BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

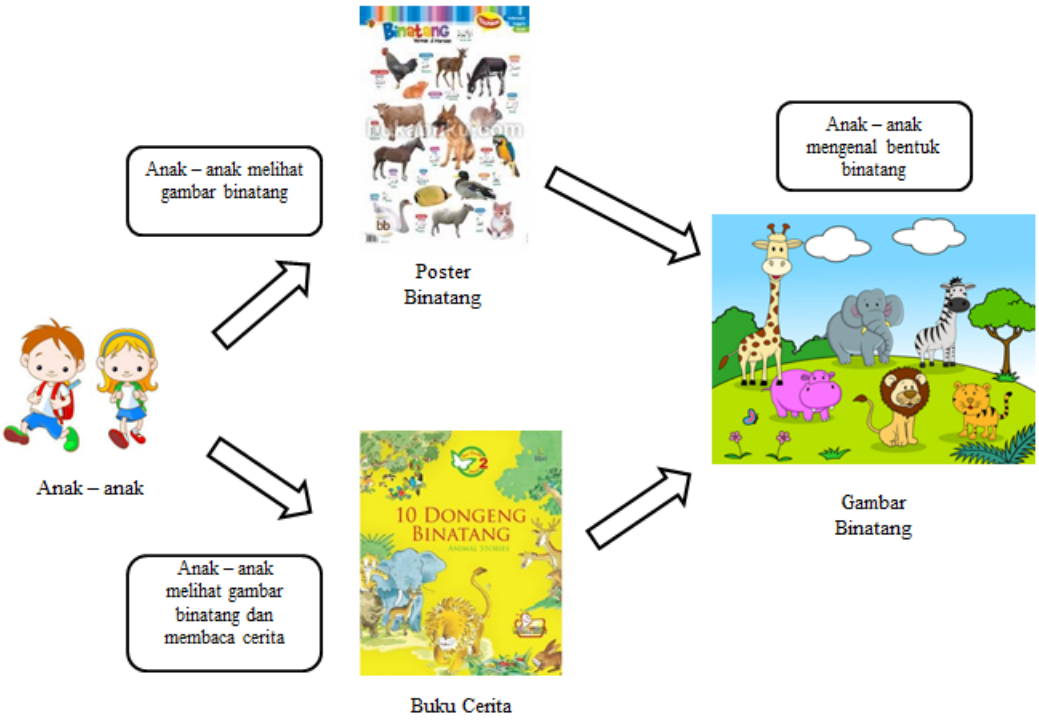
Merupakan penguraian dari suatu sistem yang sedang dibangun untuk mengidentifikasi permasalahan pada sistem dengan menentukan kebutuhan sistem tersebut. Analisis tersebut meliputi analisis masalah , analisis kebutuhan data dan analisis sistem. Sehingga dapat diusulkan perbaikan – perbaikan yang nantinya dapat menjadi acuan untuk membangun sistem yang diharapkan.

3.1.1 Analisis Masalah

Langkah awal dalam tahap analisis sistem dengan tujuan untuk mengidentifikasi masalah yang sedang berjalan pada sistem yang sudah ada. Sehingga dapat dievaluasi untuk menjadikan topik penulisan tugas akhir. Berdasarkan hasil wawancara antara penulis dengan Ibu Endah selaku Kepala Sekolah Putra Karya didapat masalah yakni penyampaian informasi tentang pengenalan binatang terhadap anak masih menggunakan poster binatang dan buku cerita yang isinya terdapat binatang, sehingga dibutuhkan sebuah media interaktif yang diharapkan dapat menambah ketertarikan anak kepada buku terutama yang mengandung banyak gambar seperti pada buku ensiklopedia *animal*. Dengan penerapan teknologi AR yang diharapkan si anak bisa mengetahui informasi yang didalamnya dengan tampilan objek menyerupai aslinya. Dan akan menjadi media pembelajaran yang interaktif kepada anak – anak.

3.1.2 Analisis Sistem Yang Berjalan

Analisis pada suatu sistem yang sedang berjalan. Sehingga dapat dijadikan acuan untuk pembangunan sistem ke tahap selanjutnya dengan tujuan yang diharapkan dapat tercapai. berikut proses sistem yang berjalan sekarang dapat di lihat pada Gambar 3.1

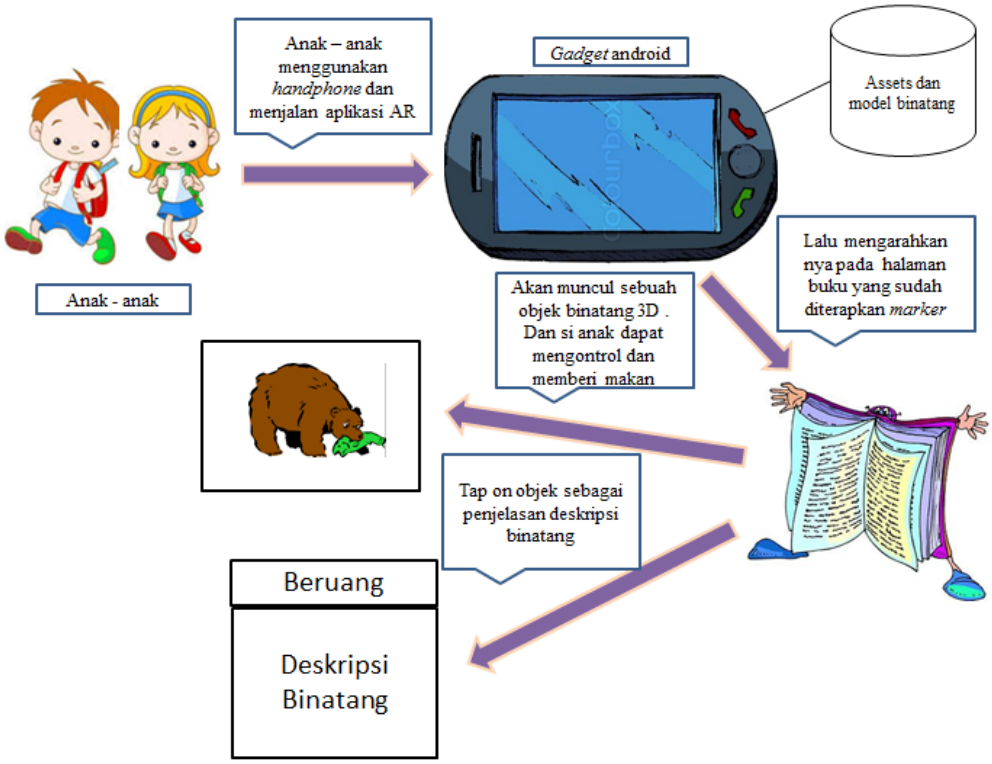


Gambar 3.1 Alur sistem yang berjalan

Sesuai dengan digambarkan pada Gambar 3.1. Anak – anak mendapatkan informasi tentang binatang dalam bentuk 2D dari poster binatang atau buku cerita yang cakupannya memiliki tema tentang binatang.

3.1.3 Analisis Arsitektur Sistem

Proses analisis arsitektur sistem yang akan di bangun oleh penulis menggunakan *multimarker* sebagai aksi memberi makan pada binatang dan menampilkan informasi berdasarkan kategori binatang yang dibangun pada perangkat *mobile*. Sehingga diharapkan dalam penggunaannya sangat mudah digunakan oleh anak – anak dimanapun dan kapanpun. Dibutuhkan sebuah buku ensiklopedia *animal* agar aplikasi ini dapat digunakan. Adapun gambar arsitektur sistem yang akan dibangun dapat dilihat pada gambar 3.2.

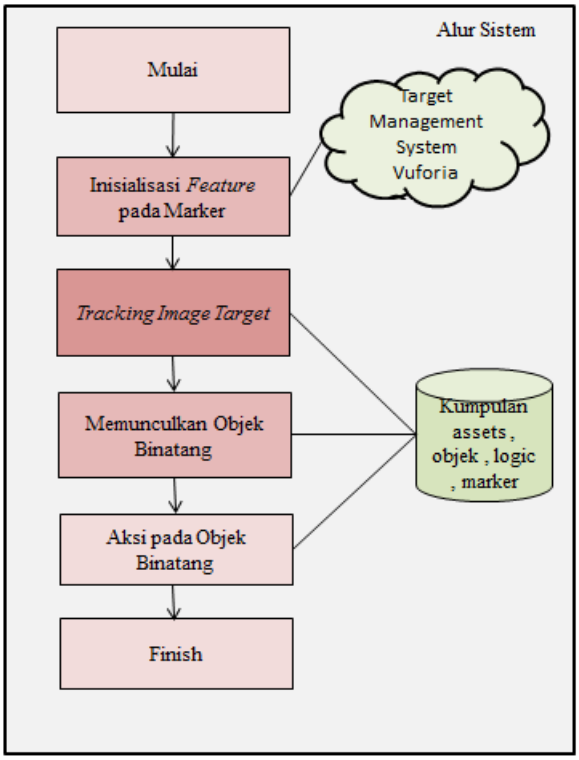


Gambar 3.2 Arsitektur Sistem

1. *User* adalah anak – anak yang berumur 4 – 7 tahun. Penggunaan dibawah pengawasan orang tua dan guru.
2. *User* akan menjalankan aplikasi pada perangkat *mobile* untuk memulai penjelajahan mengenal dunia binatang dengan implementasi teknologi AR.
3. Buku ensiklopedia *animal* merupakan media informasi tentang binatang yang terdapat banyak gambar dari binatang tersebut.
4. Menghasilkan animasi binatang , penjelasan dengan suara dan *text* sebagai informasi dan interaksi dengan *multimarker* sebagai acuan untuk memberi makan dengan metode *raycast*.

3.1.4 Analisis Alur Sistem

Menjelaskan bagaimana alur – alur sebuah sistem yang akan dibangun. Dan di definisikan pada setiap bagian yang menjadi acuan dalam membangun sebuah sistem. Adapun alur sistem pada pembangunan aplikasi ini dapat dilihat pada gambar 3.3.



Gambar 3.3 Analisis Alur Sistem

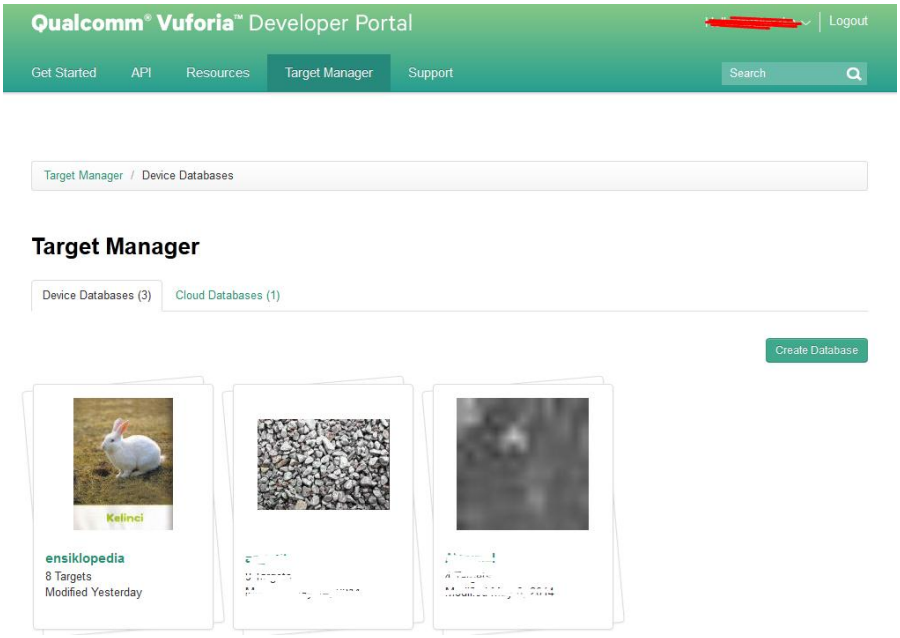
Berdasarkan gambar 3.3 dapat disimpulkan bahwa alur sistem pada analisis aplikasi ini adalah sebagai berikut :

1. Inisialisasi *Feature* pada marker merupakan proses dimana sebuah gambar dengan ekstensi .PNG dan .JPG agar dapat digunakan sebagai *marker*.
2. *Tracking Image Target* merupakan alur dari proses kamera pada *mobile* agar dapat mendeteksi sebuah *marker*.

3. Memunculkan Objek Binatang merupakan tahapan pada saat pendeteksian *image target* sesuai dengan yang ada didatabase sehingga akan muncul objek.
4. Aksi pada objek binatang menerapkan fitur *raycasting* sebagai aksi pada objek binatang agar dapat berjalan dan memberi makan.

3.1.4.1 Inisialisasi *Feature* dan *Rating* pada Marker

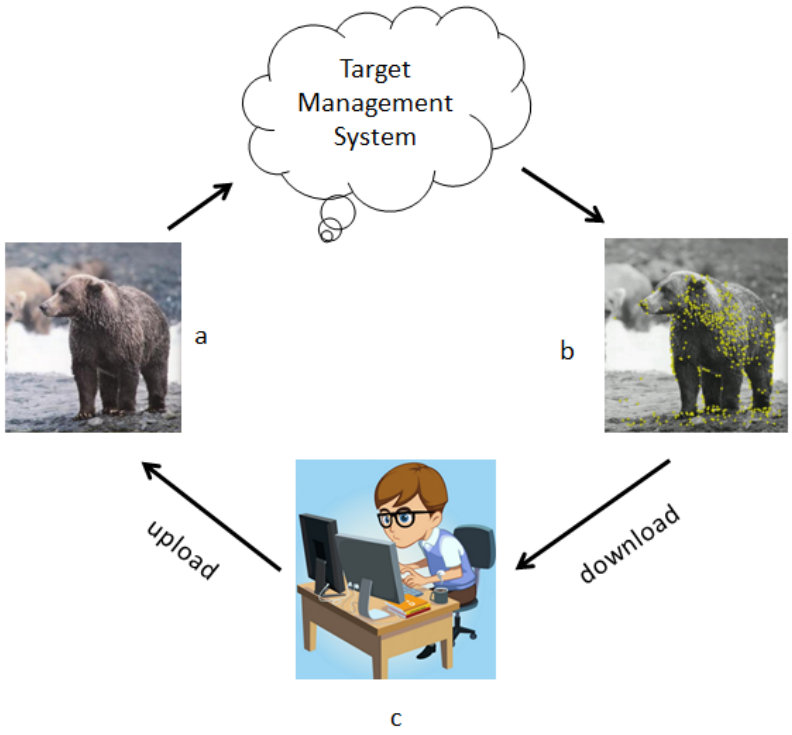
Tahapan sebuah gambar akan dijadikan sebuah penanda atau *marker* yang dalam penyususan tugas akhir ini penulis menggunakan library *vuforia* sebagai pendukung untuk membuat aplikasi *augmented reality* dan unity3d sebagai *integrated development environment*. Pada dasarnya *vuforia* sudah menyediakan *tools* sebagai inisialisasi *feature* pada *marker* yang bernama *target manager system.* Berfungsi memberikan *feature* dan *rating* pada gambar yang dijadikan *marker*. Gambar 3.4 menunjukan halaman dari *target manager.*



Gambar 3.4 *Target Manager System* pada website vuforia

Fungsi utama dari *target manager* ialah memudahkan para *developer* yang menggunakan library *vuforia* untuk membangun sebuah

aplikasi AR. Pihak *vuforia* memberikan kemudahan agar developer tidak perlu susah untuk membuat sebuah *marker* yang akan digunakan sebagai acuan munculnya sebuah objek pada *marker*. Gambar 3.5 menunjukkan alur dari bagaimana membuat sebuah marker pada *target manager.*



Gambar 3.5 Gambar *non Feature* (a) Gambar *with Feature* (b) *Developper* (c)

Deskripsi dari alur pada gambar 3.5 adalah sebagai berikut

1. *Developer* memilih gambar dengan ektensi .JPG dan .PNG dengan *image enchament Grayscale* atau *RGB* dan maksimal ukuran gambar 2.25MB.
2. Melakukan *upload* pada *target management system* dan memilih *type target* yang akan digunakan dapat dilihat pada gambar 3.6.



Gambar 3.6 *Type* Target yang dapat digunakan

3. *Developer* mengunduh ekstensi editor yang akan digunakan sebagai penggunaan pada IDE. Ekstensi yang disediakan oleh *vuforia* ada 2 yaitu .unitypackage dan .zip.

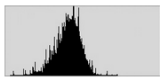
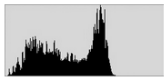
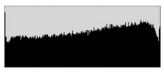
Berikut merupakan implementasi inisialisasi *feature* dan *rating* pada marker yang digunakan pada aplikasi ini. Hasil dari feature pada marker yang digunakan dapat dilihat pada gambar 3.7.



Gambar 3.7 *Feature* dan *Rating* pada *Marker* Beruang

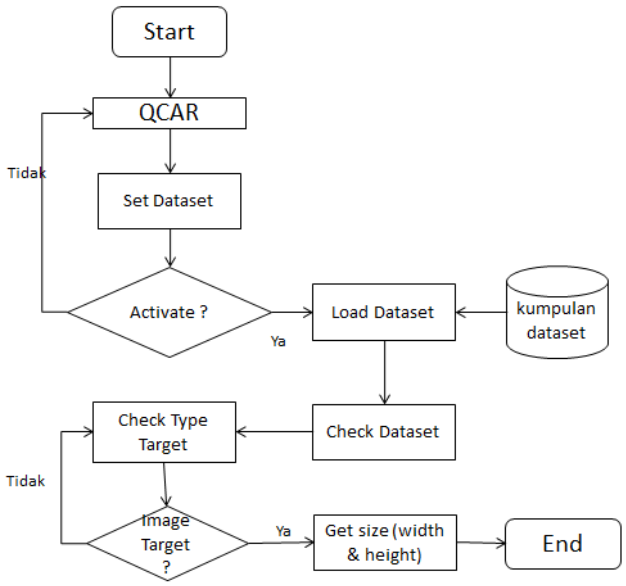
Terdapat beberapa kriteria untuk menjadikan sebuah marker mempunyai nilai *rating* yang baik. Dapat dilihat pada tabel 3.1 merupakan contoh yang mendefinisikan citra mempunyai grafik histogram berbeda – beda dan hasil dari ratingnya pun berpengaruh.

Tabel 3.1 Kriteria rating berdasarkan Histogram

Histogram	Rating
	★☆☆☆☆
	★★★☆☆
	★★★★★
	☆☆☆☆☆
	★★★★☆

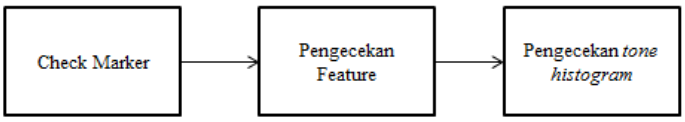
3.1.4.2 *Tracking Image Target*

Proses tracking marker. *vuforia SDK* akan memproses *image target* yang tertangkap kamera. Adapun alur pendeteksian *image target* pada unity3d dapat dilihat pada gambar 3.8.



Gambar 3.8 *Flowchart* pendeteksian *Image Target* pada Unity3d

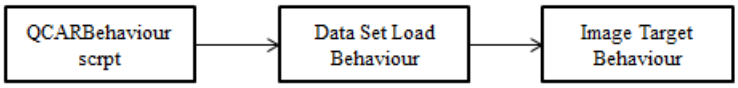
Melakukan beberapa tahapan seperti pada Gambar 3.8. *Vuforia SDK* menggunakan metode *tracking feature* dan *rating* sebagai acuan bahwa gambar yang dijadikan sebagai marker minimalnya mendapat 3-5 bintang sebagai *rating*, agar pada saat kamera di arahkan ke sebuah marker. Objek yang muncul tidak mudah hilang atau *lost tracked.* Pada gambar 3.9 merupakan alur dari proses *tracking image target*.



Gambar 3.9 Proses Tracking Image Target

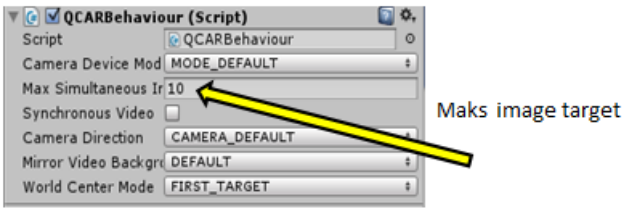
1. ***Check Marker***

Langkah awal untuk *check marker* pada *vuforia* memiliki alur seperti Gambar 3.10. Merupakan langkah – langkah pengaturan pada unity3d agar *image target* dapat dijadikan sebuah *marker* yang sebelumnya sudah mendapatkan *feature* dan *rating* pada *Target Manager System.*



Gambar 3.10 Proses *Load* Gambar

Langkah pertama pada saat *ARcamera* pada unity3d mengarah pada sebuah *image target*. *QCARBehaviour* akan melakukan pengecekan dengan nilai maksimal *image target* yang akan di deteksi. Apabila nilai yang ditentukan tidak sesuai, maka *ARkamera* hanya akan mendeteksi sesuai dengan nilai yang ditentukan dan dapat dilihat pada gambar 3.11 dan deskripsi dari setiap fungsi dapat dilihat pada tabel 3.2.



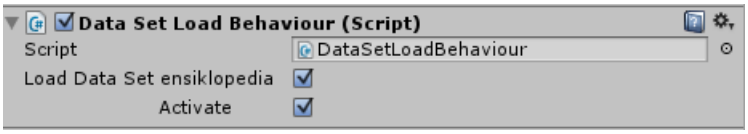
Gambar 3.11 Proses Pengecekan Maks *Image Target*

Tabel 3.2 Deskripsi QCARBehaviour (Script)

No	Nama *Method*	Keterangan
1	Camera Device Mode	Terdapat tiga opsi dalam penentuan fungsi ini terdapat *MODE_DEFAULT*, *MODE_OPTIMIZE_SPEED* dan

		MODE_OPTIMIZE_QUALITY. Yang berfungsi untuk memilih mode pada saat kamera mengarahkan pada marker dengan kualitas gambar yang baik, kecepatan yang baik tetapi marker tidak mudah *lost* atau default.
2	Max Simultaneous Image	Untuk menentukan jumlah *image target* yang dapat terdeteksi oleh *ARcamera* pada *vuforia*.
3	Synchronous Video	untuk menyinkronkan *render update* dengan *update* kamera, tidak dianjurkan untuk adegan dengan konten animasi.
4	Camera Direction	Penggunaan kamera pada hardware baik *mobile* atau *pc*. Terdapat beberapa fungsi seperti *CAMERA_DEFAULT*, *CAMERA_FRONT* dan *CAMERA_BACK*. Berfungsi untuk memilih penggunaan kamera yang akan digunakan.
5	Mirror Video Background	Sebuah efek cermin AR di mana video latar belakang yang diberikan menciptakan efek cermin.
6	World Center Mode	ada beberapa fungsi seperti *SPECIFIC_TARGET* merupakan mengatur *target* tunggal sebagai pusat, *FIRST_TARGET* ialah target pertama dipilih untuk menjadi pusat dunia lalu *CAMERA* ialah kamera tetap di tempat dan semua target bergerak

Selanjutnya *ARCamera* akan mengecek *image target* yang aktif sebagai acuan muncul nya sebuah objek pada marker tersebut, adapun gambaran untuk mengaktifkan *dataset* dapat dilihat pada gambar 3.12.dan deskripsi dari setiap fungsi dapat dilihat pada tabel 3.3.

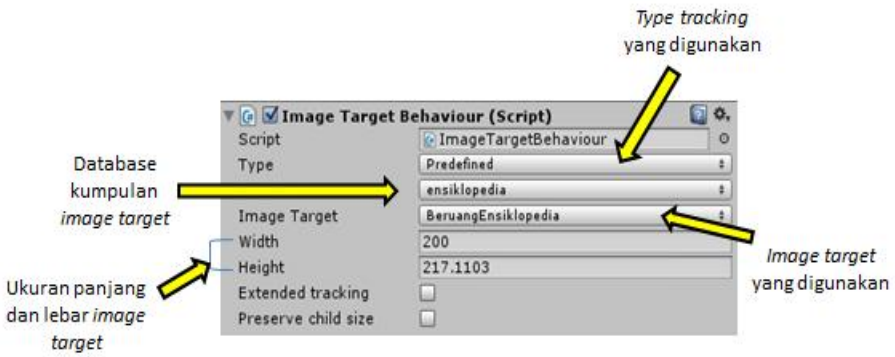


Gambar 3.12 Proses Mengaktifkan *Data Set* pada *marker*

Tabel 3.3 Deskripsi Data Set Load Behaviour (Script)

No	Nama *Method*	Keterangan
1	Load Data Set ensiklopedia	Set untuk memuat dataset ketika *scene* d iinisialisasi
2	Activate	Set untuk mengaktifkan dataset dimuat

Apabila sudah *activate* sesuai gambar 3.12. Selanjutnya penerapan gambar untuk dijadikan *image target* diaktifkan, adapun proses untuk menentukan *image target* dapat dilihat pada gambar 3.13 dan deskripsi dari setiap fungsi dapat dilihat pada tabel 3.4.



Gambar 3.13 Menentukan *marker* pada unity3d sebagai *image target*

Tabel 3.4 Deskripsi Image Target Behaviour (Script)

No	Nama *Method*	Keterangan
1	Type	Dalam pendefinisian awal vuforia mengenal 3 type *PREDEFINED* yakni pendefinisian type *image target* yang disimpan pada SD CARD, *CLOUD_RECO* yakni memanfaatkan CLOUD sebagai penyimpanan *asset* yang sudah di *apply* pada marker dan USER_DEFINED yakni pemanfaatan hak user untuk mendefinisikan *image target* sebagai acuan munculnya sebuah objek.
2	Data Set	Merupakan kumpulan *image target* dengan pendefinisian *data set* sebagai acuan utama.
3	Image Target	Disebut *marker* yang akan digunakan pada aplikasi. Sebelumnya sudah dilakukan penerapan metode *natural feature* dan *rating* pada *target management system vuforia*.
4	Width	Lebar marker yang default sudah didefinisikan pada saat proses penerapan feature pada *target management system.*
5	Height	Tinggi marker yang default sudah didefinisikan pada saat proses penerapan feature pada *target management system.*
6	Extended Tracking	Fitur ini memungkinkan pelacakan tingkat *persistance* setelah target telah terdeteksi.
7	Preserve Child Size	memungkinkan untuk mengubah skala target tanpa mempengaruhi skala relatif augmentation tersebut. dengan begitu kita dapat mengedit ekentesi file xml *image target* dan mengubat lebaran dan tinggian.

2. **Pengecekan *feature***

Kamera menemukan *image target*, proses selanjutnya untuk dapat menentukan marker tersebut digunakan sebagai *marker*. *vuforia* menggunakan metode *natural feature* dan *rating*. Artinya hasil dari menetapkan *key point* dengan berbentuk bintang. Akan menandai setiap sudut pada gambar yang terdeteksi. Berikut proses dasar yang bekerja pada *vuforia SDK* ialah mendeteksi dan melacak. Tidak seperti penanda tradisional, kode matriks dan kode QR. Sehingga *image target* tidak perlu dengan gambar hitam dan putih untuk menjadi *marker*. Bagian ini akan menunjukkan bagaimana menentukan *image target* dengan desain sasaran dengan mempertimbangkan bentuk yang tersusun.

Metode *natural feature* dan *rating* merupakan teknik penilaian seberapa baik sebuah gambar untuk dapat dideteksi oleh *ARcamera* pada *vuforia SDK*. *Rating* ini dapat dilihat pada *target manager* untuk setiap gambar yang diunggah oleh *developer* melalui web *vuforia*. Memilih ekstensi yang diinginkan sesuai bahasa pemrograman yang digunakan. Disini penulis menggunakan Unity3d sebagai pembangun aplikasi AR dan menjadikan ekstensi nya menjadi file .xml dan .dat berikut isi file tersebut.

```
 <?xml version="1.0" encoding="UTF-8"?>
<QCARConfig
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="qcar_config.xsd">
            <Tracking>
  <ImageTarget    name="Beruang"    size="200.000000
217.110260" />
      </Tracking>
</QCARConfig>
```

Lalu file .xml ini berisikan *feature* dan kesimpulan berdasarkan kategori yang di bahas pada website *vuforia*. Berikut hasil dari file tersebut dapat dilihat pada tabel 3.5.

Tabel 3.5 Feature dan Rating pada Marker

***Feature* pada Marker**	
	
High Local Contrast	**Ya**
Uniform Feature Distribution	**Ya**
High Feature Density	**Ya**
Repetitive Pattern	**Tidak**

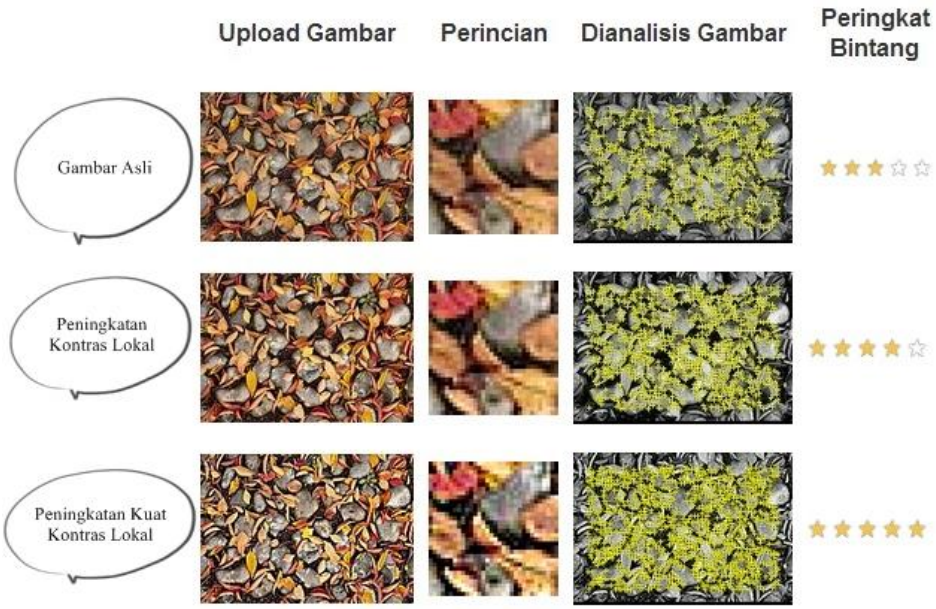
Beberapa kategori seperti *high local contrast, uniform feature distribution, high feature density dan repettive pattern.* Mempunyai jawaban antara ya dan tidak..

Rating dapat berkisar dari 0 sampai 5 untuk setiap gambar yang diberikan. Semakin tinggi *rating* dari target gambar, semakin kuat kemampuan deteksi dan pelacakan yang dikandungnya. Sebuah rating dari nol menunjukkan bahwa target tidak dilacak sama sekali oleh sistem AR, sedangkan rating bintang 5 menunjukkan bahwa sebuah gambar

dengan mudah dilacak oleh sistem AR. Adapun penjelasan dari beberapa kategori berdasarkan tabel 3.5.

a) *High Local Contrast*

Feature citra yang memiliki contrast semakin besar contrast pada citra tersebut maka semakin baik. Pada gambar 3.14 contoh dari beberapa pendefinisian *original image* peningkatan *contrast.*



Gambar 3.14 klasifikasi *high local contrast*

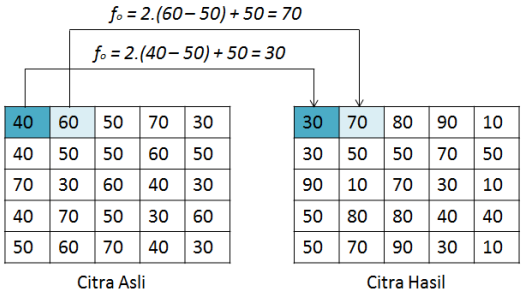
Pada Gambar 3.14 adalah contoh yang lebih praktis tentang bagaimana untuk meningkatkan kontras lokal target. Dengan menggunakan gambar dengan dua lapisan. Di latar depan adalah beberapa daun multi-warna. Latar belakang adalah permukaan bertekstur.

Adapun cara perhitungan untuk meningkatkan *contrast* pada suatu citra dengan melakukan perhitungan secara sistematis

$$f_o\,(x\,,y) = G.(x,y) - P\,) + P$$

Dimana G adalah koefisien penguatan kontras dan P adalah nilai *grayscale* yang dipakai sebagai pusat pengkotrasan.

Misal diketahui citra *grayscale* 128 warna dengan ukuran 5x5 piksel akan dilakukan operasi kontras dengan koefisien penguatan kontras G = 2 dan pusat pengkontrasan P = 50. Perhitungan fungsi kontras dilakukan sebagai berikut .



40	60	50	70	30
40	50	50	60	50
70	30	60	40	30
40	70	50	30	60
50	60	70	40	30

30	70	80	90	10
30	50	50	70	50
90	10	70	30	10
50	80	80	40	40
50	70	90	30	10

Gambar 3.15 Perhitungan mencari nilai kontras pada setiap Piksel

b) *Uniform Feature Distribution*

Semakin seimbang atau seragam fitur dalam citra atau banyaknya pola yang sama bentuk pada citra mengakibatkan penentuan titik fitur pada citra . Berikut contohnya pada gambar 3.16.

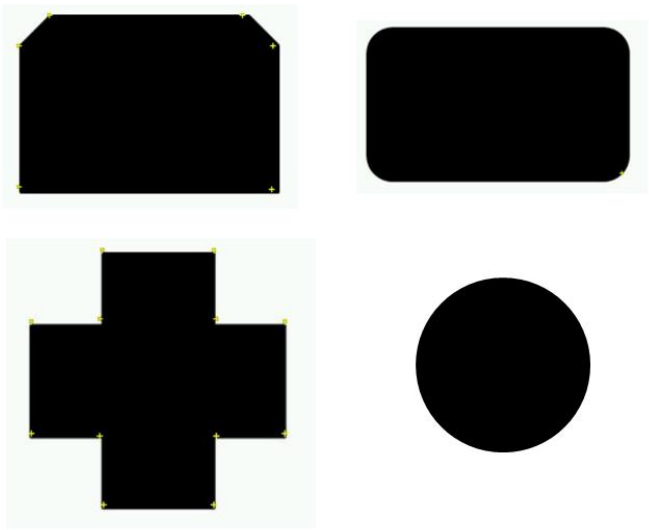


Gambar 3.16 klasifikasi *uniform feature distribution*

Pemotongan atau memilih citra yang memiliki banyak fitur dalam citra. Pada gambar 3.16 dapat dilihat pada citra a yang memiliki banyak fitur dalam beberapa pixel dibanding dengan pixel lainnya. Maka sebaiknya dilakukan *cropping* untuk daerah yang memiliki banyak fitur sehingga gambar yang digunakan sebagai *imagetarget* akan mudah dideteksi oleh sistem.

c) *High Feature Density*

Sebuah citra banyak mengandung feature yang menyebabkan rating yang dihasilkan semakin baik. Untuk mendapatkan kriteria high feature yang baik maka sebaiknya memilih sebuah citra yang memiliki banyak pola. Berikut contohnya pada gambar 3.17.



Gambar 3.17 klasifikasi *high feature density*

Dalam pendeteksian *feature density* pada gambar 3.17 sistem *target management system* mendeteksi adanya perbedaan nilai piksel pada citra. Misal pada citra *grayscale* 8x8 piksel dengan kedalaman 8 bit dengan kondisi sebagai berikut

$f(x,y) = 255$, jika $f(x,y) \geq T$

$f(x,y) = 0$, jika $f(x,y) < T$

50	50	40	20	90	120	100	50
40	130	10	200	230	50	150	150
120	20	50	230	200	60	20	25
15	180	205	225	205	230	210	65
75	210	250	210	230	205	215	30
120	100	5	220	190	80	20	95
30	40	10	240	240	50	150	40
50	40	120	90	120	140	30	110

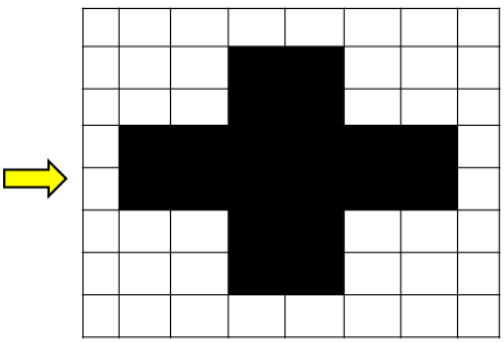
Citra *grayscale*

$$T = \frac{f_{maks} + f_{min}}{2} = \frac{250 + 10}{2} = 130$$

Bila nilai T = 130 diterapkan untuk citra tersebut diperoleh citra berikut

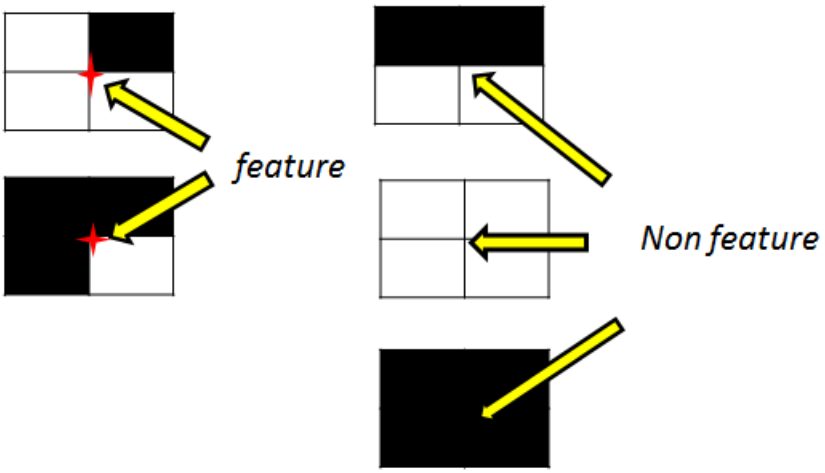
0	0	0	0	0	0	0	0
0	0	0	255	255	0	0	0
0	0	0	255	255	0	0	0
0	255	255	255	255	255	255	0
0	255	255	255	255	255	255	0
0	0	0	255	255	0	0	0
0	0	0	255	255	0	0	0
0	0	0	0	0	0	0	0

Citra Hasil Thresholding



Representasi dalam warna hitam putih

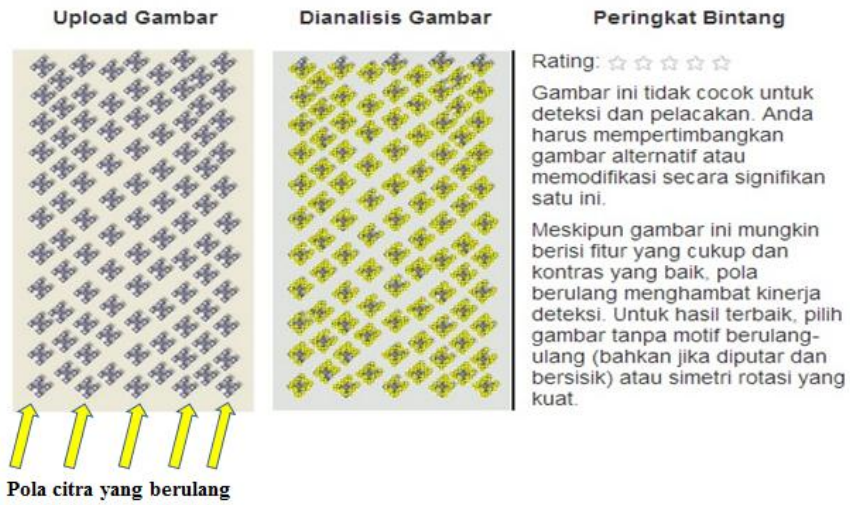
Dalam pendeteksian *fitur* pada citra diatas. Dilakukan perbandingan dengan perbandingan setiap piksel. Apabila nilai intensitas piksel pada posisi *f (x , y)* dari citra ukuran 2x2 dan terdapat perbandingan nilai entitas 1 piksel berbeda dengan piksel lainnya atau maka akan dideteksi sebagai fitur. Sebagai contoh dapat dilihat pada gambar 3.18.



Gambar 3.18 Penentuan *Feature* pada Ukuran Citra 2x2

d) *Repettive Pattern*

Banyaknya pola pada citra yang berulang yang dapat mengakibatkan proses *tracking* menjadi lambat atau bahkan sistem tidak akan dapat mendeteksi citra tersebut. Dikarenakan persamaan posisi *feature*. Berikut contohnya pada gambar 3.19.



Gambar 3.19 klasifikasi *reppetive pattern*

Meskipun beberapa gambar mengandung cukup fitur dan kontras yang baik, pola berulang menghambat kinerja deteksi. Karena akan menyulitkan sistem untuk mendeteksi posisi *feature* pada citra sebagai acuan pendeteksian. Untuk hasil terbaik, pilih gambar tanpa motif berulang-ulang (bahkan jika diputar dan bersisik) atau simetri rotasi yang kuat. Sebuah kotak-kotak adalah contoh dari pola berulang yang tidak dapat terdeteksi.

3. **Menentukan *image* histogram**

Histogram adalah grafik yang menunjukkan frekuensi kemunculan setiap nilai gradasi warna. Bila digambarkan pada koordinat kartesian maka sumbu X (absis) menunjukkan tingkat warna dan sumbu Y (ordinat) menunjukkan frekuensi kemunculan.

Sebuah histogram yang baik sangat mepresentasikan nilai rating yanng baik. Karena itu adapun perhitungan *image* histogram pada marker yang digunakan pada implementasi aplikasi ini.

Tabel 3.6 Histogram pada *Marker* dan Nilai RGB

Citra Asli
<table><tr><td>R = 90 G = 20 B = 40</td><td>R = 110 G = 60 B = 10</td><td>R =90 G = 60 B = 90</td></tr><tr><td>R = 20 G = 30 B = 40</td><td>R = 60 G = 10 B = 80</td><td>R = 30 G = 60 B = 60</td></tr><tr><td>R = 10 G = 20 B = 30</td><td>R = 20 G = 20 B = 50</td><td>R = 70 G = 30 B = 80</td></tr></table> *Ket: misal citra asli pada marker memiliki ukuran 3x3*
Histogram Citra



Perhitungan citra asli pada beruang memiliki ukuran 3x3 dan citra warna 24bit . maka langkah perhitungannya adalah sebagai berikut :

$$f_o(x,y) = \frac{f_i^R\,(x,y) + f_i^G\,(x,y) + f_i^B\,(x,y)}{3}$$

$$f_o(1,1) = \frac{90 + 20 + 40}{3} = 50$$

$$f_o(1,2) = \frac{20 + 30 + 40}{3} = 30$$

Jadi, nilai *citragrasycale* pada koordinat (1,1) ialah 50. Lakukanlah perhitungan yang sama pada koordinat lainnya. Berikut hasil secara keseluruhan.

50	60	80
30	50	50
60	30	60

Sedangkan untuk perhitungan histogram pada citra dari *grayscale* ialah. Misal citra asli diketahui ukuran 3x3 piksel dan N = 9. Mempunyai kedalaman3 bit.

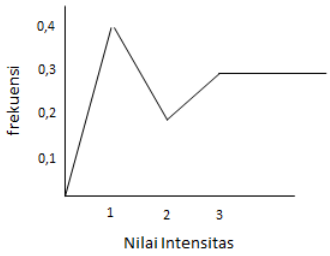
1	3	2
3	1	2
1	1	3

Untuk menggambar kurva histogram dari citra tersebut, ialah buat tabel frekuensi dari kemunculan setiap warna.

Tabel 3.7 Kemunculan warna pada koordinat x dan y

Warna(x)	1	2	3
Jumlah(y)	4	2	3
y/N	0.4	0.2	0,3

Sehingga histogram dalam diagram kartesian pada sumbu x dan y dapat dilihat pada gambar 3.20.



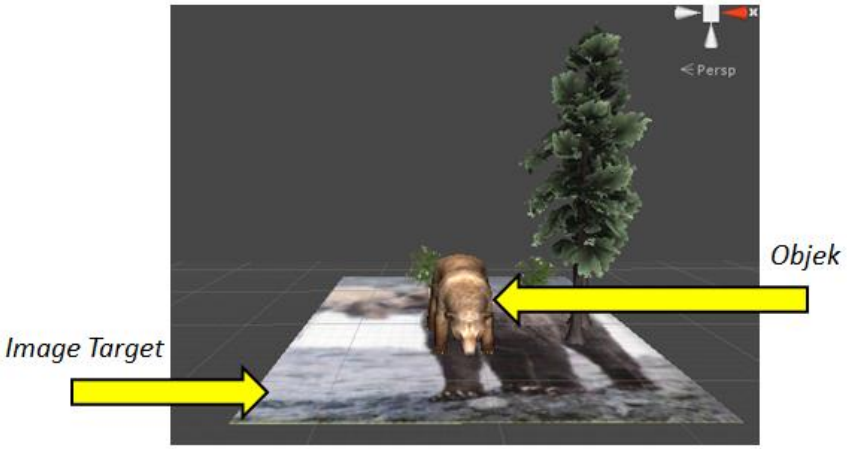
Gambar 3.20 Histogram dalam koordinat kartesian

3.1.4.3 Memunculkan Objek Binatang

Tahap pada saat memunculkan objek pada *ima*ge *target* yang sudah memiliki rating adalah *detection* dan *trackablity*. Tujuan dari tahapan ini ialah pendeteksian dengan *handhpone* dilihat pada aspek jarak maksimum dan minimum dari kamera yang mengarah pada *ima*ge *tar*get dan tracking secara *real time* dengan menggunakan *handphone* dilihat pada aspek kecepatan pada saat kamera digerakkan dengan kecepatan tertentu sehingga diketahui kecepatan maksimal agar *objek* pada *ima*ge *tar*get tetap muncul. Sudut pada saat kamera mengarah pada *ima*ge *tar*get agar diketahui sudut minimum agar object tetap terdeteksi oleh kamera dan yang terakhir ialah *occlusion*.

1. *Detection*

Tahapan dimana sebuah objek akan akan muncul pada *image target* yang sudah diterapkan *feature*. Terdapatpada *class DefaultTrackableEventHandler.* Sebelumnya sudah dijelaskan bagaimana proses pengecekan pada image target dari mulai *set marker, pengecekan image* dan feature. apabila step itu sudah benar. Maka proses selanjutnya memunculkan objek yang sudah di *apply*. Berikut implementasi pada gambar 3.21.



Gambar 3.21 Objek diatas *image target*

```
#region PUBLIC_METHODS
    /// <summary>
    ///Implementationof                          the
ITrackableEventHandler function called when the
    /// tracking state changes.
    /// </summary>
    public void OnTrackableStateChanged(

TrackableBehaviour.Status previousStatus,

TrackableBehaviour.Status newStatus)
    {
        if              (newStatus                ==
TrackableBehaviour.Status.DETECTED ||
            newStatus                             ==
TrackableBehaviour.Status.TRACKED ||
            newStatus                             ==
TrackableBehaviour.Status.EXTENDED_TRACKED)
```

```
        {            OnTrackingFound();         }
        else
        {            OnTrackingLost();        }
    }
    #endregion // PUBLIC_METHODS
```

Apabila *trackable* mendenteksi marker dengan status DETECTED , TRACKED dan EXTED_TRACKED. Selanjutkan akan memanggil fungsi *OnTrackingFound()*.

```
#region PRIVATE_METHODS
    private void OnTrackingFound()
    {
Renderer[]                    rendererComponents=
GetComponentsInChildren<Renderer>(true);
Collider[]                    colliderComponents=
GetComponentsInChildren<Collider>(true);
        // Enable rendering:
        foreach    (Renderer    component    in
rendererComponents)
        {              component.enabled = true;
}

        // Enable colliders:
        foreach    (Collider    component    in
colliderComponents)
        {              component.enabled = true;
}
        Debug.Log("Trackable          "          +
mTrackableBehaviour.TrackableName + " found");    }
```

Fungsi diatas memanggil semua objek yang sudah di *apply* pada agar muncul di atas *image target.* Berikut hasil implementasi pada gambar 3.22.



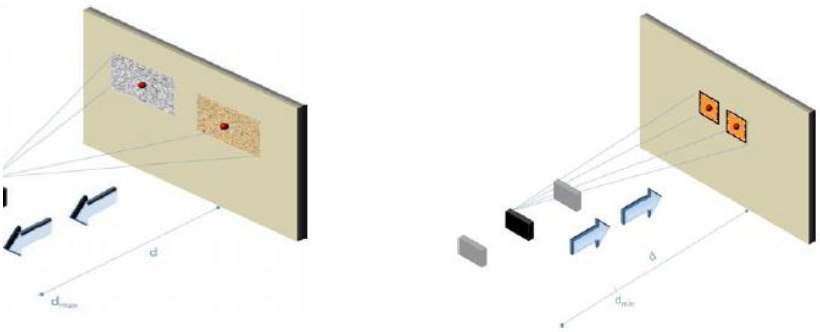
Gambar 3.22 Implementasi Munculnya Objek di atas *Image Target*

2. *Trackability*

Melakukan analisis untuk mengetahui jarak , sudut, kecepatan dan maksimun terhalang nya suatu marker (*Occlusion*) untuk dapat memunculkan objek. Berikut penjelasan berdasarkan kategori.

a) Jarak

Dalam pendetesian dalam segi jarak antara kamera dan marker. vuforia memiliki batasan tersebut. Dikarenakan perangkat keras yang digunakan sangat berpengaruh serta dalam dunia nyata seperti cahaya yang juga sangat mempengaruhi. Berikut jarak maksimum dan minimum pada gambar 3.23.



Gambar 3.23 Ilustrasi Pengujian pada Marker dengan Mengevaluasi Maksimum dan Minimum Jarak

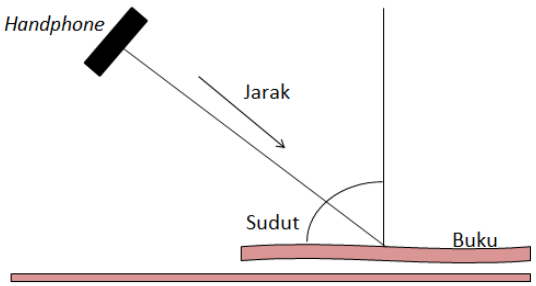
Adapun hasil yang didapat dari jarak maksimum dan minimun pada *handphone.*

Tabel 3.8 Hasil Pengujian Jarak Maksimum dan Minimum

Device	**Maksimum**	**Minimum**
Galaxy Nexus	90cm	10cm

b) Sudut

Sudut maksimum dan minum pada saat kamera mengarah pada objek. Bertujuan agar objek tidak *lost*. Adapun gambaran yang dilakukan berdasaran gambar 3.24 .



Gambar 3.24 ilustrasi pengujian dan perhitungan sudut minimun

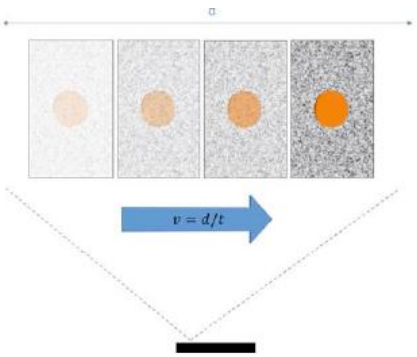
Adapun hasil yang di dapat untuk mendapat minimum sudut agar objek tidak *lost*.

Tabel 3.9 Hasil Pengujian sudut minimal agar objek tidak *lost*

Device	**Minimum Sudut**	**Jarak**
Galaxy Nexus	2’	10cm

c) Kecepatan

Bertujuan untuk melakukan test kecepatan kamera bergerak agar tidak *lost* objek. Berikut gambaran melakukan *test* seberapa maksimum kecepatan kamera bergerak agar objek tetap *tracked.* Berikut ilustrasinya pada gambar 3.25.



Keterangan
v = kecepatan
d = jarak
t = waktu

Gambar 3.25 Ilustrasi Kecepatan Handphone Bergerak

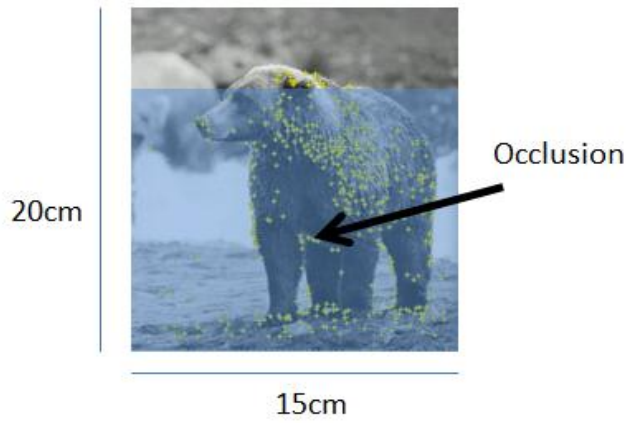
Diketahui jarak yang akan ditempuh kamera adalah 20cm dan kamera bergerak pada ujung jarak dengan waktu 2detik.

v = 20 / 2 = 10cm/sec .

d) *Occlusion*

Adanya benda atau sejenisnya yang terhalang oleh baik dari kamera dan marker yang menyebabkan tidak munculnya objek pada marker tersebut. Namun disini ada batasan maksimum *occlusion* pada marker apabila menggunakan *vuforia*. Adapun rumus untuk menghitung untuk menghitung *occlusion.*

$$\frac{Aocclusion}{Atotal}\ x\ 100 = \%Occlusion \qquad [1]$$

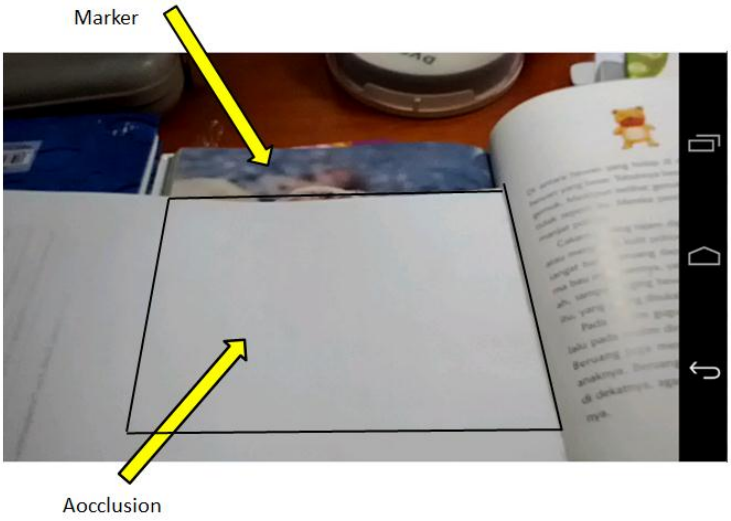


Gambar 3.26 *Occlusion* yang terjadi pada marker

$$\frac{18x15}{20x15} \text{ x } 100 = \%Occlusion$$

$$\frac{270}{300} \text{ x } 100 = 90\%$$

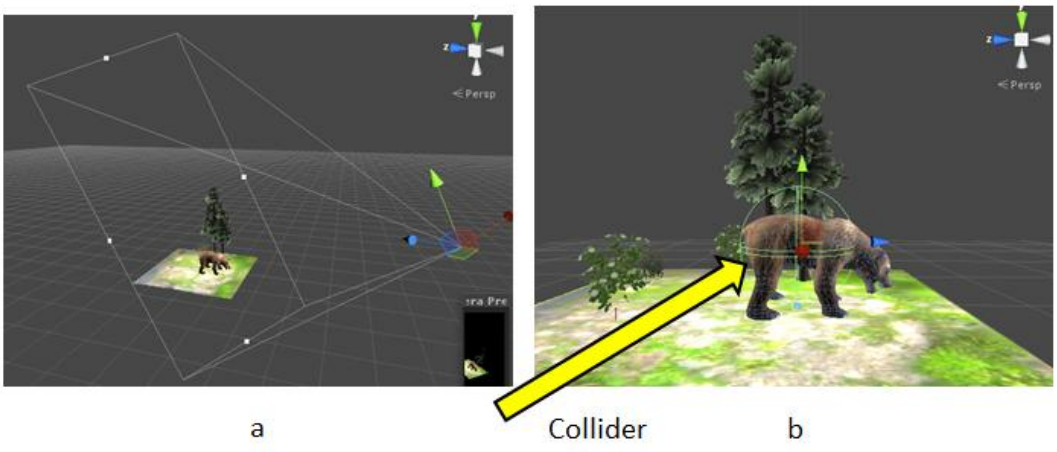
Persentasi yang terjadi akibat *occlusion* berdasarkan benda yang menghalangi marker dengan perumpamaan seperti gambar 3.26 berwarna biru ialah 90% . Berikut implementasi nya pada gambar 3.27.



Gambar 3.27 *Occlusion* pada *marker* sehingga Objek tidak muncul

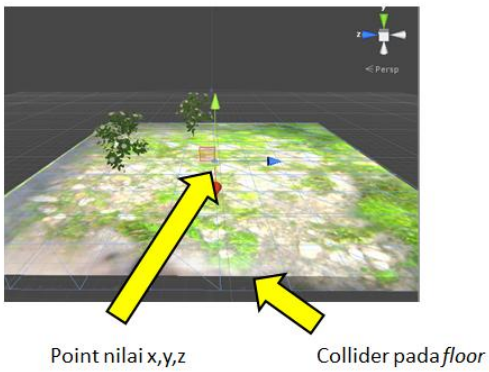
3.1.4.4 Aksi pada Objek Binatang

Merupakan tahap analisis aksi yang diterapkan pada Aplikasi AR *ini*. Sehingga objek dapat bergerak oleh kontrol dari *user*. Mendefinisikan titik awal dan arah kemudian garis imajiner yang diambil dengan parameter. Jika garis di beberapa titik bersinggungan dengan *collider* pada objek , maka akan mendapatkan obyek hit yang memberitahu apa yang dipukul (*touch)* dan nilai vector(x, y dan z) dan panjang R*ay*(*line*).



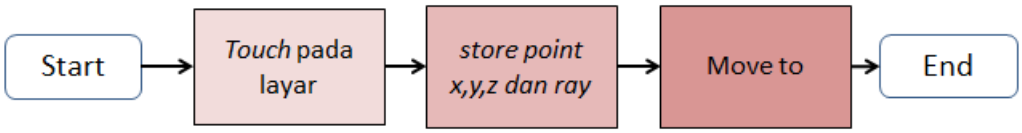
Gambar 3.28 Proyeksi *Ray* pada objek beruang (a) , Collider pada objek (b)

Pada saat *Ray*(line) sudah ditentukan . maka pada saat *touch* pada object yang memiliki *colider* akan memunculkan aksi. Untuk implementasi pada aplikasi ialah agar objek beruang akan *move to* point ke collider pada *floor* yang menyimpan *point* nilai x, y dan z dengan memanfaatkan *function RaycastHit* merupakan turunan dari *function Raycast* pada Unity3d. Gambar 3.29 menunjukkan *colider* dan point nilai x,y dan z.



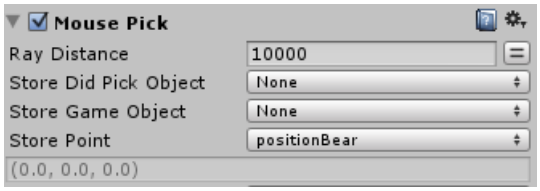
Gambar 3.29 *Collider* pada *floor* dan point nilai x,y dan z

Setelah layar pada *mobile* di *touch* pada floor maka akan menyimpan nilai *point* berupa vector. Berikut alurnya.

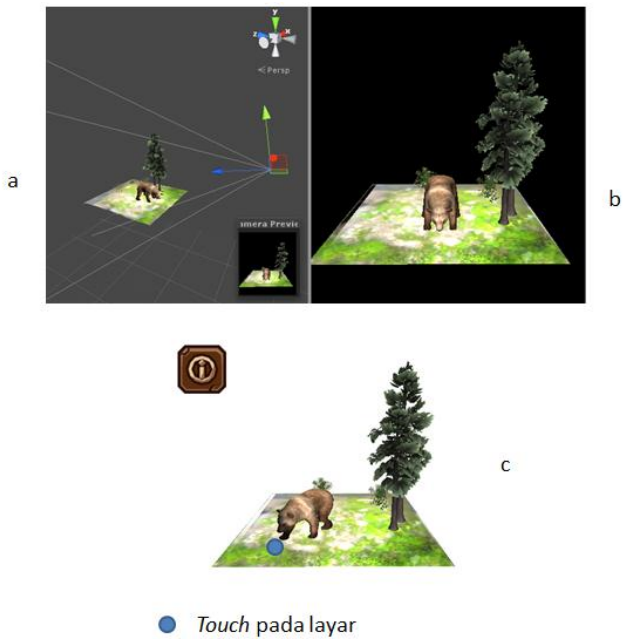


Gambar 3.30 Alur sistem untuk aksi bergerak pada binatang

Tabel 3.10 Deskripsi alur sistem untuk aksi bergerak pada binatang

No	Nama Alur	Keterangan
1	*Touch* pada layar dengan *button.* 	Pendefinisian aksi dengan menekan *touch* pada layar dan mengirimkan sebuah *event finished.*
2	Pendefinisan jarak ray dan store value x,y dan z pada posisi setelah di-*touch* pada layar. 	Inisialisasi *ray distance* dan mendefinisikan posisiBear atau objek pada value x,y dan z.
3		Mendefinisikan objek yang akan bergerak setelah *event touch* pada layar dan mengirinkan *value* nilai posisi x,y dan z.

Touch pada *floor* yang memiliki collider dengan penentuan nilai ray (line) sebagai jangkauan sebuah kamera untuk dapat tersentuh pada layar akan membuat sebuah objek bergerak. Berikut implementasi aksi pada objek binatang dengan *raycasting* pada gambar 3.31.



Gambar 3.31 proyeksi ray distance dengan kamera (a) , ray yang sudah ditentukan pada objek (b) , *move to point* pada saat *touch* pada floor (c)

3.2 Analisis Kebutuhan Non Fungsional

Analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan untuk sistem. Beberapa meliputi komponen perangkat keras apa saja yang dibutuhkan agar sistem dapat di implementasikan dengan baik. Serta menentukan spesifikasi spesifikasi masukan untuk sistem dan kebutuhan non fungsional dibagi menjadi beberapa komponen dimulai dari sasaran *user* , perangkat keras minimun dan perangkat lunak yang dibutuhkan.

Media pembelajaran mengenal dunia binatang dengan teknologi AR diharapkan anak – anak akan lebih tahu dan menggunakan *handphone* sebagai media pembelajaran. Maka dibutuhkan kriteria spesifikasi perangkat keras pada sebuah *handphone* agar sistem dapat digunakan. Berikut adalah perangkat lunak dan perangkat keras yang dibutuhkan untuk implementasi sistem.

3.2.1 Analisis Perangkat Keras

Dibutuhkan beberapa kriteria spesifikasi minimum untuk dapat mengimplementasikan aplikasi ini. Dikarenakan penerapan interaksi dengan *user* yang dimana memiliki nilai komputasi tinggi, dari inisialisasi objek dan berdiri pada perangkat *mobile.* Pada tabel 3.11 merupakan spesifikasi perangkat keras.

Tabel 3.11 Spesifikasi Perangkat Keras *Mobile*

Nama pada Sistem	Keterangan
Sistem Operasi	*Ice Cream Sandwich* (4.0)
CPU	*Dualcore* 1,2 Ghz
Memori	1 *Gigabyte*
GPU	*Adreno* 320, *PowerVR* SGX554MP4
Kamera (*back)*	5 *Mega Pixel*
SD Card	4 *Gigabyte*
Merk *Handphone*	*Opsional* asalkan dapat memenuhi minimal kriteria di atas. Merk apapun pada *handphone* tidak diperhitungkan.

3.2.2 Analisis Perangkat Lunak

Dalam sebuah sistem akan ada yang namanya perangkat lunak sebagai himpunan pendukung yang dibutuhkan untuk membangun sebuah sistem. Dalam membangun aplikasi ini dapat dilihat pada tabel 3.12.

Tabel 3.12 Perangkat Lunak Pembangun

No	Nama Perangkat Lunak	Keterangan
1	Windows 7 Ultimate 64 bit	Sistem operasi yang digunakan dalam membangun aplikasi.
2	Unity3d Versi 4.3.2f	Perangkat lunak utama untuk membangun aplikasi *ini*. dari mulai konten, assets dan

		logic.
3	3DS max 2010	Untuk pembangun model 3D binatang sebagai pendukung mengenalkan objek menyerupai bentuk aslinya.
4	Vuforia SDK	Library AR yang digunakan dalam implementasi teknologi AR untuk membangun aplikasi.
5	Android SDK	Library agar dapat di gunakan untuk perangkat *mobile* khususnya android.

3.2.3 Analisis *User*

Analsis user ini dimaksudkan untuk siapa saja yang dapat menggunakan aplikasi ini. hanya di khususkan di pergunakan hanya oleh *user* yang telah memiliki wawasan dalam penggunaan *handphone* seperti anak-anak asalkan berada di bawah pengawasan orang tua. Hak akses yang di berikan hanya melihat model binatang 3D , animasi dari binatang tersebut, mengenal binatang dengan penjelasan dengan suara dan tulisan dan memberi makan pada binatang sebagai bentuk interaksi yang interaktif. Adapun sasaran *user* dapat dilihat pada tabel 3.13.

Tabel 3.13 Sasaran *User*

No	*User*	Keterangan
1	Anak - Anak	Berumur antara 4 – 7 tahun dibawah pengawasan orang tua dan guru.

3.2.4 Spesifikasi Kebutuhan Non Fungsional

Analisis spesifikasi kebutuhan non-fungsional memiliki tugas untuk menganalisis berkaitan dengan batasan terhadap sistem yang akan dibangun. Hasil dari analisis ini diperoleh kebutuhan non-fungsional dijelaskan pada Tabel 3.14.

Tabel 3.14 Kebutuhan Non Fungsional

Nomor	Spesifikasi Kebutuhan Non Fungsional
SKPL-NF1	Sistem dibangun berbasis *mobile*
SKPL-NF2	Sistem digunakan pada hardware yang memenuhi rekomendasi minimum.

3.3 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dapat didefinsikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi sebagai alat bantu yang digunakan untuk menggambarkan sistem secara umum yang akan dibangun yaitu *Diagram Konteks* dan *Data Flow Diagram* (DFD).

3.3.1 Spesifikasi Kebutuhan Fungsional

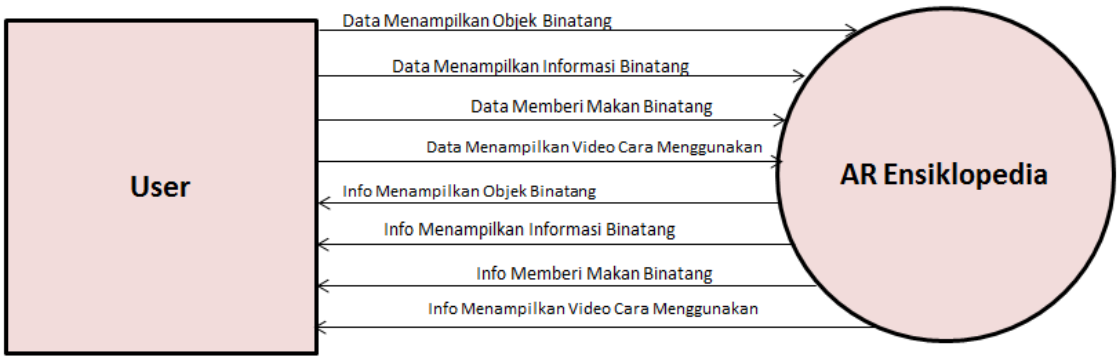
Spesifikasi kebutuhan fungsional memiliki tugas untuk menjelaskan penyediaan layanan pada sistem terhadap masukan tertentu, menjelaskan fungsionalitas sistem, serta menjelaskan secara detail dari layanan yang terdapat pada sistem. Spesifikasi kebutuhan fungsional dapat dilihat pada tabel 3.15.

Tabel 3.15 Kebutuhan Fungsional

Nomor	Spesifikasi Kebutuhan Fungsional
SKPL-F1	Sistem menampilkan objek binatang dari setiap *marker* yang berbeda – beda.
SKPL-F2	Sistem menampilkan informasi dari setiap masing – masing binatang.
SKPL-F3	*User* memberi makan pada binatang dengan mengarahkan objek binatang ke *marker* makanan.
SKPL-F4	*User* melihat video cara penggunaan aplikasi.

3.3.2 *Diagram Konteks*

Diagram konteks adalah suatu diagram yang menggambarkan ruang AR Ensiklopedia tersebut berinteraksi dengan lingkungannya. Sistem digambarkan dengan bulatan, sedangkan lingkungan diwakili oleh entitas luar yang digambarkan dengan persegi. Gambar 3.32 menggambarkan diagram konteks yang terdapat pada aplikasi *ini*.

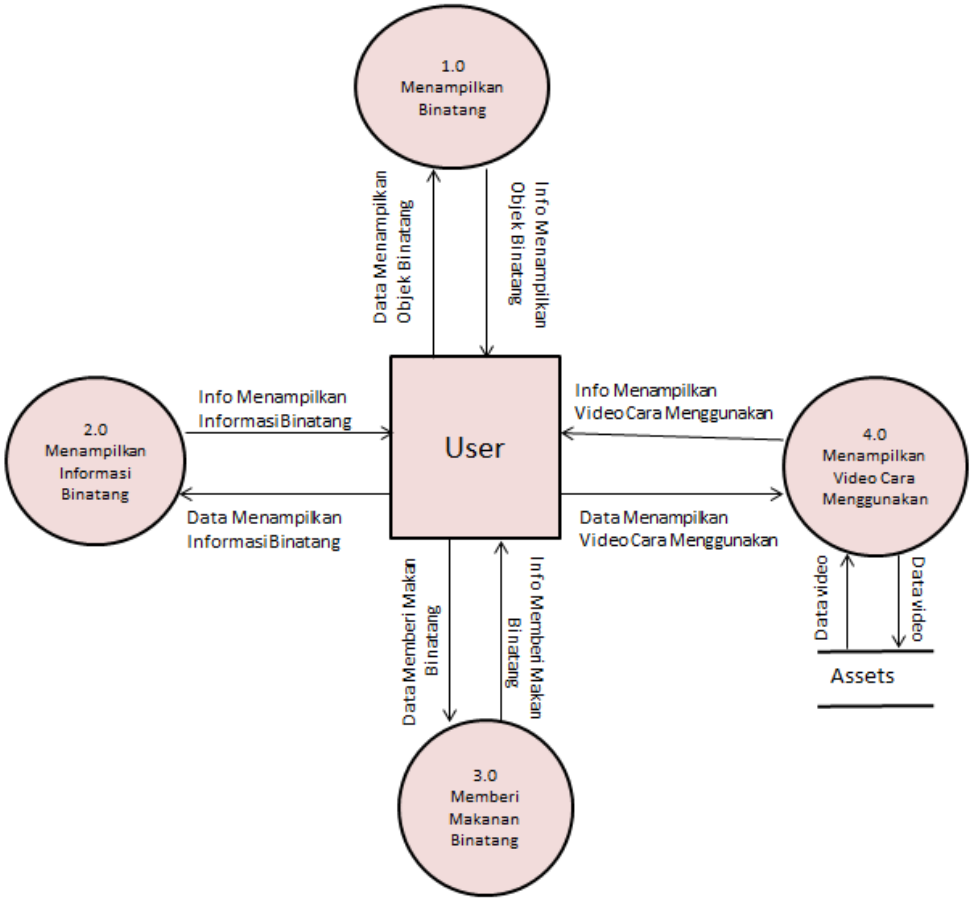


Gambar 3.32 *Diagram* konteks aplikasi

3.3.3 *Data Flow Diagram* (DFD)

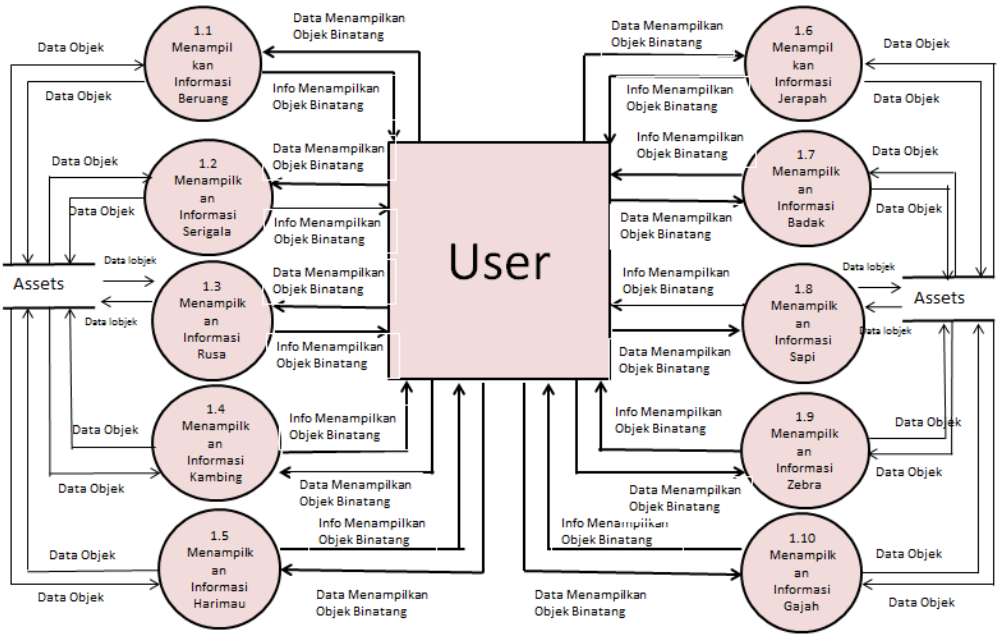
Diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas. Gambar 3.33 menjelaskan alur dari data sistem kepada pengguna.

3.3.3.1 DFD Level 1



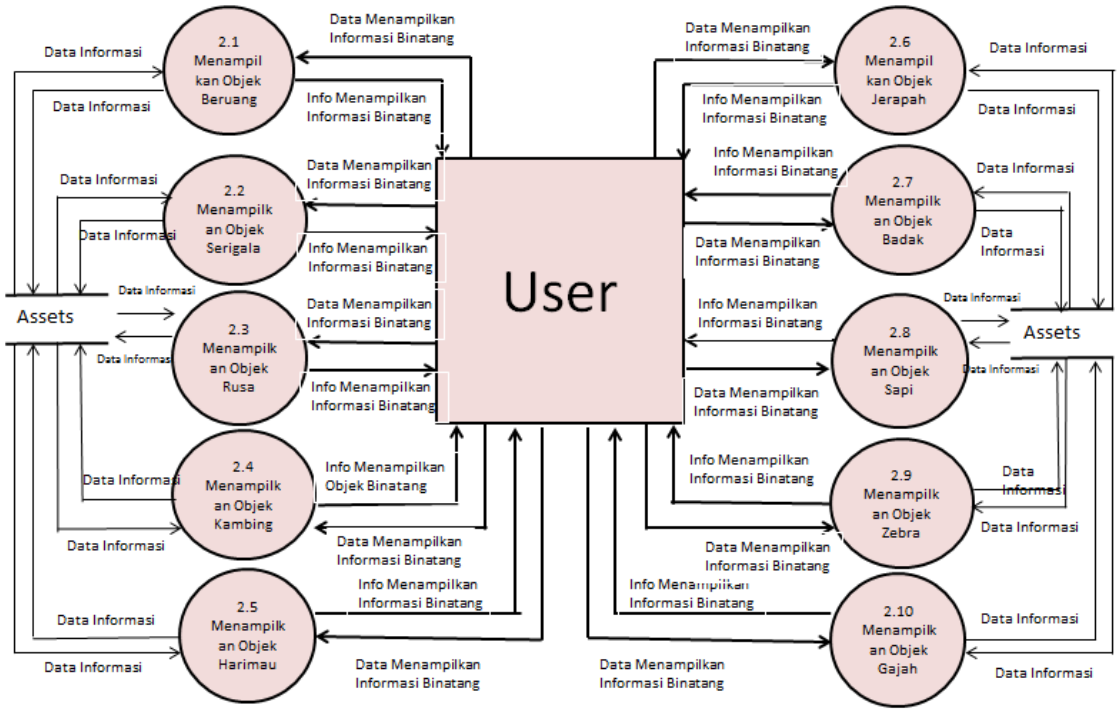
Gambar 3.33 DFD Level 1 Aplikasi

3.3.3.2 DFD Level 2 Proses 1.0 : Proses Menampikan Objek Binatang



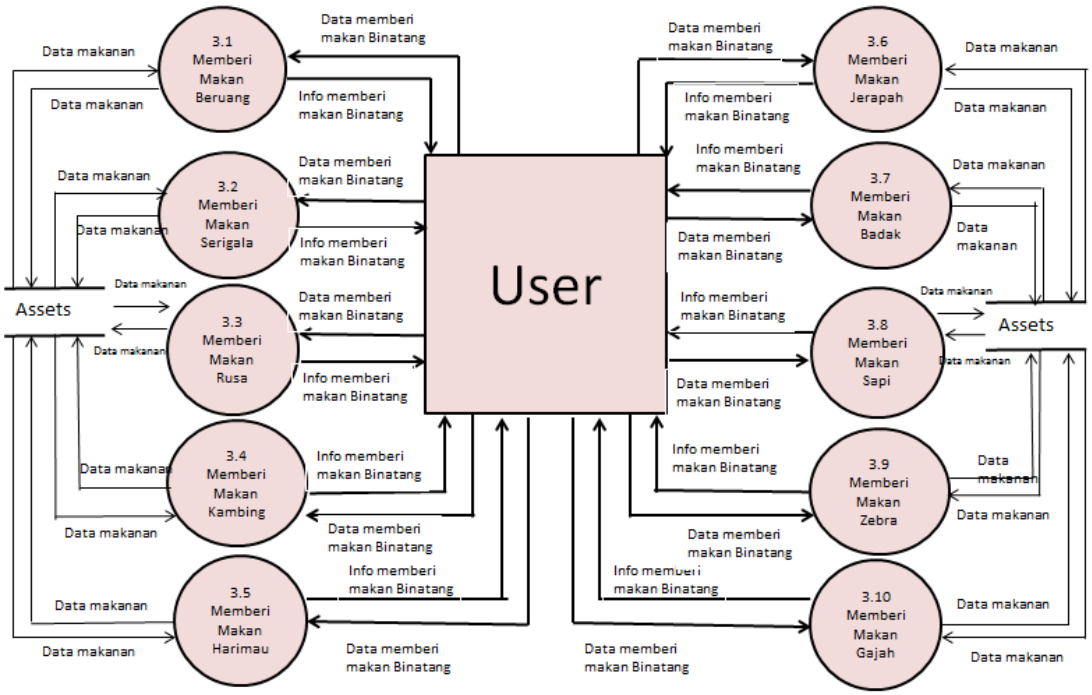
Gambar 3.34 DFD Level 2 Proses 1.0 : Menampilkan Objek Binatang

3.3.3.3 DFD Level 2 Proses 2.0 : Menampilkan Informasi Binatang



Gambar 3.35 DFD Level 2 Proses 2.0 : Menampilkan Informasi Binatang

3.3.3.4 DFD Level 2 Proses 3.0 : Memberi Makan Binatang



Gambar 3.36 DFD Level 2 Proses 3.0 : Memberi Makan Binatang

3.3.4 Spesifikasi Proses

Spesifikasi proses berfungsi untuk menjelaskan proses yang terdapat pada aplikasi *ini*. Berikut spesifikasi proses dapat dilihat pada Tabel 3.16.

Tabel 3.16 Tabel Spesifikasi Proses

No.	Proses	Keterangan
Proses Menampilkan Objek Binatang		
1	No. Proses	1.0
	Nama Proses	Menampilkan objek binatang
	Source	User
	Input	Data menampilkan objek binatang
	Output	Info menampilkan objek binatang
	Destination	User

	Logika Proses	1. *User* menyiapkan buku ensiklopedia *animal* 2. *User* mengarahkan kamera dari *handphone* ke halaman buku ensiklopedia *animal*. 3. sistem mengecek type sasaran *marker* 4. jika predefined maka sasaran adalah *marker based*. 5. Sistem mengecek database berisi kumpulan marker dan objek binatang. 6. Jika *marker* telah terinisialisasi pada database maka muncul objek binatang. 7. Jika *marker* tidak terinisialisasi pada database maka tidak muncul objek binatang.
Proses Menampilkan Informasi Binatang		
2	No. Proses	2.0
	Nama Proses	Menampilkan informasi binatang
	Source	User
	Input	Data menampilkan informasi binatang
	Output	Info menampilkan informasi binatang
	Destination	User
	Logika Proses	1. Objek binatang terdeteksi oleh kamera pada *handphone* 2. Sistem akan memunculkan tombol informasi pada dari setiap binatang yang terdeteksi.
Proses Memberi makan binatang		
3	No. Proses	3.0
	Nama Proses	Memberi makan binatang
	Source	User
	Input	Data memberi makan binatang
	Output	Info memberi makan binatang
	Destination	User
	Logika Proses	1. Objek terdeteksi oleh kamera dari *handphone* 2. Menambahkan *marker* makanan untuk binatang. 3. Jika *marker* makanan terinisialisasi pada

		database maka muncul objek makanan. 4. Jika *marker* makanan tidak terinisialisasi pada database maka tidak muncul objek makanan.
Proses Menampilkan Video Cara Penggunaan		
4	No. Proses	4.0
	Nama Proses	Menampilkan Video Cara Penggunaan
	Source	User
	Input	Data Video Cara Penggunaan
	Output	Info Video Cara Penggunaan
	Destination	User
	Logika Proses	1. Memilih menu "How to Use" 2. Sistem menampilkan video cara penggunaam aplikasi 3. Jika *user* menekan tombol *pause* maka video *pause.* 4. Jika *user* menekan tombol *play* maka video *play.*
Proses Menampilkan Objek Beruang		
5	No. Proses	1.1
	Nama Proses	Menampilkan Objek Beruang
	Source	User
	Input	Data menampilkan objek binatang
	Output	Info menampilkan objek binatang
	Destination	User
	Logika Proses	1. Kumpulan *marker* telah terinisialisasi pada *database* 2. sistem mengecek objek yang sudah di-*apply* diatas *image target.* 3. jika *marker*-nya halaman beruang maka muncul objek dan suara beruang.
Proses Menampilkan Objek Serigala		
6	No. Proses	1.2
	Nama Proses	Menampilkan Objek Serigala
	Source	User

	Input	Data menampilkan objek binatang
	Output	Info menampilkan objek binatang
	Destination	User
	Logika Proses	1. kumpulan *marker* telah terinisialisasi pada *database* 2. sistem mengecek objek yang sudah di-*apply* diatas *image target.* 3. Jika *marker*-nya halaman serigala maka muncul objek dan suara serigala
Proses Menampilkan Objek Rusa		
7	No. Proses	1.3
	Nama Proses	Menampilkan Objek Rusa
	Source	User
	Input	Data menampilkan objek binatang
	Output	Info menampilkan objek binatang
	Destination	User
	Logika Proses	1. kumpulan *marker* telah terinisialisasi pada *database* 2. sistem mengecek objek yang sudah di-*apply* diatas *image target.* 3. Jika *marker*-nya halaman rusa maka muncul objek dan suara rusa
Proses Menampilkan Objek Kambing		
8	No. Proses	1.4
	Nama Proses	Menampilkan Objek Kambing
	Source	User
	Input	Data menampilkan objek binatang
	Output	Info menampilkan objek binatang
	Destination	User
	Logika Proses	1. kumpulan *marker* telah terinisialisasi pada *database* 2. sistem mengecek objek yang sudah di-*apply* diatas *image target.* 3. Jika *marker*-nya halaman Kambing maka muncul objek dan suara Kambing

Proses Menampilkan Objek Harimau		
9	No. Proses	1.5
	Nama Proses	Menampilkan Objek Harimau
	Source	User
	Input	Data menampilkan objek binatang
	Output	Info menampilkan objek binatang
	Destination	User
	Logika Proses	1. kumpulan *marker* telah terinisialisasi pada *database* 2. sistem mengecek objek yang sudah di-*apply* diatas *image target.* 3. Jika *marker*-nya halaman Harimau maka muncul objek dan suara Harimau
Proses Menampilkan Objek Jerapah		
10	No. Proses	1.6
	Nama Proses	Menampilkan Objek Jerapah
	Source	User
	Input	Data menampilkan objek binatang
	Output	Info menampilkan objek binatang
	Destination	User
	Logika Proses	1. kumpulan *marker* telah terinisialisasi pada *database* 2. sistem mengecek objek yang sudah di-*apply* diatas *image target.* 3. Jika *marker*-nya halaman Jerapah maka muncul objek dan suara Jerapah
Proses Menampilkan Objek Badak		
11	No. Proses	1.7
	Nama Proses	Menampilkan Objek Badak
	Source	User
	Input	Data menampilkan objek binatang
	Output	Info menampilkan objek binatang
	Destination	User
	Logika Proses	1. kumpulan *marker* telah terinisialisasi pada *database* 2. sistem mengecek objek yang sudah di-

		apply diatas *image target.* 3. Jika *marker*-nya halaman Badak maka muncul objek dan suara Badak
Proses Menampilkan Objek Sapi		
12	No. Proses	1.8
	Nama Proses	Menampilkan Objek Sapi
	Source	User
	Input	Data menampilkan objek binatang
	Output	Info menampilkan objek binatang
	Destination	User
	Logika Proses	1. kumpulan *marker* telah terinisialisasi pada *database* 2. sistem mengecek objek yang sudah di-*apply* diatas *image target.* 3. Jika *marker*-nya halaman Sapi maka muncul objek dan suara Sapi
Proses Menampilkan Objek Zebra		
13	No. Proses	1.9
	Nama Proses	Menampilkan Objek Zebra
	Source	User
	Input	Data menampilkan objek binatang
	Output	Info menampilkan objek binatang
	Destination	User
	Logika Proses	1. kumpulan *marker* telah terinisialisasi pada *database* 2. sistem mengecek objek yang sudah di-*apply* diatas *image target.* 3. Jika *marker*-nya halaman Zebra maka muncul objek dan suara Zebra
Proses Menampilkan Objek Gajah		
14	No. Proses	1.10
	Nama Proses	Menampilkan Objek Gajah
	Source	User
	Input	Data menampilkan Objek Serigala
	Output	Info menampilkan Objek Serigala
	Destination	User

	Logika Proses	1. kumpulan *marker* telah terinisialisasi pada *database* 2. sistem mengecek objek yang sudah di-*apply* diatas *image target.* 3. Jika *marker*-nya halaman Gajah maka muncul objek dan suara Gajah
Proses Menampilkan Informasi Beruang		
15	No. Proses	2.1
	Nama Proses	Menampilkan informasi beruang
	Source	User
	Input	Data menampilkan informasi binatang
	Output	Info menampilkan informasi binatang
	Destination	User
	Logika Proses	1. Objek beruang terdeteksi setelah kamera diarahkan pada *marker* 2. Sistem akan memunculkan tombol *informasi* pada layar. 3. Jika *User* menekan tombol informasi pada layar. Maka sistem menampilkan informasi beruang dengan text dan suara 4. Jika *User* tidak menekan tombol informasi pada layar. Maka sistem tidak menampilkan informasi beruang.
Proses Menampilkan Informasi Serigala		
16	No. Proses	2.2
	Nama Proses	Menampilkan informasi Serigala
	Source	User
	Input	Data menampilkan informasi binatang
	Output	Info menampilkan informasi binatang
	Destination	User
	Logika Proses	1. Objek beruang terdeteksi setelah kamera diarahkan pada *marker* 2. Sistem akan memunculkan tombol *informasi* pada layar. 3. Jika *User* menekan tombol informasi pada layar. Maka sistem menampilkan informasi Serigala dengan text dan suara 4. Jika *User* tidak menekan tombol informasi pada layar. Maka sistem tidak

		menampilkan informasi Serigala.
Proses Menampilkan Informasi Rusa		
17	No. Proses	2.3
	Nama Proses	Menampilkan informasi Rusa
	Source	User
	Input	Data menampilkan informasi binatang
	Output	Info menampilkan informasi binatang
	Destination	User
	Logika Proses	1. Objek beruang terdeteksi setelah kamera diarahkan pada *marker* 2. Sistem akan memunculkan tombol *informasi* pada layar. 3. Jika *User* menekan tombol informasi pada layar. Maka sistem menampilkan informasi Rusa dengan text dan suara 4. Jika *User* tidak menekan tombol informasi pada layar. Maka sistem tidak menampilkan informasi Rusa.
Proses Menampilkan Informasi Kambing		
18	No. Proses	2.4
	Nama Proses	Menampilkan informasi Kambing
	Source	User
	Input	Data menampilkan informasi binatang
	Output	Info menampilkan informasi binatang
	Destination	User
	Logika Proses	1. Objek beruang terdeteksi setelah kamera diarahkan pada *marker* 2. Sistem akan memunculkan tombol *informasi* pada layar. 3. Jika *User* menekan tombol informasi pada layar. Maka sistem menampilkan informasi Kambing dengan text dan suara 4. Jika *User* tidak menekan tombol informasi pada layar. Maka sistem tidak menampilkan informasi Kambing.
Proses Menampilkan Informasi Harimau		
19	No. Proses	2.5

	Nama Proses	Menampilkan informasi Harimau
	Source	User
	Input	Data menampilkan informasi binatang
	Output	Info menampilkan informasi binatang
	Destination	User
	Logika Proses	1. Objek beruang terdeteksi setelah kamera diarahkan pada *marker* 2. Sistem akan memunculkan tombol *informasi* pada layar. 3. Jika *User* menekan tombol informasi pada layar. Maka sistem menampilkan informasi Harimau dengan text dan suara 4. Jika *User* tidak menekan tombol informasi pada layar. Maka sistem tidak menampilkan informasi Harimau.
Proses Menampilkan Informasi Jerapah		
20	No. Proses	2.6
	Nama Proses	Menampilkan informasi Jerapah
	Source	User
	Input	Data menampilkan informasi binatang
	Output	Info menampilkan informasi binatang
	Destination	User
	Logika Proses	1. Objek beruang terdeteksi setelah kamera diarahkan pada *marker* 2. Sistem akan memunculkan tombol *informasi* pada layar. 3. Jika *User* menekan tombol informasi pada layar. Maka sistem menampilkan informasi Jerapah dengan text dan suara 4. Jika *User* tidak menekan tombol informasi pada layar. Maka sistem tidak menampilkan informasi Jerapah.
Proses Menampilkan Informasi Badak		
21	No. Proses	2.7
	Nama Proses	Menampilkan informasi Badak
	Source	User
	Input	Data menampilkan informasi binatang

	Output	Info menampilkan informasi binatang
	Destination	User
	Logika Proses	1. Objek beruang terdeteksi setelah kamera diarahkan pada *marker* 2. Sistem akan memunculkan tombol *informasi* pada layar. 3. Jika *User* menekan tombol informasi pada layar. Maka sistem menampilkan informasi Badak dengan text dan suara 4. Jika *User* tidak menekan tombol informasi pada layar. Maka sistem tidak menampilkan informasi Badak.
Proses Menampilkan Informasi Sapi		
22	No. Proses	2.8
	Nama Proses	Menampilkan informasi Sapi
	Source	User
	Input	Data menampilkan informasi binatang
	Output	Info menampilkan informasi binatang
	Destination	User
	Logika Proses	1. Objek beruang terdeteksi setelah kamera diarahkan pada *marker* 2. Sistem akan memunculkan tombol *informasi* pada layar. 3. Jika *User* menekan tombol informasi pada layar. Maka sistem menampilkan informasi Sapi dengan text dan suara 4. Jika *User* tidak menekan tombol informasi pada layar. Maka sistem tidak menampilkan informasi Sapi.
Proses Menampilkan Informasi Zebra		
23	No. Proses	2.9
	Nama Proses	Menampilkan informasi Zebra
	Source	User
	Input	Data menampilkan informasi binatang
	Output	Info menampilkan informasi binatang
	Destination	User
	Logika Proses	1. Objek beruang terdeteksi setelah kamera diarahkan pada *marker*

		2. Sistem akan memunculkan tombol *informasi* pada layar. 3. Jika *User* menekan tombol informasi pada layar. Maka sistem menampilkan informasi Zebra dengan text dan suara 4. Jika *User* tidak menekan tombol informasi pada layar. Maka sistem tidak menampilkan informasi Zebra.
Proses Menampilkan Informasi Gajah		
24	No. Proses	2.10
	Nama Proses	Menampilkan informasi Gajah
	Source	User
	Input	Data menampilkan informasi binatang
	Output	Info menampilkan informasi binatang
	Destination	User
	Logika Proses	1. Objek beruang terdeteksi setelah kamera diarahkan pada *marker* 2. Sistem akan memunculkan tombol *informasi* pada layar. 3. Jika *User* menekan tombol informasi pada layar. Maka sistem menampilkan informasi Gajah dengan text dan suara 4. Jika *User* tidak menekan tombol informasi pada layar. Maka sistem tidak menampilkan informasi Gajah.
Proses Memberi Makan Beruang		
25	No. Proses	3.1
	Nama Proses	Memberi makan Beruang
	Source	User
	Input	Data memberi makan binatang
	Output	Info memberi makan binatang
	Destination	User
	Logika Proses	1. Objek beruang dan objek makanan terdeteksi oleh *kamera* 2. *User* mengontrol rusa ke objek makanan. 3. Jika makanan Beruang ialah ikan, maka Beruang makan. 4. Jika makanan Beruang adalah rumput, maka Beruang tidak makan.

		5. Jika makanan Beruang adalah daging, maka Beruang tidak makan.
Proses Memberi Makan Serigala		
26	No. Proses	3.2
	Nama Proses	Memberi makan Serigala
	Source	User
	Input	Data memberi makan binatang
	Output	Info memberi makan binatang
	Destination	User
	Logika Proses	1. Objek Serigala dan objek makanan terdeteksi oleh *kamera* 2. *User* mengontrol rusa ke objek makanan. 3. Jika makanan Serigala ialah ikan, maka Serigala makan. 4. Jika makanan Serigala adalah rumput, Serigala rusa tidak makan. 5. Jika makanan Serigala adalah daging, maka Serigala makan.
Proses Memberi Makan Rusa		
27	No. Proses	3.3
	Nama Proses	Memberi makan Rusa
	Source	User
	Input	Data memberi makan binatang
	Output	Info memberi makan binatang
	Destination	User
	Logika Proses	1. Objek Rusa dan objek makanan terdeteksi oleh *kamera* 2. *User* mengontrol rusa ke objek makanan. 3. Jika makanan Rusa ialah rumput, maka Rusa makan. 4. Jika makanan Rusa adalah ikan, maka Rusa tidak makan. 5. Jika makanan Rusa adalah daging, maka Rusa tidak makan.

Proses Memberi Makan Kambing		
28	No. Proses	3.4
	Nama Proses	Memberi makan Kambing
	Source	User
	Input	Data memberi makan binatang
	Output	Info memberi makan binatang
	Destination	User
	Logika Proses	1. Objek Kambing dan objek makanan terdeteksi oleh *kamera* 2. *User* mengontrol Kambing ke objek makanan. 3. Jika makanan Kambing ialah rumput, maka Kambing makan. 4. Jika makanan Kambing adalah ikan, maka Kambing tidak makan. 5. Jika makanan Kambing adalah daging, maka Kambing tidak makan.
Proses Memberi Makan Harimau		
29	No. Proses	3.5
	Nama Proses	Menampilkan informasi Harimau
	Source	User
	Input	Data memberi makan binatang
	Output	Info memberi makan binatang
	Destination	User
	Logika Proses	1. Objek Harimau dan objek makanan terdeteksi oleh *kamera* 2. *User* mengontrol Harimau ke objek makanan. 3. Jika makanan Harimau ialah daging, maka Harimau makan. 4. Jika makanan Harimau adalah rumput, maka Harimau tidak makan. 5. Jika makanan Harimau adalah ikan, maka harimau tidak makan.

Proses Memberi Makan Jerapah		
30	No. Proses	3.6
	Nama Proses	Memberi makan Jerapah
	Source	User
	Input	Data memberi makan binatang
	Output	Info memberi makan binatang
	Destination	User
	Logika Proses	1. Objek beruang dan objek makanan terdeteksi oleh *kamera* 2. *User* mengontrol rusa ke objek makanan. 3. Jika makanan Jerapah ialah rumput, maka Jerapah makan. 4. Jika makanan Jerapah adalah ikan, maka Jerapah tidak makan. 5. Jika makanan Jerapah adalah daging, maka Jerapah tidak makan.
Proses Memberi Makan Badak		
31	No. Proses	3.7
	Nama Proses	Memberi makan Badak
	Source	User
	Input	Data memberi makan binatang
	Output	Info memberi makan binatang
	Destination	User
	Logika Proses	1. Objek Badak dan objek makanan terdeteksi oleh *kamera* 2. *User* mengontrol rusa ke objek makanan. 3. Jika makanan Badak ialah rumput, maka Badak makan. 4. Jika makanan Badak adalah ikan, maka Badak tidak makan. 5. Jika makanan Badak adalah daging, maka Badak tidak makan.
Proses Memberi Makan Sapi		
32	No. Proses	3.8
	Nama Proses	Memberi makan Sapi

	Source	User
	Input	Data memberi makan binatang
	Output	Info memberi makan binatang
	Destination	User
	Logika Proses	1. Objek Sapi dan objek makanan terdeteksi oleh *kamera* 2. *User* mengontrol rusa ke objek makanan. 3. Jika makanan Sapi ialah rumput, maka Sapi makan. 4. Jika makanan Sapi adalah ikan, maka Sapi tidak makan. 5. Jika makanan Sapi adalah daging, maka Sapi tidak makan.
Proses Memberi Makan Zebra		
33	No. Proses	3.9
	Nama Proses	Memberi makan Zebra
	Source	User
	Input	Data memberi makan binatang
	Output	Info memberi makan binatang
	Destination	User
	Logika Proses	1. Objek Zebra dan objek makanan terdeteksi oleh *kamera* 2. *User* mengontrol Zebra ke objek makanan. 3. Jika makanan Zebra ialah rumput, maka Zebra makan. 4. Jika makanan Zebra adalah ikan, maka Zebra tidak makan. 5. Jika makanan Zebra adalah daging, maka Zebra tidak makan.
Proses Memberi Makan Gajah		
34	No. Proses	3.10
	Nama Proses	Memberi makan Gajah
	Source	User

	Input	Data memberi makan binatang
	Output	Info memberi makan binatang
	Destination	User
	Logika Proses	1. Objek Gajah dan objek makanan terdeteksi oleh *kamera* 2. *User* mengontrol Gajah ke objek makanan. 3. Jika makanan Gajah ialah rumput, maka Gajah makan. 4. Jika makanan Gajah adalah ikan, maka Gajah tidak makan. 5. Jika makanan Gajah adalah daging, maka Gajah tidak makan.

3.3.5 Kamus Data

Kamus data merupakan sebuah daftar yang tersusun dari elemen data yang berhubungan dengan sistem. Kamus data dibuat berdasarkan arus data yang ada di data flow diagram dapat dilihat pada tabel 3.17.

Tabel 3.17 Kamus Data

No	Proses	Keterangan
1	Nama	Data menampilkan objek binatang.
	Where used/how used	*User* Proses 1.0 , Proses 1.1, Proses 1.2 ,Proses 1.3,Proses 1.4,Proses 1.5,Proses 1.6,Proses 1.7,Proses 1.8,Proses 1.9,Proses 1.10
	Deskripsi	Data yang digunakan ketika menampilkan objek binatang
	Struktur data	- Halaman buku pada buku ensiklopedia sebagai *marker*
	Penjelasan struktur data	- *Marker {* gambar beruang | gambar serigala | gambar rusa | gambar kambing | gambar harimau | gambar jerapah | gambar badak | gambar sapi | gambar zebra | gambar gajah }

2	Nama	Data menampilkan informasi binatang
	Where used/how used	*User* Proses 2.0 , Proses 2.1, Proses 2.2 ,Proses 2.3,Proses 2.4,Proses 2.5,Proses 2.6,Proses 2.7,Proses 2.8,Proses 2.9,Proses 2.10
	Deskripsi	Data yang digunakan ketika menampilkan informasi binatang
	Struktur data	- Halaman buku pada buku ensiklopedia sebagai *marker*
	Penjelasan struktur data	- *Marker {* gambar beruang | gambar serigala | gambar rusa | gambar kambing | gambar harimau | gambar jerapah | gambar badak | gambar sapi | gambar zebra | gambar gajah }
3	Nama	Data memberi makan binatang
	Where used/how used	*User* Proses 3.0 , Proses 3.1, Proses 3.2 ,Proses 3.3,Proses 3.4,Proses 3.5,Proses 3.6,Proses 3.7,Proses 3.8,Proses 3.9,Proses 3.10
	Deskripsi	Data yang digunakan ketika memberi makan binatang
	Struktur data	- Halaman buku pada buku ensiklopedia sebagai *marker , marker* makanan
	Penjelasan struktur data	- *Marker {* gambar beruang | gambar serigala | gambar rusa | gambar kambing | gambar harimau | gambar jerapah | gambar badak | gambar sapi | gambar zebra | gambar gajah gambar makanan}
4	Nama	Data Menampilkan Video Cara Menggunakan
	Where used/how used	*User* – Proses 4.0
	Deskripsi	Data yang digunakan ketika menampilkan cara penggunaan
	Struktur data	- *Video*
	Penjelasan struktur data	- *Video* Cara Penggunaan
5	Nama	Data Menampilkan Objek Beruang
	Where used/how used	*User* - Proses 1.1

	Deskripsi	Data yang digunakan ketika menampilkan Objek Beruang
	Struktur data	- *Marker*
	Penjelasan struktur data	- *marker {* gambar beruang }
6	Nama	Data Menampilkan Objek Serigala
	Where used/how used	*User* - Proses 1.2
	Deskripsi	Data yang digunakan ketika menampilkan Objek 3D Serigala
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar serigala}
7	Nama	Data menampilkan Objek Rusa
	Where used/how used	*User* - Proses 1.3
	Deskripsi	Data yang digunakan ketika menampilkan Objek 3D Rusa
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar rusa}
8	Nama	Data menampilkan Objek Kambing
	Where used/how used	*User* - Proses 1.4
	Deskripsi	Data yang digunakan ketika menampilkan Objek 3D Kambing
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Kambing }
9	Nama	Data menampilkan Objek Harimau
	Where used/how used	*User* - Proses 1.5
	Deskripsi	Data yang digunakan ketika menampilkan Objek 3D Harimau

	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Harimau }
10	Nama	Data menampilkan Objek Jerapah
	Where used/how used	*User* - Proses 1.6
	Deskripsi	Data yang digunakan ketika menampilkan Objek 3D Jerapah
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Jerapah }
11	Nama	Data menampilkan Objek Badak
	Where used/how used	*User* - Proses 1.7
	Deskripsi	Data yang digunakan ketika menampilkan Objek 3D Badak
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Badak }
12	Nama	Data menampilkan Objek Sapi
	Where used/how used	*User* - Proses 1.8
	Deskripsi	Data yang digunakan ketika menampilkan Objek 3D Sapi
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Sapi}
13	Nama	Data menampilkan Objek Zebra
	Where used/how used	*User* - Proses 1.9
	Deskripsi	Data yang digunakan ketika menampilkan Objek 3D Zebra
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Rusa | gambar makanan}
14	Nama	Data menampilkan Objek Gajah

	Where used/how used	*User* - Proses 1.10
	Deskripsi	Data yang digunakan ketika menampilkan Objek 3D Gajah
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Gajah }
15	Nama	Data menampilkan informasi Beruang
	Where used/how used	*User* - Proses 2.1
	Deskripsi	Data yang digunakan ketika menampilkan informasi Beruang
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Beruang} - *mouseEvent* {*TouchButtonDown}*
16	Nama	Data menampilkan informasi Serigala
	Where used/how used	*User* - Proses 2.2
	Deskripsi	Data yang digunakan ketika menampilkan informasi Serigala
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Serigala} - *mouseEvent* {*TouchButtonDown}*
17	Nama	Data menampilkan informasi Rusa
	Where used/how used	*User* - Proses 2.3
	Deskripsi	Data yang digunakan ketika menampilkan informasi Rusa
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Rusa} - *mouseEvent* {*TouchButtonDown}*
18	Nama	Data menampilkan informasi Kambing
	Where used/how used	*User* - Proses 2.4
	Deskripsi	Data yang digunakan ketika menampilkan informasi Kambing
	Struktur data	- *marker*

	Penjelasan struktur data	- *marker* {gambar Kambing} - *mouseEvent* {*TouchButtonDown}*
19	Nama	Data menampilkan informasi Harimau
	Where used/how used	*User* - Proses 2.5
	Deskripsi	Data yang digunakan ketika menampilkan informasi Harimau
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Harimau } - *mouseEvent* {*TouchButtonDown}*
20	Nama	Data menampilkan informasi Jerapah
	Where used/how used	*User* - Proses 2.6
	Deskripsi	Data yang digunakan ketika menampilkan informasi Jerapah
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Jerapah} - *mouseEvent* {*TouchButtonDown}*
21	Nama	Data menampilkan informasi Badak
	Where used/how used	*User* - Proses 2.7
	Deskripsi	Data yang digunakan ketika menampilkan informasi Badak
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Badak } - *mouseEvent* {*TouchButtonDown}*
22	Nama	Data menampilkan informasi Sapi
	Where used/how used	*User* - Proses 2.8
	Deskripsi	Data yang digunakan ketika menampilkan informasi Sapi
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Sapi } - *mouseEvent* {*TouchButtonDown}*
23	Nama	Data menampilkan informasi Zebra
	Where used/how used	*User* - Proses 2.9

	Deskripsi	Data yang digunakan ketika menampilkan informasi Zebra
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Zebra} - *mouseEvent* {*TouchButtonDown}*
24	Nama	Data menampilkan informasi Gajah
	Where used/how used	*User* - Proses 2.10
	Deskripsi	Data yang digunakan ketika menampilkan informasi Gajah
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Gajah} - *mouseEvent* {*TouchButtonDown}*
25	Nama	Data memberi makan Beruang
	Where used/how used	*User* - Proses 3.1
	Deskripsi	Data yang digunakan ketika memberi makan Beruang
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Beruang | gambar makanan} - *mouseEvent* {*TouchButtonDown}*
26	Nama	Data memberi makan Serigala
	Where used/how used	*User* - Proses 3.2
	Deskripsi	Data yang digunakan ketika memberi makan Serigala
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Serigala | gambar makanan} - *mouseEvent* {*TouchButtonDown}*
27	Nama	Data memberi makan Rusa
	Where used/how used	*User* - Proses 3.3
	Deskripsi	Data yang digunakan ketika memberi makan Rusa

	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Rusa | gambar makanan} - *mouseEvent* {*TouchButtonDown}*
28	Nama	Data memberi makan Kambing
	Where used/how used	*User* - Proses 3.4
	Deskripsi	Data yang digunakan ketika memberi makan Kambing
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Kambing | gambar makanan} - *mouseEvent* {*TouchButtonDown}*
29	Nama	Data memberi makan Harimau
	Where used/how used	*User* - Proses 3.4
	Deskripsi	Data yang digunakan ketika memberi makan Harimau
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Harimau | gambar makanan} - *mouseEvent* {*TouchButtonDown}*
30	Nama	Data memberi makan Jerapah
	Where used/how used	*User* - Proses 3.5
	Deskripsi	Data yang digunakan ketika memberi makan Jerapah
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Jerapah | gambar makanan} - *mouseEvent* {*TouchButtonDown}*
31	Nama	Data memberi makan Badak
	Where used/how used	*User* - Proses 3.6
	Deskripsi	Data yang digunakan ketika memberi makan Badak
	Struktur data	- *marker*

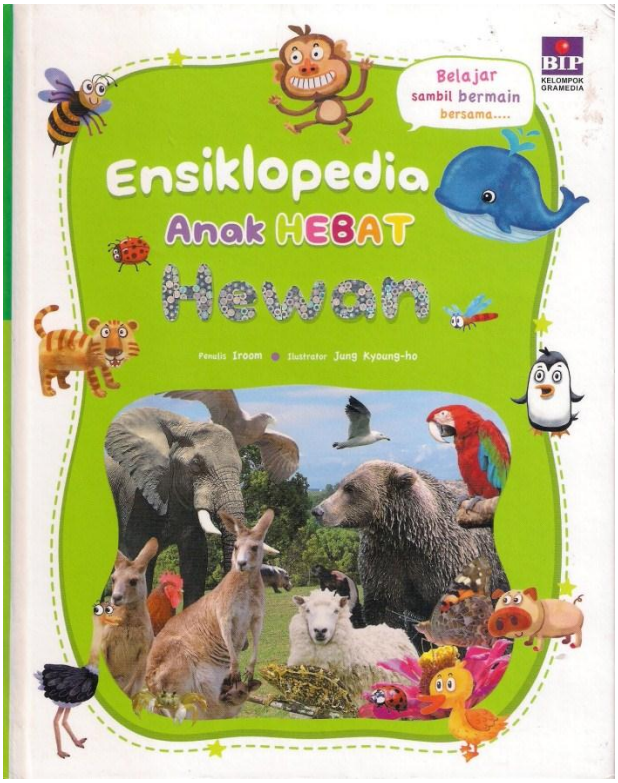
	Penjelasan struktur data	- *marker* {gambar Badak | gambar makanan} - *mouseEvent* {*TouchButtonDown}*
32	Nama	Data memberi makan Sapi
	Where used/how used	*User* - Proses 3.7
	Deskripsi	Data yang digunakan ketika memberi makan Sapi
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Sapi | gambar makanan} - *mouseEvent* {*TouchButtonDown}*
33	Nama	Data memberi makan Zebra
	Where used/how used	*User* - Proses 3.9
	Deskripsi	Data yang digunakan ketika memberi makan Zebra
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Zebra | gambar makanan} - *mouseEvent* {*TouchButtonDown}*
34	Nama	Data memberi makan Gajah
	Where used/how used	*User* - Proses 3.10
	Deskripsi	Data yang digunakan ketika memberi makan Gajah
	Struktur data	- *marker*
	Penjelasan struktur data	- *marker* {gambar Gajah | gambar makanan} - *mouseEvent* {*TouchButtonDown}*

3.4 Perancangan Sistem

Perancangan sistem dimulai setelah tahap terhadap sistem selesai dilakukan. Perancangan dapat didefinisikan sebagai proses bagaimana aplikasi itu akan ditampilkan dengan melakukan perencanaan dan penggambaran awal. Yang nantinya akan menjadi satu dan utuh serta berfungsi sesuai harapan.

3.4.1 Buku Ensiklopedia yang Digunakan

Sebagai implementasi dan pemanfaatan sebuah buku tentang binatang, penulis menggunakan buku ensiklopedia yang tentu memiliki banyak gambar dan informasi yang terfokus. Adapun buku yang digunakan sebagai penerapan pada penelitian ini. Dapat dilihat pada gambar 3.37.



Gambar 3.37 Buku Ensiklopedia Anak Hebat "Hewan"

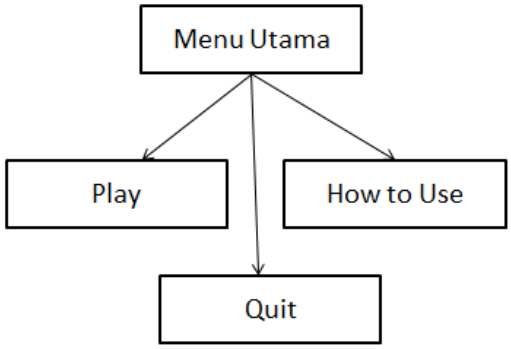
Buku ini menceritakan tentang 60 jenis hewan yang disukai anak – anak. Diharapkan berbagai pengetahuan dalam buku ini dapat menambah wawasan anak – anak tentang hewan – hewan tersebut. selain itu buku ini juga memberikan pengetahuan tentang lingkungan, agar anak – anak mencintai dan memelihara lingkungan.

3.4.2 Struktur Menu

Perancangan menu dibuat dengan harapan agar pengguna dapat mengunakan laboratorium virtual dengan mudah tanpa kesulitan, sehingga memudahkan pemakai dalam memilih menu aplikasi yang sedang berjalan.

3.4.2.1 Struktur Menu *Front end/ User*

Perancangan struktur menu merupakan gambaran jalur pemakaian aplikasi user pengguna dalam menggunakan aplikasi mobile ini. Perancangan Struktur menu pada aplikasi *ini* dapat dilihat pada gambar 3.38.



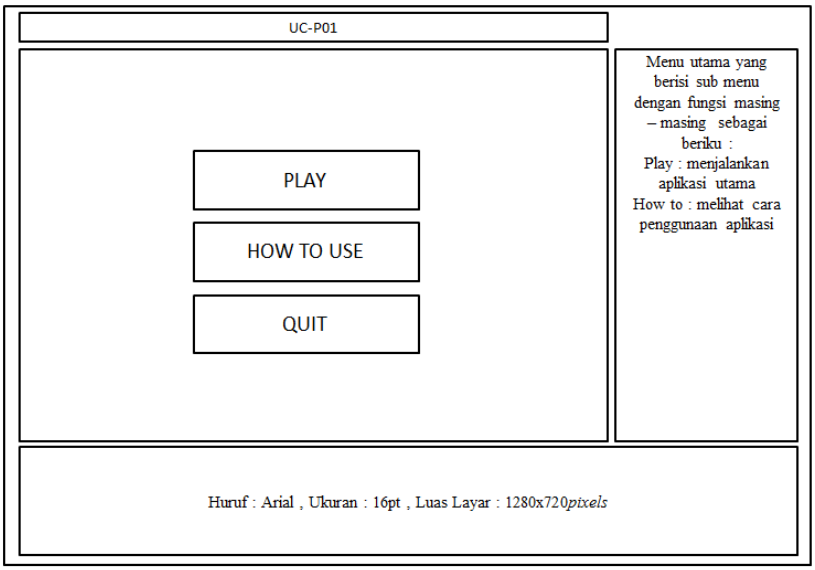
Gambar 3.38 Perancangan Menu pada Aplikasi

3.4.3 Perancangan Grafis Antarmuka

Perancangan antarmuka (*interface*) menggambarkan bagaimana perangkat lunak berkomunikasi dengan dirinya sendiri, dengan sistem yang berinteroperasi dengannya dan dengan manusia yang menggunakannya..

3.4.3.1 Rancangan Tampilan Menu Utama

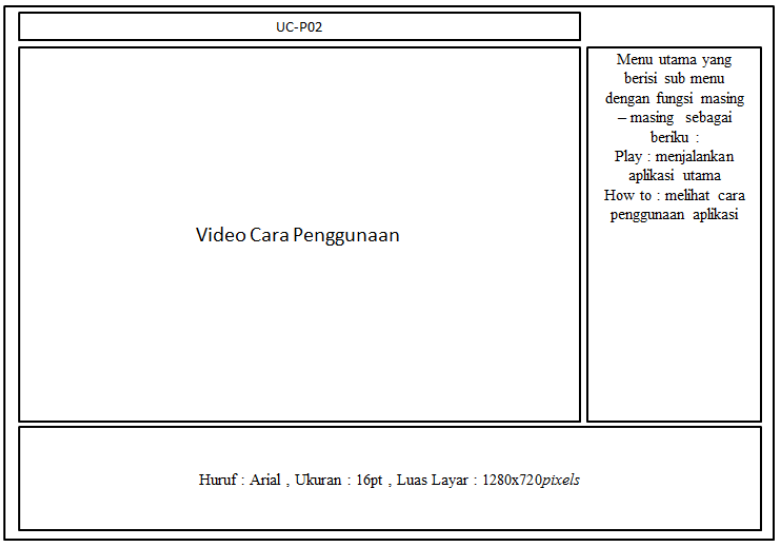
Rencana dari racangan tampilan pada aplikasi ini untuk mendeskripsikan setiap form. Adapun rancangan tampilan sebagai berikut



Gambar 3.39 Perancangan Tampilan Menu Utama

3.4.3.2 Rancangan Tampilan *How to Use*

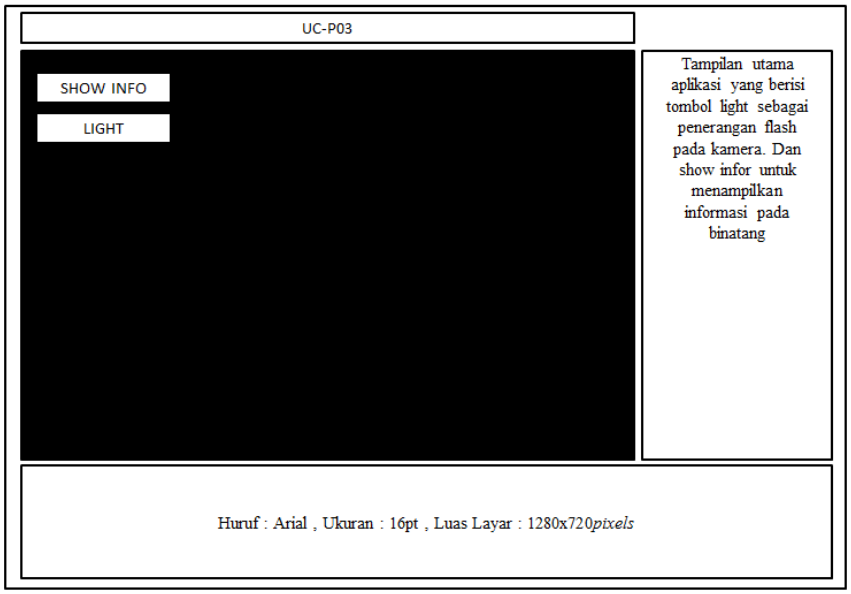
Gambaran umum rancangan tampilan untuk melihat video cara penggunaan dari aplikasi.



Gambar 3.40 Perancangan Menu pada cara penggunaan

3.4.3.3 Rancangan Tampilan *Tracking Marker*

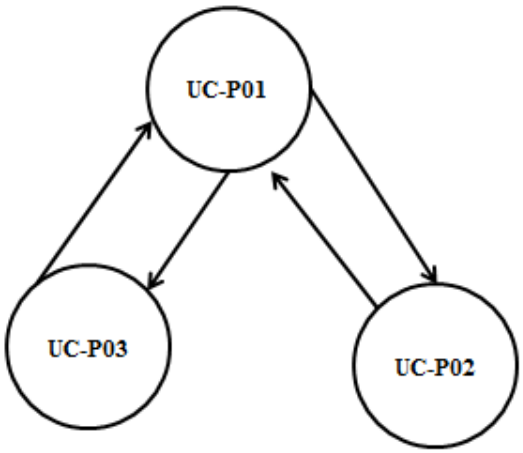
Gambaran umum dari rancangan tampilan utama dari aplikasi *ini* .



Gambar 3.41 Perancangan Menu pada Aplikasi Utama

3.4.3.4 Jaringan Semantik

Jaringan semantik merupakan jaringan data dan informasi yang menunjukan hubungan antar berbagai objek. Dapat di lihat pada gambar 3.42.



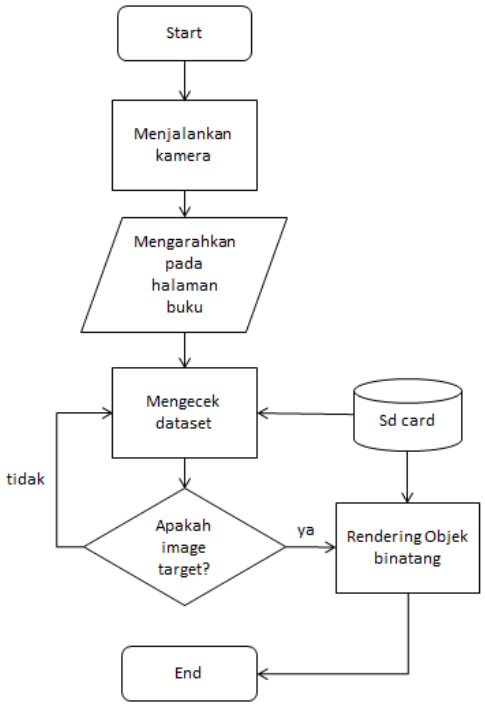
Gambar 3.42 Jaringan Semantik Aplikasi

3.4.3.5 Perancangan Prosedural

Pada bagian ini akan dibuat deskripsi rinci dari perangkat lunak, yaitu algoritma yang akan digunakan oleh masing-masing prosedur yang ada pada masing-masing modul. Perancangan prosedural menjelaskan tentang bagaimana program aplikasi berjalan. Adapun deskripsi rincinya adalah sebagai berikut :

3.4.3.5.1 *Flowchart* Menampilkan Objek Binatang

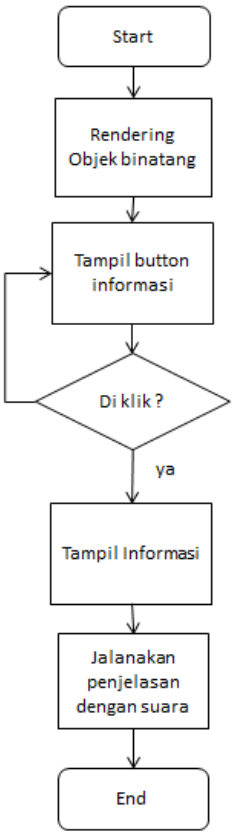
Flowchart pada gambar 3.43 menunjukkan proses dari sistem untuk menampilkan objek binatang dari masing – masing marker.



Gambar 3.43 *Flowchart* menampilkan objek binatang

3.4.3.5.2 *Flowchart* Menampilkan Informasi Binatang

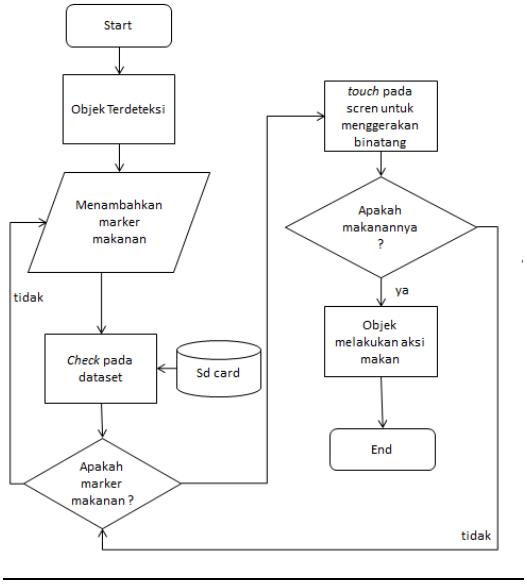
Flowchart pada gambar 3.44 menunjukkan proses dari *inputan user* yang ingin menampilkan informasi dari objek binatang.



Gambar 3.44 *Flowchart* menampilkan informasi binatang

3.4.3.5.3 *Flowchart* Memberi Makan Binatang

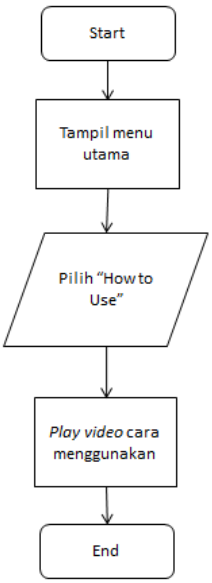
Flowchart pada gambar 3.43 menunjukkan proses dari *inputan user* yang *mengontrol binatang* untuk diberi makan.



Gambar 3.45 *Flowchart* Memberi objek binatang

3.4.3.5.4 *Flowchart* Menjalankan Video Cara Penggunaan

Flowchart pada gambar 3.44 menunjukkan proses dari *user* memilih menu "how to use" untuk melihat bagaimana aplikasi *ini* digunakan.



Gambar 3.45*Flowchart* menampilkan Video Cara Penggunaan