BAB 3

ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Sistem yang dibuat merupakan informasi mobil berbasis augmented reality yang mana objek mobil yang ditampilkan pada brosur berupa gambar sebagai markernya yang kemudian di scan dengan aplikasi AR yang dibuat sehingga keluar objek mobil dalam bentuk 3D

Aplikasi ini dibuat dengan mengambil latar dilingkungan nyata yang kemudian gabungkan dengan obyek-obyek 3D melalui kamera. Orientasi dan posisi marker akan dideteksi dan ditangkap oleh kamera. Setelah marker terdeteksi oleh kamera, maka akan didapatkan matriks transformasi yang dapat digunakan untuk transformasi seluruh obyek yang ada dalam aplikasi.

3.1.1 Analisis Masalah

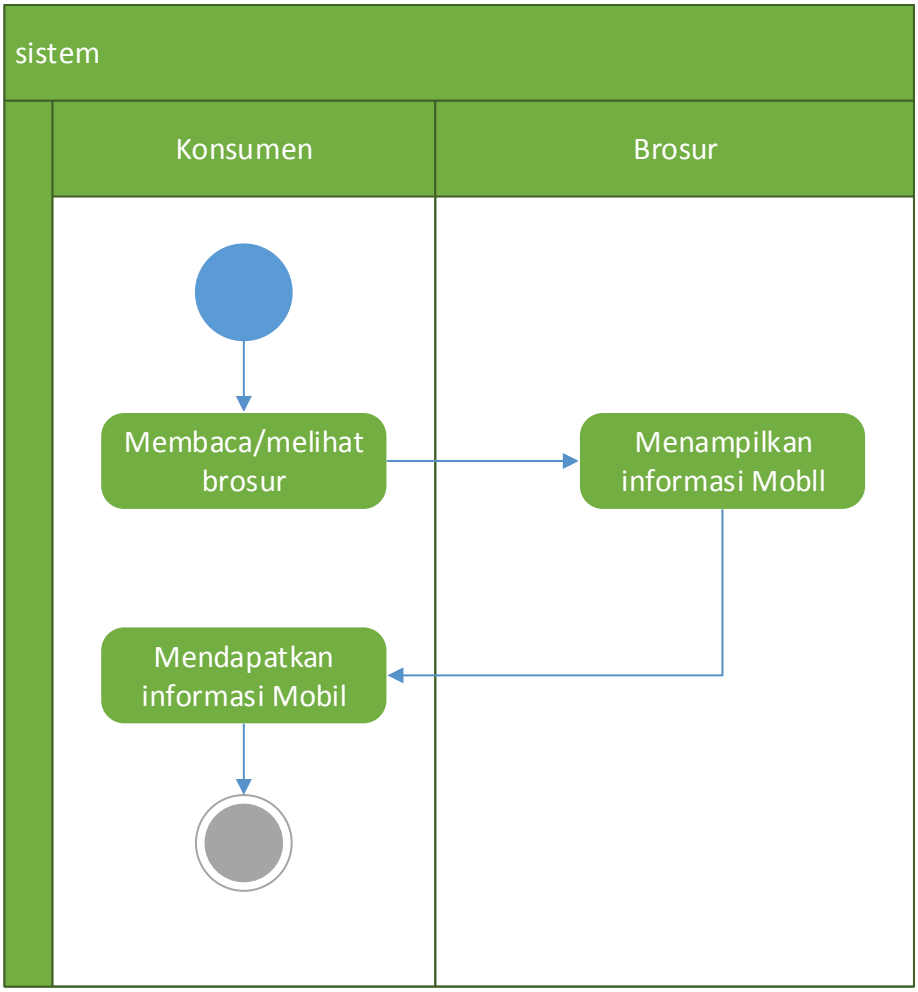
Analisis masalah yang dimaksud disini adalah jika konsumen ingin melihat mobil yang ingin dibeli secara detail maka konsumen harus datang langsung ke dealer selain itu karena keterbatasan waktu ,konsumen enggan datang langsung ke dealer. Serta media promosi saat ini masih berupa brosur konfensional yang hanya menampilkan objek kendaraan hanya sebatas gambar tanpa mengetahui tampilan kendaraan tersebut secara nyata.

Dengan teknologi yang berkembang sekarang ini, khususnya teknologi mobile yang unggul dengan sisi ruang geraknya (mobilitas), user friendly, serta praktis sehingga dapat menutupi kekurangan yang terdapat pada media saat ini. Merujuk terhadapa kekeurangan yang ada maka dibangun aplikasi mobile dengan menggunakan teknologi Augmented Realty (AR). Aplikasi ini bertujuan menampilkan informasi secara real time berupa bentuk 3D maupun video di layar ponsel. Identifikasi pengenalan gambar diambil dari kamera ponsel seluler dapat dikenali sebagai markerless. Markerless AR inilah yang berfungsi untuk

menampilkan informasi dari mobil yang dipasarkan dan ditampilkan informasi secara real time di layar ponsel seluler dalam hal ini smarthphone berbasis android.

3.1.2 Analisis Sistem yang Berjalan

Analisis Sistem atau analisis proses adalah tahapan yang memberi gambaran tentang sistem yang sedang berjalan sekarang. Analisis ini bertujuan untuk memberi gambaran yang lebih detail bagaimana cara kerja darisistem yang sedang berjalan. Prosedur pada proses media yang sedang berjalan dalam hal ini konsumen sekarang dapat di lihat seperti pada Gambar 3.1

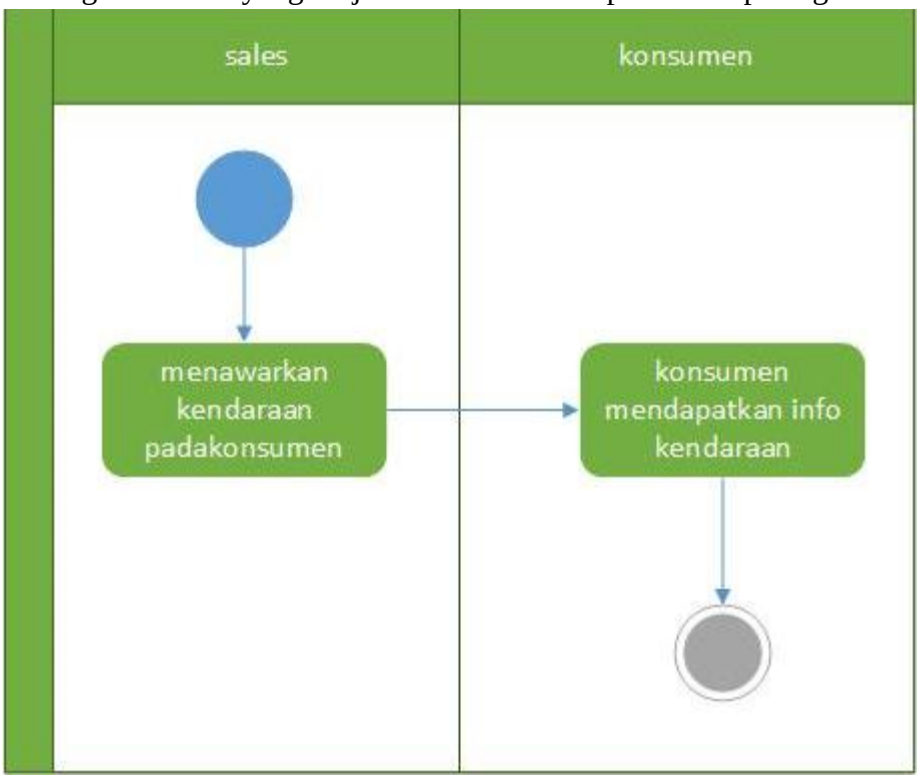


Gambar 3. 1 sistem yang berjalan

1. Konsumen adalah objek yang ingin mendapatkan informasi yang terdapat pada Brosur.

2. Brosur atau sebagai alat yang memberikan informasi dengan berisikan gambar yang ditampilkan masih dalam bentuk 2D.

Sedangkan sistem yang berjalan di sisi sales dapat dilihat pada gambar 3.2



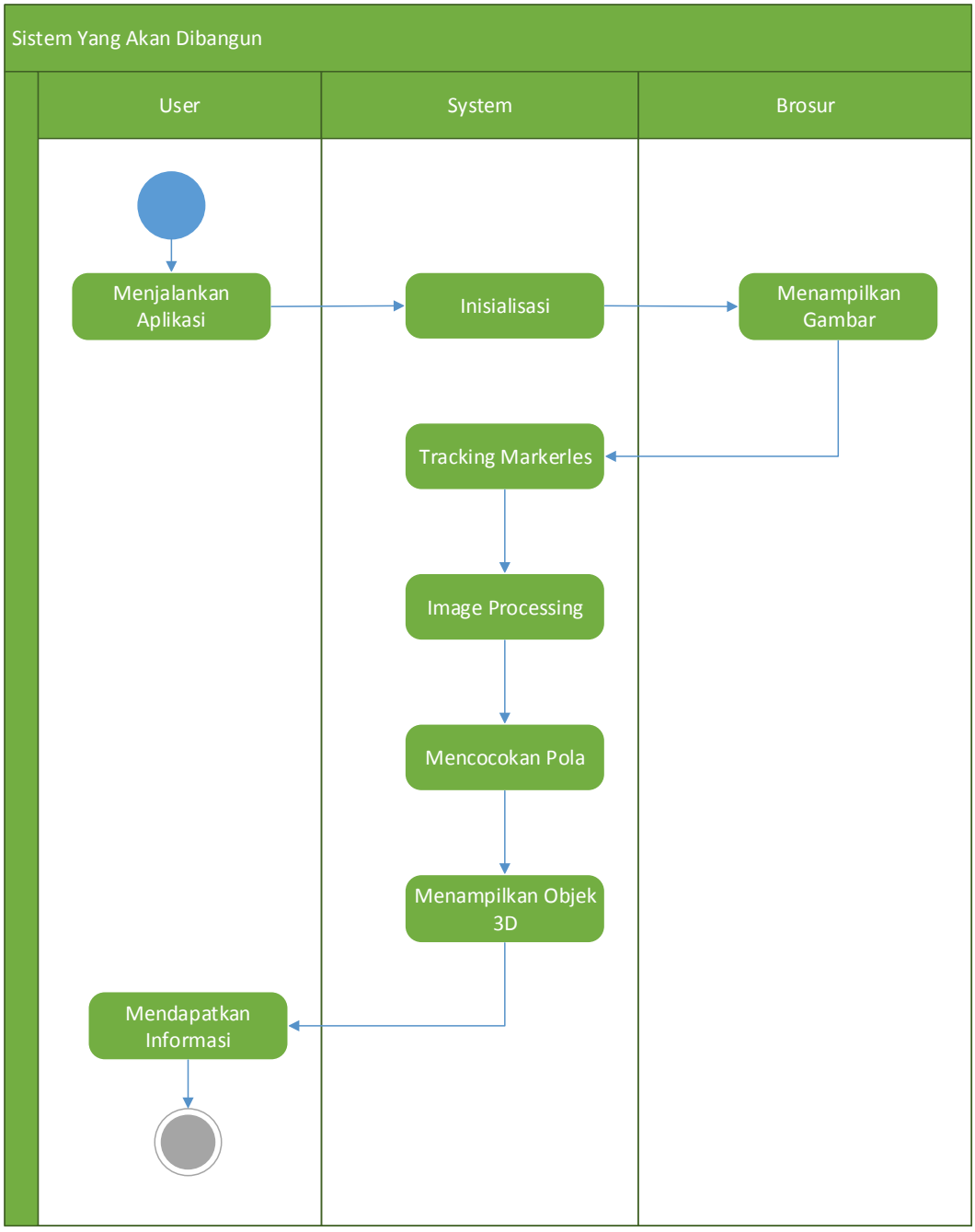
Gambar 3. 2 sistem yang berjalan di sisi sales

1. Sales adalah objek yang memberikan informasi pada konsumen
2. Knsumen adalah objek yang mendapatkan info kendaraan yang dijual dari sales

3.1.3 Analisis Sistem yang Akan Dibangun

Sistem yang akan dibangun adalah suatu aplikasi *mobile* menggunakan *platform* Android dengan teknologi *Augmented Reality*. Teknologi dengan aplikasi mobile dengan mudah diakses dimana saja dan sangat berkembang pada masa kini sehingga memudahkan *user* untuk mengakses aplikasi ini. Keunggulan dari teknologi *augmented reality* juga sangat menonjol dalam segi menampilkan suatu informasi secara *realtime* sehingga dapat diterapkan kedalam aplikasi, untuk dapat memberikan solusi dari permasalahan yang ada. Tujuan yang ingin dicapai dari perancangan aplikasi ini adalah dapat menampilkan informasi dari brosur secara

realtime. Proses aplikasi yang akan di bangun dengan menggunakan sistem markerles dapat dilihat pada Gambar 3.3 yang menggambarkan alur yang akan di rancang.



Gambar 3. 3 Sistem yang akan dibangun

1. *User* adalah seorang pengguna yang akan menggunakan akses handphone dalam mendapatkan informasinya.

2. Pada saat *user* menjalankan aplikasi, aplikasi ini akan menjalankan kamera telepon seluler secara otomatis untuk melakukan *tracking*marker serta mensingkronkan marker dengan kamera.
3. Brosur merupakan media informasi yang didalamnya terdapat marker yang memiliki bentuk yang berbeda dan yang nantinya akan di *tracking* oleh *handphone*.
4. Menghasilkan Animasi 3D yang merupakan objek yang dihsilkan oleh smarthphone yang telah memiliki aplikasi *Augmented Reality markerles*

3.1.4 Analisis Markerles

Markerless merupakan salah satu teknologi tracking. Tracking jenis ini menggunakan gambar sebagai markernya. Proses tracking ini menggunakan tekstur gambar (berkas jenis gambar) yang disimpan dalam database sebagai sumber referensinya, dan membandingkan tekstur yang tertangkap oleh kamera dengan tekstur gambar yang ada di database markernya.

3.1.5 Analisis algoritma

Analisis algoritma ini menjelaskan algoritma yang digunakan vuforia dalam mendektesi marker berikut adalah gambar alur dalam merubah gambar biasa menjadi marker

3.1.5.1 Algoritma FAST Corner detection

FAST (Feture Form Accelerated segment Test) adalah suatu algoritma yang dikembangkan oleh Edward Rosten, Reid Porter, and Tom Drummond. FAST *corner detection* ini dibuat dengan tujuan mempercepat waktu komputasi secara *real-time* dengan konsekuensi menurunkan tingkat akurasi pendeteksian sudut[6] .

FAST *corner detection* dimulai dengan menentukan suatu titik *p* pada koordinat (x_p , y_p) pada citra dan membandingkan intensitas titik *p* dengan 4 titik di sekitarnya. Titik pertama terletak pada koordinat (x, y_{p-3}), titik kedua terletak pada

koordinat (x_{p+3}, y), titik ketiga terletak pada koordinat (x, y_{p+3}), dan titik keempat terletak pada koordinat (x_{p-3}, y_p).

Jika nilai intensitas di titik *p* bernilai lebih besar atau lebih kecil daripada intensitas sedikitnya tiga titik disekitarnya ditambah dengan suatu intensitas batas ambang, maka dapat dikatakan bahwa titik *p* adalah suatu sudut. Setelah itu titik *p* akan digeser ke posisi(+1') dan melakukan intensitas keempat titik disekitarnyalagi. Iterasi ini terus dilakukan sampai semua titikppada citra sudah dibandingkan

FAST *corner detecteion* bekerja pada suatu citra sebagai berikut:

1. Tentukan sebuah titik *p* pada citra dengan posisi awal (,)
2. Tentukan keempat titik. Titik pertama (*n=1) terletak pada koordinat (,-3)* titik kedua(*n*=2) terletak pada koordinat +3, titik ketiga terletak pada koordinat (*n*=3) terletak pada koordinat (- 3,), titik keempat (*n*=4) terletak pada koordinat (- 3')
3. Bandingkan intensitas titikpusat *p* dengan keempat titikdisekitar. Jika terdapat paling sedikit 3 titik yang memenuhi syarat berikut, maka titik pusat *p*

$$C_p = \begin{cases} 1, & I_n < I_p - t \text{ or } I_n > I_p + t \\ 0, & otherwise \end{cases} \qquad (13)$$

Dengan

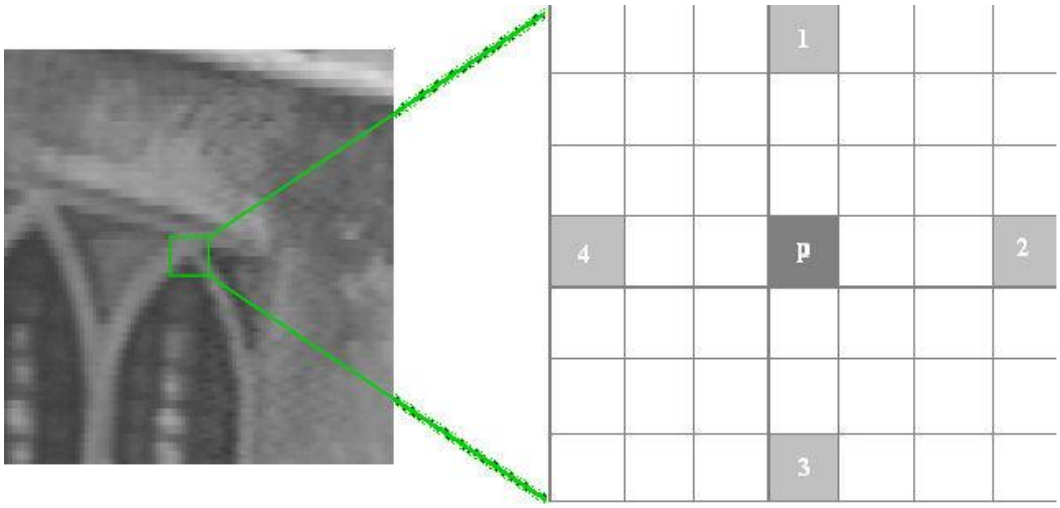
C_p =Keputusan titik p sebagai sudut, nilai 1 menunjukan bahwa titik merupakan suatu sudut, dan nilai 0 menunjukan bahwa titik bukanlah sudut

I_n =nilai intensitas piksel ke-n

I_p =nilai intensitas titik p

t =batas ambang nilai intensitas yang ditoleransi

4. Ulangi proses sampai seluruh titik pada citra sudah dibandingkanintensitasnya



3.1.5.2 Harris Corner Detection

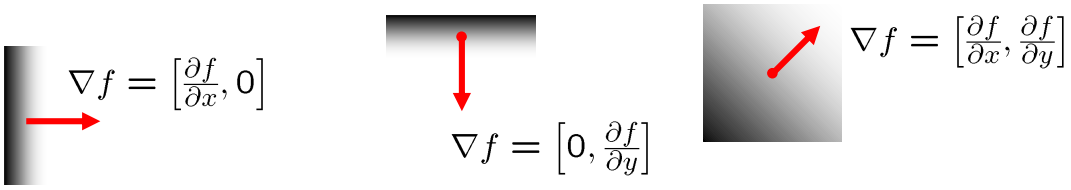
Harris corner detector (detektor sudut Harris) adalah detektor titik (sudut) yang populer karena mampu menghasilkan nilai yang konsisten walau dengan adanya rotasi, skala, variasi pencahayaan maupun *noise* pada gambar. Detektor sudut Harris didasarkan pada fungsi autokorelasi sinyal lokal di mana fungsi autokorelasi lokal akan menghitung perubahan lokal dari sinyal.

Untuk menentukan suatu titik dapat kita anggap sebagai titik sudut adalah jika kedua titik tersebut sisinya bertemu (misalkan yaitu titik dimana bertemunya dua buah garis miring). Titik sudut tidak akan bias didefinisikan pada piksel tunggal, karena disana hanya ada satu gradien per titik. Perilaku gradien ini jika kita cuplik dalam sebuah jendela kecil dapat dikategorikan bedasarkan statistiknya sebagai berikut :

- **Konstan**: Jika hanya sedikit atau tidak ada perubahan kecerahan (*brightness*)
- **Sisi/tepi/garis**: Jika terjadi perubahan kecerahan yang kuat pada satu arah
- **Flow** : garis Paralel

- **Pojok (*corner*)**: Jika terjadi perubahan kecerahan yang kuat dalam arah orthogonal

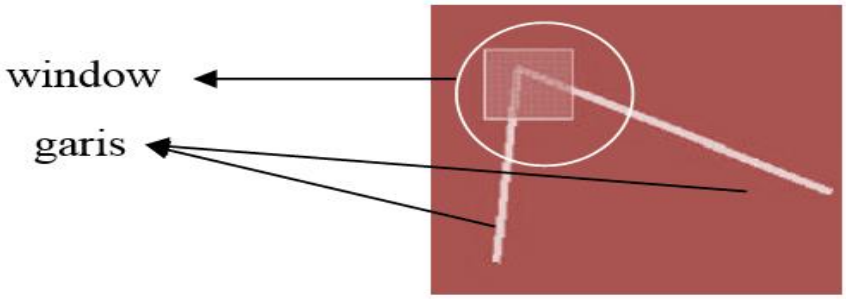
Gradien dari suatu citra adalah sebagai berikut : $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$



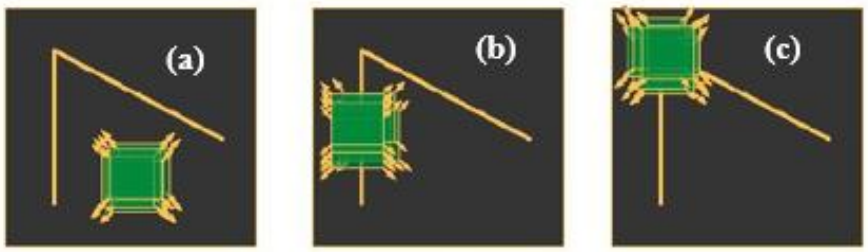
Gambar 3. 4 Gradien garis

Selanjutnya dapat ditentukan bahwa pada jendela pencuplikan untuk suatu titik disebut sebagai sudut adalah jika baik Ix = $\frac{\partial f}{\partial x}$ maupun Iy = $\frac{\partial f}{\partial y}$ tinggi.

Pada metode *harris corner detection* sutu titik pojok(sudut) akan mudah dikenali dengan menggunakan suatu nilai intensitas pada jendela (*window*) yang kecil yaitu pada susutu titik sudut dengan menggeser *window* tersebut ke segala arah sehingga didapatkan nilai perbedaan yang besar.

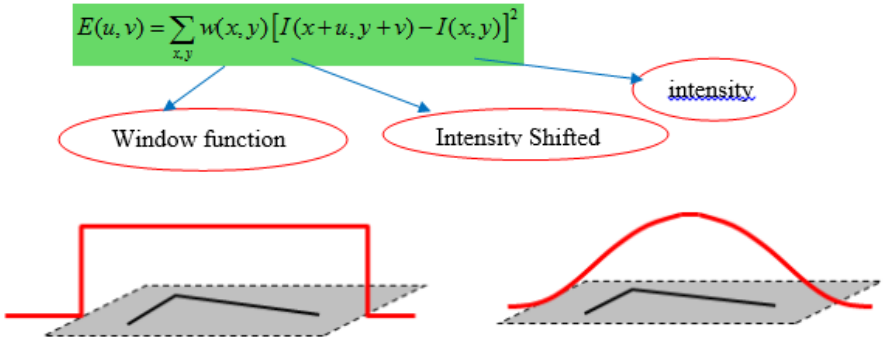


Gambar 3. 5 pencuplikan (windowing)



a. **Konstan/Flat**: tidak ada perubahan pada semua arah
b. **Sisi/tepi/garis** : tidak ada perubahan sepanjang arah garis
c. **Pojok (*corner*)**: terdapat perubahan yang signifikan pada segala arah

Secara Matematis :



Fungsi windowing W(x,y) → bernilai 1 jika didalam jendela atau 0 jika di luar atau menggunakan bisa juga dengan menggunkan fungsi Gaussian.

Nilai pada bagian *Intensity Shifted* dan *intensity* untuk bagian yg flat/konstan akan bernilai kecil, sedangkan untuk daerah yang mengandung corner akan bernilai besar. Yang ingin dicari adalah daerah dengan nilai E(u,v) yang besar.

Untuk perubahan kecil [u,v] kita gunakan pendekatan bilinier: $E(u,v) \cong [u,v] \quad M \begin{bmatrix} u \\ v \end{bmatrix}$

Dimana M adalah matrik 2x2 dihitung dari *image derivatives* (turunan gambar)

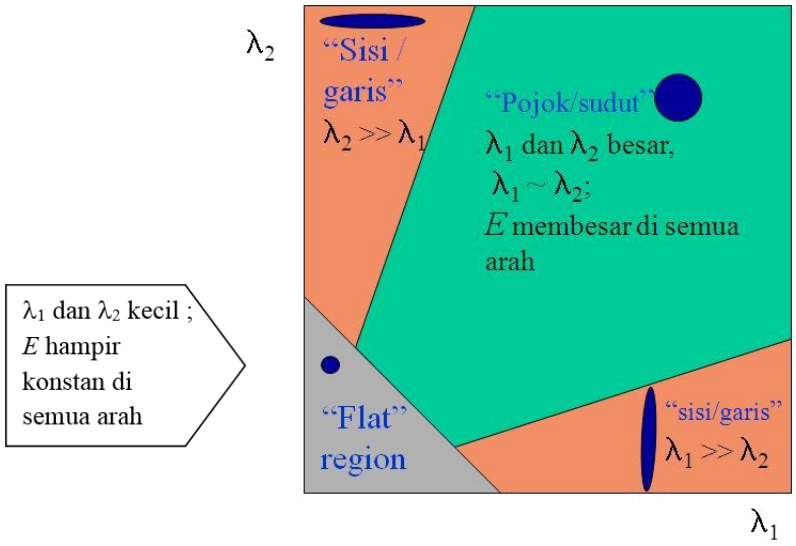


Intensitas perubahan pergeseran jendela: analisis eigenvalue

$$E(u,v) \cong [u,v] \quad M \quad \begin{bmatrix} u \\ v \end{bmatrix}$$

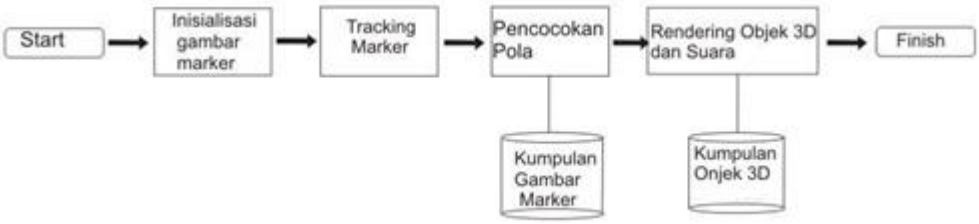
λ_1, λ_2 adalah *eigenvalue* dari M

Klasifikasi titik citra menggunakan nilai eigen M



3.1.6 Analisis Penentuan Markerless

Analisis alur sistem ini menjelaskan bagaimana proses yang dilakukan seperti yang terlihat pada Gambar 3.5



Gambar 3. 6 Diagram alur sistem augmented reality

Secara garis besarnya, dalam perancangan ada tiga bagian utama yaitu:

1. Inisialisasi Gambar Marker

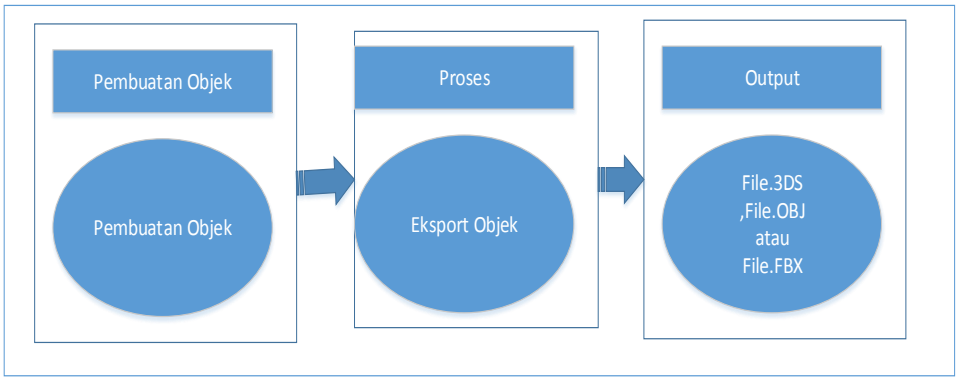
2. *Tracking* Markerless
3. *Rendering* Objek.

3.1.6.1 Inisialisasi

Pada tahap ini ditentukan marker yang akan digunakan, sumber input objek 3D yang akan digunakan .Pada bagian ini objek 3D diinisialisasi terlebih dahulu karena loading objek 3D memerlukan waktu yang cukup lama. Adapun tahap inisialisasi yaitu Inisialisasi Model 3D, Inisialisasi Video , Inisialisasi Marker yang digunakan dan Inisialisasi Informasi Text.

3.1.5.1.1 Inisialisasi Model 3D

Model 3D yang akan ditampilkan di-load terlebih dahulu pada engine *Unity3D.* Agar aplikasi dapat menampilkan objek 3D tertentu tanpa merubah atau membangun ulang aplikasi, diperlukan sebuah file format *.3DS , .OBJ atau .FBX* untuk menentukan objek 3D yang akan di-load di Unity3D. Pembuatan objek 3D sampai dengan ringgingnya menggunakan software Blender, setelah di buat lalu diekspor ke format *.3DS , .OBJ atau .FBX* agar dapat dieksport melalui Unity3D.

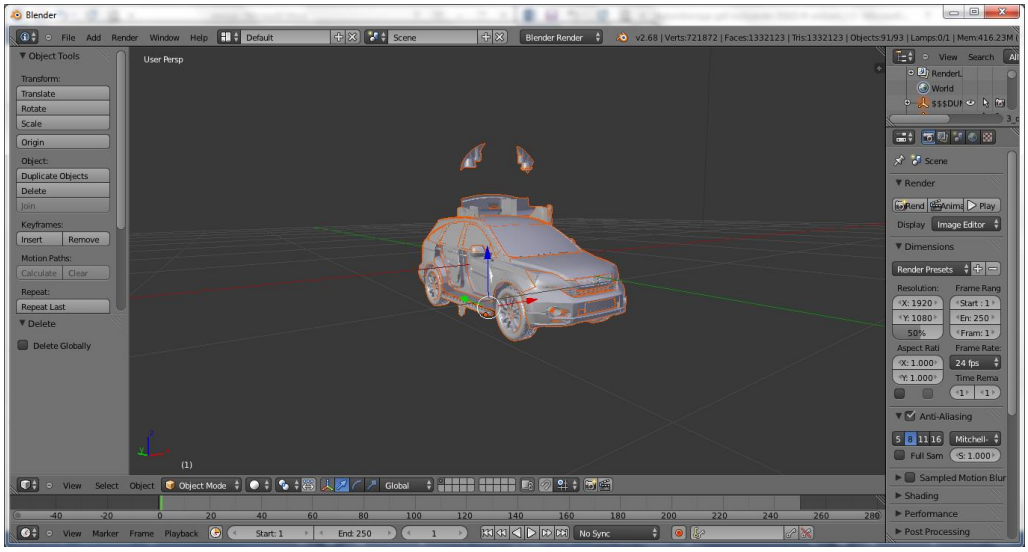


Gambar 3. 7 insialisasi model 3d

Tahapan dalam menganalisis bentuk dan format dari objek 3D yang digunakan.

3.1.5.2.1 Analisi Object 3D

Objek 3D meupakan objek yang dijadikan sebuah *output*yang hasilkan dihasilkan pada tahap proses *tracking* marker. Dalam analisis objek 3D yang digunakan pada *vuforia* harus berupa objek 3D yang berekstensi file .fbx obj dan .3ds sehingga file 3d tersebut dapat ditampilka pada unity engine gambar 3.6 menunjukan pembuatan objek 3D dengan menggunakan software Blender



Gambar 3. 8 Pembuatan Objek 3d deangan blender

Dalam proses pembentukan objek 3D ada beberapa langkah yang dilakukan :

1. Menyesuaikan objek 3D yang dibuat dengan menambahkan tekstur yang dibutuhkan.
2. Mengekspor objek 3D yang telah dirancang menjadi format .fbx Pada tahap mengekspor objek 3D dapat dilakukan dengan menggunaka software blender itu sendiri.

3.1.7 Inisialisasi

Pada tahap ini ditentukan marker yang akan digunakan, sumber input video nya, dan objek 3D yang akan digunakan .Pada bagian inisialisasi ini, objek 3D

diinisialisasi terlebih dahulu karena loading objek 3D memerlukan waktu yang cukup lama

3.1.7.1 Model 3D

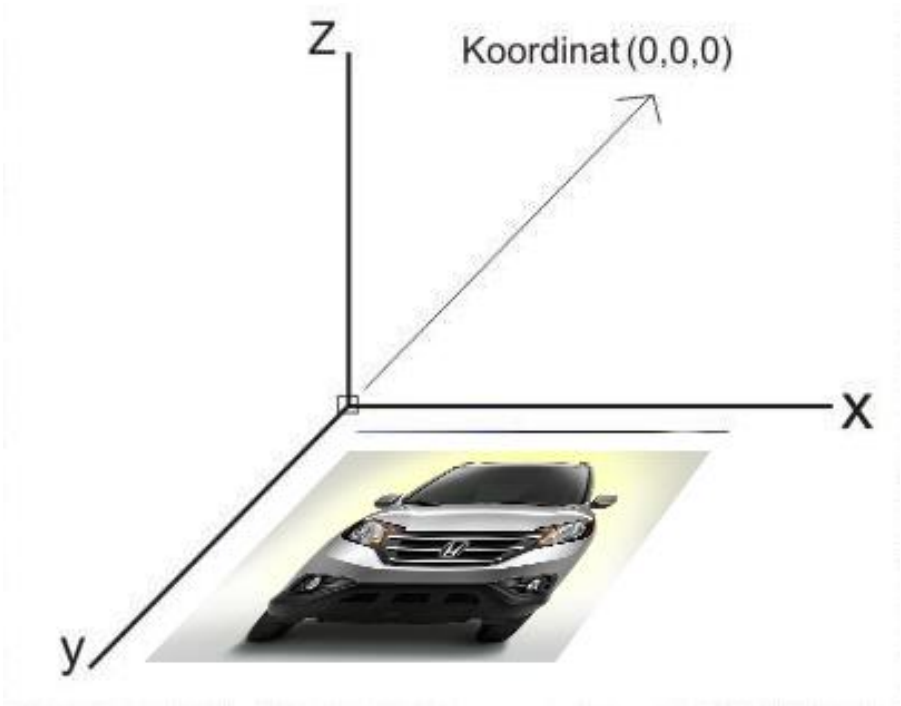
Model 3D yang akan ditampilkan di-load terlebih dahulu engine *Unity3D*. Agar aplikasi dapat menampilkan objek 3D tertentu tanpa merubah atau membangun ulang aplikasi, diperlukan sebuah file konfigurasi untuk menentukan objek 3D yang akan di-load sesuai dengan pola marker yang dideteksi

3.1.8 Perancangan Marker

Tahapan dalam menganalisis bentuk dan isi pada marker yang digunakan serta *marker* seperti apa saja yang digunakan

3.1.9 Analisis Marker

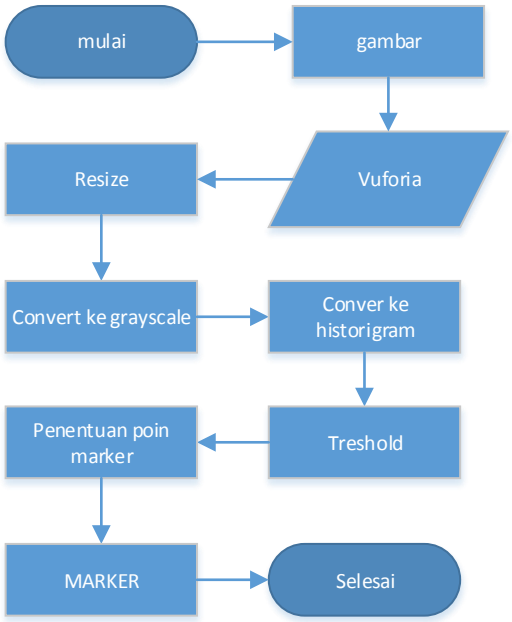
Marker merupakan perangkat keras lainnya yang merupakan salah satu bagian dari *Auigmented Reality* (AR) yang digunakan untuk membuat suatu aplikasi *Augmented Reality* (AR), pada umumnya marker merupakan sebuah persegi hitam dan ditengahnya terdapat bentuk atau ciri yang membedakan antara satu marker dengan marker lain yang akan menjadi *keypoint*, marker yang terdapat ditengah bisa berbentuk gambar, angka, huruf, atau apa saja, didalam pola marker yang merupakan ilustrasi hitam dan putih dengan batas hitam yang tebal dan latar belakang putih digunakan agar komputer mengenali posisi dan orientasi marker dan menciptakan dunia virtual 3D yaitu pada titik (0,0,0) dari 3 sumbu yaitu X,Y, dan Z.



Gambar 3. 9 Koordinat pada komputer

3.1.10 Pembuatan Markers

Marker yang digunakan pada analisis ini didapatkan dari gambar yang terdapat pada brosur yang sudah ada. Proses pembuatan gambar marker menjadi marker dilakukan oleh pihak Vuforia dengan cara mengupload gambar yang akan di-convert, melalui marker engine yang disediakan oleh pihak Vuforia. Setelah gambar di convert menghasilkan file dengan format .unitypackage. File tersebut kemudian dijadikan masukan pada Engin *Unity3D* untuk mendeteksi gambar yang dijadikan marker. Berikut tahapan atau alur dalam pembuatan marker



Gambar 3. 10 alur pembuatan marker

1. **Resize**

   Setelah kita mengunggah Gambar ke situs vuforia maka sistem secara otomatis akan memperkecil gambar ke ukuran 320*320



Gambar 3. 11 Proses resize gambar

Algoritma Resize

```
Read(image,imgx,imgy)
Max_x<- 250 (maksimal nilai lebar gambar) Max_y-
<250(maksimal nilai panjang gambar)

If imgx>=Max_x
then  Scale =
Max_x div imgx
new_imgx = imgx
* scale
new_imgy = imgy
*scale endif

If new_imgy > Max_y then
     Scale = Max_y/new_imgy
     new_imgx =
new_imgx*scale
    new_imgy = new_imgy*scale
    endif

    If imgx<Max_x then
          new_imgx =
imgx;     If imgy<Max_y
then
    new_imgy = imgy;
```

2. Grayscale

Grayscale merupakan proses mengubah gambar menjadi monokrom dengan nilai intensitas 0 untuk hitam, 255 untuk putih, dan abu-abu untuk nilai yang berada pada rentang nilai 0 sampai 255. Dimana rumus secara garis besarnya adalah new pixel = (Red + Green +Blue)/3.



Gambar 3. 12 proses grayscale gambar

Algoritma Grayscale

```
Read (image,x,y)

For (i<-0 ; i<x; i++) do

 For (j<-0; j<y; j++) do

   Getpixel(i,j);

    Int grayscale = (int)((originalcolor.R)+
(originalcollor.G)+(originalcolor.B)/3)

  Newimage(i,j,grayscale)

End for End for
```

3. Historigram

Proses Histogram adalah suatu proses perataan, dimana distribusi derajat keabuan pada suatu gambar dibuat rata.



Gambar 3. 13 algoritma historigram

Berikut algoritma historigram

```
Deklarasi : (A,int N,int M) /*A=array citra N, M=panjang,Lebar
gambar*/
/*inisialisasi Hist[0,255] dengan 0*/
for(i=0;i<=255;i++)
Hist[i]=0
endfor
/*mencacah frekuensi kemunculan*/
for(k=0;k<=255;k++)
for (i=0;i<=N-1;i++)
for(j=0,j<=M-1;j++)
if(A[i,k]==i-1)
Hist[i]=Hist[i]+1
endif
endfor
endfor
endfor
```

tahap selanjutnya sistem akan melakukan proses merubah historigram ke treshold

5. Ambang batas (treshold)

Dalam Library Vuforia sebelum objek diconvert kedalam point-point marker yang dapat menghasilkan titik sudut marker, marker diubaha kedalam bentuk Threshold. Proses threshold ini yang menjadi acuan untuk proses selanjutnya, yaitu proses penentu titik marker. Berikut contoh perubahan gambar dari gambar asli ke gambar Threshold.



Gambar 3. 14 threshold

Berikut algoritma threshold

Algoritma *Threshold*

```
Tmean = 0 /*deklarasi nilai awal t*/
for (k = 0; k < level; k++
For x=0 to panjang_pixel-1 do
For y=0 to lebar_pixel-1 do
N= image[i],[y]
If n<=T then
new_x=0
new_y=0
t=k;
else
new_x=255
new_y=255
end if
endfor
endfor
```

6. Penetuan poin poin marker

Gambar Threshold tersebut kemudian diproses untuk menentukan letak point marker dan pencocokan pola menggunakan metode Fast Corner Detection .

Dari bentuk gambar Threshold maka selanjutnya menyimpan titik point kedalam gambar Threshold tersebut yang bertujuan untuk menetukan banyaknya jumlah dari titik point, berikut gambar yang telah diproses.



Gambar 3. 15 marker yang diproses

Semakin banyak bintang yang terdapat pada maker, maka semakin baik pola tersebut dijadikan marker, sehingga kamera ponsel dapat dengan cepat mendeteksi marker.

Berukut fungsi dari metode Fast Corner Detection yang digunaka untuk menentukan point-point marker pada gambar yang telah menjadi treshold.

```
clear all;
close all;
clc;
%% membaca citra
% filename=jazz.jpg';
```

```
filename='jazz.bmp';
I=imread(filename);
[x,y,z]=size(I);
if z>1
Ig=rgb2gray(I);
else
Ig=I;
end
%%
%% Proses Corner Detection
Ig=double(Ig);
sigma=2; rad=5;
% Compute x and y derivatives of image
dx=[-1 0 1; -1 0 1; -1 0 1];
dy=dx';
Ix=conv2(Ig,dx,'same');
Iy=conv2(Ig,dy,'same');
% Compute products of derivatives at every pixel
Ix2=Ix.*Ix;
Iy2=Iy.*Iy;
Ixy=Ix.*Iy;
% Compute the sums of the product of derivatives
% at each pixel
h=fspecial('gaussian',max(1,fix(6*sigma)),sigma);
Sx2=conv2(Ix2,h,'same');
```

```
Sy2=conv2(Iy2,h,'same');
Sxy=conv2(Ixy,h,'same');
% Compute the response of the detector at each pixel
k=0.04; % k=0.04 - 0.06
R=zeros(size(Ig));
% Compute the response of detector at each pixel
R=(Sx2.*Sy2 - Sxy.^2)./(Sx2 + Sy2 + eps);
ambang=12;
% Threshold on value of R. Compute nonmax suppresion
% Size of mask
sze = 2*rad+1;
% Grey-scale dilate
mx = ordfilt2(R,sze^2,ones(sze));
% Find maxima
R = (R==mx)&(R>ambang);
[row,col] = find(R);
imshow(I);
hold on;
points=[col row];
plot(points(:,1),points(:,2),'rp');
title('Detected Corner');
```

3.1.11 Analisi Tracking marker

Analisis *Tracking* merupakan analisis yang mendeskripsikan bagaimana proses *augmented realit y*pada sistem dapat berjalan, dari mulai inisialisasi,

tracking marker, sampai dengan proses *rendering* objek 3D dengan metode *tracking*marker yang di terapkan oleh sistem tersebut.

Proses Tracking beberapa objek yang dapat dilacak dan diregistrasi oleh QCAR (Qualcom Augmented Reality) SDK Vuforia Unity. Dalam proses pelacakan ada beberapa parameter untuk menentukan objek yang akan dilacak. MarkerTracker.css adalah satu dari banyaknya proses pelacakan. Kode fragmen pada script MarkerTracker.css di bawah ini menggambarkan penciptaan 4 penanda marker yaitu :

1. Marker Honda jazz
2. Marker Honda civic
3. Marker Honda Crv
4. Marker Honda City
5. Marker Honda Brio

Berikut fungsi yang digunakan untuk melakukan pelacakan terhadap 5 marker yang dibuat untuk memunculkan onjek 3D

```
// inisialisasi tracking marker:

    QCAR::TrackerManager& trackerManager =
        QCAR::TrackerManager::getInstance();
    QCAR::Tracker* trackerBase =
        trackerManager.initTracker(QCAR::Tracker::MARKER_TRACKER);
    QCAR::MarkerTracker* markerTracker =
        static_cast<QCAR::MarkerTracker*>(trackerBase);
    if (markerTracker == NULL)
    {
        LOG("Gagal diinisialisasi MarkerTracker.");
        return 0;
    }
    LOG("Sukses diinisialisasi MarkerTracker.");
```

```
    // Membuat frame markers:

          if   (!markerTracker->createFrameMarker(0,   "Markerjazz",
QCAR::Vec2F(50,50)) ||

                !markerTracker->createFrameMarker(1,  "Markercity",
QCAR::Vec2F(50,50)) ||

               !markerTracker->createFrameMarker(2,  "Markercivic",
QCAR::Vec2F(50,50)) ||

!markerTracker->createFrameMarker(3,"MarkerBrio",
QCAR::Vec2F(50,50)) ||

                !markerTracker->createFrameMarker(4,  "Markercrv",
QCAR::Vec2F(50,50)))

    {

        LOG("Gagal menemukan marker.");

        return 0;

    }

    LOG("Sukses menemukan Marker.");

  }
```

3.1.12 Implementasi Mobile

Pada analisis tahapan implementasi *mobile* ini menjelaskan bahwa bagaimana *scenario* aplikasi yang dibangun dapat berjalan pada sebuah *flatform mobile* .Aplikasi *Augmented Reality* yang telah dirancang unity 3d dapat digunakan atau dapat diimplementasikan dalam aplikasi mobile harus dengan menggunakan sebuah*library*, *libray* tersebut yaitu berupa Vuforia SDK(software development kit) untuk Android, dimana didalam *library* tersebut berupa fungsifungsi yang digunakan untuk memanggil *scenario AR*serta menyertakan *class* untuk mengatur segala inisialisasi pada framelayout dan untuk mengatur segala aktivitas aplikasi *mobile*.



Gambar 3. 16 marker yang diproses

3.2 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional menggambarkan kebutuhan sistem yang menitik beratkan pada property perilaku yang dimiliki oleh sistem, diantaranya kebutuhan perangkat keras , perangkat lunak, serta *user* sebagai bahan analisis kekurangan dan kebutuhan yang harus di penuhi dalam perancangan sistem yang akan di terapkan.

3.2.1 Analisis Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem yang merupakan sebuah himpunan perangkat yang mendukung atau dibutuhkan untuk membangun aplikasi *augmented rality* ini.

Tabel 3. 1 Kebutuhan Perangkat Lunak Pembuatan Aplikasi

1	**Sistem operasi Window 7 Ultimate.**
2	Library Vuforia for untiy

3	Blender version 2.70
4	Unity-4.3.3.
5	Java JDK 6.1

Tabel 3. 2 Kebutuhan Perangkat Lunak Implementasi Aplikasi

1	Minimal Sistem operasi Android 2.3 Ginggerbread

3.2.2 Analisis Perangkat Keras

Komputer/ *handphone* dan *webcam* adalah sebuah perangkat keras yang tidak luput dari perangkat lunak sebagai interaksinya. Perangkat lunak memberikan sebuah perintah-perintah terhadap perangkat keras agar dapat berjalan dengan baik. Dalam pembangunan aplikai *Brosur AR* ini menggunakan perangkat keras sebagai pendukungnya adalah sebagai berikut

Perangkat keras yang digunakan untuk membangun aplikasi ini adalah sebagai berikut:

Tabel 3. 3 kebutuhan perangkat keras pembuatan aplikasi

1	**PC/Leptop**
2	Kamera Device 2 Megapixel
3	Proccesor intel Core i5
4	Ram 4 GB
5	Harddisk 750GB
6	Mouse dan keyboard
7	Printer

Tabel 3. 4 Kebutuhan Perangkat Keras Implementasi Aplikasi

1	**Smartphone ber os android**
2	Aplikasi berjalan pada smarthphone dengan minimum Procesor Armv7
3	Smarthpone yang digunakan minimal mempunyai ram 524mb
4	Kamera Ponsel Seluler minimal 3 Megapixel

3.2.3 Analisis *User*

Aplikasi yang dibuat dapat dioprasikan oleh setiap user yaitu sales dari honda indosal pasteur atau konsumen itu sendiri yang mempunyai sebuah prangkat smartphone berbasis android serta mengerti dalam hal pengoprasiannya.

3.3 Anlisis kebutuhan Fungsional

Kebutuhan fungsiona berhubungan dengan fitur software yang akan dibuat atau dikembangkan. Berikutini adalah tahapan analisis kebutuhan fungsional Sistem Aplikasi *brosur berbasis AR*.

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik.

Analisis yang dilakukan dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain identifikasi aktor, *usecase diagram, skenario, sequence diagram, activity diagram, classdiagram*.

3.3.1 Identifikasi aktor

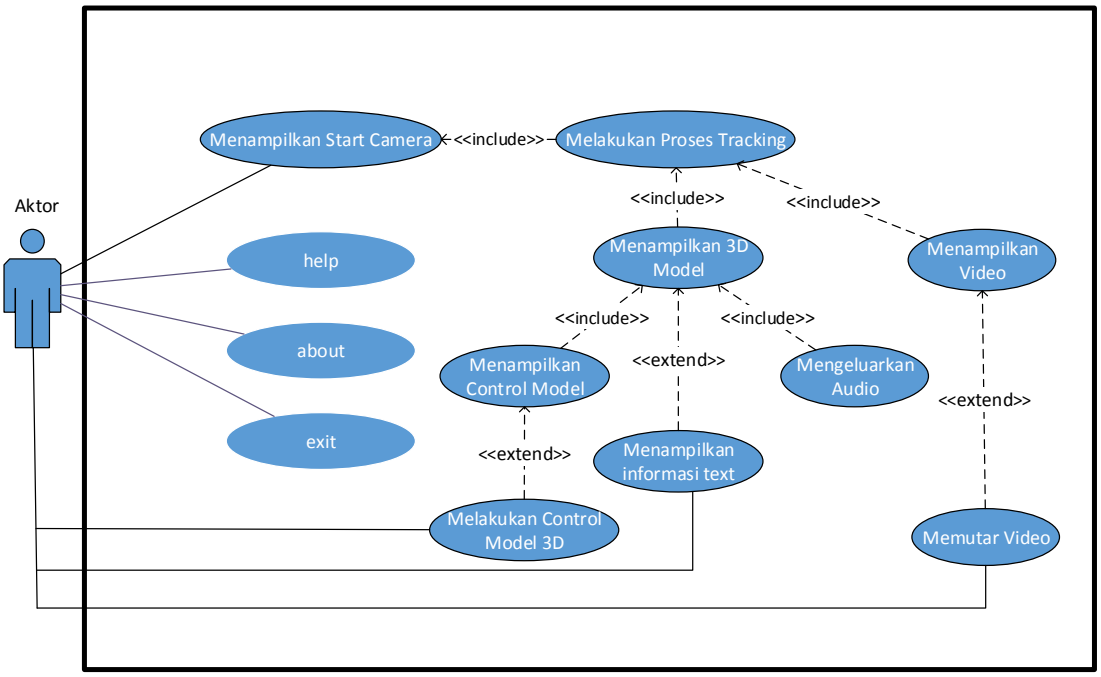
Aktor yang berperan dalam menjalakan aplikasi yaitu ini konsumen dan sales dari honda indosal pasteur yang mana user menggunakan sebuah smartphon untuk menjalankan palikasi ini.

3.3.2 *Usecase* Diagram

Usecase Diagram merupakan konstruksi untuk mendeskripsikan hububungan-hubungan yang terjadi antar aktor dengan aktivitas yang terdapat pada sistem. Sasaran pemodelan *usecase* diantaranya adalah sebagai berikut:

Nama : Aplikasi Augmented Reality

Aktor : User



Gambar 3. 17 Use case diagram aplikasi

3.3.2.1 Definisi Usecase

Use casediagram merupakan konstruksi untuk mendeskripsikan hubungan-hubungan yang terjadi antar aktor dengan aktivitas yang terdapat pada sistem.

Sasaran pemodelan *use case* diantaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan yang disepakati antara pemakai dan pengembang

Tabel 3. 5 Defenisi Use Case

Use Case Aplikasi User		
No	**Use Case**	**Deskripsi**
UCU-01	Menampilkan Start Camera	Fungsional untuk memulai mengambil gambar menggunakan ponsel.
UCU-02	Melakukan Proses Tracking	Fungsional untuk membandingkan gambar yang berasal dari ponsel dengan gambar yang ada pada file tracking.
UCU-03	Menampilkan 3D Model	Fungsional memunculkan 3D Model dari objek yang ditrack.
UCU-04	Mengeluarkan Audio	Fungsional memunculkan Suara dari Model 3D serta info mobil tertentu yang berasal dari ponsel.
UCU-05	Menampilkan Control Model	Fungsional memunculkan control untuk interaksi pengguna terhadap 3D Model.

Use Case Aplikasi User		
No	**Use Case**	**Deskripsi**
UCU-06	Melakukan Control Model 3D	Melakukan fungsi Rotasi dan Scale terhadap model 3D.
UCU-07	Menampilkan Video	Fungsional memunculkan Video dari objek yang ditrack.
UCU-08	Memutar Video	Fungsional memutarkan video.

3.3.2.2 Skenario Usecase

Skenario *Usecase* mendeskripsikan urutan langkah-langkah dalam proses bsnis, baik yang dilakukam aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor.

Tabel 3. 6 use case Start camera

Identifikasi	
Nama Usecase	Start
Aktor	Pengguna *User*
Tujuan	User menyentuh tombol start untuk kemudian sistem menjalankan kamera AR
Keadaan awal	Sistem menampilkan Main Menu

Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1.menekan tombol start.	
	2. Menjalankan Kamera AR.

Tabel 3. 7 Use Case Tracking

Identifikasi	
Nama Usecase	Tracking
Aktor	Pengguna *User*
Tujuan	Mencocokan gambar yang diambil melalui kamera telepon seluler dengan data yang berada pada file tracking sistem.
Keadaan awal	Sistem menampilkan gambar marker pada aplikasi di layar telepon seluler.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1.Menyorotkan Kamera Telepon Seluler ke Objek	
	2. Melakukan pencocokan pola marker dengan data yang berada pada file tracking.
Skenario Gagal Tracking	
Aksi Aktor	**Reaksi Sistem**
	1. “Layar Kosong”.

Keadaan Akhir	Sistem menampilkan gambar yang akan di *tracking.*

Tabel 3. 8 Skenario Use Case Lihat objek mobil

Identifikasi	
Nama *Usecase*	Lihat objek mobil
Aktor	Pengguna *User*
Tujuan	Menampilkan object 3d mobil
Scenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menjalankan aplikasi dan meyorotkan kamera pada gambar.	
	2. pencocokan pola / *tracking* gambar Marker
	3. Pemilihan objek 3D .
Keadaan Akhir	Aplikasi menampilkan objek 3D mobil
Skenario Gagal menampilkan video	
Aksi Aktor	**Reaksi Sistem**
	1.”Layar Kosong”
Keadaan Akhir	Aplikasi tetap menampilkan gambar yang sedang di*tracking.*

Tabel 3. 9 Sekenario menampilkan Control Objek

Identifikasi	
Nama Use Case User	Menampilkan Control Model
Aktor	Pengguna *User*
Tujuan	Menampilkan fungsi control sebagai interaksi pengguna terhadap objek 3D Model Mobil menggunakan teknologi Augmented Reality.
Skenario Utama	
Aksi Aktor	**Reaksi Sitem**
1. Menyorotkan kamera brosur.	
	2. Data hasil pengenalan pola cocok dengan data didatabase sehingga menampilkan 3D Model dan Control model dari Mobil tersebut secara real time dengan menggunakan teknologi Augmented Reality.
Keadaan Akhir	Aplikasi menampilkan tombol control dari Mobil secara real time, tombol control ditampilkan berdasarkan model 3D yang ditampilkan.
Skenario Gagal	
Aksi Aktor	**Reaksi Sitem**
	1.Tidak tampil informasi

Keadaan Akhir	Aplikasi menampilkan gambar yang sedang di tracking.

Tabel 3. 10 Skenario Usecase User Melakukan Control Model 3D

Identifikasi	
Nama Use Case User	Melakukan Control Model 3D
Aktor	Pengguna *User*
Tujuan	Merotasi dan memperbesar ukuran objek 3D Model mobil menggunakan teknologi Augmented Reality.
Skenario Utama	
Aksi Aktor	**Reaksi Sitem**
1. Menekan Tombol Control.	
	2. Melakukan fungsi control sesuai fungsi yang di gunakan yaitu rotasi dan *scale* terhadap 3D Model Mobil tersebut secara real time dengan menggunakan teknologi Augmented Reality.
Keadaan Akhir	3D Model rotasi dan scale secara real time.
Skenario Gagal	

Aksi Aktor	**Reaksi Sitem**
	1.Tidak tampil informasi
Keadaan Akhir	Aplikasi menampilkan gambar yang sedang di tracking.

Tabel 3. 11 Skenario Usecase User Menampilkan Video

Identifikasi	
Nama Use Case User	Menampilkan Video
Aktor	Pengguna *User*
Tujuan	Menampilkan Video Mobil menggunakan teknologi Augmented Reality.
Skenario Utama	
Aksi Aktor	**Reaksi Sitem**
1. Menyorotkan kamera brosur.	
	2. Data hasil pengenalan pola cocok dengan data didatabase sehingga menampilkan Video Mobil tersebut yang dapat dimainkan.
Keadaan Akhir	Aplikasi menampilkan Video dari Mobil secara real time.

Skenario Gagal	
Aksi Aktor	**Reaksi Sitem**
	1.Tidak tampil informasi
Keadaan Akhir	Aplikasi menampilkan gambar yang sedang di tracking.

Tabel 3. 12 Skenario Usecase User Memutar Video

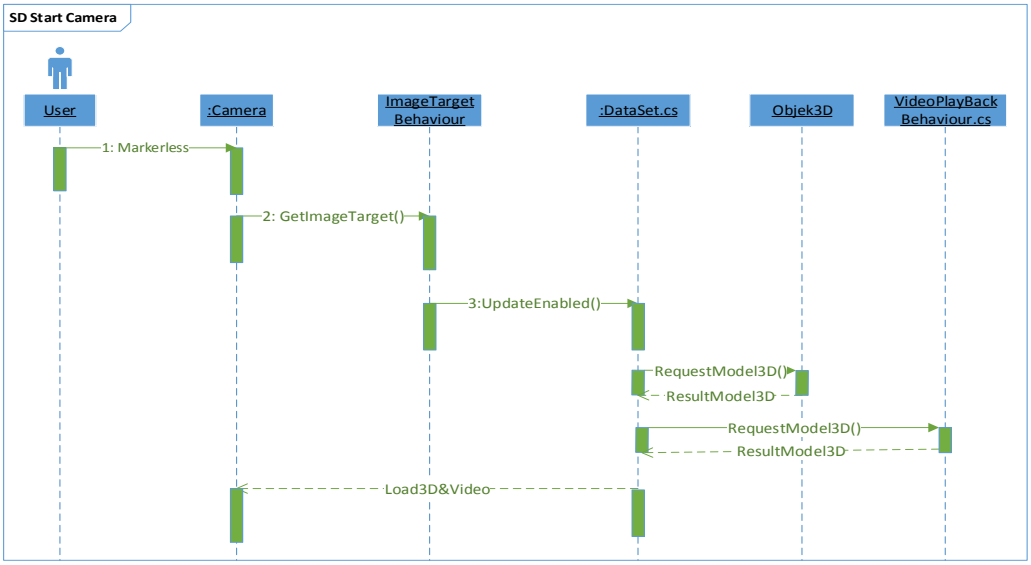
Identifikasi	
Nama Use Case User	Memutar Video
Aktor	Pengguna *User*
Tujuan	Memutar video menggunakan teknologi Augmented Reality.
Skenario Utama	
Aksi Aktor	**Reaksi Sitem**
1. Menekan Tombol Play .	
	2. Melakukan pemutaran video mobil honda secara real time dengan menggunakan teknologi Augmented Reality.

Keadaan Akhir	Video mulai diputar secara real time.
Skenario Gagal	
Aksi Aktor	**Reaksi Sitem**
	1.Tidak tampil informasi
Keadaan Akhir	Aplikasi menampilkan gambar yang sedang di tracking.

3.3.3 Sequence Diagram

Adapun sequence diagram yang digunakan dalam aplikasi Brosur AR ini adalah sebagai berikut.

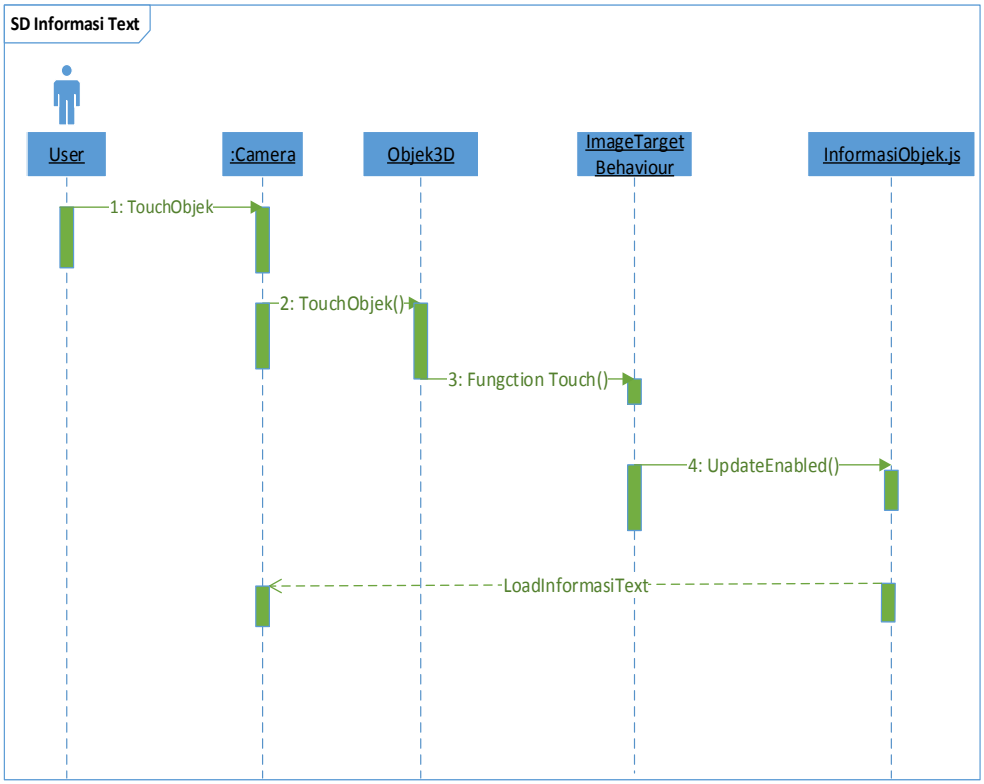
3.3.3.1 Sequence Diagram Start AR



Gambar 3. 18 class diagram

Pada gambar diatas dijelaskan bahwa user menscan markerless dengan kamera kemudian dideteksi oleh system. Pada class ImageTargetBehaviour mendeteksi marker yang ada didepan kamera kemudian memanggil dataset yang ada di class DataSet, setelah itu akan dimunculkan objek 3D .

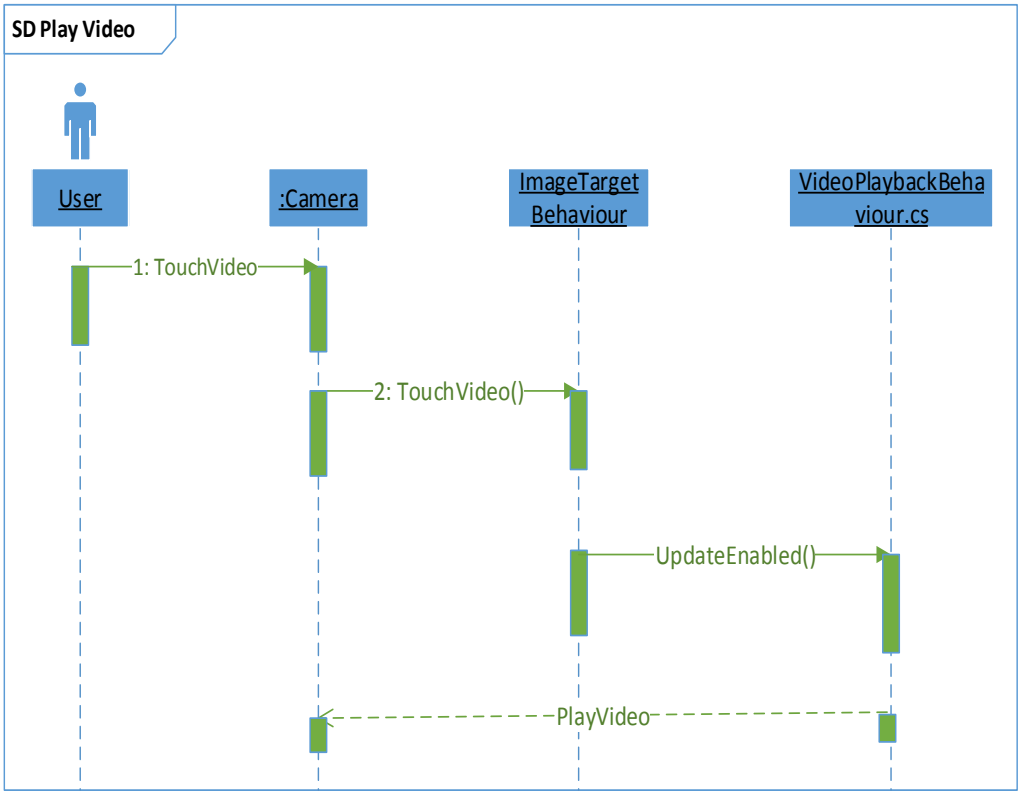
3.3.3.2 Sequence Diagram Informasi Objek



Gambar 3. 19 Sequence Diagram Informasi Objek

Pada gambar diatas dijelaskan bahwa user menyentuh objek Augmented Reality kemudian pada class ImageTargetBehaviour memangil fungsi informasiObjek untuk menampilkan informasi berupa text.

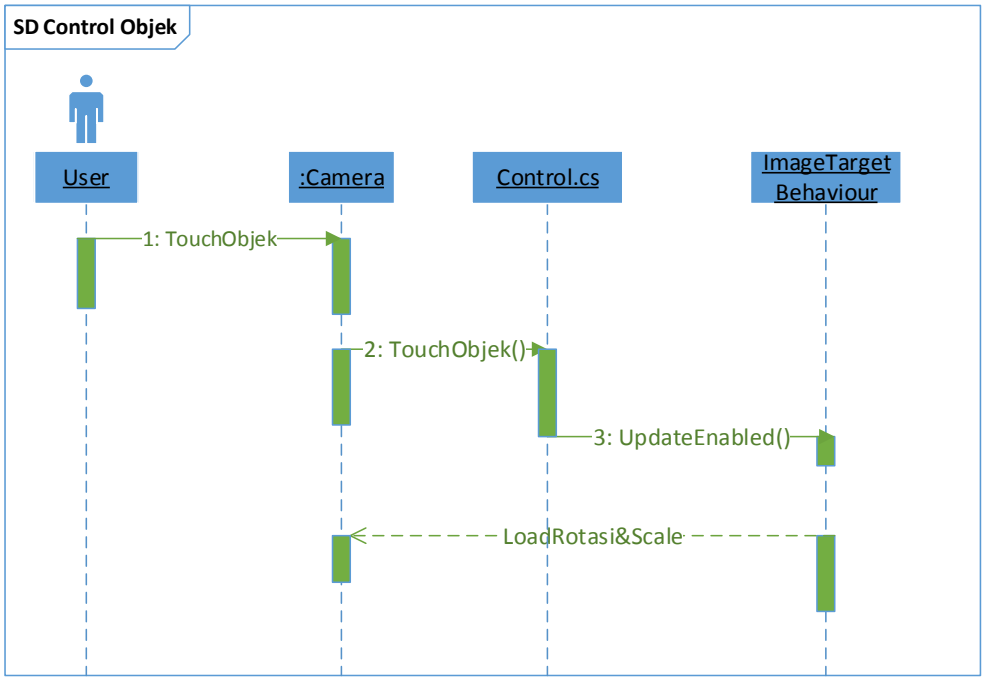
3.3.3.3 Sequence Diagram Play Video



Gambar 3. 20 Sequence Diagram Play Video

Pada gambar diatas dijelaskan bahwa user menyentuh video Augmented Reality kemudian pada class ImageTargetBehaviour memangil fungsi VideoPlayBackBehaviour untuk memainkan video.

3.3.3.4 Sequence Diagram Control Objek



Gambar 3. 21 Sequence Diagram Control Objek

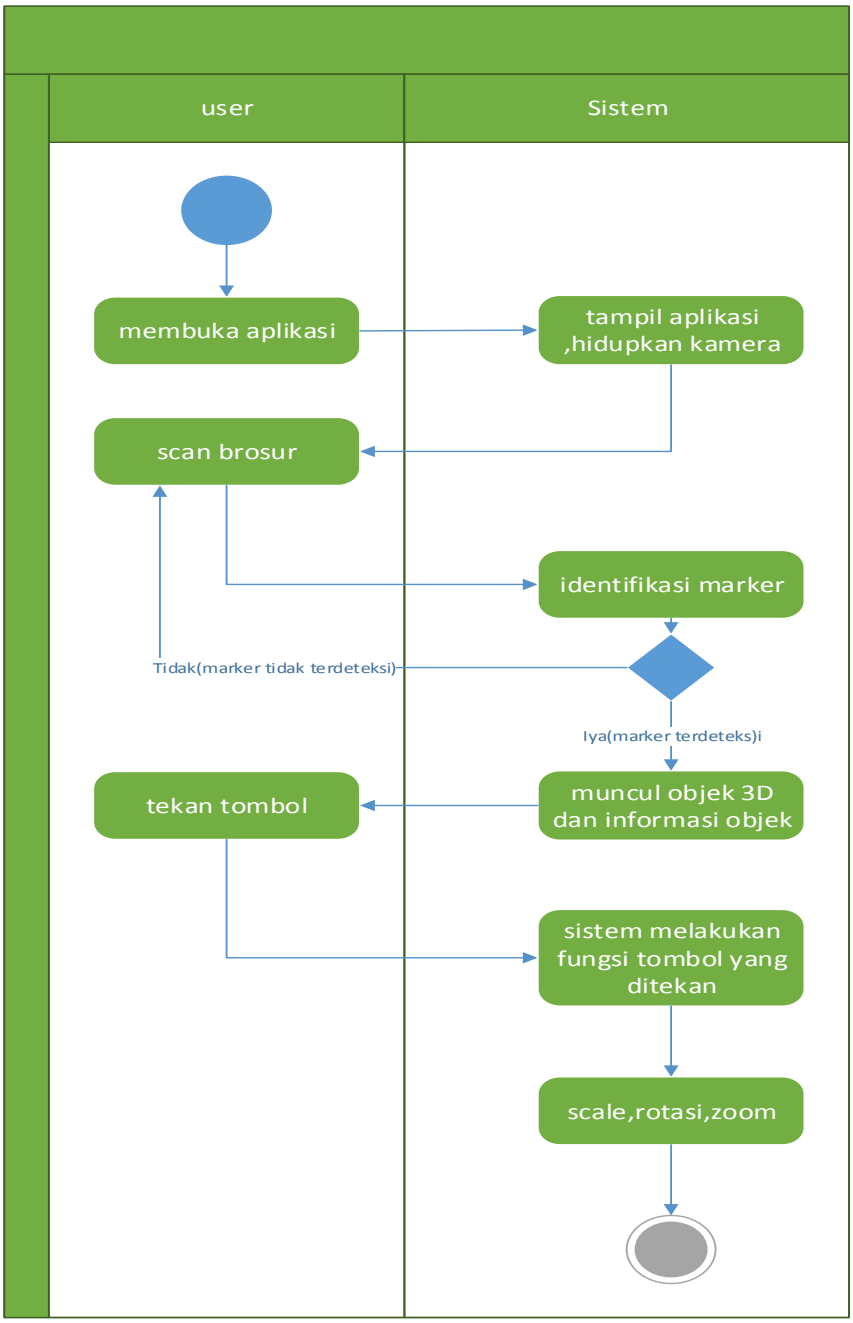
Pada gambar diatas dijelaskan bahwa user menyentuh tombol control kemudian pada class ImageTargetBehaviour memangil fungsi Control untuk rotasi dan zoom in ,zoom out objek.

3.3.4 Activity Diagram

Adapun activity diagram dari penjabaran masing-masing use case diagram apat dijelaskan pada activity diagram di bawah ini.

3.3.4.1 Activity DiagramTracking Marker

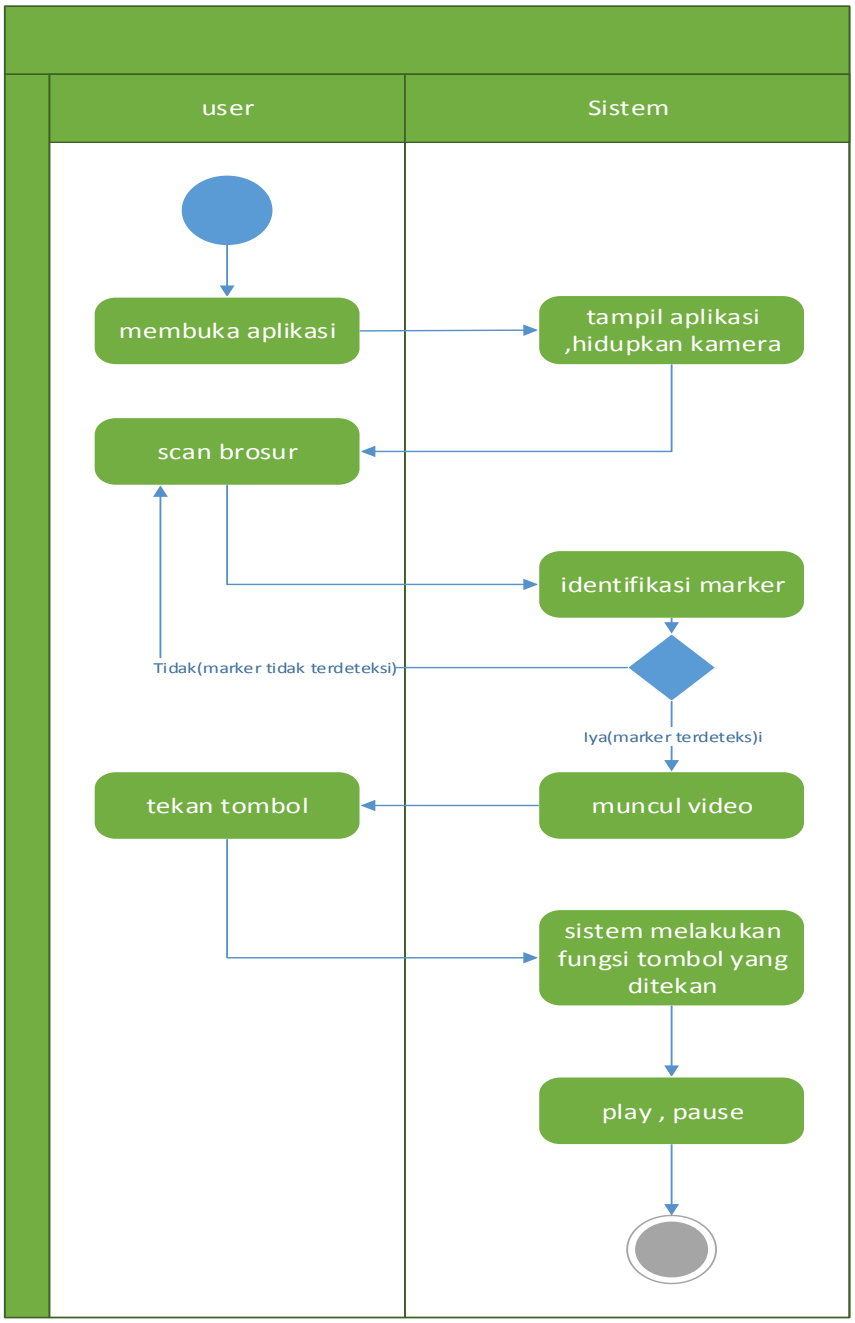
Aktivity diagram Tracking Marker ini menjelaskan aliram kerja aktor pengguna pada saat kamera menyorot gambar agar dapat di-*tracking* yang akan dicocokan melalui aplikasi *mobile. Aktivity diagram Tracking Marker*dapat di lihat pada Gambar 3.13



Gambar 3. 22 Activity diagram menampilkan objek 3d dan informasi objek

Dari gambar 3.13 dapat diketahui bahwa alur aktivitas yang menunjukkan proses jalannya sistem aplikasi AR secara detail mulai dari user membuka aplikasi sampai system mengeluarkan object, dan melakukan interaksi terhadap AR. Gambar 3.8 actifity diagram

Sedangkan activity diagram untuk menampilkan video adalah sebagai berikut :

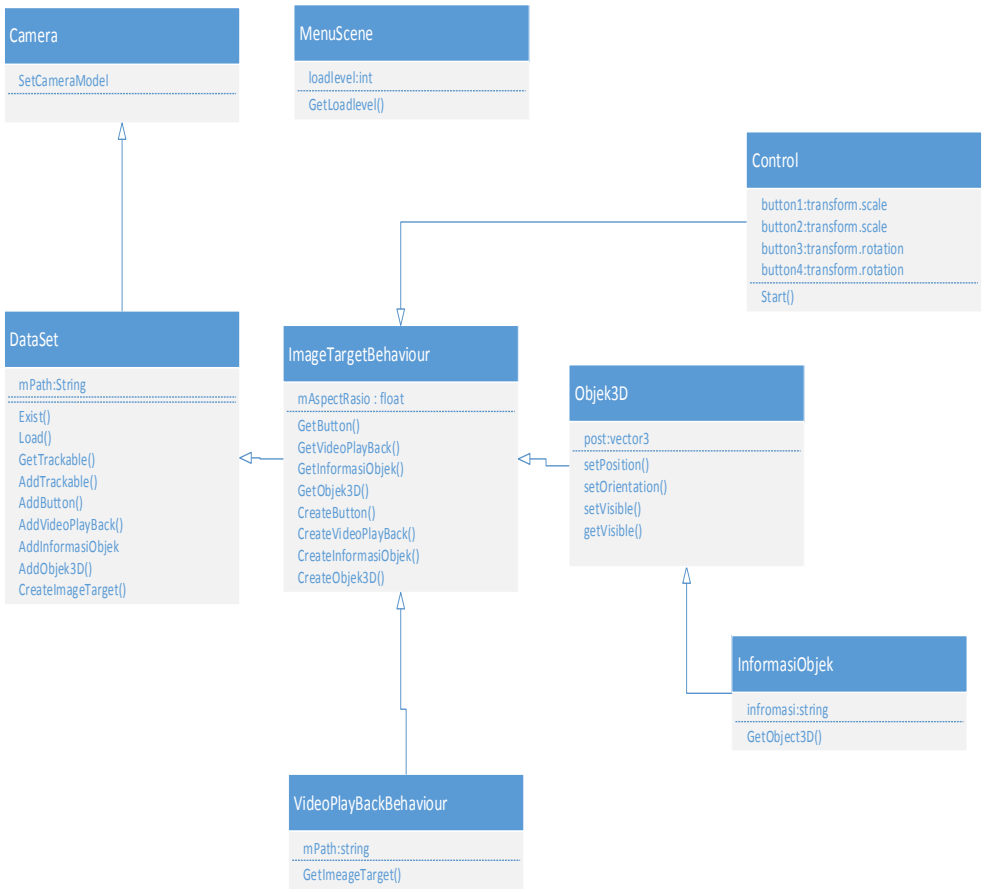


Gambar 3. 23 Activity diagram Menampilkan Video

3.3.5 Class diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (*attribut* atau *property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda atau fungsi). Berikut adalah *class diagram* dari sistem aplikasi *broaur AR*.

Berikut adalah class diagram yang dibangun



Gambar 3. 24 class diagram

a. Class diagram Image Target

imageTargets	**<<boundary>>**
+imageTargets()	Konstruktor untuk menginisialisasi semua atribut class imageTargets.
+initTracker()	Menginisialisasi fungsi tracker untuk mengenali marker
+deinitTracker()	Proses untuk memberhentikan fungsi tracker bila marker sudah dikenali.
+loadTrackerData()	Proses untuk meload data tracker yang diambil dari Target Management System vuforia.
+destroyTrackerData()	Method untuk menghapus data tracker sebelumnya setelah digunakan
-startcamera()	Proses untuk menjalankan kamera.

imageTargetsRenderer	**<<boundary>>**
+getData()	*Method* ini digunakan untuk mengambil data marker mobil pada dataSet.

+initRendering()	*Method* ini digunakan untuk menginisialisasi fungsi render image pada Brosur AR.
+onDrawFrame()	Method ini digunakan untuk mengaktifkan fungsi untukmelakukan penggambaran kembali marker pada layar utama.

3.4 Perancangan antarmuka

Antar muka merupakan tampilan dari suatu program aplikasi Brosur AR yang berperan sebagai media komunikasi yang digunakan sebagai sarana berdialog antara program dengan pengguna. Sistem yang akan dibangun diharapkan menyediakan interface yang mudah dipahami dan digunakan oleh pengguna.

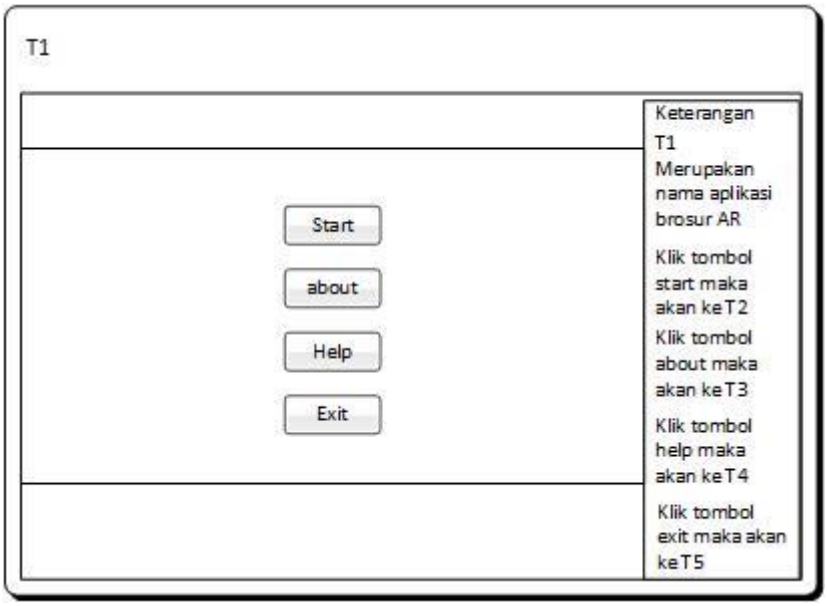
Perancangan antarmuka mendeskripsikan rencana tampilan dari setiap form yang akan digunakan pada aplikasi. Perancangan antarmuka terdiri dari perancangan menu utama, perancangan start AR, perancangan How To Use dan perancangan About.

3.4.1 Analisi Grafis antarmauka aplikasi brosur AR

Gambar dibawah menunjukan desain antarmuka aplikasi brosur augmented reality yang akan dibangun

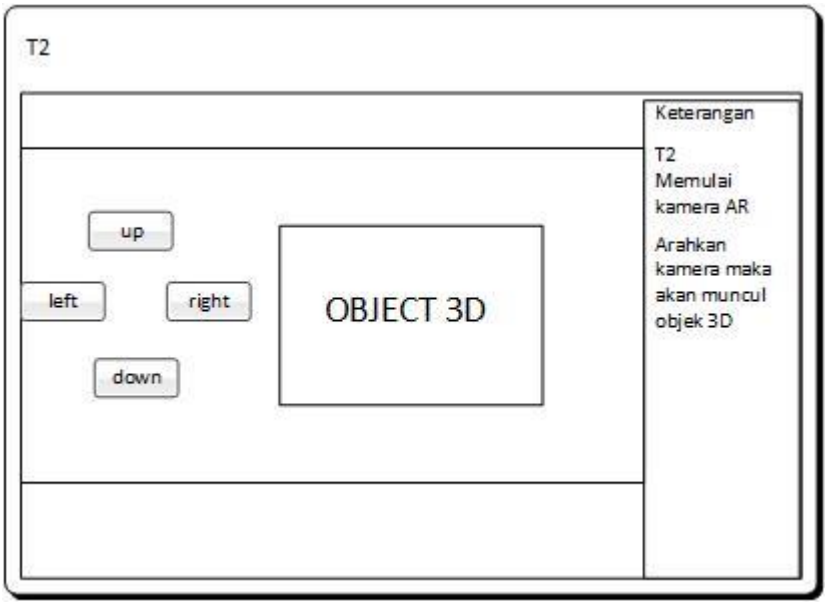
3.4.1.1 Analisis antarmuka

pada tampilan awal aplikasi yang akan dibuat ialah terdapat tombol start, help dan exit , seperti yang terihat pada gambar dibawah



Gambar 3. 25 perancangan antarmuka Menu utama

Ketika user menekan tombol start maka aplikasi akan mulai menjalankan kamera untuk selanjutnya mendeteksi marker seperti terlihat gambar dibawah



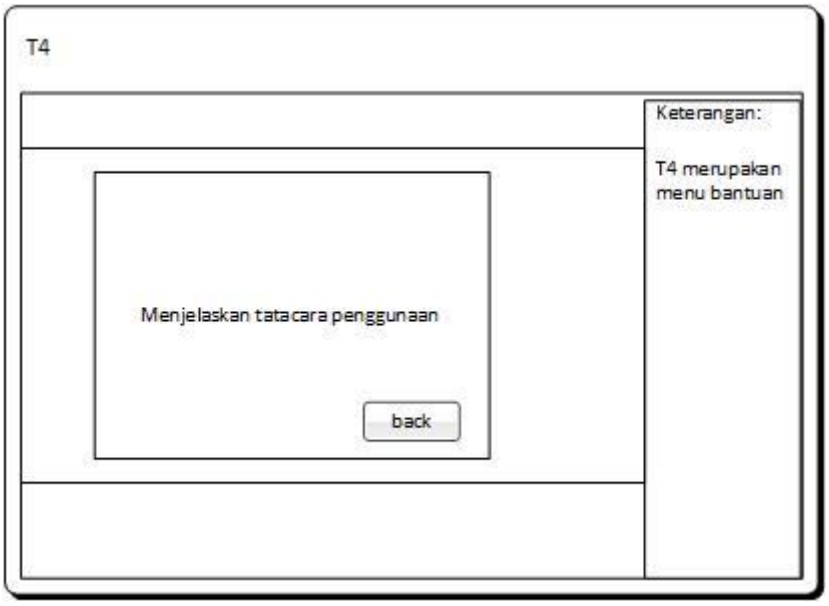
Gambar 3. 26 perancangan antarmuka aplikasi

Ketika user mengarahkan kamera pada marker yang disediakan maka akan keluar objek 3D, untuk selanjutnya ketika user menekan tombol about maka user akan masuk pada tampilan sekilas tentang aplikasi dan ketika menekantombol back maka akan kembali ke T1



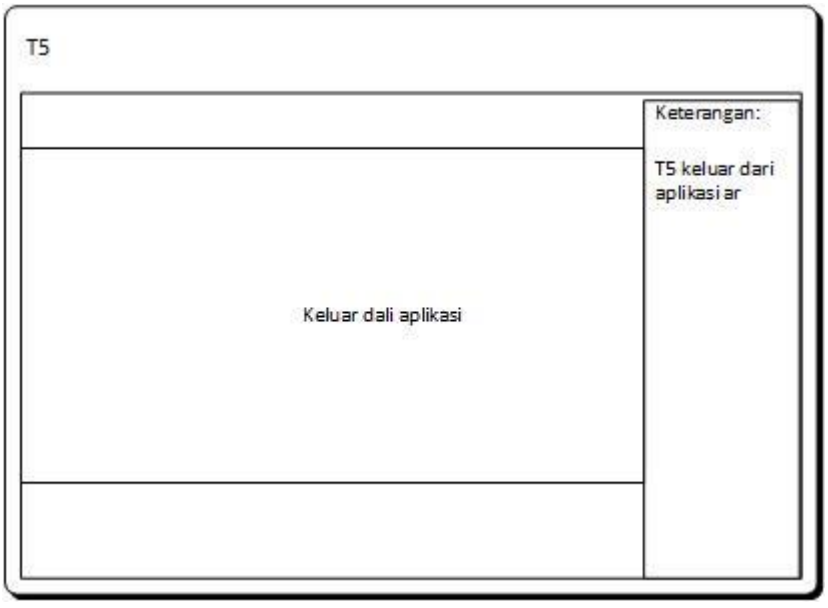
Gambar 3. 27 perancangan antarmuka aplikasi

Selanjutnya ketika user menekan tombol help maka user akan dibawa ke tampilan tentang bantuan cara menggunakan aplikasi



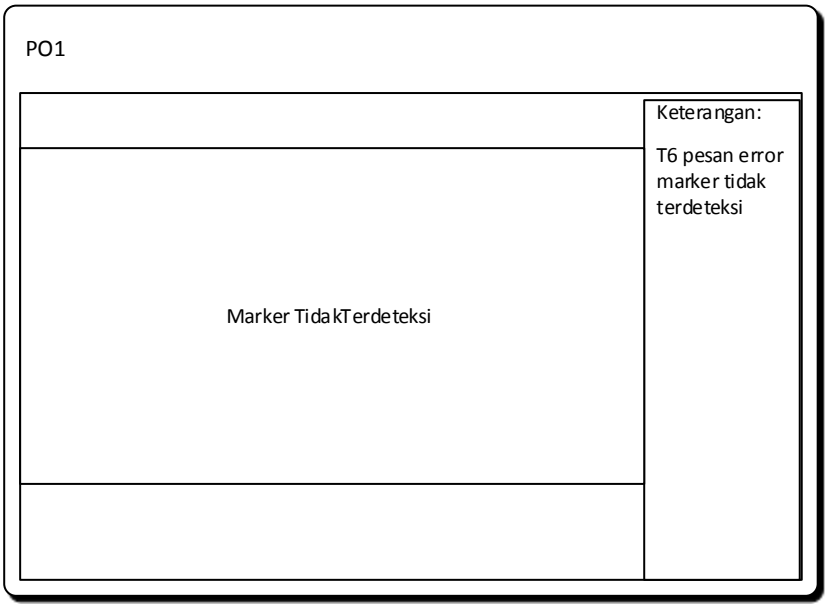
Gambar 3. 28 perancangan antarmuka help

selanjutnya ketika user tekan tombol exit maka aplikasi akan keluar.



Gambar 3. 29 perancangan antarmuka exit

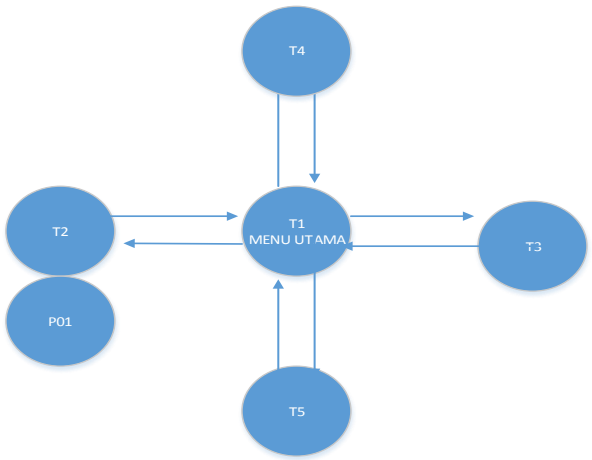
Perancangan Pesan error



Gambar 3. 30 Pesan error

3.4.1.2 Jaringan Semantik

Jaringan semantik menggambarkan keterhubungan nagivasi menu dari satu halaman ke halaman lainnya. Jaringan semantik pada aplikasi Brosur AR dapat dilihat pada gambar 3.24



Gambar 3. 31 Jaringan Semantik