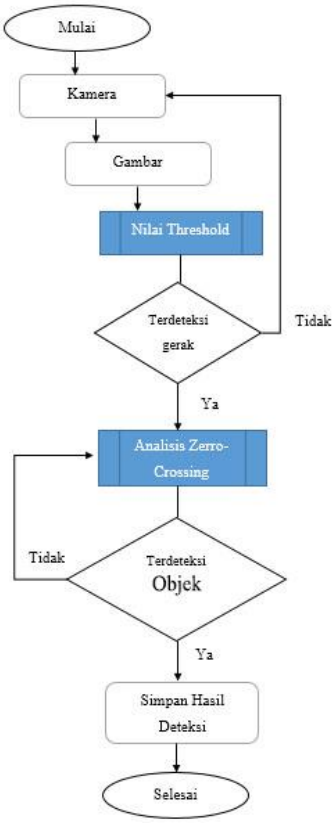
BAB 3
ANALISA DAN PERANCANGAN

3.1 Analisis Masalah

Tahapan awal untuk membangun *home controlling system,* yaitu dengan menentukan metode yang akan diterapkan dalam pendeteksian citra. Deteksi citra merupakan kemampuan untuk mengenali objek pada gambar yang ter-*capture* melalui sebuah kamera digital seakan-akan kamera adalah sebagai sensor pendeteksi gerak. Mengimplementasikan metode *Marr-Hilderth* sebagai deteksi tepi objek.

3.1.1 Proses Pendeteksian Objek

Dalam menentukan nilai objek yang akan dideteksi menggunakan *Marr-Hilderth,* terlebih dahulu melalui tahapan-tahapan yang terlihat pada Gambar 3.1.



Gambar 3.1 Flow Chart Proses Deteksi Objek

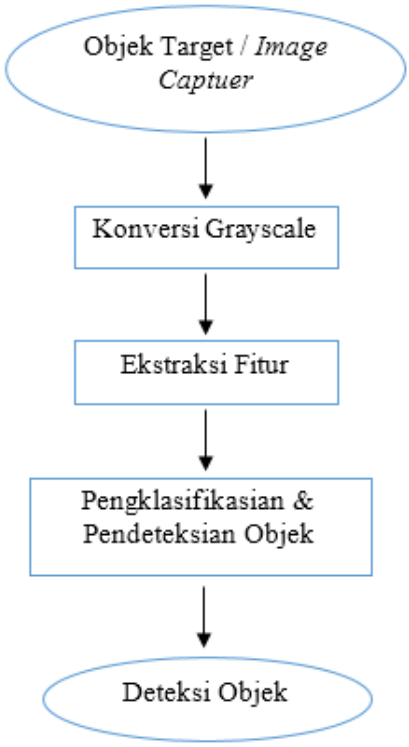
3.1.2 Analisis Klasifikasi Pendeteksian Objek

Untuk klasifikasi objek yang akan disegmentasi dapat dikelompokkan dengan menggunakan tabel agar objek yang akan menjadi perhatian dapat teridentifikasi. Dalam menganalisis objek yang akan dideteksi yaitu dengan mengklasifikasikan suatu objek yang tertangkap oleh *webcam*. Untuk klasifikasinya adalah sebagai berikut :

Tabel 3.1 Klasifikasi analisis segmentasi objek

Objek	Citra	Kegunaan segmentasi	Acuan yang digunakan
Area taman	Manusia, halaman, jalan, lampu pijar, pohon	Pelacakan manusia	Gerakan dan perubahan warna
Area ruang tamu	Manusia, sofa, pintu, jendela	Pelacakan manusia	Gerakan dan perubahan warna

Segmentasi sebagai tahapan awal dalam membagi suatu objek ke dalam region yang telah ditentukan, dapat diketahui alur dalam pengklasifikasian sebagai berikut :



Gambar 3.2 Segmentasi langkah awal sistem klasifikasi

Keterangan Gambar 3.2 adalah sebagai berikut :

1. Objek target yaitu dimana citra yang akan dianalisis merupakan objek yang ditangkap oleh kamera webcam.
2. Konversi grayscale yaitu merubah intensitas warna RGB menjadi warna keabuan.
3. Ekstraksi fitur, perbandingan antara lebar dan panjang objek, warna rata-rata suatu objek, bentuk pada objek dan tekstur pada objek dengan menggunakan metode Marr-Hilderth.
4. Pengklasifikasian dan pendeteksian, objek yang dapat ditentukan apakah objek yang memengaruhi berubahnya nilai baik warna maupun tekstur dan bentuk pada objek adalah objek dari menusia ?. maka dapat diketahui setelah diklasifikasi yang akan dijelaskan disubbab pengklasifikasian objek .

1.1.2.1 Analisis Objek Target

Objek target merupakan citra pada objek yang ter-*capture* oleh kamera webcam yang akan dianalisis sehingga menjadi matriks atau bisa disebut kernel yang memiliki bobot tertentu yang diperoleh dari setiap piksel yang ada pada objek tersebut. Piksel-piksel yang ada pada objek yang sudah dianalisis menjadi suatu bobot nilai kedalam matriks akan di bandingkan dengan nilai objek yang sudah di-*difault* atau memiliki nilai awal pada saat objek belum dilakukan pendeteksian.

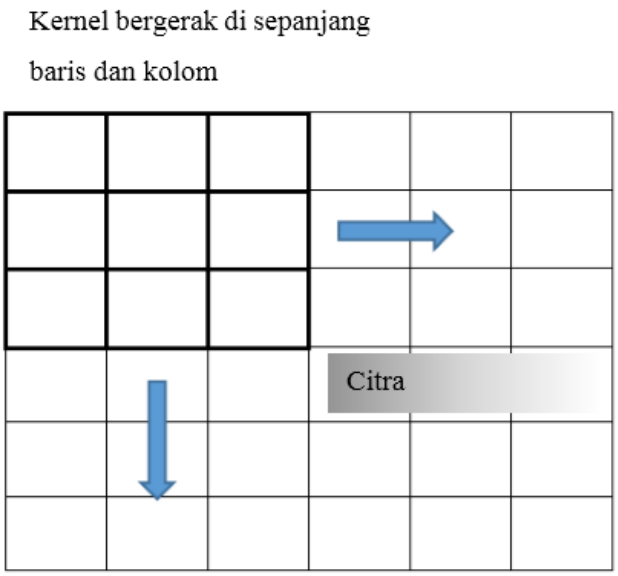
Proses dalam menentukan atau memperoleh suatu nilai pada operas ketetanggaan piksel yang melibatkan matriks atau bisa disebut kernel yang mempresentasikan pembobotan. Untuk mencapai tujuan objek yang menjadi target dapat ditentukan menggunakan konvolusi yang dapat berbentuk bujur sangkar maupun persegi panjang.

Kernel yang akan digunakan untuk menentukan nilai konvolusi yaitu dengan memilih kernel konvolusi yang berukuran 3x3.

-1	0	1
-2	0	2
-1	0	1

Gambar 3.3 Kernel konvolusi 3x3

Pada pemrosesan untuk mencari nilai konvolusi, kernel digeser sepanjang baris dan kolom dalam citra, sehingga dapat diperoleh nilai baru pada citra keluaran.



Gambar 3.4 Proses konvolusi sepanjang kolom dan baris

Rumus yang digunakan untuk menentukan nilai konvolusi adalah sebagai berikut :

$$\boldsymbol{g(x,y) = \sum_{p=-m2}^{m2} \sum_{q=-n2}^{n2} h(p+m2+1, q+n2+1) f(y-p, x-q)} \quad \textbf{(3.1)}$$

Dalam hal ini untuk keterangan formula diatas :

a. m2 adalah separuh dari tinggi kernel (m2 = floor(m/2)).
b. n2 adalah separuh dari lebar kernel (n2 = floor(n/2)).
c. Floor menyatakan pembulatan kebawah.
d. h menyatakan kernel, dengan indeks dimulai dari 1.

Untuk gambaran implementasi persamaan $\boldsymbol{g(x,y) = \sum_{p=-m2}^{m2} \sum_{q=-n2}^{n2} h(p+m2+1, q+n2+1) f(y-p, x-q)}$ (3.1) dalam menentukan nilai konvolusi kita ambil contoh Gambar 3.5 yang di ambil nilai piksel resolusinya 8x8.



Gambar 3.5 Contoh Object

	0	1	2	3	4	5	6	7	8		m
0	65	50	55	58	29	16	17	52	54		
1	78	68	60	55	57	51	54	54	99		
2	60	60	62	56	52	72	55	52	59		
3	28	54	58	51	58	59	59	36	43		
4	86	72	13	82	57	46	55	57	52		
5	03	05	57	53	59	58	35	46	54		
6	95	66	06	12	24	18	93	95	62		
7	99	01	53	84	34	94	86	61	37		
8	67	09	65	13	83	94	62	13	72		
n											

1	0	1
-2	0	2
-1	0	1

Gambar 3.6 a) kernel citra target, b) kernel konvolusi 3x3

Maka nilai piksel hasil konvolusi adalah sebagai berikut :

g(x, y) = (-1 x 62) + (0 x 60) + (1 x 60) + (-2 x 60) + (0 x 68) + (2 x 78) + (-1 x 55) + (0 x 50) + (1 x 65)

= 44

Lakukan terus berulang sehingga dengan demikian,nilai yang telah di konvolusikan akan seperti pada Gambar 3.7.

	0	1	2	3	4	5	6	7	8
0	65	50	55	58	29	66	67	52	54
1	78	44	22	42	-16	-35	28	-81	99
2	60	-16	24	23	-36	-4	60	-37	59
3	28	-39	0	16	4	-3	55	31	43
4	86	12	-15	10	59	27	13	3	52
5	53	54	-2	34	20	-19	-14	-4	54
6	95	120	-23	81	-27	-166	-9	92	62
7	99	133	-66	52	-57	-152	70	119	37
8	67	59	65	63	83	94	62	63	72

Gambar 3.7 Nilai citra keluaran

Dari hasil analisi implementasi dalam menentukan nilai konvolusi dalam piksel citra yang menjadi target dapat dilakukan dengan cara yang telah dijelaskan. Selain itu untuk menentukan nilai konvolusi kernel objek atau piksel citra harus dilakukan dengan memfilter menjadi kernel yang mempunyai baris dan kolom harus bernilai ganjil.

1.1.2.2 Konversi Grayscale

Misal diambil ukuran piksel dengan resolusi 5x5 dengan nilai RGB yang diperoleh dari setiap blok nilai citra keluaran, nilai tersebut dapat dilihat pada Gambar 3.8:

187 ,261 ,217	171 ,253 ,192	168 ,136 ,228	197 ,233 ,191	213 ,162 ,188
175 ,198 ,232	186 ,204 ,248	184 ,165 ,153	161 ,209 ,193	163 ,238 ,178
252 ,169 ,131	164 ,229 ,102	186 ,151 ,192	177 ,168 ,184	189 ,187 ,172
192 ,176 ,184	139 ,188 ,199	166 ,248 ,205	199 ,213 ,174	177 ,216 ,157
119 ,142 ,224	104 ,156 ,165	184 ,209 ,179	121 ,174 ,225	118 ,153 ,167
135 ,175 ,196	198 ,204 ,221	174 ,163 ,246	195 ,187 ,189	186 ,218 ,199

Gambar 3.8 RGB Nilai Citra keluaran dengan ukuran 5x5

Dengan menggunakan rumus mencari rata-rata nilai RGB dengan membagi 3 untuk mengkonversi dari citra warna ke citra grayscale, setelah itu nilai rata-rata yang telah didapatkan harus dibulatkan.

F(1.1) = (187 ,261 ,217) = (187 + 261 + 217) / 3 = 221.66 = 222

F(1.2) = (171 ,253 ,192) = (171 + 253 + 192) / 3 = 205.33 = 205

F(1.3) = (168 ,136 ,228) = (168 + 136 + 228) / 3 = 177.33 = 177

F(1.4) = (197 ,233 ,191) = (197 + 233 + 191) / 3 = 207

F(1.5) = (213 ,162 ,188) = (213 + 162 + 188) /3 = 187.66 = 188

Untuk piksel selanjutnya dilakukan perhitungan seperti yang dijelaskan, sehingga dapat kita lihat hasil perhitungannya pada Gambar 3.9.

222	205	177	207	188
202	213	167	188	193
184	165	176	176	183
184	175	206	195	183
162	142	191	173	146

Gambar 3.9 Nilai Citra Grayscale

Dari hasil diatas untuk gambaran objek gambar yang telah dikonversi ke grayscale adalah sebagai berikut :



Gambar 3.10 citra asli & citra yang dokonversi ke *grayscale*

1.1.2.3 Analisis Ekstraksi Fitur

Analisis ekstraksi fitur yaitu berhubungan dengan pengukuran atau perbandingan bentuk suatu objek, warna pada objek dan tekstur yang terkandung dalam objek citra. Untuk mencari suatu nilai perbandingan dan bentuk suatu objek dapat diketahui dengan menggunakan mencari nilai ambang (*threshold*) dimana nilai intensitasnya hanya terdiri dari 0 dan 1.

Dalam menentukan nilai ambang (*threshold*) yaitu dengan menggunakan pengambangan basis adaptif (*basic adapter thresholding*) dapat dicari dengan menentukan nilai T untuk titik tengah antara tigkat keabuan maksimum dan minimum dalam subcitra.

Untuk proses analisis dengan menggunakan *basic adapter thresholding* dapat dilakukan dengan sampel resolusi dengan nilai yang ditentukan pada objek yang ter-*capture* oleh kamera:



Gambar 3.11 sample objek *grayscale*

Diketahui :

L = 256

P = 193 x 173

t = 100

Keterangan :

L : resolusi keabuan

P = resolusi spasial

t = threshold

222	205	177	207	188
202	213	167	188	193
184	165	176	176	183
184	175	206	195	183
162	142	191	173	146

Gambar 3.12 Nilai piksel contoh objek

Lakukan proses adapter *thresholding* dengan membagi citra menjadi subcitra ukuran 2x2

Penyelesaian :

F(1,1)

1. Ambil subcitra 2x2 $\begin{bmatrix} 222 & 205 \\ 202 & 213 \end{bmatrix}$
2. Hitung Mean

$$mean = \bar{x} = \frac{222 + 205 + 202 + 213}{4} = 210$$

3. Hitung Var

$$var = \frac{(222 - 210)^2 + (205 - 210)^2 + (202 - 210)^2 + (213 - 210)^2}{4 - 1}$$

$$= 57.50$$

F(1.2)

1. Ambil subcitra 2x2 $\begin{bmatrix} 205 & 177 \\ 213 & 167 \end{bmatrix}$
2. Hitung Mean

$$mean = \bar{x} = \frac{205 + 177 + 213 + 167}{4} = 191$$

3. Hitung Var

$$var = \frac{(205 - 191)^2 + (177 - 191)^2 + (213 - 191)^2 + (167 - 191)^2}{4 - 1}$$

$$= 354.33$$

F(1.3)

1. Ambil subcitra 2x2 $\begin{bmatrix}177 & 207\\167 & 188\end{bmatrix}$
2. Hitung Mean

$$mean = \bar{x} = \frac{177 + 207 + 167 + 188}{4} = 185$$

3. Hitung Var

$$var = \frac{(177-185)^2 + (207-185)^2 + (167-185)^2 + (188-185)^2}{4-1}$$

$$= 214.47$$

F(1.4)

1. Ambil subcitra 2x2 $\begin{bmatrix}207 & 188\\188 & 193\end{bmatrix}$
2. Hitung Mean

$$mean = \bar{x} = \frac{207 + 188 + 188 + 193}{4} = 194$$

3. Hitung Var

$$var = \frac{(207-194)^2 + (188-194)^2 + (188-194)^2 + (193-194)^2}{4-1}$$

$$= 61.53$$

Jika var > 100, maka :

Gunakan nilai T tersebut untuk melakukan *basic thresholding*. Dan diperoleh *threshold* = 210 dan subcitra output = 0

Jika tidak, maka

Subcitra output = 1

Sehingga didapatkan nilai piksel biner seperti pada Gambar 3.13

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

Gambar 3.13 Citra Biner

Gambar 3.14 berikut menunjukkan hasil *thresholding* dengan metode *global adaptive thresholding*. Gambar kiri citra asli dan gambar kanan adalah hasil *adaptive thresholding* dengan membagi menjadi 4 subcitra. Pada gambar hasil *adaptive thresholding*, bentuk tangan terlihat lebih jelas.



Gambar 3.14 (a) Citra Asli, (b) hasil adaptif *thresholding*

1.1.2.4 Analisis Klasifikasi dan Pendeteksian Objek

Analisis klasifikasi objek merupakan tahap akhir dalam menentukan citra objek yang menjadi target dalam penelitian. Untuk mengklasifikasi objek yang akan dipilih harus melalui beberapa tahap yang sudah dijelaskan pada subbab sebelumnya.

Untuk menerapkan meode dari operator Marr-Hilderth dapat kita cari dengan menentukan nilai konvolusi dengan menggunakan operator Marr-Hilderth. Berikut ini adalah langkah-langkah dalam mendapatkan nilai konvolusi dengan menggunakan Marr-Hilderth untuk contoh citra input yang telah dideteksi

perubahan objek dengan pengambilan citra inputan ukuran 5x5 dari hasil ekstraksi nilai piksel *grayscale* berikut ini :

222	205	177	207	188
202	213	167	188	193
184	165	176	176	183
184	175	206	195	183
162	142	191	173	146

Gambar 3.15 Piksel *Grayscale*

1. Misalkan ukuran kernel untuk *filter* adalah 3 x 3, maka turunan kedua dari *filter Gaussian* dengan σ = 1, dengan x, y adalah koordinat filter dan σ adalah standar deviasi, dimana :

	y		
x	(-1, -1)	(0, -1)	(1, -1)
	(-1, 0)	(0, 0)	(1, 0)
	(-1, 1)	(0, 1)	(1, 1)

Maka filter H =

$\frac{1}{\sqrt{2\pi(1)^2}}\left(2-\frac{(-1^2)+(-1^2)}{1^2}\right)e^{\frac{(-1^2)+(-1^2)}{2(1)^2}}$	$\frac{1}{\sqrt{2\pi(1)^2}}\left(2-\frac{(0^2)+(-1^2)}{1^2}\right)e^{\frac{(0)+(-1^2)}{2(1)^2}}$	$\frac{1}{\sqrt{2\pi(1)^2}}\left(2-\frac{(1^2)+(-1^2)}{1^2}\right)e^{\frac{(1^2)+(-1^2)}{2(1)^2}}$
$\frac{1}{\sqrt{2\pi(1)^2}}\left(2-\frac{(-1^2)+(0^2)}{1^2}\right)e^{\frac{(-1^2)+(0^2)}{2(1)^2}}$	$\frac{1}{\sqrt{2\pi(1)^2}}\left(2-\frac{(0^2)+(0^2)}{1^2}\right)e^{\frac{(0^2)+(0^2)}{2(1)^2}}$	$\frac{1}{\sqrt{2\pi(1)^2}}\left(2-\frac{(1^2)+(0^2)}{1^2}\right)e^{\frac{(1^2)+(0^2)}{2(1)^2}}$
$\frac{1}{\sqrt{2\pi(1)^2}}\left(2-\frac{(-1^2)+(1^2)}{1^2}\right)e^{\frac{(-1^2)+(1^2)}{2(1)^2}}$	$\frac{1}{\sqrt{2\pi(1)^2}}\left(2-\frac{(0^2)+(1^2)}{1^2}\right)e^{\frac{(0^2)+(1^2)}{2(1)^2}}$	$\frac{1}{\sqrt{2\pi(1)^2}}\left(2-\frac{(1^2)+(1^2)}{1^2}\right)e^{\frac{(1^2)+(1^2)}{2(1)^2}}$

0	-0.2420	0
-0.2420	0.7979	-0.2420
0	-0.2420	0

2. Konvolusi citra input dengan *filter* h diatas

0	0	0	0	0
0	0*222 + (-0.242*205) + (0*177) + (-0.242*202) + (-0.7979*213) + (-0.242*167) + (0*184) + (-0.242*+165) + (0*176)	0*205) + (-0.242*177) + (0*207) + (-0.242*213)+ (-0.7979.*167) + (-0.242*188) + (0*165) + (-0.242*176) + (0*176)	…	0
0	…	…	…	0
0	…	…	…	0
0	0	0	0	0

$$\nabla^2 G_\sigma * I =$$

-399169.92	-426647.12	-426614.03	-441948.46	-426612.53	-458876.30	-466959.32
-424210.75	-499228.08	-472607.43	-443559.56	-495193.62	-468563.38	-457276.26
-342747.97	-461295.73	-419362.44	-353226.31	-510505.26	-526648.17	-483095.29
-502483.13	-470160.83	-460502.57	-486352.15	-462896.79	-510498.84	-493571.43
-475821.20	-447595.12	-516975.19	-454856.65	-431462.64	-474194.64	-451629.29
-508897.99	-526657.99	-558893.72	-516969.55	-437924.33	-377400.88	-476631.55
-524222.22	-358855.02	-528235.03	-474203.72	-570215.22	-471012.90	-407253.77

3. Lakukan *Zero Crossing Detection* , dimana untuk setiap *subimage* pada koordinat (x,y) ukuran 3x3, dihitung intl, int2, in3, dan int 4 sebagai berikut:

   Intl = (-399169.92-426647.12-424210.75-499228.08) = -1749255.86

   Int2 = -426647.12 -426614.03-499228.08-472607.43) = -1825096.66

   lnt3 = (-424210.75 -499228.08-342747.97-461295.73) = -1727482.52

   Int4 = (-499228.08-461295.73-472607.43-472607.43) = -1852493.68

   Maxval = max(int1,int2,int3,int4) = -1852493.68

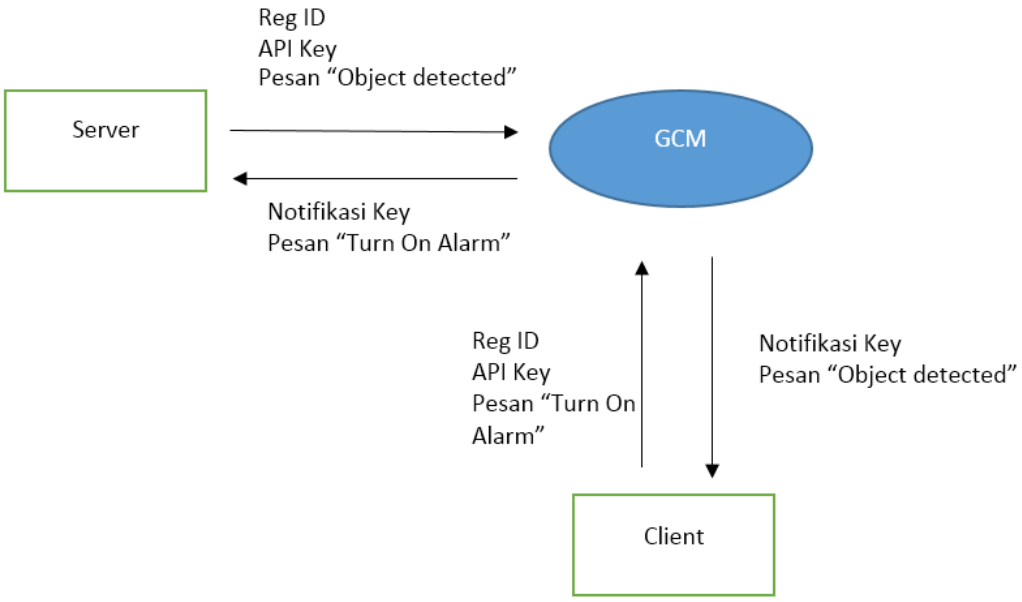
   Minval = min(int1,int2,int3,int4) = -1727482.52

   Jka maxval > 0 dan minval < 0 maka citra output G(x,y) = 255 (tepi) jika tidak G(x,y) = 0

   jadi karena maxval < 0 dan minval < 0, maka G(x,y) = 0 (bukan tepi).

3.1.3 Analisis Komunikasi Backend dan Frontend

Analisis komunikasi *backend* dan *frontend* merupakan cara komunikasi antara komputer server dan mobile dengan *google cloud messaging* yang berfungsi untuk mengirim notifikasi dari server ke client dengan mengirimkan registrasi ID dan API key yang disediakan oleh google devloper, begitu pula dari client ke server ialah mobile mengirim reistrasi ID dan notifikasi key saja, data yang dikirim berupa pesan yang nantinya akan dibaca oleh server dan di terjemahkan kedalam bahasa mesin guna melakukan aksi yang harus dilakukan oleh aplikasi server. Untuk alur komunikasi yang sedang berjalan dapat dilihat pada Gambar 3.16 berikut :



Gambar 3.16 Komunikasi Client Server dengan GCM (Google Cloud Messagging)

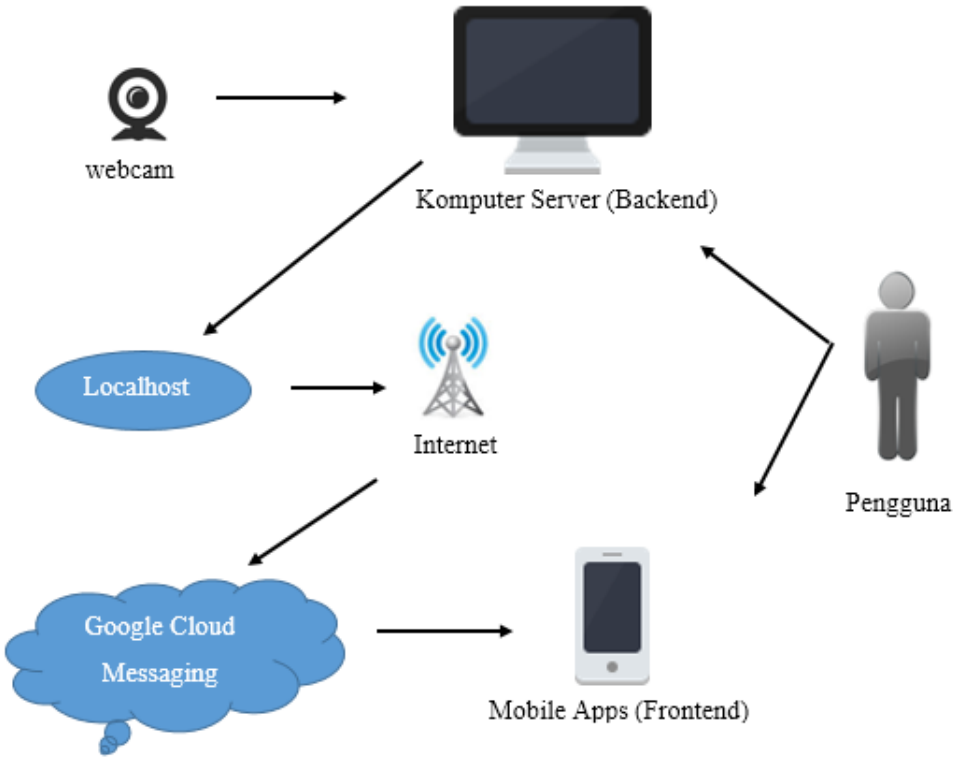
Keterangan :

Reg ID : 857065105553

API Key : `AIzaSyDZq_TJX3Eb1GBGUMWKBVz0-J9Hjzoqkuo`

3.2 Deskripsi Umum Sistem

Deskripsi umum adalah gambaran umum mengenai sistem yang akan dibangun, berikut adalah alur gambaran sistem yang akan dibangun :



Gambar 3.17 Deskripsi Umum Sistem

3.2.1 Sistem Backend

Sistem backend merupakan sistem atau aplikasi yang akan dibangun berupa aplikasi desktop yang berfungsi untuk membantu kamera webcam untuk mengolah gambar yang di-capture untuk dianalisis, sehigga dapat diketahui nilai citra pada objek yang ter-*capture* oleh kamera webcam. Sistem ini dibangun dengan menggunakan bahasa pemrograman J2SE (*Java 2 Standard Edition*) dan Bahasa pemrograman Android.

3.2.2 Sistem *Frontend*

Sistem *Frontend* merupakan sistem atau aplikasi yang akan dibangun untuk platform di mobile yang berfungsi untuk menampung notifikasi bahawa terdapat

data baru yang masuk ke *cloud storage* GCM. Sistem atau aplikasi ini dibangun dengan menggunakan bahasa pemrograman java Android.

3.2.3 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non-fungsional merupakan kebutuhan yang berhubungan dengan perangkat keras (*hardware*) yang menunjang kerja sistem yang dibangun oleh *software devloper*. Sistem operasi membutuhkan perangkat keras (*hardware*) yang *suport* dengan sistem operasi yang akan digunakan. Serta disesuaikan dengan aplikasi yang dibangun baik spesifikasi *hardware* yang digunkan maksimum maupun minimum.

1. Analisis kebutuhan perangkat keras (*Hardware*)
2. Analisis kebutuhan perangkat lunak (*Software*)
3. Analisis kebutuhan pengguna (*user/brainware*)

3.2.3.1 Analisis Kebutuhan Perangkat Keras

Untuk spesifikasi kebutuhan perangkat keras disini terbagi menjadi 2 perangkat keras berbeda yang akan digunakan, maka dapat diuraikan sebagai berikut :

1. Spesifikasi Kebutuhan Perangkat Keras Komputer
2. Spesifikasi Kebutuhan Perangkat Keras Mobile (Android)

3.2.3.2 Spesifikasi Kebutuhan Perangkat Komputer

Spesifikasi kebutuhan perangkat keras komputer dapat kita uraikan dengan menggunakan tabel spesifikasi kebutuhan perangkat keras yang dapat dilihat pada Tabel 3.2 Spesifikasi Perangkat Keras yang Dibutuhkan.

Tabel 3.2 Spesifikasi Perangkat Keras yang Dibutuhkan

No	Perangkat Keras	Spesifikasi
1	Processor	Kecepatan 2.5 GHz
2	Monitor	Monitor 19’, resolusi 1024x768
3	VGA On-board	Kecepatan 512 Mb
4	Optical Drive DVD-ROM	Standar
5	Keyboard	Standar

6	Mouse	Standar
7	Memori	1024 Mb
8	Kamera *webcam*	5Mp
9	Koneksi Internet	368 Kb/s

Spesifikasi perangkat keras minimum pada perangkat lunak yang akan dibangun pada unit personal komputer agar dapat menjalankan aplikasi optimal seperti yang terlihat pada Tabel 3.3 Spesifikasi Minimum Perangkat Keras yang Dibutuhkan.

Tabel 3.3 Spesifikasi Minimum Perangkat Keras yang Dibutuhkan

No	Perangkat Keras	Spesifikasi
1	Processor	Kecepatan 2.5 GHz
2	Monitor	Monitor 19', resolusi 1024x768
3	VGA On-board	Kecepatan 512 Mb
4	Optical Drive DVD-ROM	Standar
5	Keyboard	Standar
6	Mouse	Standar
7	Memori	1024 Mb
8	Kamera *webcam*	1.2 Mp
9	Koneksi Internet	40 Kb/s

3.2.3.3 Spesifikasi Kebutuhan Perangkat Keras Mobile Android

Spesifikasi kebutuhan perangkat keras mobile Android dapat kita uraikan dengan menggunakan tabel spesifikasi kebutuhan perangkat keras yang dapat dilihat pada Tabel 3.4 Spesifikasi Perangkat Keras Mobile Android yang Dibutuhkan**.**

Tabel 3.4 Spesifikasi Perangkat Keras Mobile Android yang Dibutuhkan

No	Perangkat Keras	Spesifikasi
1	Processor	Kecepatan 1.5 GHz
2	Sistem operasi	OS Android Versi jelly beans

3	Ram	1 Gb
4	Penyimpanan data	8 Gb
5	Koneksi internet	HSDPA, 2G, 3G

Spesifikasi perangkat keras minimum pada perangkat lunak yang akan dibangun pada unit personal komputer agar dapat menjalankan aplikasi optimal seperti yang terlihat pada Tabel 3.5 Spesifikasi Minimum Perangkat Keras Mobile Android yang Dibutuhkan.

Tabel 3.5 Spesifikasi Minimum Perangkat Keras Mobile Android yang Dibutuhkan

No	Perangkat Keras	Spesifikasi
1	Processor	Kecepatan 1.5 GHz
2	Sistem operasi	OS Abdroid versi froyo 2.2
3	Ram	256 Mb
4	Penyimpanan data	2 Gb
5	Koneksi internet	HSDPA, 2G

3.2.4 Analisis Kebutuhan Perangkat Lunak

Sistem operasi yang dibutuhkan yang digunakan untuk mendukung sistem controlling rumah dengan memanfaatkan kamera webcam sebagai sensor dengan metode zero crossing menggunakan GCM sebagai notifikasi Android baik untuk membangun maupun dalam pengimplementasiannya, adalah sebagai berikut :

a. Sistem Operasi Windows 7 32bit
b. Jdk 1.7 32bit atau lebih tinggi lagi
c. Jre 1.7 32bit atau lebih tinggi lagi
d. Netbeans 7.3
e. Android Froyo 2.2

3.2.5 Analisis Kebutuhan Pengguna (*User / Brainware*)

Analisis dan spesifikasi kebutuhan diperlukan agar kemampuan perangkat lunak yang dibangun menjadi jelas. Beberapa analisis dan kebutuhan yang berkaitan dengan perangkat lunak yang akan dibangun nanti yaitu analisis dan kebutuhan pengguna. Adapun karakteristik dari pengguna adalah sebagai berikut :

Pengguna minimal mampu menggunakan keyboard sebagai sarana penginputan not angka pada sebuah lagu.

a. Pengguna dapat membaca.
b. Pengguna minimal mempunyai pengetahuan mengenai komputer.
c. Pengguna minimal dapat menggunakan smartphone Android.

Berdasarkan analisis pengguna, dapat diambil kesimpulan bahwa pengguna yang ada telah memenuhi syarat sebagai pengguna sistem yang akan dikembangkan, sehingga tidak diperlukan pelatihan khusus mengenai penggunaan sistem, cukup berupa dokumen atau buku panduan untuk membantu menjalankan perangkat lunak.

3.3 Analisis Kebutuhan Fungsional

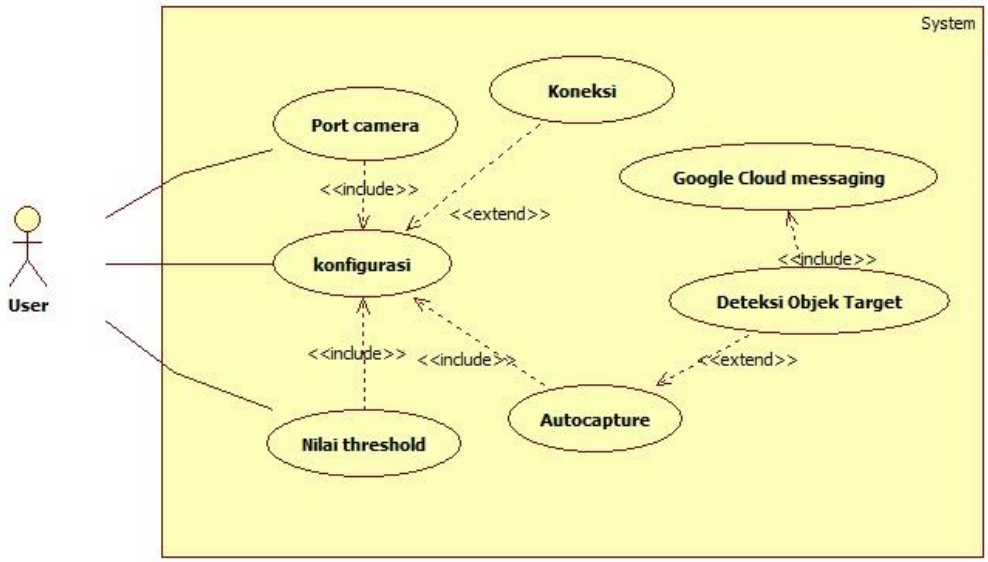
Analisis kebutuhan fungsional merupakan proses analisis sistem yang akan dibangun dengan menggunakan metode pemodelan tertentu yang bertujuan untuk memudahkan dan membantu dalam memabangun sistem.

Pemodelan dan perancangan yang akan digunakan dalam penelitian disini ialah dengan menggunakan pemodelan visual dengan UML (*Unified Modelling Language*), dimana pendekatan dalam memodelkannya yaitu dengan OOP *(Object Oriented Programming*).

3.3.1 Use Case Diagram Desktop (*Backend*)

Usecase merupakan gambaran umum sistem yang dimodelkan dengan komponen aktor, usecase dan asosiasi yang menghubungakan komponen-komponen tersebut. Use case sebagai model dalam sistem *controlling* rumah dengan memanfaatkan kamera *webcam* sebagai sensor dengan metode *Marr-Hilderth* menggunakan GCM sebagai notifikasi Android. Berikut adalah use case

yang menjelaskan kegiatan dalam sistem desktop (*backend*) dapat kita lihat pada Gambar 3.18 *Use Case Diagram Backend* :



Gambar 3.18 *Use Case Diagram Backend*

3.3.2 Skenario Use Case Desktop (*Backend*)

Skenario use case adalah deskripsi yang menerangkan kegiatan apa saja yang dilakukan aktor pada use case diagram yang ada di dalam sistem desktop (*backend*).

Tabel 3.6 Skenario Usecase Konfigurasi

Nama	Konfigurasi
Deskripsi	Konfigurasi merupakan fungsionalitas untuk mengecek koneksi maupun mengaktifkan Alarm
Aktor	User
Pre-condition	User menekan tombol Konfigurasi
Post-condition	Sistem menampilkan Form Konfigurasi

Aktor	Sistem
1. User menekan Tombol konfigurasi	

	2. Sistem menampilkan Form konfigurasi
3. User menceklist autocapture dan Turn On Alarm	
	4. Sistem menampilakn informasi

Tabel 3.7 Skenario *Use Case Port Camera*

Nama	Port Kamera
Deskripsi	Dimana user dapat memilih port kamera yang terhubung atau yang telah terdeteksi oleh sistem
Aktor	User
Pre-condition	User menekan tombol Port Kamera
Post-condition	Sistem menampilkan Form Port Kamera

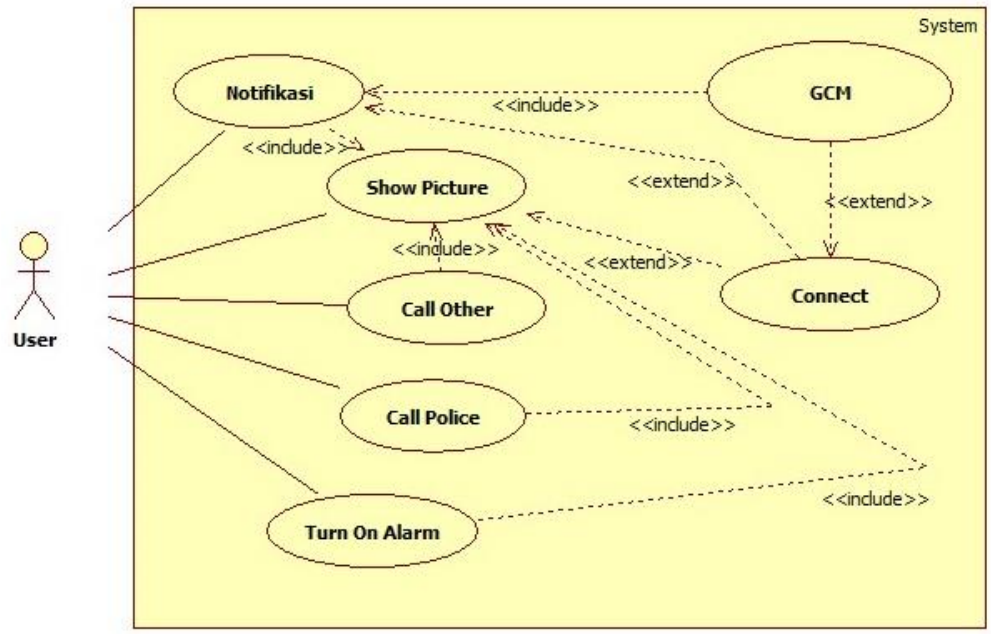
Aktor	Sistem
1. User menekan tombol Port Kamera	
	2. Sistem menampilkan form pilih Port Kamera
	3. Sistem mengecek ketersediaan port kamera
	4. Sistem menampilkan informasi port yang dipilih
5. User memilih port kamera	

Tabel 3.8 Skenario Use Case Nilai *Threshold*

Nama	Nilai Thershold	
Deskripsi	Mengurangi dan menambah nilai threshold pada objek yang terekam oleh kamera webcam	
Aktor	User	
Pre-condition	User menggerser ke kanan atau ke kiri	
Post-condition	Sistem mengubah gambar pada layar kamera	
Aktor		Sistem
1. User menggeser slider kekanan dan kekiri.		
		2. Sistem menambah dan mengurangi nilai threshold

3.3.3 Use Case Diagram Mobile (*Frontend*)

Usecase diagram mobile (*Frontend*) menjelaskan mengenai komponen yang ada pada sistem mobile, kegiatan seperti apa yang dapat dilakukan di dalam sistem *controlling* rumah dengan memanfaatkan kamera *webcam* sebagai sensor dengan metode *Marr-Hilderth* menggunakan GCM sebagai notifikasi Android. Berikut adalah use case yang menjelaskan kegiatan dalam sistem mobile Androis (*Frontend*) dapat kita lihat pada Gambar 3.19 *Use Case Diagram Mobile (Frontend)*.



Gambar 3.19 *Use Case Diagram Mobile (Frontend)*

3.3.4 Skenario Use Case Diagram Mobile (*Frontend*)

Skenario use case adalah deskripsi yang menerangkan kegiatan apa saja yang dilakukan aktor pada use case diagram yang ada di dalam sistem *mobile* (*Frontend*).

Tabel 3.9 Skenario Use Case Notifikasi

Nama	Notifikasi
Deskripsi	Peringatan bahwa objek target terdeteksi, atau sesuatu tertangkap oleh webcam
Aktor	User
Pre-condition	User men-tap tampilan notifikasi pada Android
Post-condition	Sistem menampilkan *Activity* Main Menu

Aktor	Sistem
	1. Sistem menampilkan notifikasi
2. User men-tap notifikasi	

	3. Sistem menampilkan Activity Main Menu

Tabel 3.10 Skenario Use Case *Show Pictures*

Nama	*Show Pictures*
Deskripsi	Fungsionalitas yang menampilkan image yang telah tertangkap
Aktor	User
Pre-condition	User men-tap tombol *Show Pictures*
Post-condition	Sistem menampilkan *Activity* kedua

Aktor	Sistem
1. User men-tap *Show Pictures*	
	2. Sistem menampilkan gambar yang dikirim.
3. User dapat menggeser ke kanan dan ke kiri untuk melihat gambar yang lain	

Tabel 3.11 Skenario Use Case *Call Police*

Nama	Call Police
Deskripsi	User dapat langsung menghubungi pihak yang berwajib
Aktor	User
Pre-condition	User men-tap tombol *Call Police*
Post-condition	Sistem menghubungi nomor tujuan

Aktor	Sistem
1. User men-tap Call Police	

	2. Sistem menampilkan dialog service telepon

Tabel 3.12 Skenario Use Case *Call Others*

Nama	Call Others
Deskripsi	User dapat langsung menghubungi orang terdekat atau yang lainnya.
Aktor	User
Pre-condition	User men-tap tombol *Call Others*
Post-condition	Sistem menampilkan kontak yang akan dihubungi

Aktor	Sistem
1. User men-tap Call Others	
	2. Sistem menampilkan layanan kontak nomor

Tabel 3.13 Skenario Use Case *Turn On Alarm*

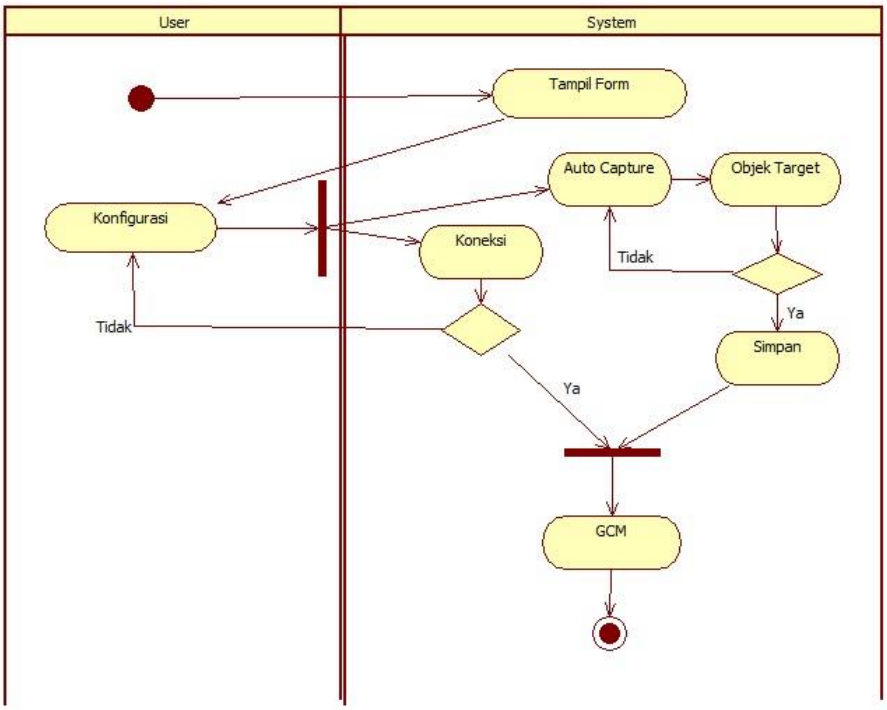
Nama	Turn On Alarm
Deskripsi	TOmbol untuk mengaktifkan alarm ketika image menangkap orang yang tidak diketahui.
Aktor	User
Pre-condition	User men-tap tombol Turn On Alarm
Post-condition	Sistem menampilkan informasi aktif atau tidaknya Alarm

Aktor	Sistem
1. User men-tap Turn On Alamr	
	2. Sistem menampilkan informasi koneksi.

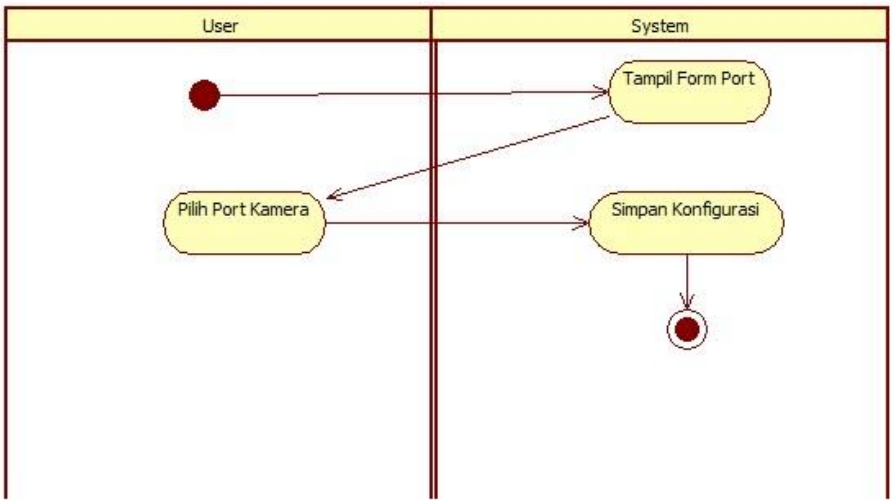
	3. Sistem menampilkan informasi alarm sdah diaktifkan.

3.3.5 Activity Diagram Desktop (*Backend*)

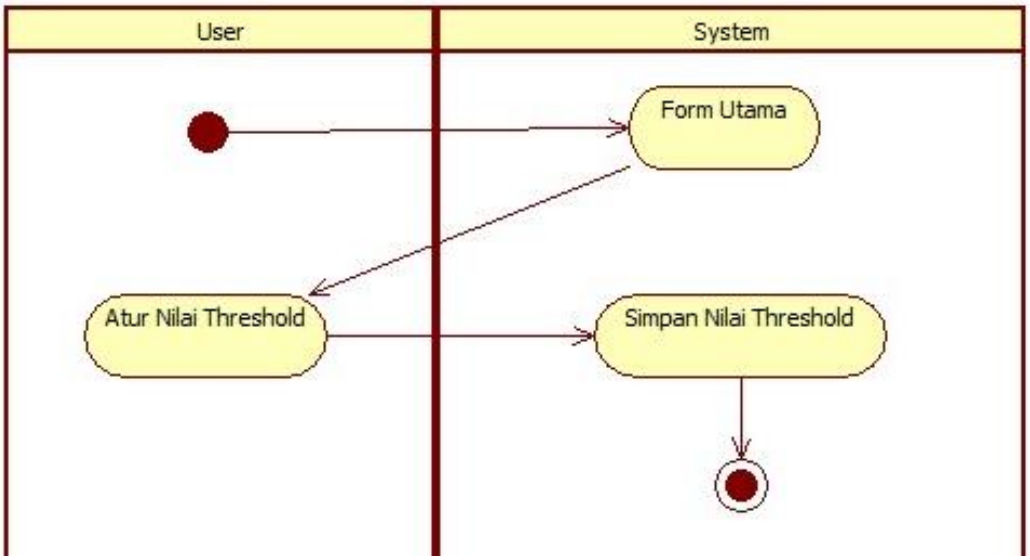
Untuk mengetahui alur kerja dari setiap kegiatan user dengan sistem dapat diketahui dengan menggunakan *activity diagram*. Berikut adalah *activity diagram* yang digunakan untuk mendeskripsikan kegiatan user dan sistem pada sistem desktop (*backend*) sebagai berikut :



Gambar 3.20 *Activity* Diagram Konfigurasi



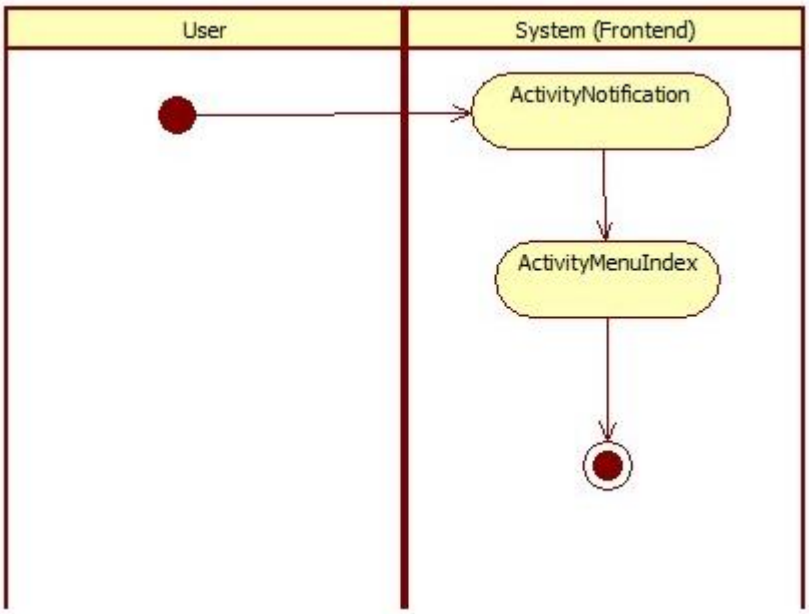
Gambar 3.21 *Activity Diagram Port* Kamera



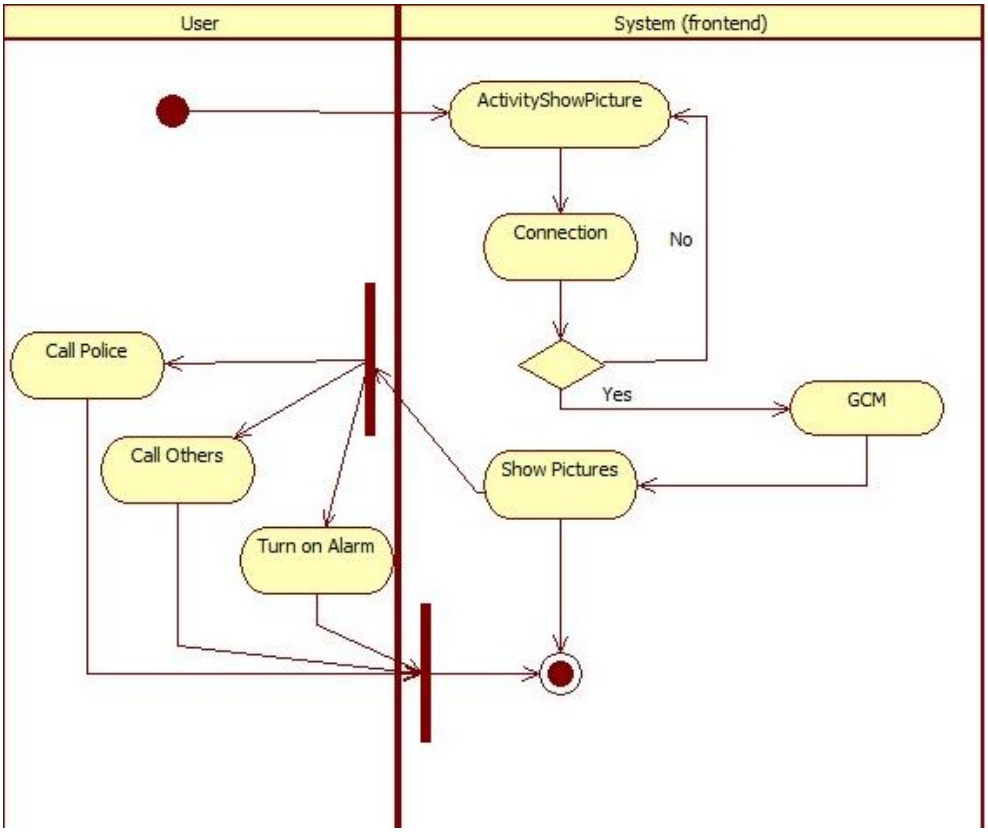
Gambar 3.22 *Activity Diagram* Nilai *Threshold*

3.3.6 Activity Diagram Mobile (*Frontend*)

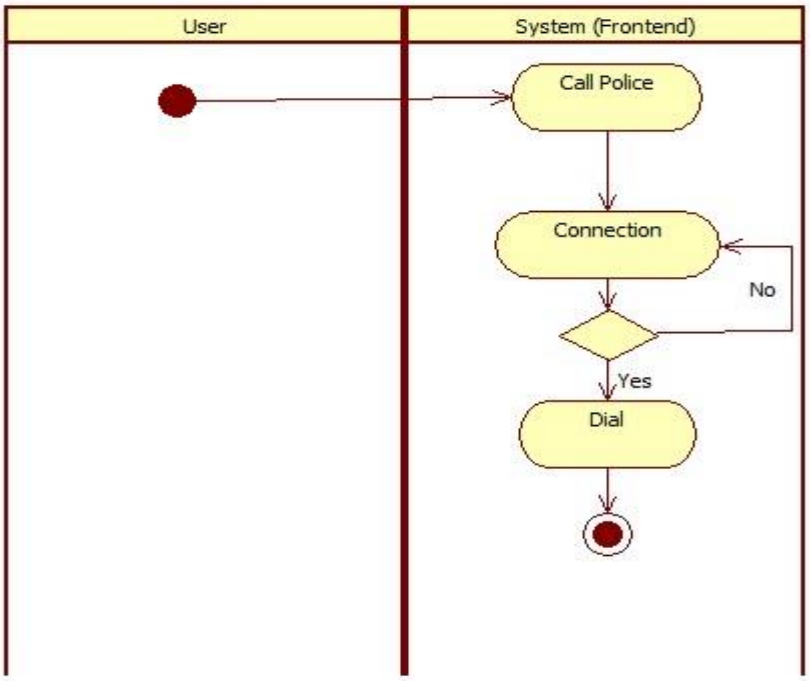
Berikut adalah *activity diagram* yang digunakan untu mendeskripsikan kegiatan user dan sistem pada sistem Mobile (*Frontend*) sebagai berikut :



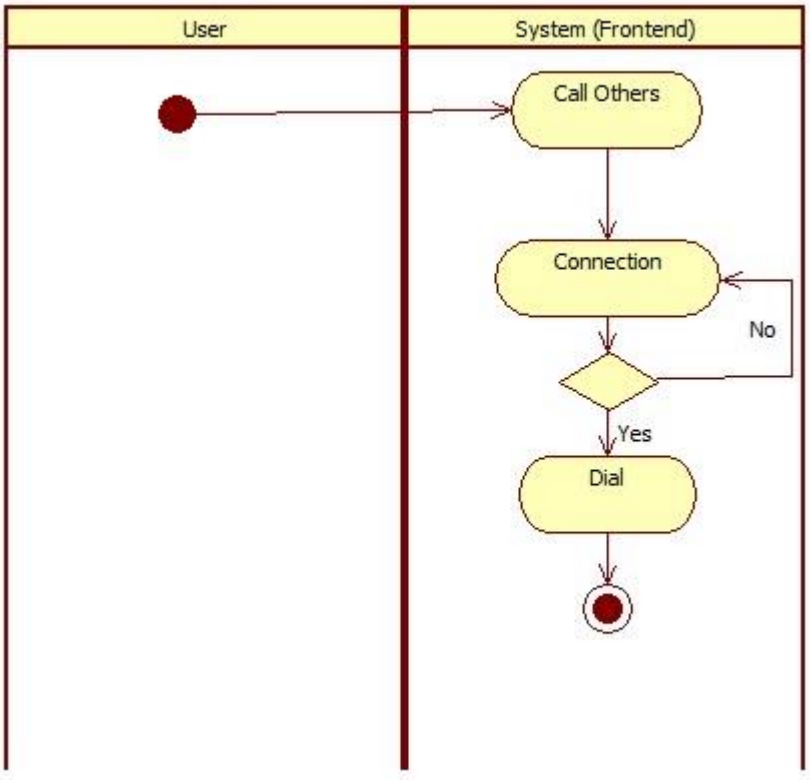
Gambar 3.23 *Activity Diagram* Notifikasi



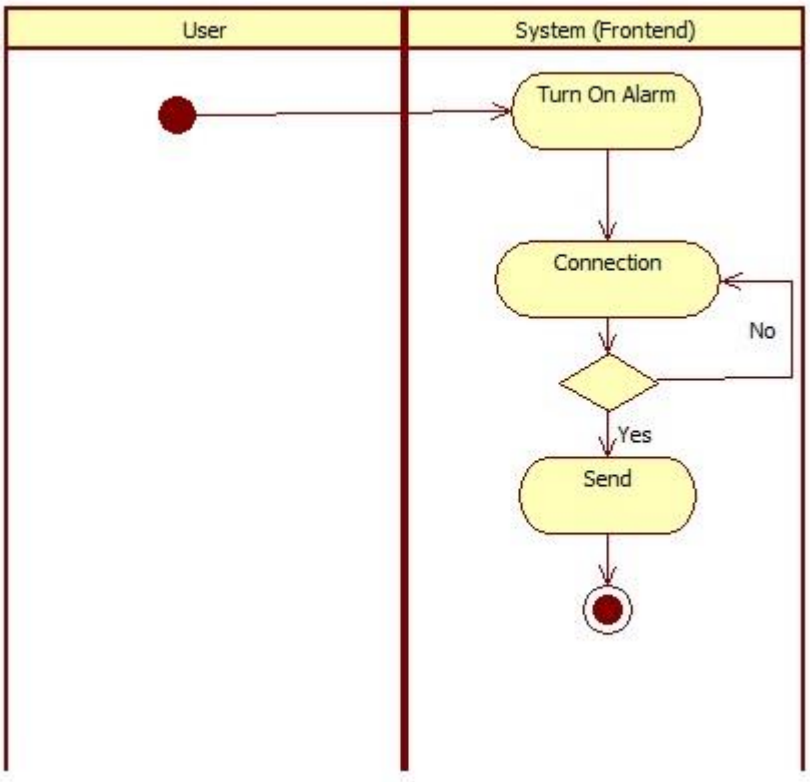
Gambar 3.24 *Activity Diagram Show Picture*



Gambar 3.25 ***Activity Diagram Call Police***



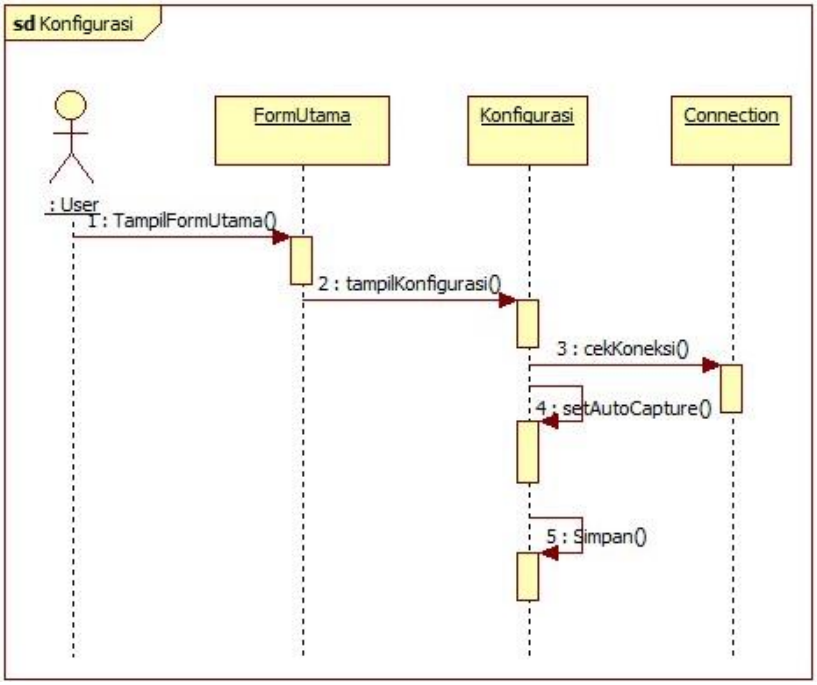
Gambar 3.26 ***Activity Diagram Call Others***



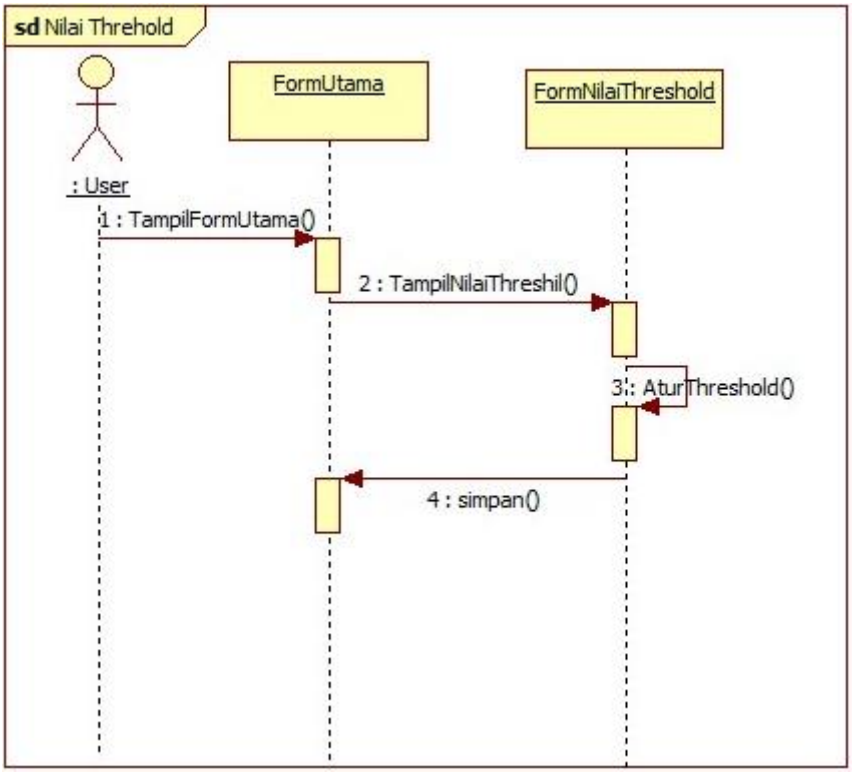
Gambar 3.27 *Activity Diagram Turn On Alarm*

3.3.7 Sequances Diagram Desktop (Backend)

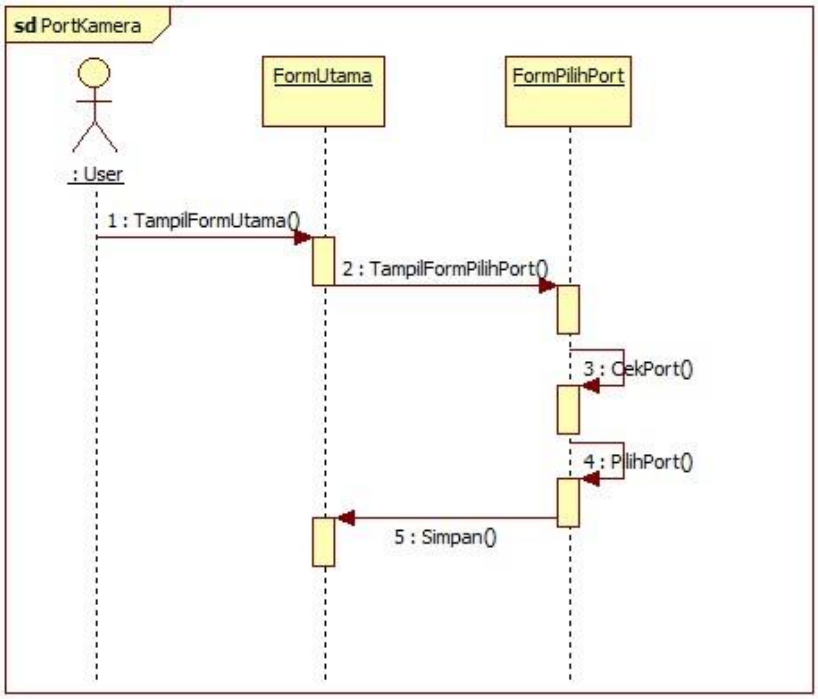
Sequances diagram Desktop *beckend* menjelaskan secara detail urutan proses yang dilakukan sistem untuk mencapai tujuan dari *Use Case beckend*. Interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi dan informasi yang diperlukan oleh masing-masing operasi.



Gambar 3.28 *Sequences Diagram* Konfigurasi



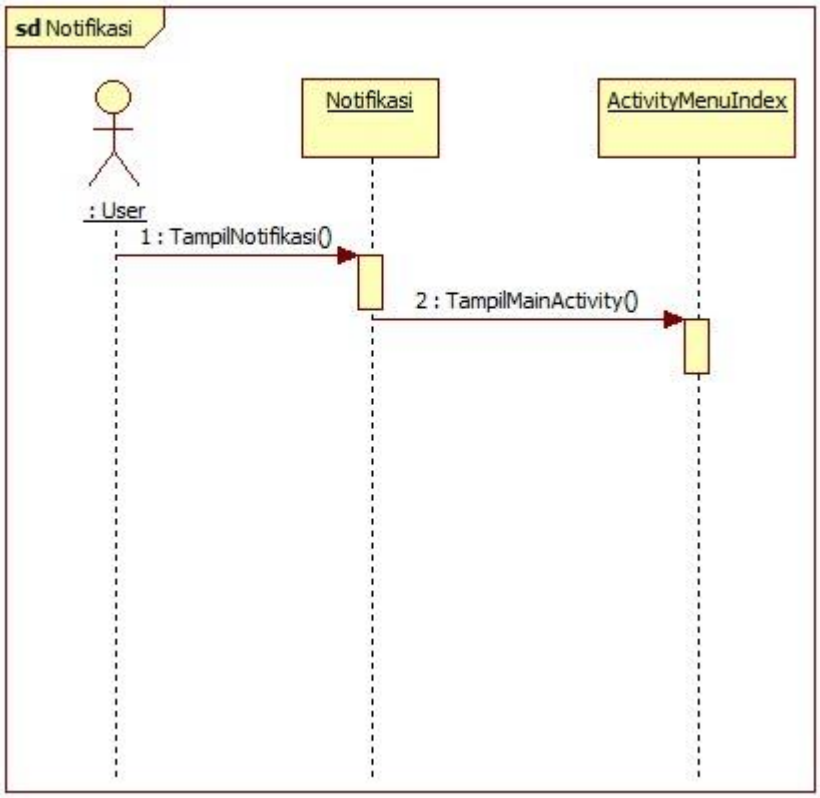
Gambar 3.29 *Sequences Diagram* Nilai Threshold



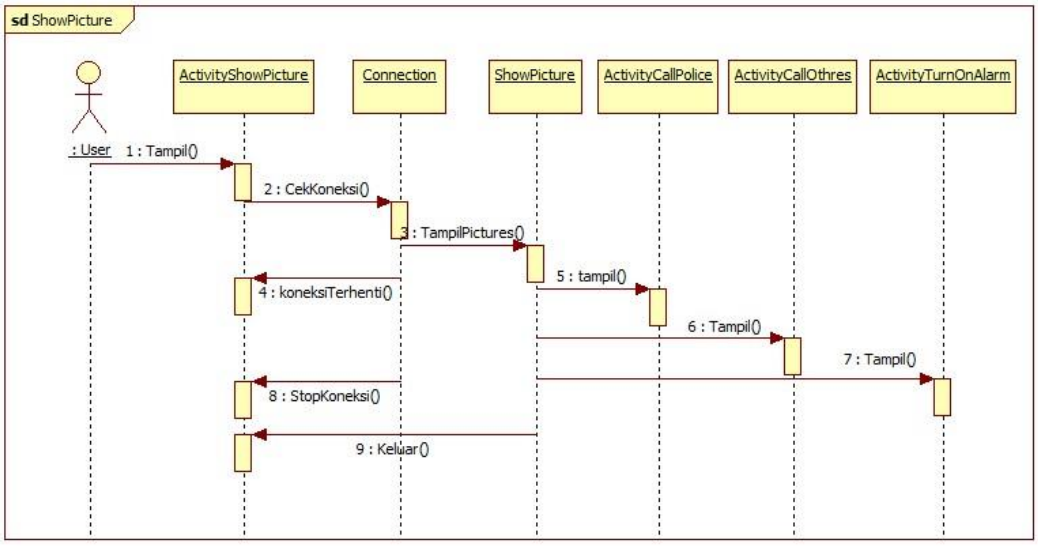
Gambar 3.30 *Sequences Diagram* Port Kamera

3.3.8 Sequances Diagram Mobile (*Frontend*)

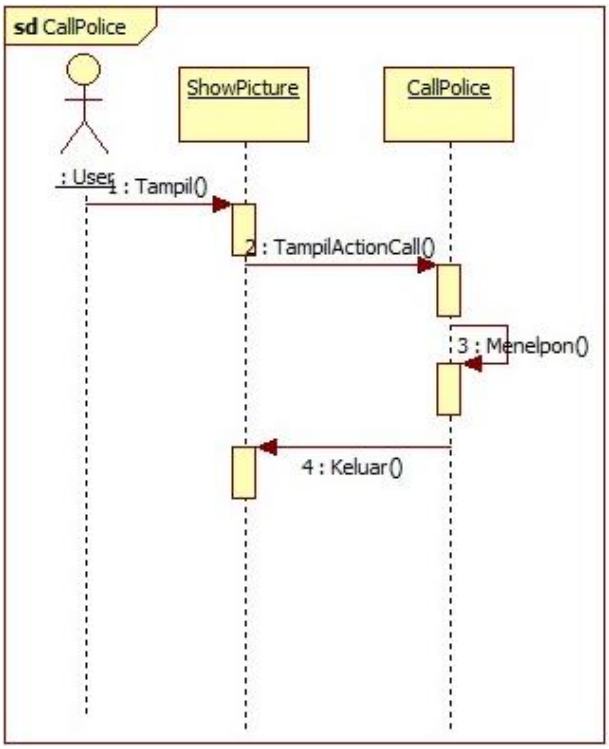
Sequances diagram Desktop *frontkend* menjelaskan secara detail urutan proses yang dilakukan sistem untuk mencapai tujuan dari *Use Case frontend*. Interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi dan informasi yang diperlukan oleh masing-masing operasi.



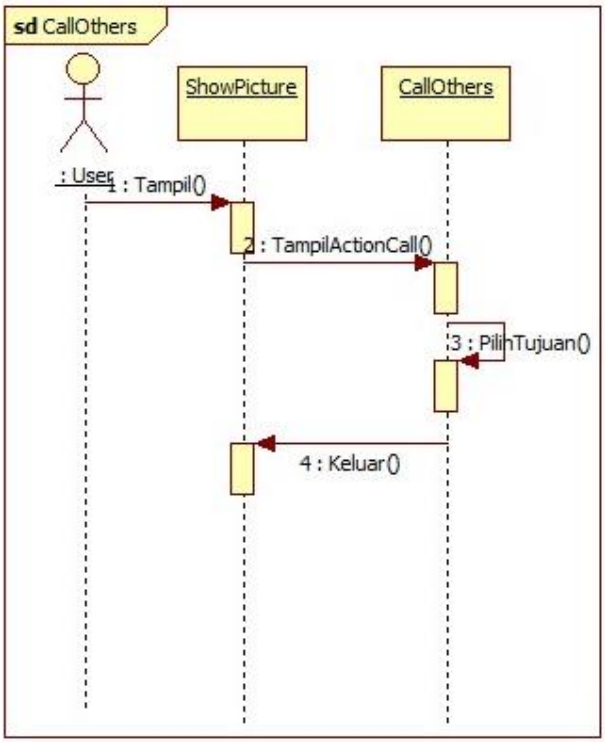
Gambar 3.31 *Sequences Diagram* Notifikasi



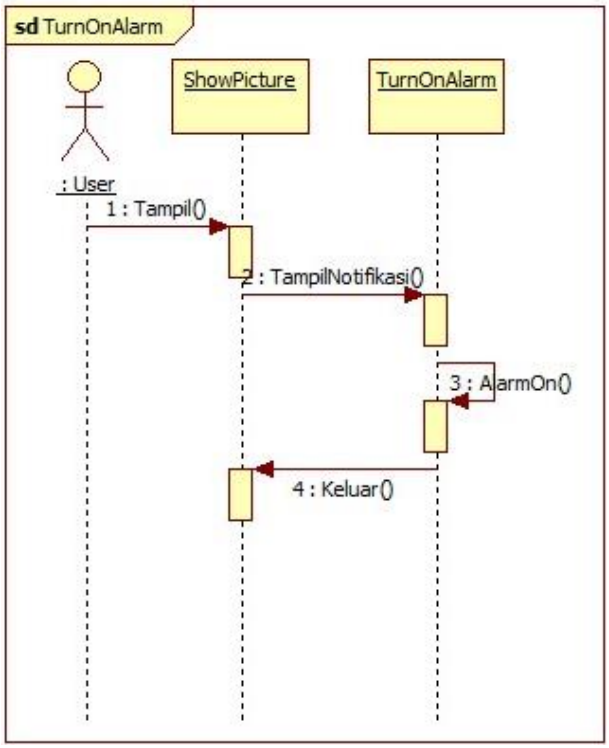
Gambar 3.32 *Sequences Diagram Show Pictures*



Gambar 3.33 ***Sequences Diagram Call Police***



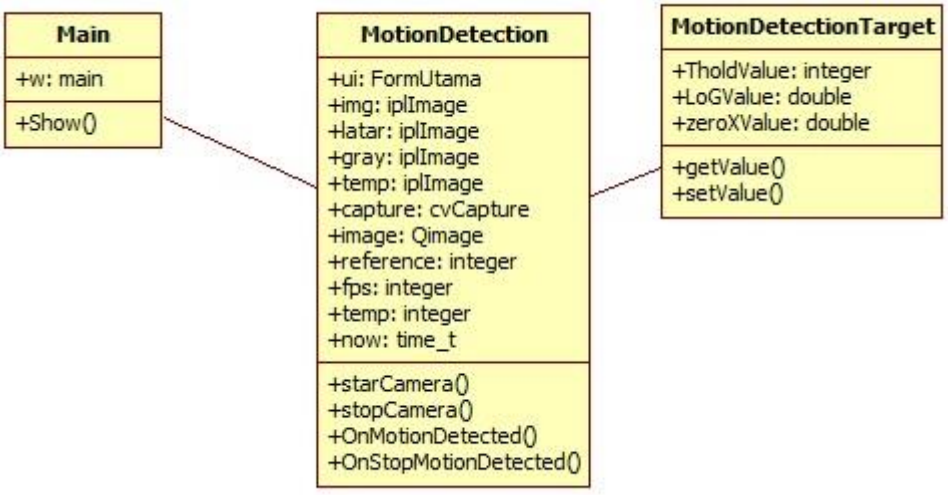
Gambar 3.34 ***Sequences Diagram Call Others***



Gambar 3.35 *Turn On Alarm*

3.3.9 Class Diagram Desktop (*Backend*)

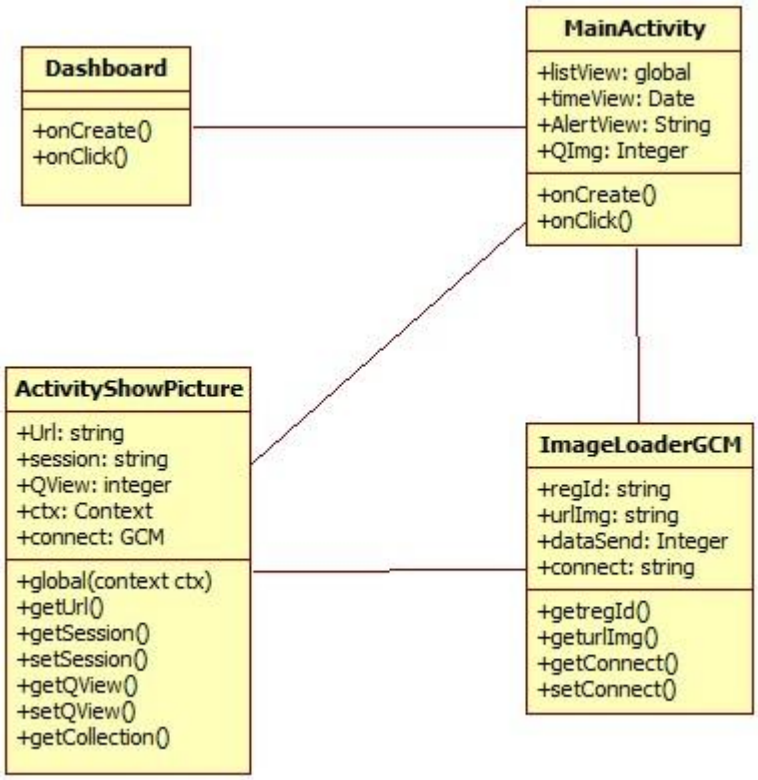
Class Diagram Desktop beckend merupakan kelas-kelas yang ada pada sistem yang memiliki hubungan antar satu kelas dengan kelas yang lainnya.



Gambar 3.36 *Class Diagram Desktop* (*Backend*)

3.3.10 Class Diagram Mobile (*Frontend*)

Class Diagram Desktop frontend merupakan kelas-kelas yang ada pada sistem yang memiliki hubungan antar satu kelas dengan kelas yang lainnya.



Gambar 3.37 *Class Diagram Mobile (Frontend)*

3.4 Perancangan Sistem

Pernacangan sistem merupakan penggambaran, perancangan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh. Tahampan ini meliputi konfigurasi komponen-komponen yang ada dalam perangkat lunak yang akan dibangun dengan penjelasan sebagia berikut.

3.4.1 Perancangan Arsitektur Perangkat Lunak

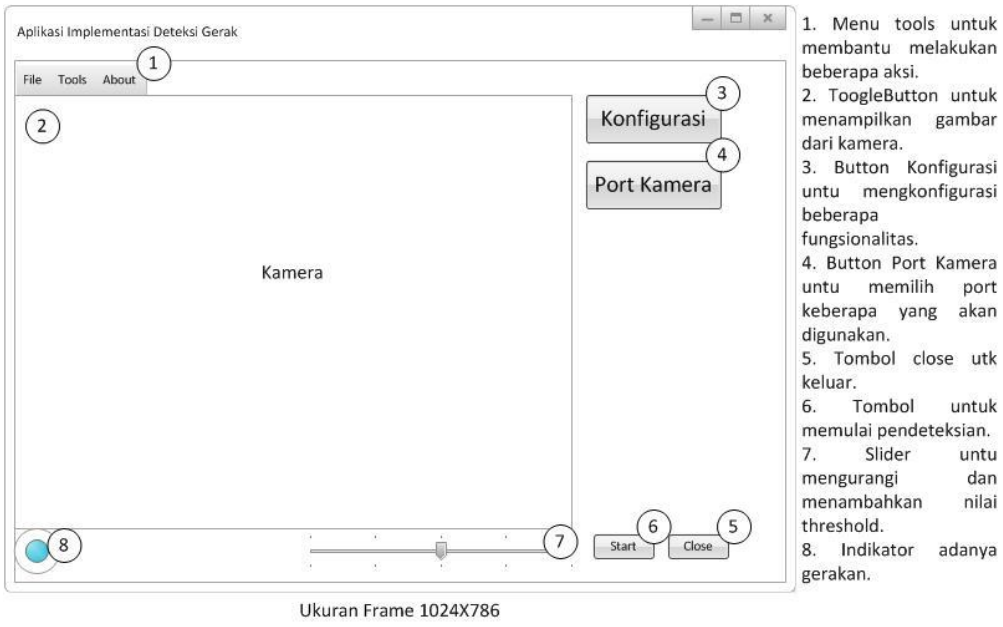
Perancangan arsitektur merupakan rancangan atau tata letak komponen-komponen yang akan dibuat sehingga dapat tergambarkan sistem yang akan dibangun. Perancangan yang akan dibangun terdiri dari 2 (dua) aplikasi yaitu

backend untuk aplikasi *desktop* dan *Frontend* sebagai aplikasi *mobile* Android sebagai alat *control* yang berupa notifikasi.

3.4.2 Perancangan Antarmuka Perangkat Lunak

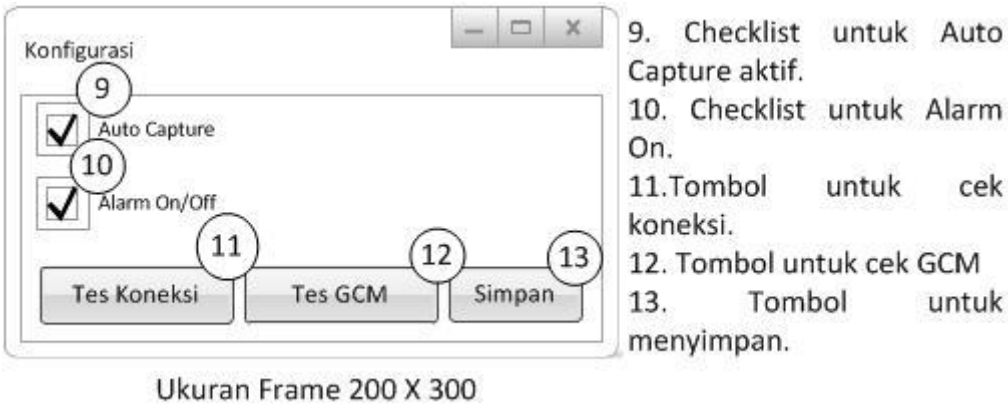
Perancangan antarmuka perangkat lunak bertujuan untuk mempermudah dalam mendisain tampilan sistem yang akan dibangun, sehingga sistem dapat digunakan dengan mudah oleh pengguna (*user friendly*).

1. Antarmuka Form Utama *Backend* (FB01)



Gambar 3.38 Form Utama

2. Antarmuka Form Konfigurasi *Backend* (FB02)



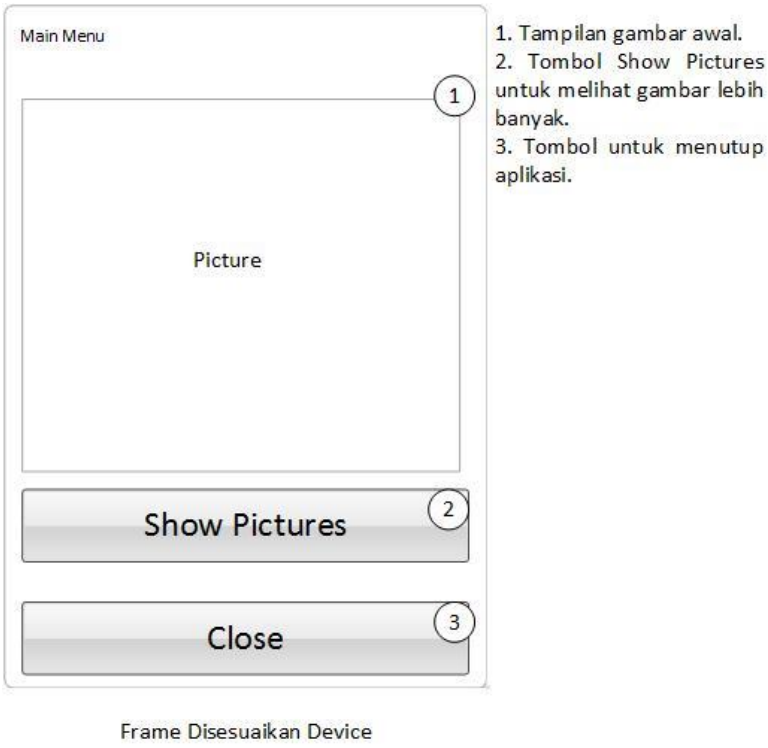
Gambar 3.39 Form Konfigurasi

3. Antarmuka Form Port Kamera *Backend* (FB03)



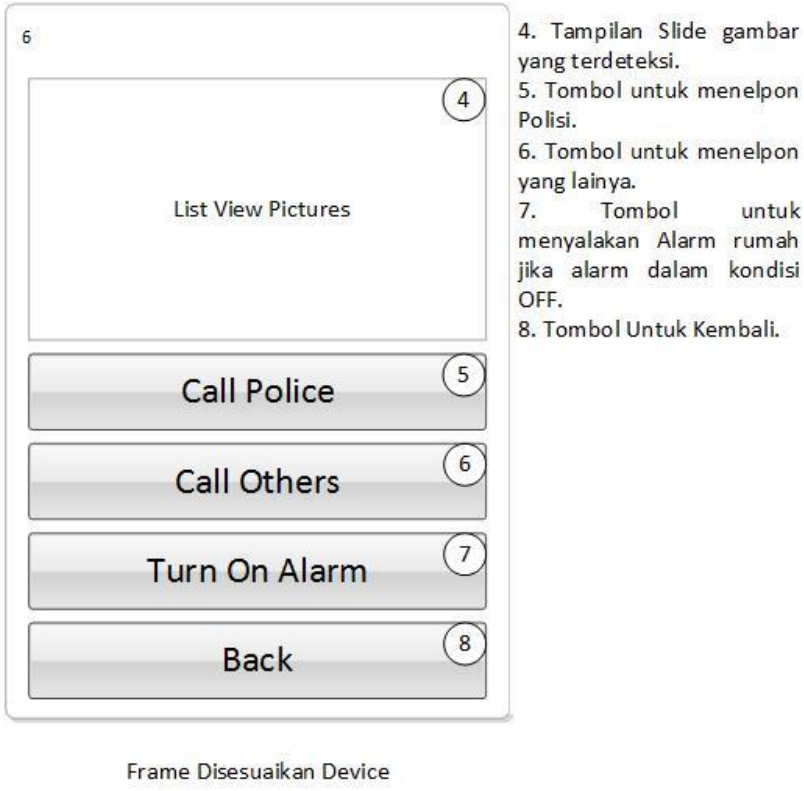
Gambar 3.40 Form Port Kamera

4. Antarmuka Activity Main Menu *Frontend* (FF04)



Gambar 3.41 Activity Main Menu

5. Antarmuka Activity *Show Pictures Frontend* (FF05)



Gambar 3.42 Activity *Show Pictures*

3.4.3 Perancangan Antarmuka Pesan

Perancangan antarmuka pesan merupakan desain untuk antarmuka yang menampilkan peringatan ketika terjadi kesalahan pengguna dalam mengoperasikan sistem dan pada saat sistem terjadi *error* atau *lost connection*.



Gambar 3.43 Pesan MB01



Gambar 3.44 Pesan MB02



Gambar 3.45 Pesan MB03



Gambar 3.46 Pesan MB04



Gambar 3.47 Pesan MB05



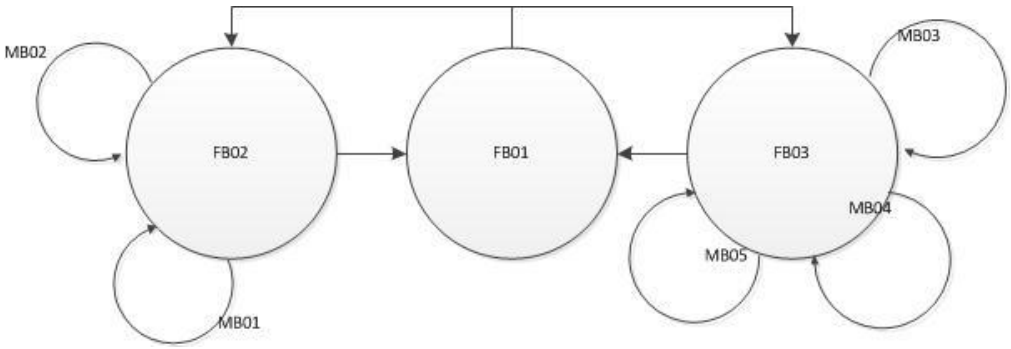
Gambar 3.48 Pesan MF01



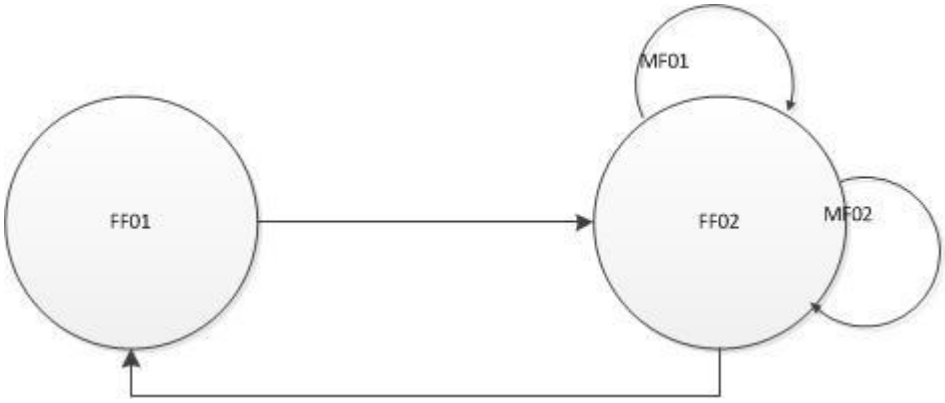
Gambar 3.49 Pesan MF02

3.5 Jaringan Semantik

Jaringan semantik merupakan jaringan yang menghubungakan antara objek satu dengan objek yang lainya, yang satu sama lain saling memberikan informasi dan data.



Gambar 3.50 Jaringan Semantik *Backend*



Gambar 3.51 Jaringan Semantik *Frontend*