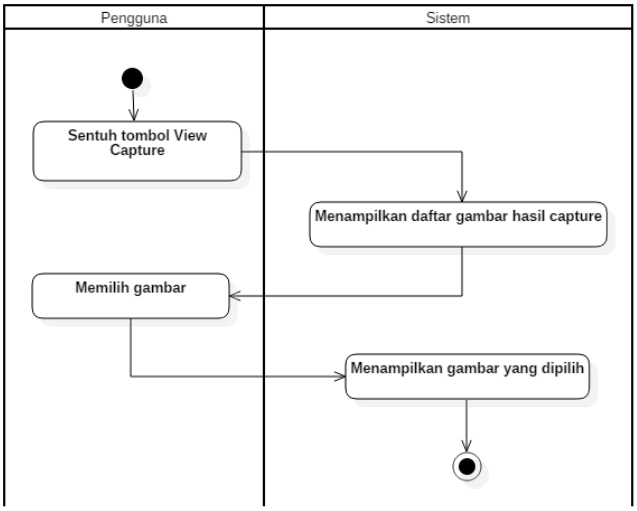
Activity diagram merupakan bagian dari penggambaran sistem secara fungsional menjelaskan proses-proses logika atau fungsi yang terimplementasi oleh kode program. *Activity diagram* memodelkan *event-event* yang terjadi didalam suatu *use case* dan digunakan untuk pemodelan aspek dinamis dari sistem. Berikut adalah masing-masing *activity diagram* untuk subsistem *mobile.*



Gambar 3. 20 ***Activity Diagram Live View***



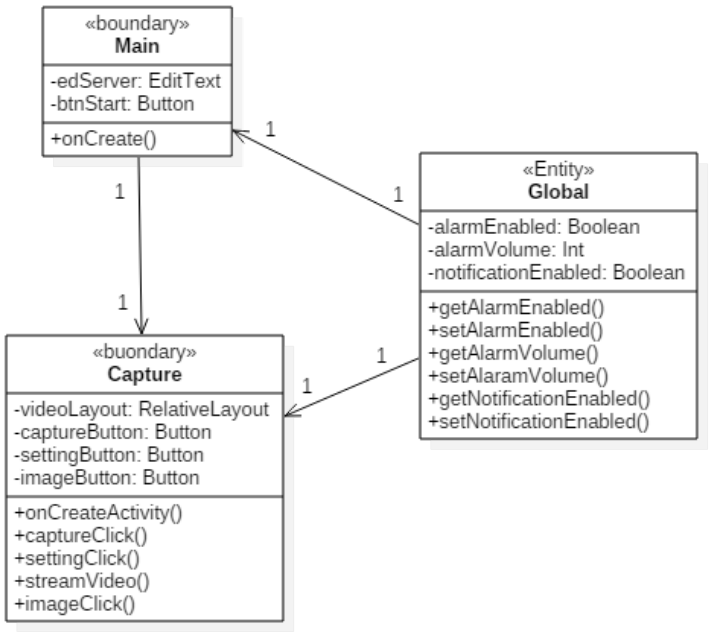
Gambar 3. 21 ***Activity Diagram Capture Image***



Gambar 3. 22 ***Activity Diagram View Captured Image***

3. *Class Diagram*

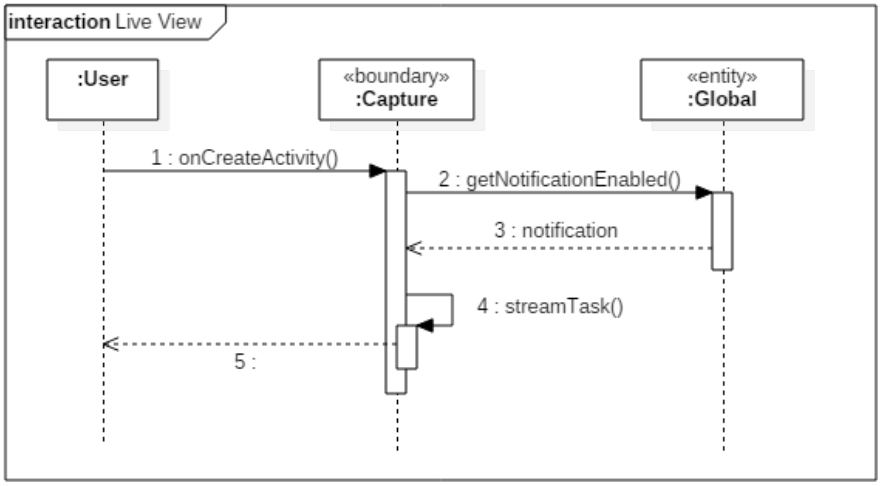
Class Diagram adalah sebuah spesifikasi dari fungionalitas yang menghasilkan objek dan merupakan inti dari pengembangan aplikasi ini. Gambar 3.45 menjelaskan class diagram dari sistem yang akan dibangun.



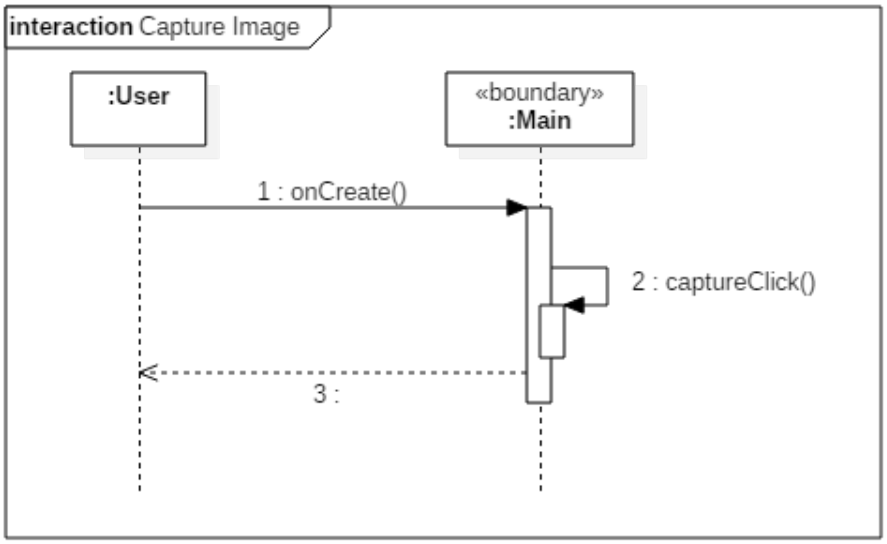
Gambar 3. 23 ***Class Diagram Mobile***

4. *Sequance Diagram*

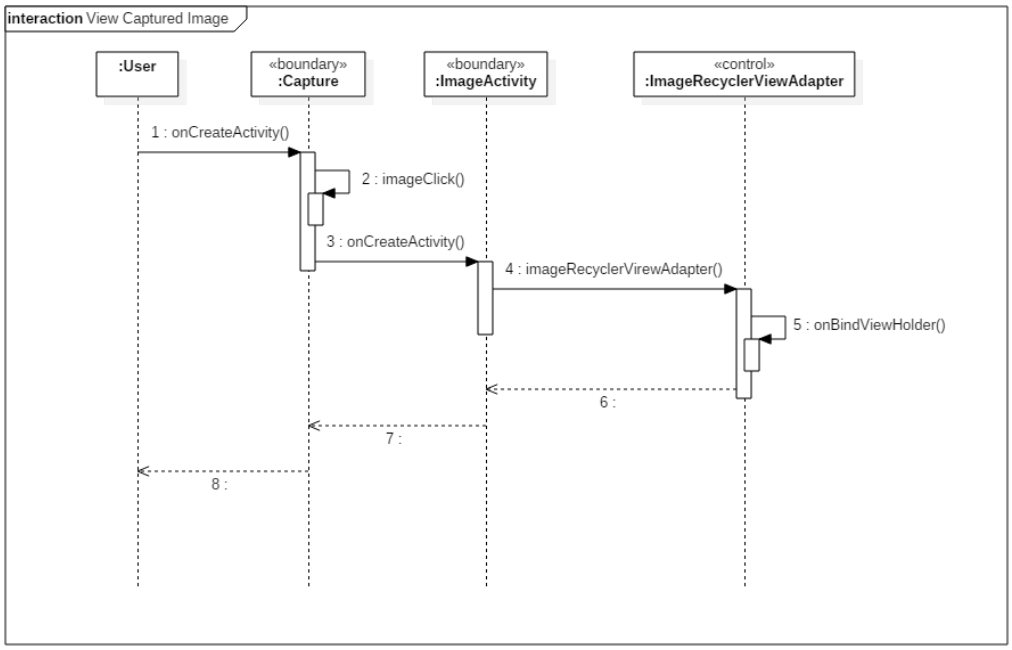
Sequance diagram digunakan untuk menggambarkan arus pekerjaan, pesan yang disampaikan, dan bagaimana elemen-elemen di dalamnya bekerja sama dari waktu ke waktu untuk mencapai suatu hasil. *Sequence diagram* dari sistem yang akan dibangun dapat dilihat pada gambar-gambar berikut.



Gambar 3. 24 *Sequence Diagram Live View*



Gambar 3. 25 *Sequence Diagram Capture Image*



Gambar 3. 26 *Sequence Diagram View Captured Image*

3.2 Perancangan Sistem

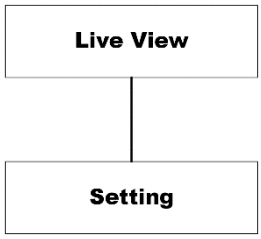
Di dalam perancangan sistem ini, hal yang akan dibahas adalah perancangan struktur menu, perancangan antarmuka, dan jaringan semantik untuk masing-masing subsistem.

3.2.1 Perangcangan Struktur Menu

Perancangan struktur menu merupakan gambaran jalur pemakaian aplikasi sehingga aplikasi yang dibangun mudah dipahami dan mudah digunakan.

3.2.1.2 Perancangan Struktur Menu Subsistem *Desktop*

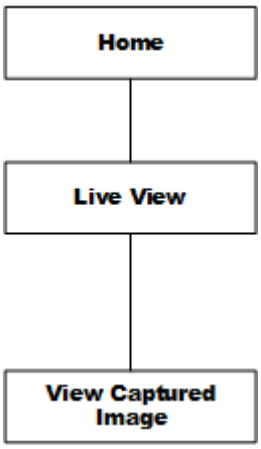
Perancangan struktur menu subsistem *desktop* dapat dilihat pada Gambar 3.32.



Gambar 3. 27 Struktur Menu Subsistem *Desktop*

3.2.1.3 Perancangan Struktur Menu Subsistem *Mobile*

Perancangan struktur menu sub sistem *mobile* dapat dilihat pada Gambar 3.33.



Gambar 3. 28 Struktur Menu Subsistem *Mobile*

3.2.2 Perancangan Antarmuka

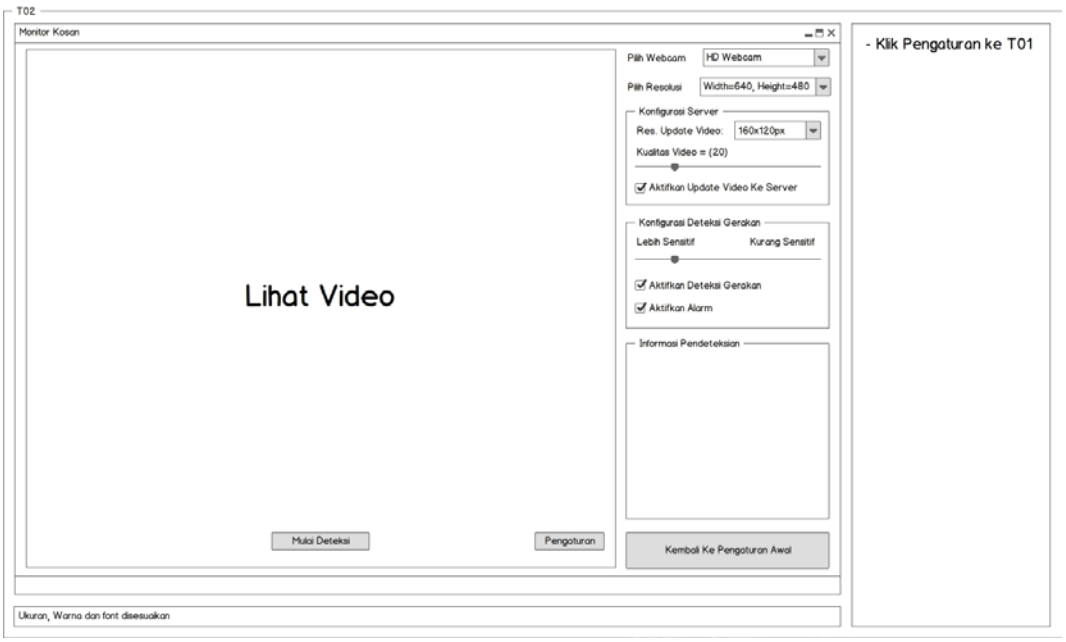
Sistem yang akan dibangun diharapkan menyediakan interface (antarmuka) yang mudah dipahami dan digunakan oleh user. Berikut ini adalah perancangan antarmuka dari kedua sub sistem yang akan dibuat.

3.2.2.2 Perancangan Antarmuka Subsistem *Desktop*

Gambar-gambar berikut merupakan perancangan antarmuka untuk subsistem *desktop* yang akan dibuat

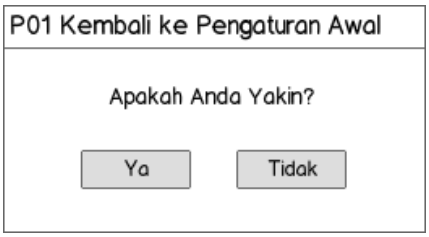


Gambar 3. 29 Perancangan Antarmuka Menu Utama



Gambar 3. 30 Perancangan Antarmuka Tampil *Setting*

3.2.2.3 Perancangan Pesan Subsistem *Desktop*



Gambar 3. 31 Perancangan Pesan Kembali ke Pengaturan Awal

3.2.2.4 Perancangan Antarmuka Subsistem *Moblie*

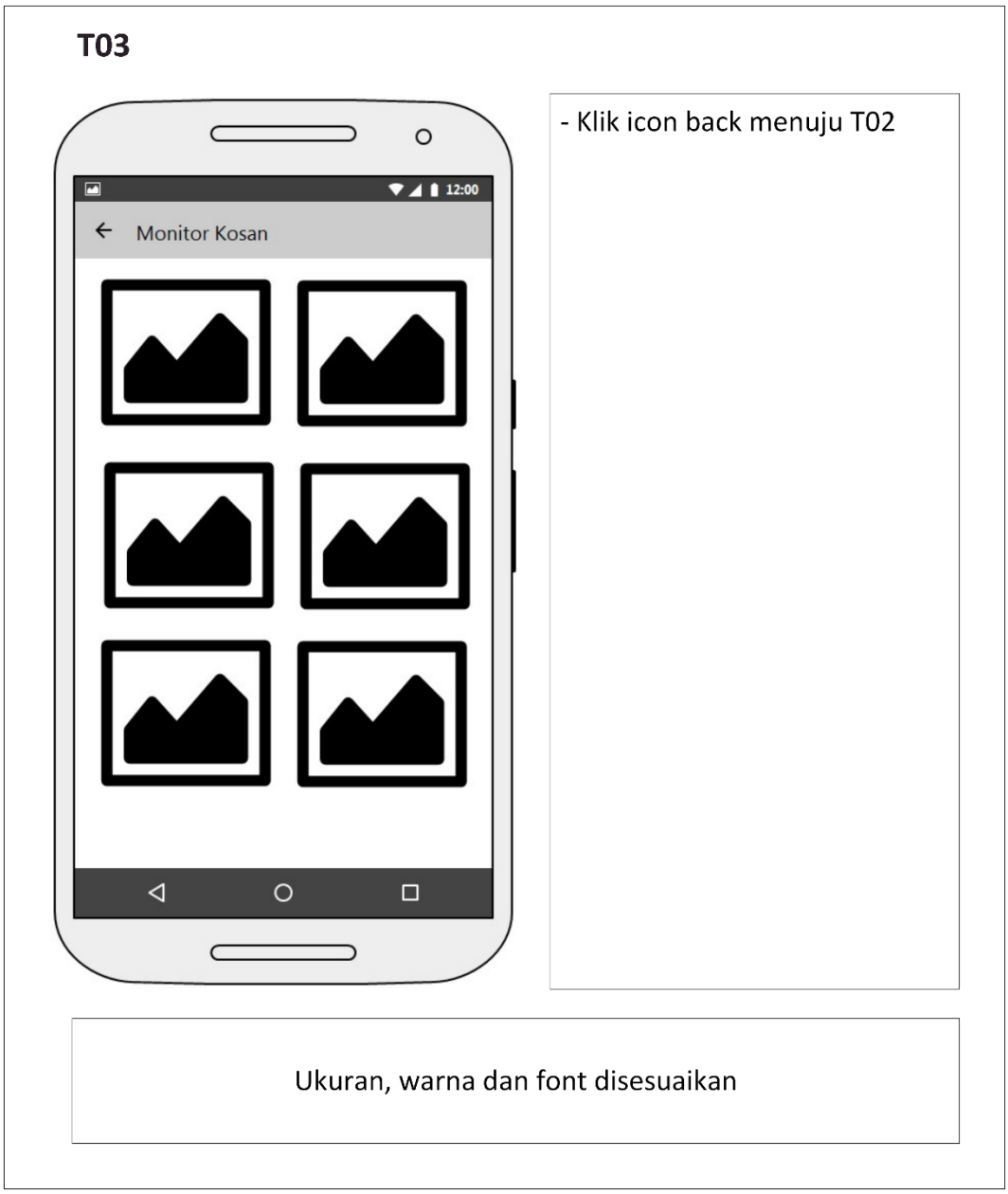
Gambar-gambar berikut merupakan perancangan antarmuka untuk subsistem *mobile* yang akan dibuat



Gambar 3. 32 Perancangan Antarmuka Menu Utama



Gambar 3. 33 Perancangan Antarmuka *Live View*



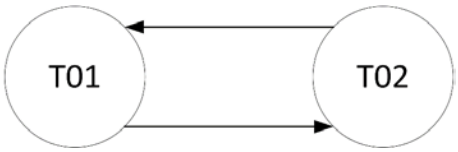
Gambar 3. 34 Perancangan Antarmuka *View Captured Image*

3.2.3 Jaringan Semantik

Jaringan semantik merupakan gambaran pengetahuan grafis yang menunjukkan hubungan antar objek (lingkaran) dan garis yang menggambarkan formasi antar objek tersebut

3.2.3.1 Jaringan Semantik Subsistem *Desktop*

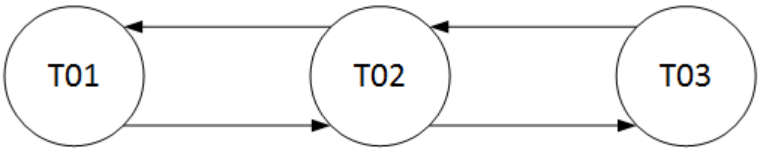
Jaringan semantik untuk sub sistem *desktop* dapat dilihat pada Gambar 3.142.



Gambar 3. 35 Semantik Subsistem *Desktop*

3.2.3.2 Jaringan Semantik Subsistem *Mobile*

Jaringan semantik untuk sub sistem *mobile* dapat dilihat pada Gambar 3.47



Gambar 3. 36 Semantik Subsistem *Mobile*