# BAB III
# ANALISIS DAN PERANCANGAN SISTEM

### 3.1.1 Analisis Sistem

Analisis sistem merupakan penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan. Analisis merupakan tahapan yang paling penting, karena kesalahan dalam tahap ini akan menyebabkan kesalahan di tahap selanjutnya.

### 3.1.2 Analisis Masalah

Analisis masalah adalah langkah awal dari analisis sistem. Langkah ini diperlukan untuk mengetahui permasalahan apa saja yang terjadi pada sistem yang telah berjalan.

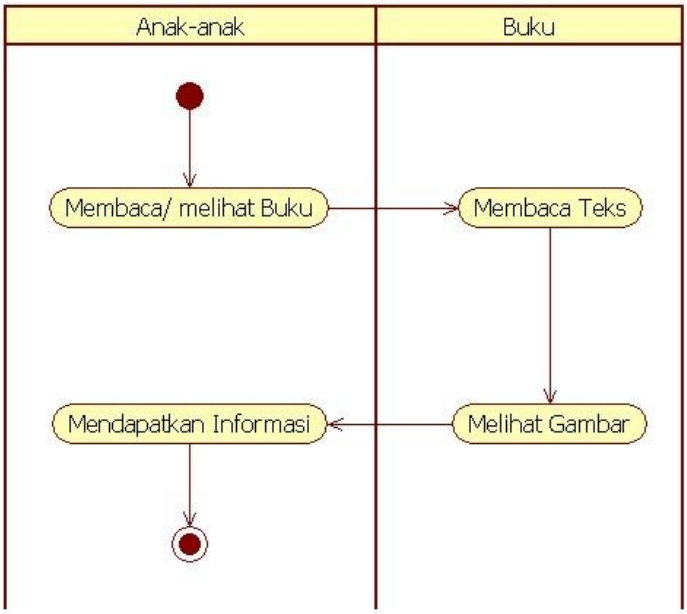
Analisis masalah yang dimaksud disini adalah sulit diperolehnya media yang dapat memperkenalkn sosok hewan beserta suara pada buku-buku pembelajaran anak, dikarenakan masih digunakannya media buku yang hanya dapat memperlihatkan satu sisi atau sudut pandang yang searah dan kurang interaktif dalam pembelajarannya.

Dengan teknologi yang berkembang pada saat ini, khususnya teknologi *Augmented reality (AR)* yang unggul dengan virtual animasinya yang dapat menggambungkan antara objek dunia nyata dengan objek *virtual*, dapat dijadikan sarana yang dapat menutupi kekurangan yang terdapat pada media pembelajaran saat ini. Berkembangnya teknologi AR ini telah berkembang keberbagai bentuk yang dapat diaplikasikan baik dalam bentuk aplikasi*web, desktop* maupun *mobile*. Dengan merujuk kepada kekurangan yang ada maka di bangunlah alpikasi yang berbasiskan mobile yang dikarenakan memiliki kelebihan yang dapat digunakan secara efisien jika di bandingkan dengan media lain. Dengan menggunakan teknologi *Augmented Reality* (AR) di harapkan dapat menjembatani kebutuhan di

dalam proses pembelajaran yang lebih interaktif dikarenakan memiliki kelebihan-kelebihan yang dapat menjawab masalah yang dihadapi. *Augmented Realitymulti marker*merupakan jawaban yang cocok dalam masalah yang sedang di hadapi, sebab dengan *multi marker*dapat merealisasikan objek yang lebih dari satu objek yang di hasilkan. Aplikasi ini bertujuan untuk menampilkan informasi secara *real time* berupa hewan virtual serta suara sebagai pendukung dengan menggunakan tehnik marker yang lebih dari satu.

### 3.1.3 Analisis Sistem yang Berjalan

AnalisisSistemataуanalisisprosesadalahtahapanyangmemberi gambarantentangsistemyangsedang berjalan sekarang.Analisisinibertujuanuntuk memberi gambaranyanglebihdetailbagaimanacarakerjadarisistemyang sedang berjalan.Prosedur pada proses media yang sedang berjalan sekarang dapat di lihat seperti pada Gambar 3.1.




**Gambar 3.1 Alur sistem yang sedang berjalan**

1. Anak-anak adalah objek yang ingin mendapatkan informasi yang terdapat pada buku.

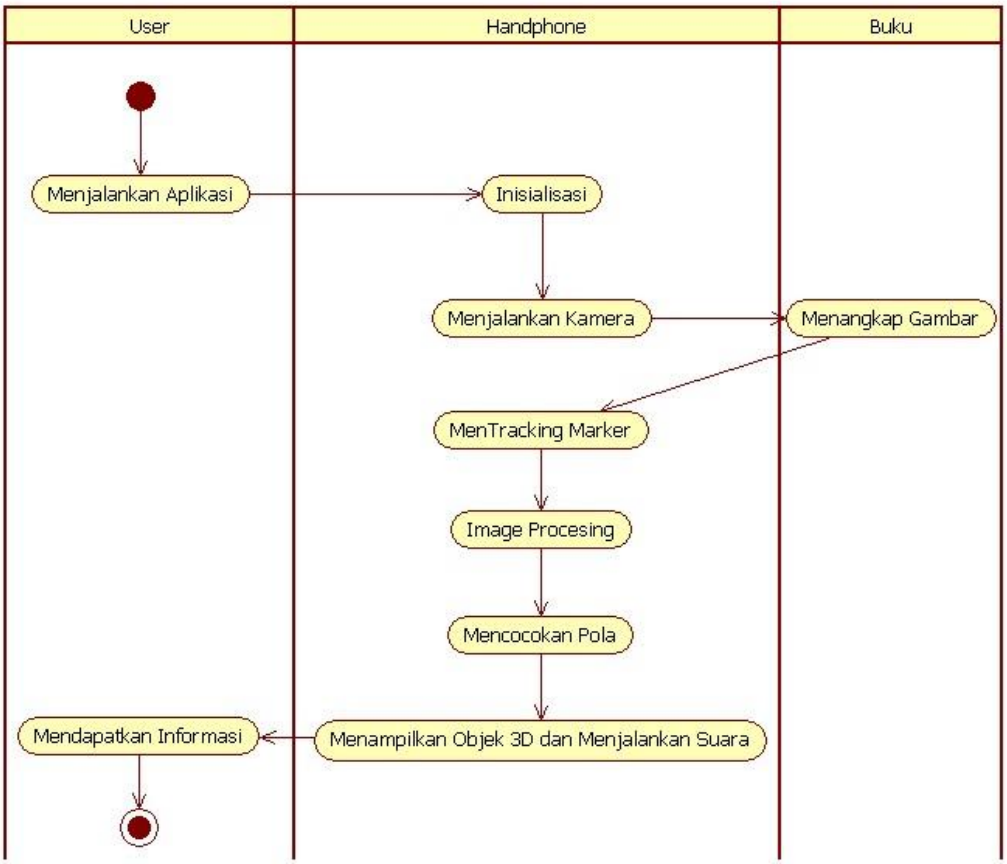
2. Media buku atau sebagai alat yang memberikan informasi dengan berisikan gambar yang ditampilkan masih dalam bentuk 2D.

Dari gambaran prosedur media yang didapat untuk mengakses informasi pada buku, teknologi *augmented reality* bisa dijadikan sebagai media alternatif sehingga konten informasi yang didapat dari buku dapat menjadi lebih menarik dan interaktif.

### 3.1.4 Analisis Arsitektur Sistem

Sistem yang akan dibangun adalah suatu aplikasi *mobile* menggunakan *platform* Android dengan teknologi *Augmented Reality*. Teknologi dengan aplikasi mobiledengan mudah diakses dimana saja dan sangat berkembang pada masa kini sehingga memudahkan *user* untuk mengakses aplikasi ini. Keunggulan dari teknologi *augmented reality* juga sangat menonjoldalam 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 buku yang bersifat*realtime.*

Proses aplikasi yang akan di bangun dengan menggunakan sistem multi marker dapat dilihat pada Gambar 3.2 yang menggambarkan alur yang akan di rancang.

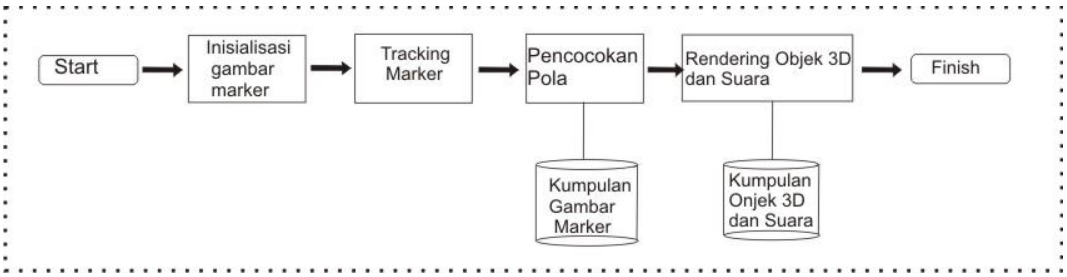



**Gambar 3.2 Arsitektur Sistem**

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 men-*tracking*marker serta mensingkronkan marker dengan kamera.
3. Buku merupakan media informasi yang didalamnya terdapat marker yang memiliki bentuk yang berbeda dan yang nantinya akan di *tracking* oleh *handphone*.
4. Menghasilkan Animasi 3D dan suara yang merupakan hasil yang dihsilkan oleh media aplikasiyang telah memiliki aplikasi *Augmented Reality Multi Marker* di dalamnya.

### 3.1.5 Analisis Alur Sistem

Analisis alur sistem ini menjelaskan bagaimana proses yang dilakukan seperti yang terlihat pada Gambar 3.3 .




**Gambar 3.3 Diagram alur sistem *Augmented Reality***

Secara garis besarnya, dalam perancangan ada tiga bagian utamayaitu:

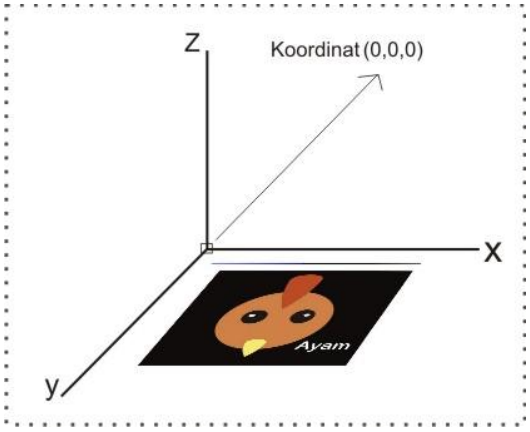
1. Inisialisasi Gambar Marker
2. *Tracking* Marker
3. *Rendering* Objek.

### 3.1.6 Perancangan Marker

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

#### 3.1.6.1 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 3.4 Koordinat pada komputer**

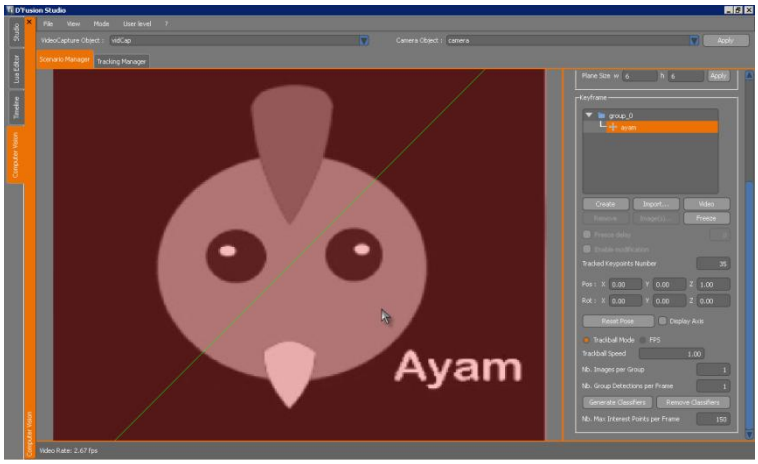
### 3.1.6.2 Pembuatan Marker

Dalam tahapan proses input *Augmented Reality(AR)* di perlukan sebuah marker dalam tahap proses pengaplikasiannya. Adapun cara dalam dalam pembuatannya adalah dengan membuat sebuah objek persegi empat yang memiliki objek lain di dalamnya baik itu berupa gambar, teks maupun angka dengan warna yang berbeda dengan latar belakang persegi tersebut. Contoh marker tersebut dapat di lihat seperti pada Gambar 3.5 yang menunjukan bentuk marker. Ukuran marker yang digunakan dalam analisis ini adalah 402 x 402 px serta gambar yang digunakan berupa file format .jpg.Dalam tahap pembuatan marker untuk digunakan oleh aplikasi, tahap yang dilakukan adalah merubah file tersebut menjadi berupa file gambar yang meiliki object.bin yang didalamnya berisikan informasi dari marker tersebut.




**Gambar 3.5 Bentuk *Marker***

Tahapan yang dilakukan adalah merubah file tersebut dengan menggunakan *Computer Vision*yang telah disediakan *D'Fusion Studio*agar seluruh informasi marker tersebut dapat dibuat.



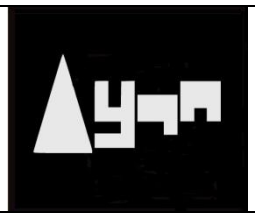
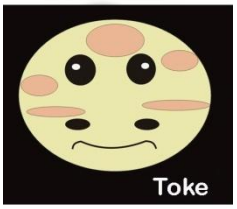
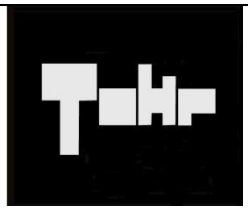

**Gambar 3.6 Gambar Pembentukan Marker dengan *Computer Vision***

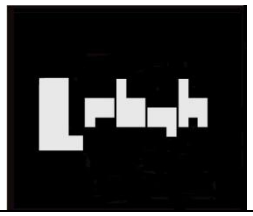
### 3.1.6.3 Marker yang digunakan

Dalam tahap analisis marker ini menggunakan 2 jenis marker yang berbeda agar dapat membedakan marker yang digunakan baik untuk objek 3D ataupun suara seperti yang ditunjukan pada Tabel 3.1.

**Tabel 3.1 Jenis marker yang digunakan.**

| Marker | Nama pada database | Tujuan |
|---|---|---|
|   | ayam | Menghasilkan objek 3D Ayam  |
| | Suara ayam | Menghasilkan Suara Ayam |

| | | |
|---|---|---|
|  | | |
|  | bison | Menghasilkan objek 3D Bison  |
|  | Suara bibson | Menghasilkan Suara Bison |
|  | toke | Menghasilkan objek 3D Toke  |
|  | Suara toke | Menghasilkan Suara Toke |
| | lebah | Menghasilkan objek 3D Lebah |

| | | |
|---|---|---|
|  | |  |
|  | Suara lebah | Menghasilkan Suara Lebah |

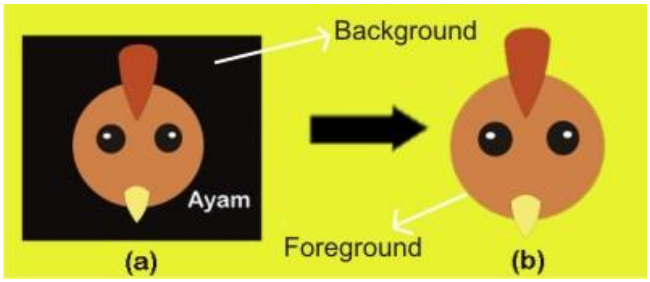
Dari Tabel 3.1 dapat dilihat ada dua jenis marker yang berbeda untuk menentukan jenis marker yang di gunakan sebagai inputan awal yang nantinya menghasilkan output 3D maupun suara. Dari marker tersebut dapat mudah di kenali mana yang termasuk untuk objek 3D maupun suara, karena telah di beri label di masing-masing marker supaya mudah di kenali. Ukuran marker secara normal minimal berukuran 320x240 px dan bentuk marker secara umum harus kotak karena berfungsi sebagai kunci pendeteksian sudut yang nantinya akan dikenali sebagai garis marker.

### 3.1.6.4 Isi Marker

Seperti yang telah di bahas sebelumnya, isi dari marker ini dapat berupa gambar , teks maupun angka yang memiliki tekstur warna yang berbeda dengan latar belakangnya. Fungsi dari gambar yang berada di tengah marker persegi empat tersebut adalah digunakan sebagai pengenal pada saat proses men-*tracking*oleh kamerayang berisi informasi *keypooin* yang nantinya diproses untuk menyamakan pola dengan file pada sistem.

Pola marker yang tersebut terdiri dari background dan foreground dimana dari kedua bagian tersebut memiliki fungsinya masing, untuk background digunakan sebagai batas terluar dari marker atau yang memiliki warna hitam agar tidak tercampur dengan foreground dan foreground sendiri adalah gambar yang

terdapat di dalam background atau didalam kotak hitam yang dijadikan sebagai acuan dalam mendapatkan *keypoint*.



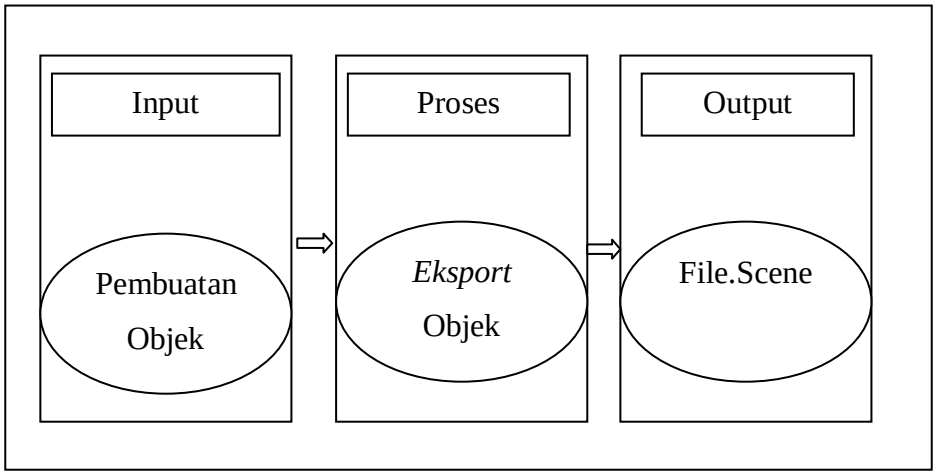

**Gambar 3.7(a) Marker dengan latar belakang (b) Isi dari marker sebagai acuan untuk menghasilkan keypoint.**

### 3.1.7 Perancangan Objek 3D

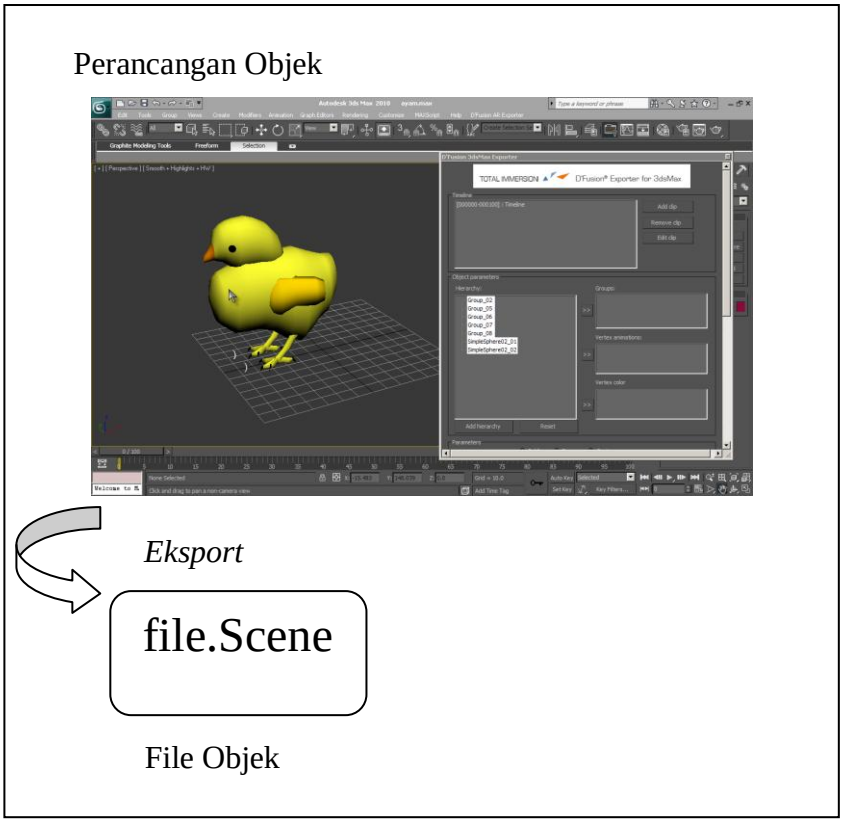
Tahapan dalam menganalisis bentuk dan format dari objek 3D yang digunakan.

#### 3.1.7.1 Analisis 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 *D'fusion Studio* harus berupa objek 3D yang berekstensi file .scene, agar file tersebut dapat diimpor.




**Gambar 3.8 Gambar Proses pembentukan data objek**




**Gambar 3.9 Gambar *Ekport* objek 3D**

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 .scene.

Pada tahap mengekspor objek 3D, tidak dapat secara langsung dengan menggunakan *eksporter* bawaan aplikasi pembuat objek dapat di ekspor. Untuk ekspor objek 3D terlebih dahulu harus meng-*install*aplikasi bawaan dari D'Fuision berupa eksporter untuk 3DSmax agar objek dapat ditampilkan dan sesuai dengan yang telah dibuat serta file tersebut dapat berupa file.scene.

### 3.1.8 Analisis objek Suara

Pada tahap analisis objek suara yang digunakan terhadap aplikasi berupa objek suara yang memiliki file.ogg. File .ogg merupakan sebuah file yang dapat digunakan oleh *tools*pengembang AR D'fusion Studio selain file .wav.

### 3.1.9 Analisis Tracking Marker

Analisis *Tracking*merupakan analisis yang mendeskripsikan bagaimana proses *augmented reality*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.

#### 3.1.9.1 InisialisasiGambar Marker

Inisialisasi adalah merupakan tahapan-tahapan pertama pada sebuah alat yang digunakan sebagai media untuk menangkap gambar *video* , adapun fungsi atau *method*yang digunakan system *library* aplikasi untuk mengaktifkan vidio *Camera* secara *real time.*

```
local camera = Camera(scene:getCurrentCamera())
local vidCap =
VideoCapture(scene:getObjectByName("vidCap"))
```

memanggil fungsi kamera pada hardware, dengan input-nya adalah getCurrentCamera() dimana fungsi tersebut merupakan fungsi yang digunakan untuk memanggil *scenario* kamera yang memungkinkan untuk digunakan.

`VideoCapture(scene:getObjectByName(“VidCap”)` merupak an fungsi yang digunakan untuk memanggil fungsi *video*yang digunakan untuk menangkap objek marker dengan input-an getObjectByname(“VidCap”) yang fungsinya adalah mengaktifkan *video capture*pada skenariodengan nama variable yang telah diberikan yaitu “VidCap” .




**Gambar 3.10 Inisialisasi Gambar Secara *Real Time***

### 3.1.9.2 *Tracking* Marker

*Tracking* marker adalah proses dimana mendeteksi sebuah marker yang kemudian terintegrasi untuk menghasilkan objek 3D maupun suara.Didalam prosesnya setiap objek marker dan objek 3D beserta suara akan di inisialisasikkan terlebih dahulu agar dapat mudah untuk digunakan dalam proses pencocokan pola marker.

Misalkan fungsi yang digunakan :

```
gtrackingStatusAyam = 0 – ayam
gtrackingStatusSuaraAyam = 0 – Suara ayam
localtrackingObject2 =
Object3D(scene:getObjectByName("Tracking_object"))
local suaraAyam =
Sound(scene:getObjectByName("suaraAyam"))
local trackingObjectIndexAyam = 1
local trackingObjectIndexSuaraAyam = 2
```

1. `gtrackingstatus1 = 0` adalah merupakan status terhadap suatu objek marker, apakah terdeteksi ataupun sebaliknya. Nilai 0 merupakan nilai awal yang diset agar inisialisasi masih dalam tahap belum ada objek marker yang terdeteksi maka jika telah terjadi pendeteksian secara *default* sistem akanmerubah nilai 0 menjadi 1.
2. `trackingObjekAyam` dan suaraAyam adalah inisialisasi untuk menghubungkan objek 3D ayam dan suara ayam agar dapat digunakan sebagai lokasi untuk melakukan proses *rendering*.
3. `trackingObjectIndexAyam`maupun trackingObjectIndexSuaraAyammerupakan `getObjectByName("Tracking_object".`tahap *indexing* objek agar dapat dibedakan antara satu dengan objek lainnya.
   Fungsi `Object3D(scene:getObjectByName("Tracking_object"))`ad alahmerupakan fungsi yang digunakan untuk mendapatkan informasi

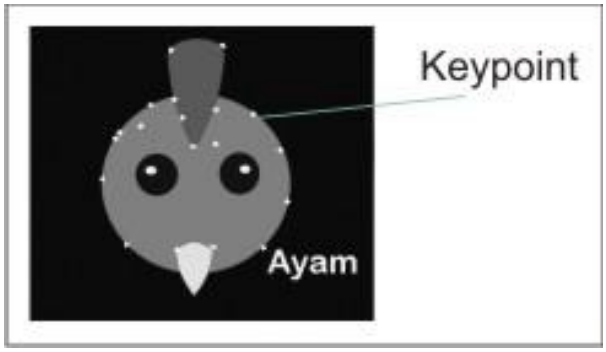
objek 3D agar dapat dikalkulasikan pada saat objek 3D diletakan, objek 3D tersebut akan didesuaikan objeknya berdasarkan fungsi variable “Tracking_object” dengan fungsi yang digunakan adalah

Setelah objek telah diinisialisasi selanjutnya sistem akan melakukan proses persiapan pelacakan dengan mengintegrasikan setiap acuan yang ada pada file trackeng.xml. File Tracker.xml itu sendiri merupakan file yang berisikan informasi-informasi objek marker pada saat proses pembuatan marker yang digunakan sebagai acuan untuk penyamaan pola marker.

```
errorStatus, trackingIndex =
MLTPlugin:startTracking("tracker/tracker.xml",
vidCap:getVidCapID(), camera)
```

`startTracking(“tracker/tracker.xml”,vidCap:getCapID(),camera)` merupakan fungsi yang digunakan untuk memulai pelacakan dengan mengacu terhadap file *tracker*.xml, dimana isi dari *tracker* tersebut adalah merupakan informasi-informasi mengenai gambar marker yang telah diinisialisasikan pada tahap pembuatan.

Fungsi tersebut berfungsi untuk mengintegrasikan informasi yang dihasilkan melalui kamera secara realtime yang kemudian menghasilkan pola-pola marker atau *keypoint* berdasarkan *key frame* dari objek marker yang terdeteksi sesuai dengan file object.bin dari setiap marker.




**Gambar 3.11*Keypoint* Citra**

### 3.1.9.3 Penyamaan Pola

Penyamaan pola adalah merupakan tahapan dimana proses menyamakan polaberdasarkan sebuah *keyframe*pada tahap sebelumnya.Proses disini diasumsikan bahwa tahap untuk proses *tracking* benar-benar telah berjalan.Setiap aktivitas dalam status telah terjadi tracking ataupun tidak terjadi tracking terhadap marker, secara keseluruhan *library* yang melakukan setiap prosesnya.

```
errorStatus, gtrackingStatusAyam =
MLTPlugin:getTargetStatus(trackingIndex,
trackingObjectIndexAyam)
errorStatus, gtrackingStatusSuaraAyam =
MLTPlugin:getTargetStatus(trackingIndex,
trackingObjectIndexsuaraAyam)
errorStatus, gtrackingKeyFrameIndex =
MLTPlugin:getRecognizedKeyFrameIndex(trackingIndex,
trackingObjectIndex)

if (gtrackingStatusAyam == 1) then

MLTPlugin:getTargetPos(trackingIndex,
trackingObjectIndexAyam, trackingPosition,
trackingOrientation)

trackingObjectAyam:setPosition(trackingPosition, camera)

trackingObjectAyam:setOrientation(trackingOrientation,
camera)

if not trackingObjectAyam:getVisible() then
trackingObjectAyam:setVisible(true) end

else

if trackingObjectAyam:getVisible() then
trackingObjectAyam:setVisible(false) end

end
if (gtrackingStatusSuaraAyam == 1 ) then

if not suaraAyam:isPlaying() then suaraAyam:play() end

else

if suaraAyam:isPlaying() then suaraAyam:stop() end

end
```

Fungsidari getTargetStatus(trackingIndex,trackingObjectIndexAyam) / `getTargetStatus(trackingIndex, trackingObjectIndexSuaraAyam)` merupakan fungsi yang dugunakan untuk mendapatkan status bahwa pelacakan target telah diterima. `getrRecognizedKeyFrameIndex(trackingIndex, trackingObjectIndex)` merupakan fungsi yang digunakan untuk mendapakan hasil suatu objek apakah dikenali atau tidak pada saat *tracking*.

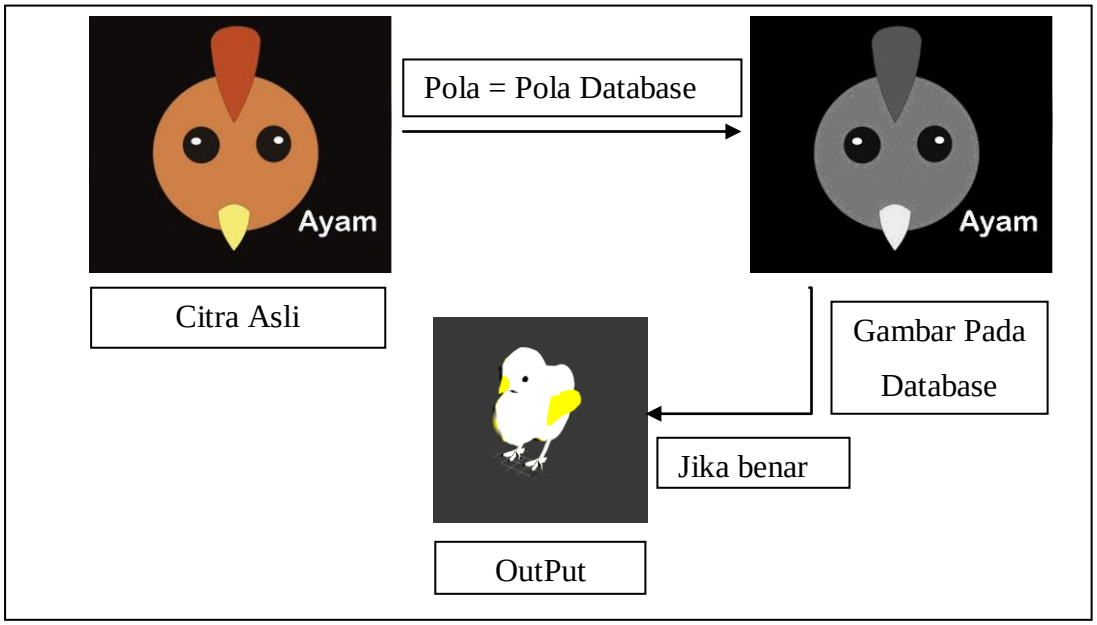
Pada proses tersebut terjadi proses pencocokan pola gambar merker terhadap objek yang menjadi keluaran.

1. File Objek 3D

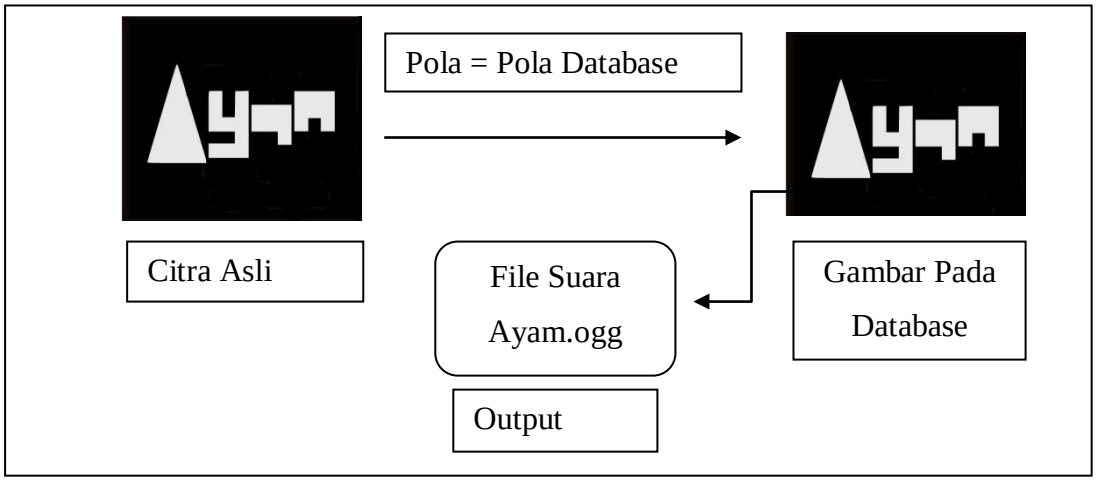
   Jika *tracking* ada (`gtrackingStatusAyam==1` ) maka set pola 3D ayam (`trackingObjectAyam`) yang sesuai dengan marker ayam (`trackingObjectIndexAyam`) dengan target posisi (`setPosition(trackingPosition, camera)`) orientasi (`setOrientation(trackingOrientation, camera)`) kemudian set objek 3D ayam menjadi ada (`setVisible(false)`), jika tidak terjadi *tracking* maka set 3D ayam menjadi hilang (`setVisible(true)`).

2. File suara

   Jika *tracking* ada ( `gtrackingStatusSuaraAyam==1` ) dan suara sedang tidak berjalan (`not suaraAyam:isPlaying()` ) maka keluarkan suara ayam (`suaraAyam:play()`), jika suara sedang berjalan (`suaraAyam:isPlaying()` ) maka set suara menjadi berhenti (`suaraAyam:stop()`).




**Gambar 3.12Gambar Penyamaan Pola Marker Objek 3D**




**Gambar 3.13 Gambar Penyamaan Pola Marker Suara**

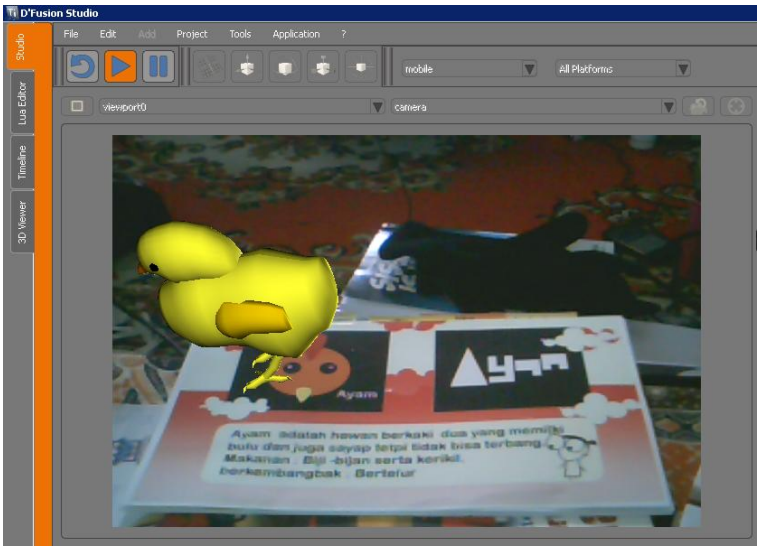
### 3.1.9.4 *Rendering* Objek

Setelah mendapatkan pola-pola terhadap marker pada proses pencocokan pola, maka sistem akan menempatkan bentuk 3D dan suara yang sesuai dengan objek gambar marker. Posisi animasi 3D tersebut akan terlihat berada di atas gambar marker.

```
MLTPlugin:getTargetPos(trackingIndex,
trackingObjectIndexAyam, trackingPosition,
trackingOrientation)
trackingObjectAyam:setPosition(trackingPosition, camera)
trackingObjectAyam:setOrientation(trackingOrientation,cam
era)

trackingObjectAyam:getVisible() then
trackingObject2:setVisible(true)
-- sound --
suaraAyam : play()
```

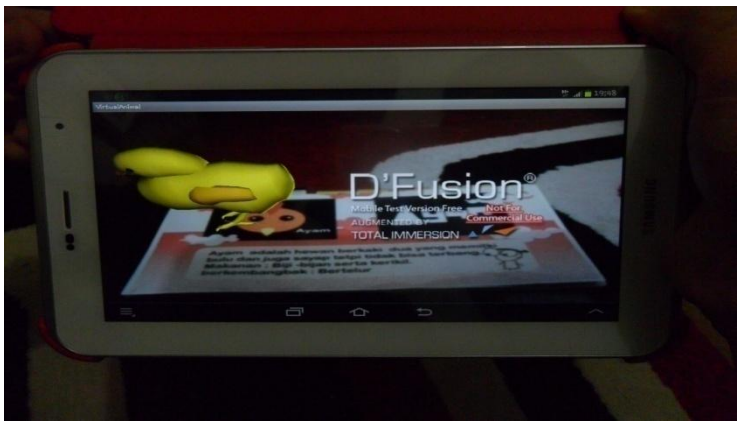
`getTargetPos(trackingIndex,trackingObjectIndexAyam, trackingPosition,trackingOrientation)` merupakanfungsiyang digunakan untuk mendapatkan informasi untuk meletakan objek 3D diatas marker. `setPosition(tarckingPosition, camera)` merupakan fungsi yang digunakan untuk mengatur posisi objek terhadap kamera. `setOrientation(trackingOrientation, camera)` merupakan fungsi yang digunakan untuk mengatur orientasi objek terhadap kamera. `getVisible()` fungsi yang digunakan unrtuk menampilkan objek 3D agar dapat terlihat dengan objek tersebut yang akan diset dengan fungsi `setVisible(true)` agar dapat terlihat. Agar suara dapat dimunculkan, maka fungsi untuk suara tersebut diset *play*().




**Gambar 3.14*Rendering* objek 3D dan suara**

### 3.1.9.5 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 menggunakan D'fusion *Studio* dapat digunakan atau dapat diimplementasikan dalamaplikasi mobileharus dengan menggunakan sebuah*library*, *libray* tersebut yaitu berupa D'Fusion *Mobile* SDK untuk Android, dimana didalam *library* tersebut berupa fungsi-fungsi yang digunakan untuk memanggil *scenario AR*serta menyertakan *class* tiComponent untuk mengatur segala inisialisasi pada framelayout dan untuk mengatur segala aktivitas aplikasi *mobile*.




**Gambar 3.15 Gambar Implemtasi *Mobile* AR**

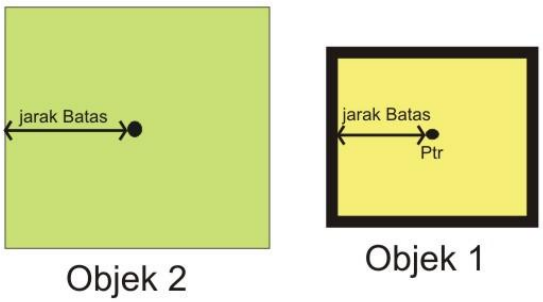
### 3.1.9.6 Analisis *Occlusion*

Pada analisis metode ini menjelaskan bahwaada tidaknya *occlusion* dalam penampilan objek 3D, secara sederhana *occlusion* mendefinisikan dimana suatu objek marker tidak dapat terdeteksi dikarenakan terhalang oleh objek lain. Akan tetapi ada beberapa persyaratan bahwa objek tersebut mengalami *occlusion* yaitu dimana persamaan 5 dan 6 terpenuhi.

$$02x - batas \leq 01x \leq 02x + batas \qquad (5)$$

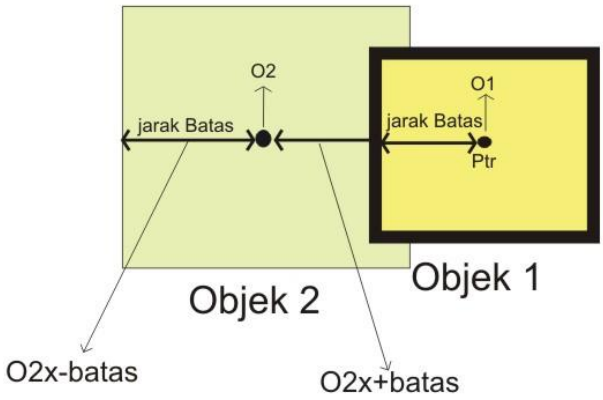
$$02y - batas \leq 01y \leq 02y + batas \qquad (6)$$

Untuk menjelaskan persamaan (5) agar tidak terjadi event akan diuraikan seperti pada Gambar 3.16. Misalkan titik tengah objek1 diwakili oleh matrik O1(x,y) dan objek 2 diwakilili oleh matrik O2(x,y).




**Gambar 3.16Objek matrik O1(x,y) dan O2(x,y)**

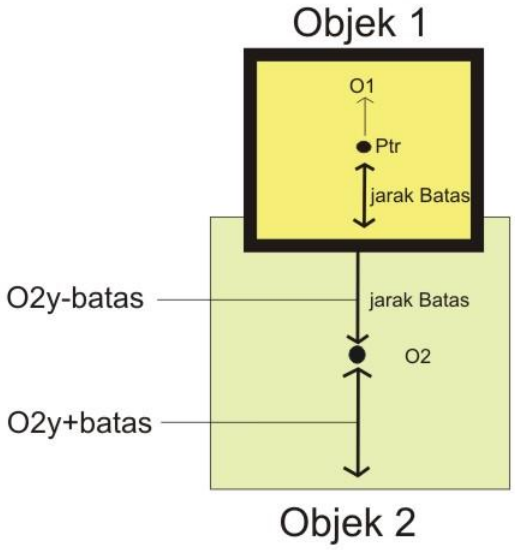
Untuk persamaan (5) tidak terjadi *event*jika :




**Gambar 3.17Objek matik O1 Tidak terjadi event terhadap persamaan (5)**

O2x-batasadalah merupakan setengah bagian dari objek matrik O2 yang tidak menutupi koordinat Ptr pada Objek O1, kemudian persinggungan antara matrik O1 lebih kecil dari pada batas objek O2x+batas, maka dari hal tersebut titik objek matrik O1 tidak terjadi *event.*

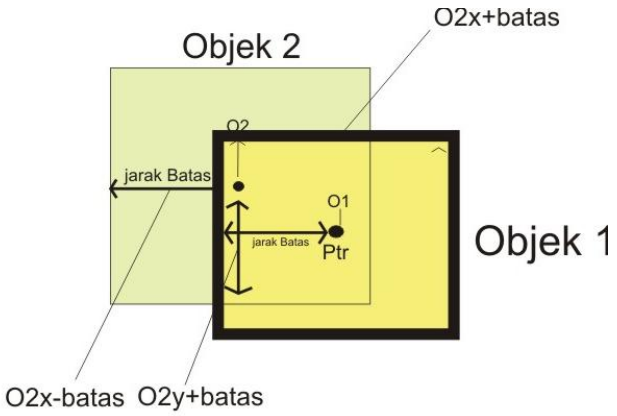
Untuk persamaan (6) tidak terjadi *event*jika :




**Gambar 3.18 Objek O1 tidak terjadi event terhadap persamaan (6)**

O2y-batas adalah merupakan setengah bagian dari objek matrik O2(x,y) yang tidak menutupi koordinat Ptr pada Objek O1(x,y), kemudian persinggungan antara matrik O1 lebih kecil dari pada batas objek O2y+batas, maka dari hal tersebut titik objek matrik O1 tidak terjadi *event.*

Untuk persamaan (5) dan (6) dapat tepenuhi dan terjadi *event*jika :




**Gambar 3.19 Objek matrik O1(x,y) yang memenuhi persamaan (5) dan (6)**

Dari Gambar 3.19 Dapat terlihat bahwa objek matrik O1(x,y) melebihi batas dari objek O2(x,y) O2x+batas, O2y+batas dan O2x-batas, O2y-batas, oleh karena itu objek matrik O1 terjadi *event*karena telah memenuhi persamaan (5) dan (6) yaitu Objek matrik O1(x,y) berada di dalam wilayah objek matrik O2(x,y).

## 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 *Virtual Animal* ini. Perangkat lunak tersebut adalah sebagai berikut ;

1) Sistem operasi Window Xp Propesional.
2) D'Fusion Studio 3.25
3) Eclipse Juno

4) Autodesk 3ds Max 2010
5) Corel Draw x4
6) Photoshop CS3
7) Sistem operasi Android 2.2

### 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 *Virtual Animal* inimenggunakan perangkat keras sebagai pendukungnya adalah sebagai berikut :

1) *Proccesor* intel Pentium T2390
2) Ram 1GB
3) *Mouse* dan *keyboard*
4) Back camera 3 Mega Pixel
5) buku bergambar
6) Tabelt Samsung Galaxy Tab 2

### 3.2.3 Analisis *User*

Analsis *user* ini dimaksudkan untuk siapa saja yang dapat menggunakan aplikasi *Virtual Animal* ini. Disini hanya di khususkan di pergunakan hanya oleh *user* yang telah memiliki pengawasan dalam penggunaan hardware baik seperti guru, orang tua maupun anak-anak asalkan untuk anak-anak berada di bawah pengawasan orang tua. Hak akses yang di berikan hanya melihat animasi 3D dan suara yang di hasilkan oleh aplikasi.

## 3.3 Anlisis kebutuhan Fungsional

Kebutuhanfungsionalberhubungan denganfitursoftwareyangakandibuat ataudikembangkan. Berikutini adalahtahapananalisiskebutuhanfungsionalSistem Aplikasi *mobile virtual animal*.

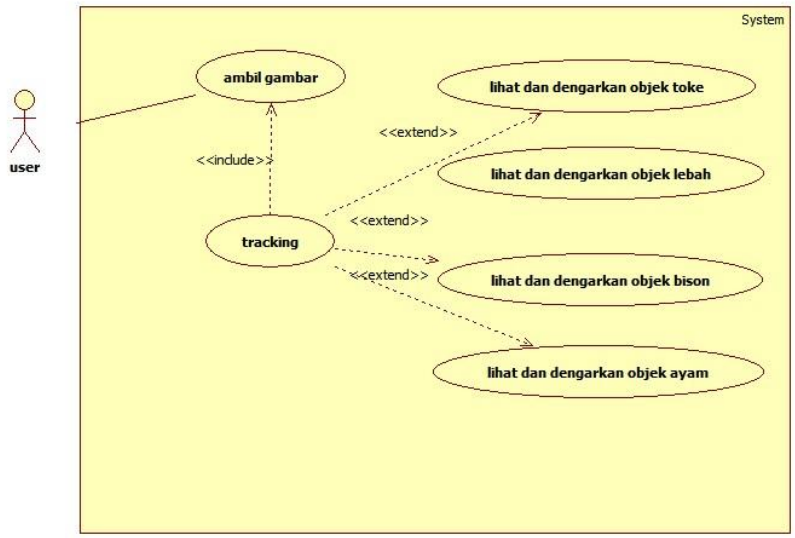
Analisiskebutuhan fungsionalmenggambarkanproses kegiatanyangakan diterapkan dalamsebuahsistemdanmenjelaskankebutuhanyang diperlukan sistem

agar sistem dapatberjalan dengan baik.

Analisisyangdilakukandimodelkandenganmenggunakan UML (*Unified Modeling Language*).Tahap-tahappemodelandalamanalisistersebutantara lain identifikasi aktor, *usecase diagram, skenario, sequence diagram, activity diagram, classdiagram*.

### 3.3.1 *Usecase*Diagram

*Usecase Diagram* merupakan konstruksi untuk mendeskripsikan hububungan-hubungan yangterjadiantaraktordenganaktivitas yangterdapat pada sistem. Sasaranpemodelan*usecase*diantaranyaadalahmendefinisikan kebutuhan fungsionaldanoperasionalsistemdenganmendefinisikanskenariopenggunaansistem yangakandibangun.Dari hasila nalisis aplikasi yang ada maka *use case diagram* untuk aplikasi ini dapatdilihat pada Gambar 3.20 :




**Gambar 3.20Gambar *UsecaseDiagram* Aplikasi**

#### 3.3.1.1 DefinisiUsecase

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

Sasaran pemodelan *use case* diantaranya adalah mendefinisikan

kebutuhan fungsional dan operasional sistem dengan mendefinisikanskenariopenggunaanyangdisepakatiantarapemakaidan pengembang.

**Tabel 3.2 Definisi *Usecase***

| *Usecase Aplikasi User* | | |
|---|---|---|
| No | UseCase | Deskripsi |
| UCU-01 | Ambil gambar | Fungsional untiuk Mengambil Gambar menggunakan kamera telpon seluler. |
| UCU-02 | Tracking | Fungsional untuk membandingkan gambar yang diambil kamera telepon dengan gambar yang ada pada file tracking. |
| UCU-03 | Lihat dan Dengarkan objek Ayam | Fungsional untuk menampilkan informasi objek ayam. |
| UCU-04 | Lihat dan Dengarkan objek Bison | Fungsional untuk menampilkan informasi objek bison. |
| UCU-05 | Lihat dan Dengarkan objek Toke | Fungsional untuk menampilkan informasi objek toke. |
| UCU-06 | Lihat dan Dengarkan objek Lebah | Fungsional untuk menampilkan informasi objek lebah. |

### 3.3.1.2 Skenario *Usecase*

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

#### 3.3.1.2.1 Skenario *Usecase User*

Skenario proses-proses yang terdapat dalam *usecase diagram* aplikasi *user*.

**Tabel 3.3 Skenario Use Case Ambil Gambar**

| **Identifikasi** | |
|---|---|
| Nama Usecase | Ambil Gambar Marker |
| Aktor | Pengguna *User* |
| Tujuan | Menjalankan aplikasi dengan menyorotkan kamera ke pada objek tertentu atau gambar, agar dapat berfungsi untuk mengambil gambar oleh sistem dan dilakukan proses pencocokan pola. |
| Keadaan awal | Sistem menampilkan gambar yang akan di *tracking.* |
| **Skenario Utama** | |
| **Aksi Aktor** | **Reaksi Sistem** |

| | |
|---|---|
| 1.Menyorotkan aplikasi *AnimalVirtual*pada marker. | |
| | 2. Menjalankan Kamera telepon seluler untuk deteksi gambar markeryang akan diproses. |
| | 3.Apabila data sesuai dengan gambar yang ada file tracking maka akan keluar animasi dan suara hewan yang sesuai dengan gambar. |
| **Skenario Gagal Ambil Gambar** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| | 1. "Layar Kosong". |
| **Keadaan Akhir** | Aplikasi menampilkan gambar yang sedang di tracking. |

**Tabel 3.4*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 3.5 Skenario *Use Case* Lihat dan Dengarkan Objek Ayam**

| **Identifikasi** | |
|---|---|
| Nama *Usecase* | Lihat dan Dengarkan objek Ayam. |
| Aktor | Pengguna *User* |
| Tujuan | Menampilkan dan mendengarkan suara hewan. |

| **Scenario Utama** | |
|---|---|
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Menjalankan aplikasi dan meyorotkan kamera pada gambar. | |
| | 2. pencocokan pola / *tracking* gambar Marker |
| | 3. Pemilihan objek 3D atau suara. |
| **Keadaan Akhir** | Aplikasi menampilkan hewan virtual dan suara hewan tersebut secara *real time* |
| **Skenario Gagal Lihat dan Dengar Objek Ayam** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| | 1.”Layar Kosong” |
| **Keadaan Akhir** | Aplikasi tetap menampilkan gambar yang sedang di*tracking.* |

**Tabel 3.6 Skenario *Use Case* Lihat dan Dengarkan Objek Bison**

| **Identifikasi** | |
|---|---|
| Nama *Usecase* | Lihat dan Dengarkan objek Bison. |
| Aktor | Pengguna *User* |
| Tujuan | Menampilkan dan mendengarkan suara hewan. |
| **Scenario Utama** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Menjalankan aplikasi dan meyorotkan kamera pada gambar. | |
| | 2. pencocokan / *tracking* pola gambar Marker |
| | 3. Pemilihan objek 3D atau suara. |
| **Keadaan Akhir** | Aplikasi menampilkan hewan virtual dan suara hewan tersebut secara *real time* |
| **Skenario Gagal Lihat dan Dengar Objek Bison** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| | 1.”Layar Kosong” |
| **Keadaan Akhir** | Aplikasi tetap menampilkan gambar yang sedang di*tracking.* |

**Tabel 3.7 Skenario *Use Case* Lihat dan Dengarkan Objek Toke**

| **Identifikasi** | |
|---|---|
| Nama *Usecase* | Lihat dan Dengarkan objek Toke. |
| Aktor | Pengguna *User* |

| | |
|---|---|
| Tujuan | Menampilkan dan mendengarkan suara hewan. |
| **Scenario Utama** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Menjalankan aplikasi dan meyorotkan kamera pada gambar. | |
| | 2. pencocokan pola/ *tarcking* gambar Marker |
| | 3. Pemilihan objek 3D atau suara. |
| **Keadaan Akhir** | Aplikasi menampilkan hewan virtual dan suara hewan tersebut secara *real time* |
| **Skenario Gagal Lihat dan Dengar Objek Toke** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| | 1.”Layar Kosong” |
| **Keadaan Akhir** | Aplikasi tetap menampilkan gambar yang sedang di*tracking.* |

**Tabel 3.8 Skenario *Use Case* Lihat dan Dengarkan Objek Lebah**

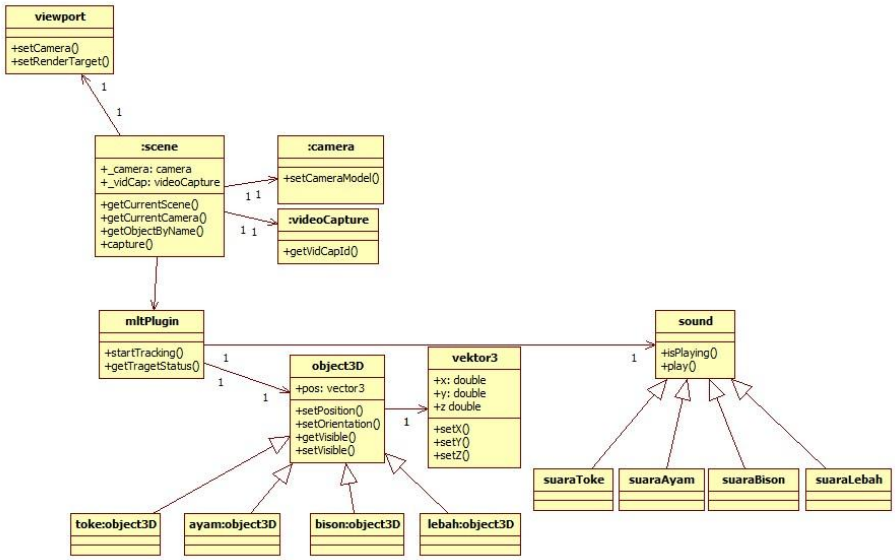
| **Identifikasi** | |
|---|---|
| Nama *Usecase* | Lihat dan Dengarkan objek Lebah. |
| Aktor | Pengguna *User* |
| Tujuan | Menampilkan dan mendengarkan suara hewan. |
| **Scenario Utama** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Menjalankan aplikasi dan meyorotkan kamera pada gambar. | |
| | 2. pencocokan pola/ *tracking* gambar Marker |
| | 3. Pemilihan objek 3D atau suara. |
| **Keadaan Akhir** | Aplikasi menampilkan hewan virtual dan suara hewan tersebut secara *real time* |
| **Skenario Gagal Lihat dan Dengar Objek Lebah** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| | 1.”Layar Kosong” |
| **Keadaan Akhir** | Aplikasi tetap menampilkan gambar yang sedang di*tracking.* |

### 3.3.2 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 *AnimalVirtual*.

### 3.3.2.1 *Class Diagram AnimalVirtual*




**Gambar 3.21Gambar *Class Diagram AnimalVirtual***

**Tabel 3.9Deskripsi *Class DiagramAnimalVirtual***

| *User Application* | | |
|---|---|---|
| *Class* | Jenis *Class* | Deskripsi |
| *scene* | *Control* | Class yang berisikan yang mengontrol scenario utama dan class lain. |
| mltPlugin | *Control* | Class yang berisikan method untuk memulai tracking dan pencocokan pola. |
| *camera* | *Control* | Class yang berisikan metod-metod untuk fungsionalitas *camera*. |
| videoCapture | *Control* | Class yang berisikan metod-metod untuk fungsionalitas *videoCapture*. |
| Object3D | *Entity* | Class yang berisikan metod-metod untuk menampilkan sebuah objek. |
| Ayam:object3D | *Entity* | Class yang berisikan metod-metod untuk menampilkan sebuah objek ayam. |
| Bison:object3D | *Entity* | Class yang berisikan metod-metod untuk |

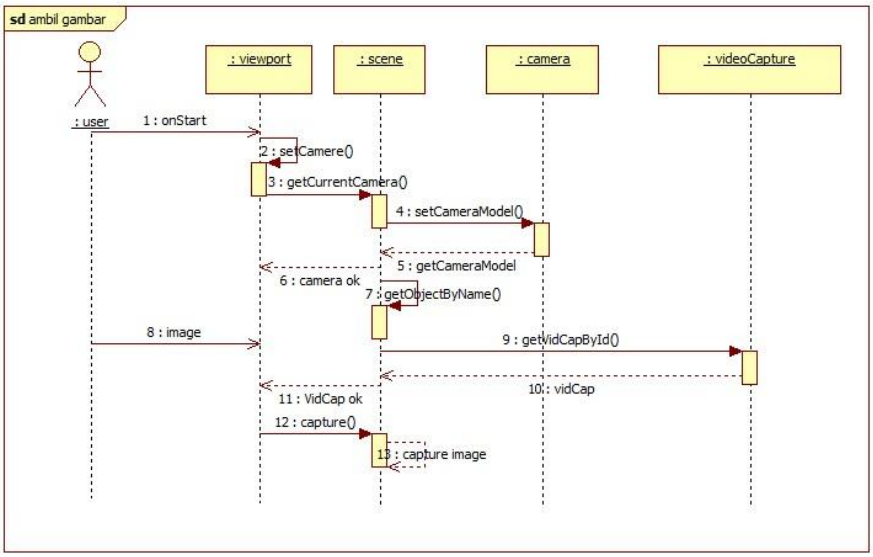
| | | |
|---|---|---|
| | | menampilkan sebuah objek bison. |
| Toke:object3D | *Entity* | Class yang berisikan metod-metod untuk menampilkan sebuah objek toke. |
| Lebah:object3D | *Entity* | Class yang berisikan metod-metod untuk menampilkan sebuah objek lebah. |
| *Sound* | *Entity* | Class yang berisikan metod-metod untuk menghasilkan suara. |
| suaraAyam:sound | *Entity* | Class yang berisikan metod-metod untuk menghasilkan suara. |
| suaraBison:sound | *Entity* | Class yang berisikan metod-metod untuk menghasilkan suara. |
| suaraToke:sound | *Entity* | Class yang berisikan metod-metod untuk menghasilkan suara. |
| suaraLebah:sound | *Entity* | Class yang berisikan metod-metod untuk menghasilkan suara. |
| *viewport* | *bounday* | Class yang berisikan fungsionalitas untuk menampilkan objek. |
| Vektor3 | *control* | Class yang berisikan metod-metod untuk menghasilkan nilai posisi objek. |

### 3.3.3 Sequence Diagram

*Sequencediagram*merupakan sebuah diagram yang menggambarkaninteraksiantarmasing-masing objek pada setiap*use case*dalamurutan waktu.Interaksiiniberupa pengiriman serangkaian dataantarobjek-objekyangsalingberinteraksi.

#### 3.3.3.1 *Sequence Diagram* Ambil Gambar

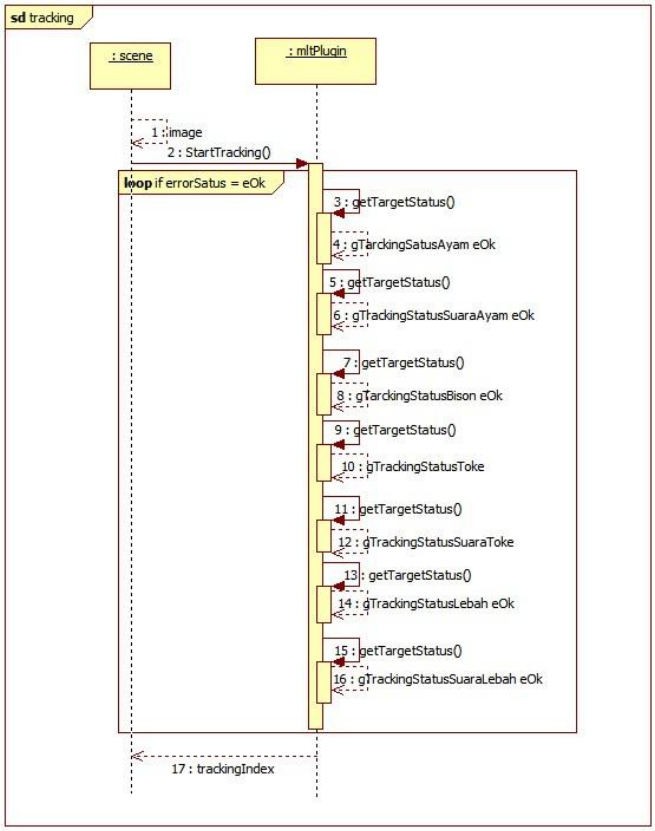
*Sequence diagram*Ambil Gambar menunjukkan proses yang berfungsi untuk mengambil gambar yang digunakan sebagai inisialisasi awal.*Sequence diagram*Ambil Gambar dapatdilihat pada Gambar 3.22.




**Gambar 3.22Gambar *Sequence* Ambil Gambar**

### 3.3.3.2 *Sequence Diagram Tracking*

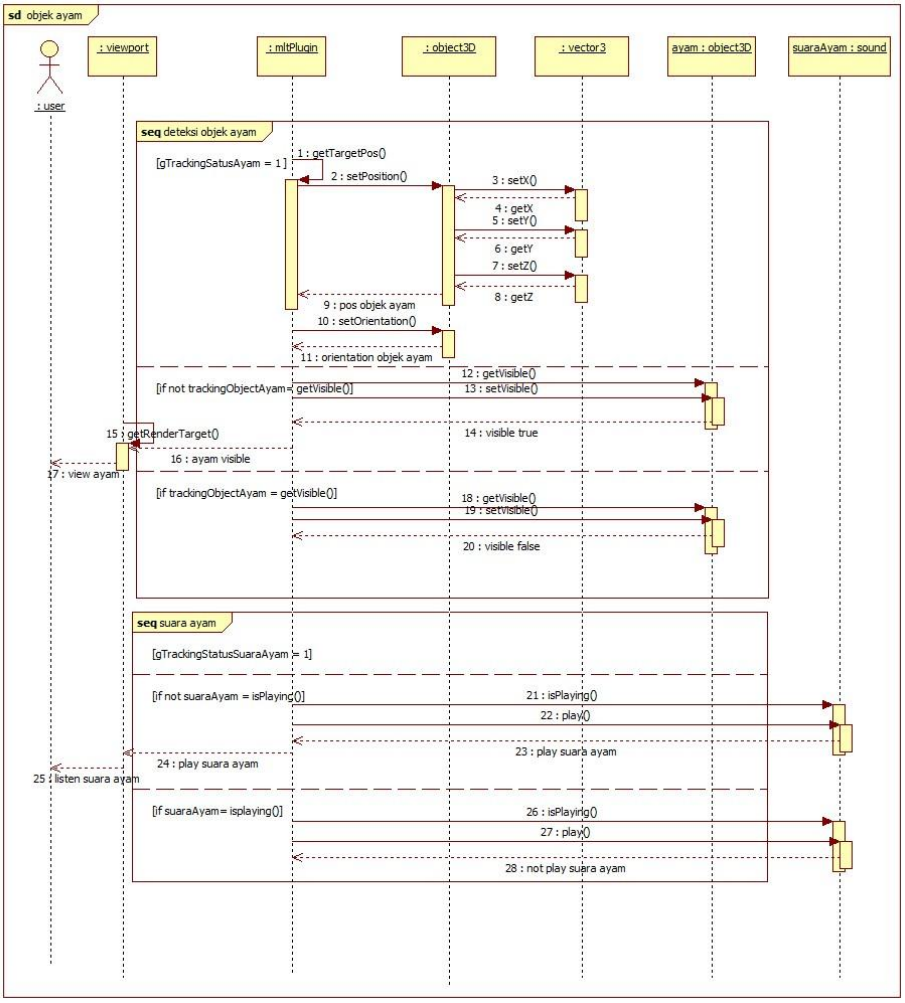
*Sequence diagram tracking*menunjukkan proses untuk pencocokan pola agar dapat menghasilkan objek yang diharapkan.*Sequence diagramtracking*dapatdilihat pada Gambar 3.23.




**Gambar 3.23Gambar *Sequence Diagram Tracking***

### 3.3.3.3 *Sequence diagram* Lihat dan Dengarkan Objek Ayam

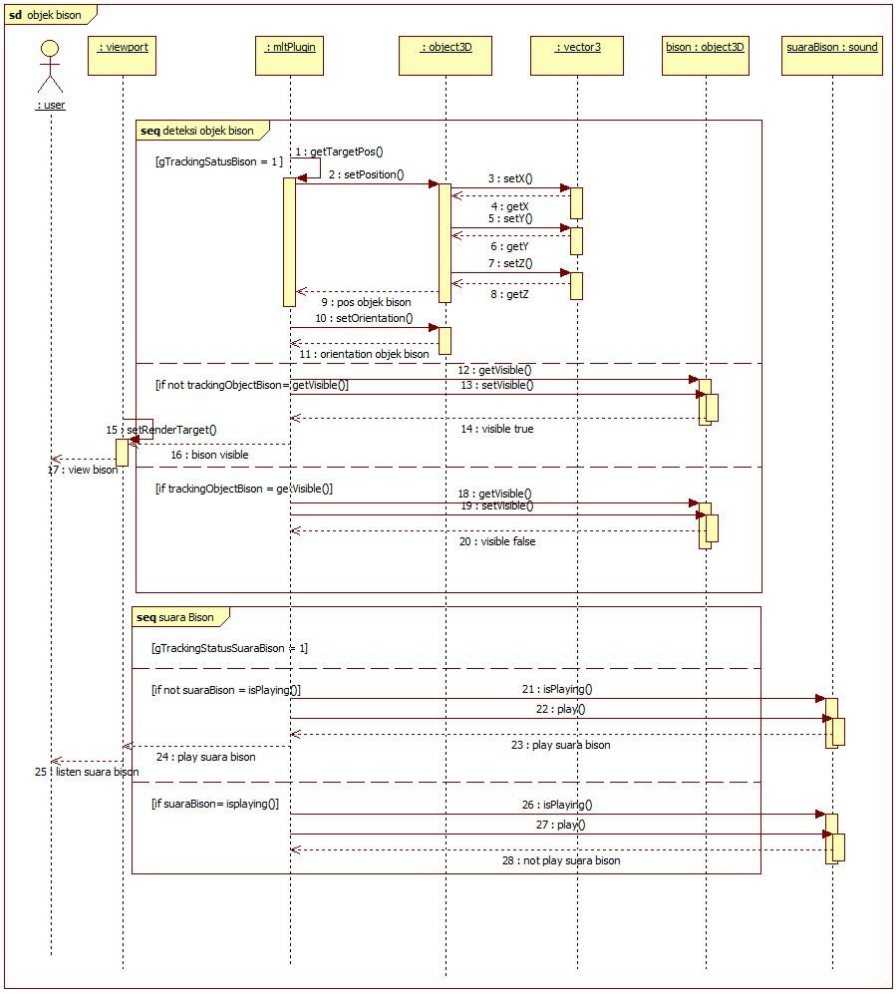
*Sequence diagram*Lihat dan Dengarkan Objek Ayam menunjukkan proses yang berfungsi untuk menampilkan objek ayam dan menghasilkan suara ayam.*Sequence diagram*Lihat dan Dengarkan Objek Ayamdapatdilihat pada Gambar 3.24.




**Gambar 3.24Gambar *Sequence diagram* Lihat dan Dengarkan objek Ayam.**

### 3.3.3.4 *Sequence diagram* Lihat dan Dengarkan Objek Bison

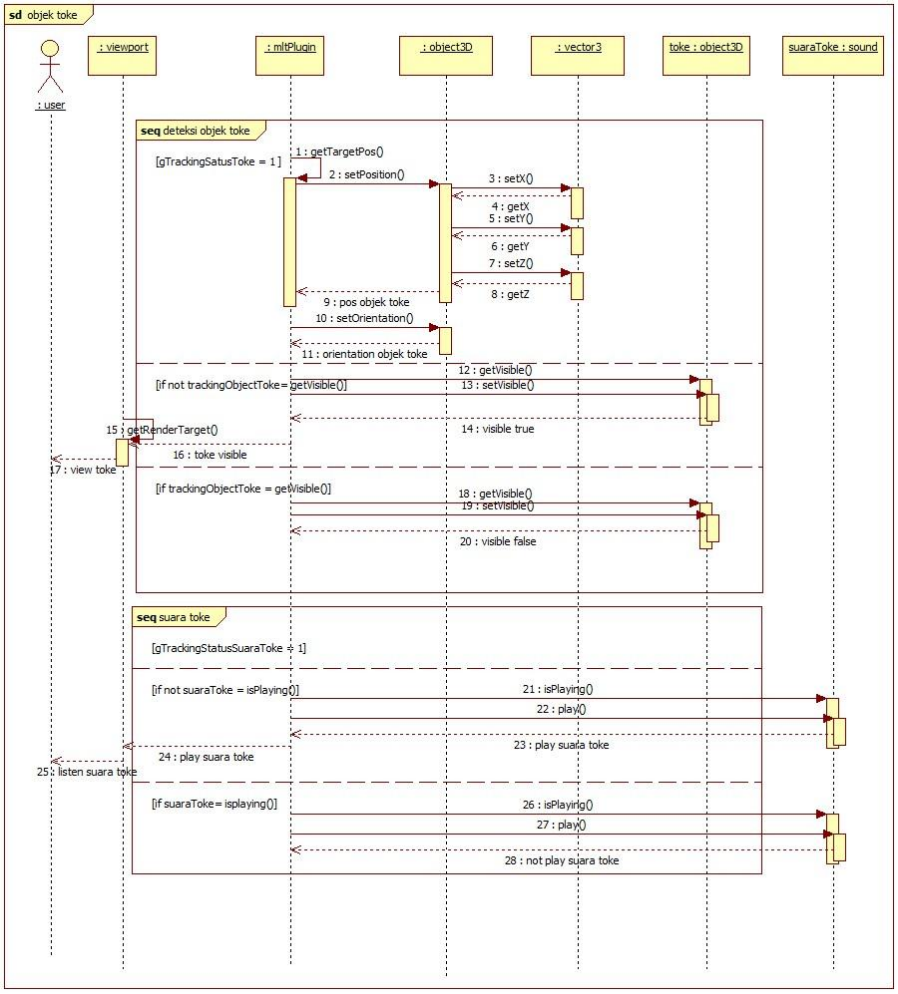
*Sequence diagram*Lihat dan Dengarkan Objek Bison menunjukkan proses yang berfungsi untuk menampilkan objek bison dan menghasilkan suara bison.*Sequence diagram*Lihat dan Dengarkan Objek Bisondapatdilihat pada Gambar 3.25.




**Gambar 3.25Gambar *Sequence diagram* Lihat dan Dengarkan Objek Bison.**

### 3.3.3.5 *Sequence diagram* Lihat dan Dengarkan Objek Toke

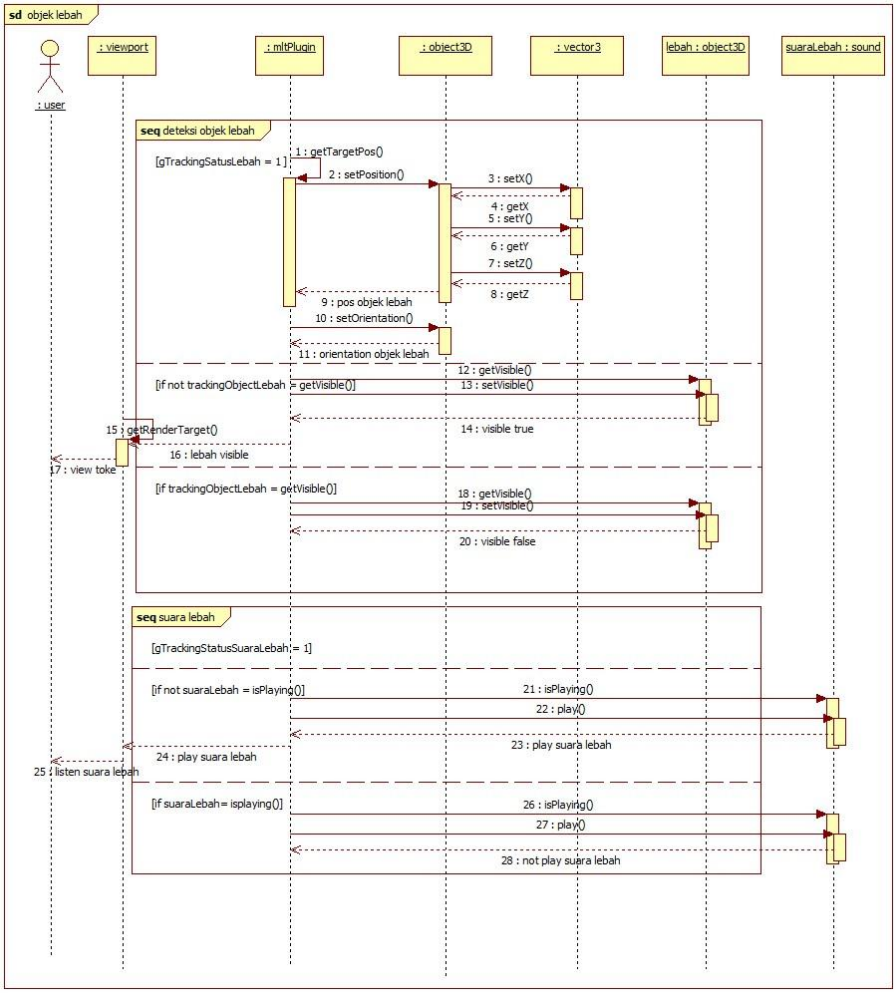
*Sequence diagram*Lihat dan Dengarkan Objek Toke menunjukkan proses yang berfungsi untuk menampilkan objek toke dan menghasilkan suara toke.*Sequence diagram*Lihat dan Dengarkan Objek Tokedapatdilihat pada Gambar 3.26.




**Gambar 3.26Gambar *Sequence diagram* Lihat dan Dengarkan Objek Toke.**

### 3.3.3.6 *Sequence diagram* Lihat dan Dengarkan Objek Lebah

*Sequence diagram*Lihat dan Dengarkan Objek Lebah menunjukkan proses yang berfungsi untuk menampilkan objek lebah dan menghasilkan suara lebah.*Sequence diagram*Lihat dan Dengarkan Objek Lebahdapatdilihat pada Gambar 3.27.



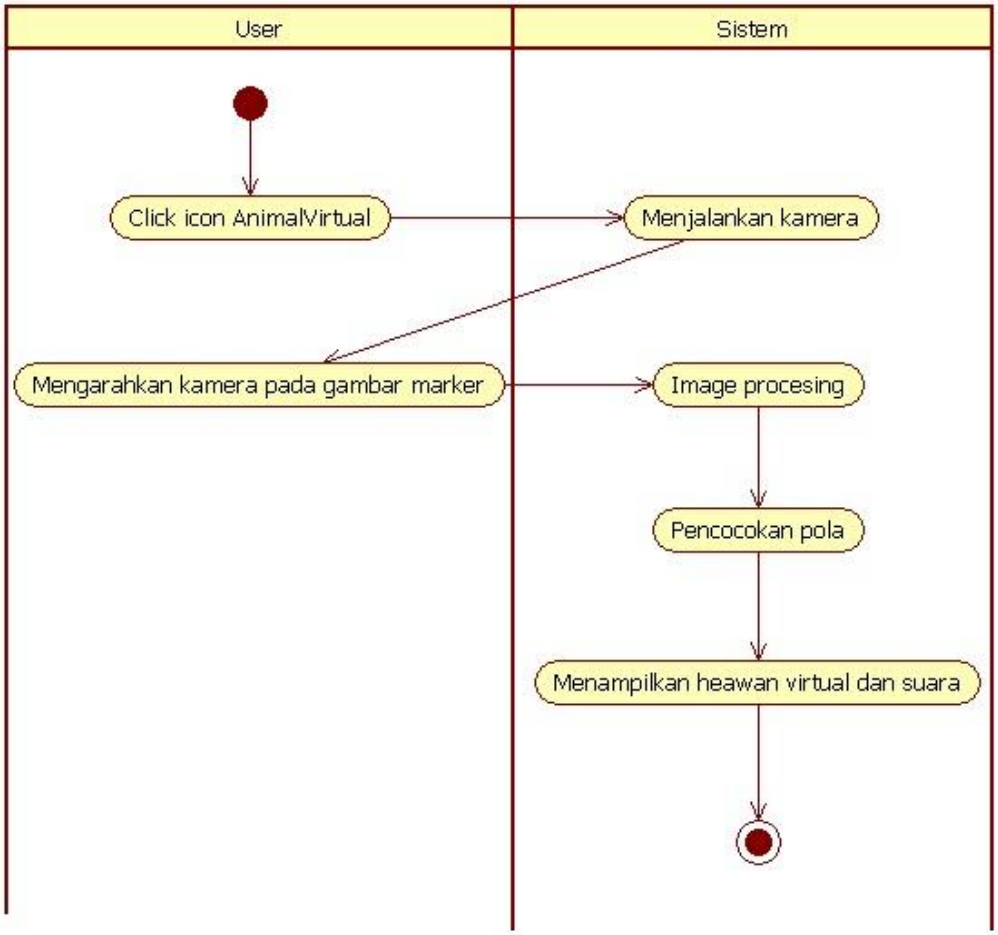

**Gambar 3.27Gambar *Sequence diagram* Lihat dan Dengarkan Objek Lebah.**

### 3.3.4 Activity Diagram

*Activity diagram*merupakandiagramyang memodelkanalirankerja atau *workflow*dariurutanaktifitas dalam suatu prosesyang mengacu pada *use case diagram* yang ada. Berikut ini penjelasan dari *activity diagram*:

#### 3.3.4.1 *Activity DiagramTrackingMarker*

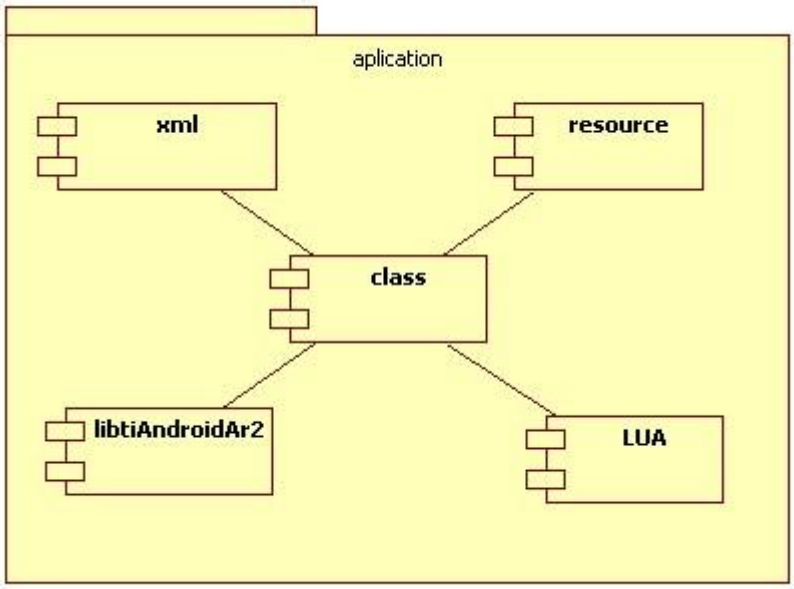
*Aktivity diagram Tracking Marker* ini menjelaskan aliram kerja aktor pengguna pada saat kamera menyorot gambar agar dapat di-*tracking* yang akan dicocokan melalui aplikasi *mobile*. *Aktivity diagram Tracking Marker*dapat di lihat pada Gambar 3.28.




**Gambar 3.28 Gambar *Activity diagramTracking Marker***

### *3.3.5 Component Diagram*

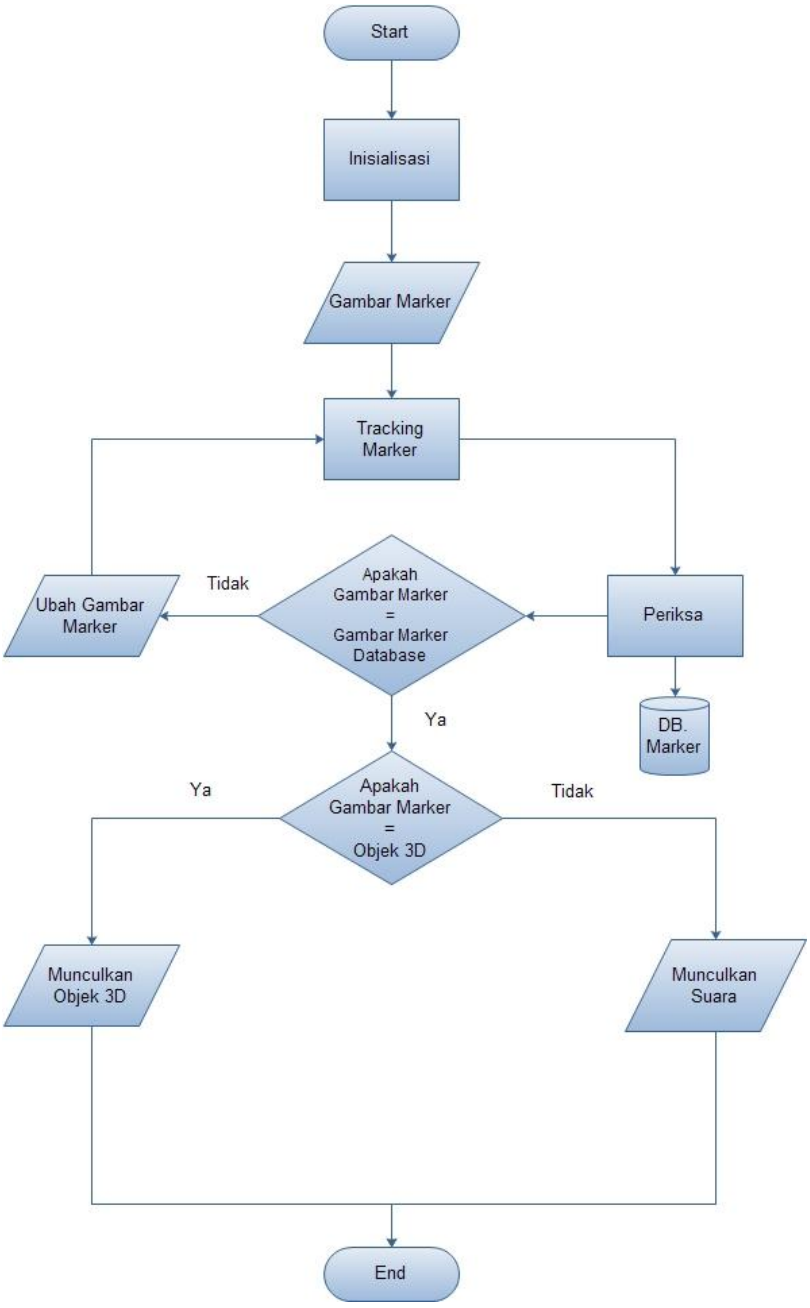
Diagram komponen atau *component diagram* digunakan untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem. Adapun Diagram komponen yang digunakan dalam perancangan sistem dapat dilihat pada Gambar 3.29.




**Gambar 3.29Gambar *Component Diagram* Aplikasi *Animal Virtual***

### 3.3.6 Flowchart Diagram

*Flowchart* adalah penggambaran secara grafik dari langkah-langkah danurut-urutan prosedur dari suatu program.*Flowchart* menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yanglebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalampengoperasian.




**Gambar 3.30Gambar Alur *Flowchart*Aplikasi sistem**

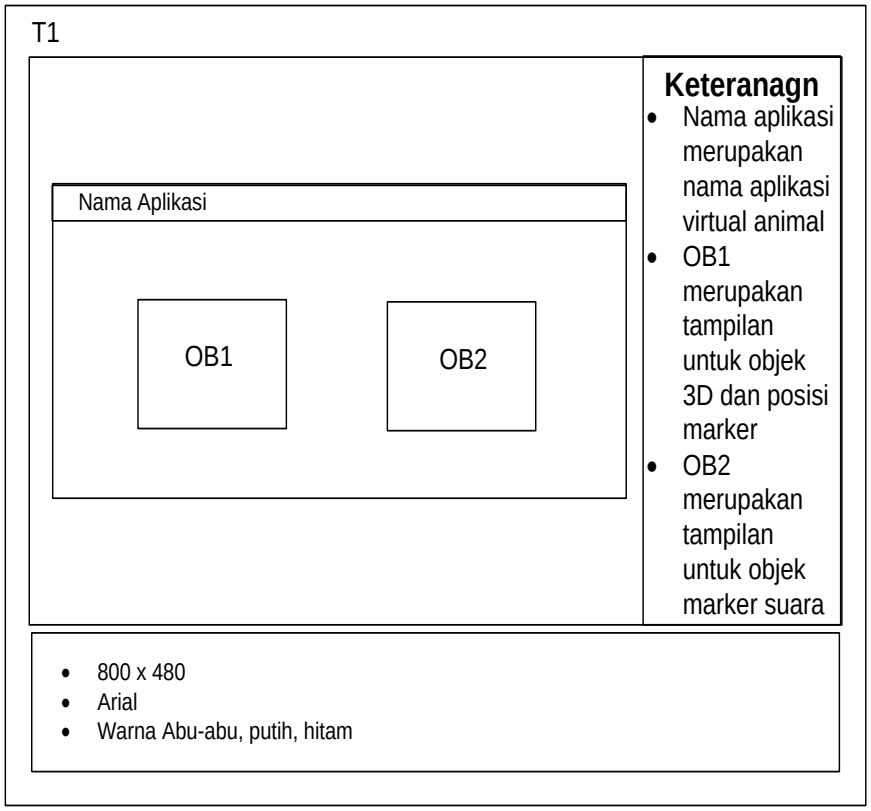
## 3.4 Perancangan Sistem

Perancangan akan dimulai setelah tahap analisis terhadap sistem selesai dilakukan. Perancangan dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi.

### 3.4.1 Perancangan Grafis Antarmuka

Tahap perancangan desain bertujuan untuk mencari bentuk yang optimal dari aplikasi yang akan dibangun dengan pertimbangan faktor-faktor permasalahan dan kebutuhan yang ada pada sistem seperti yang telah ditetapkan pada tahap analisis. Dalam tahap ini upaya yang dilakukan yaitu dengan cara mengkombinasikan penggunaan teknologi perangkat keras dan perangkat lunak yang tepat sehingga diperoleh desain yang optimal dan mudah di implementasikan.

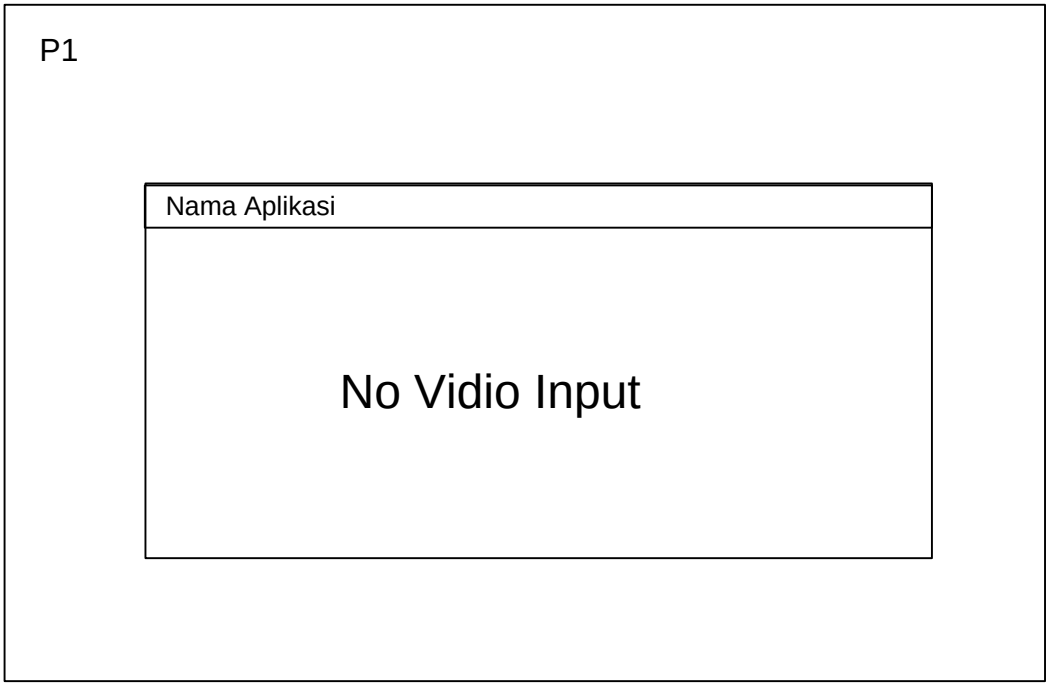
#### 3.4.1.1 Analisis Antarmuka Aplikasi AnimalVirtual




**Gambar 3.31Gambar Perancangan Antarmuka aplikasi *AnimalVirtual***

### 3.4.1.2 Perancangan Pesan

Pada Gambar 3.32merupakan perancangan pesan yang terdapat pada aplikasi *AnimalVirtual*. Adapun tampilan pesan yang ada adalah sebagai berikut :




**Gambar 3.32Gambar Perancangan Pesan**

Pesan tersebut mengindikasikan bahwa camera yang ada pada *hardware*tidak terdeteksi oleh aplikasi atau kamera tidak *support*terhadap aplikasi.