BAB III
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam 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-perbaikannya. Sistem yang dibuat merupakan edukasi berbasis *augmented reality* yaitu pengenalan binatang untuk anak. Aplikasi yang dibuat seolah-olah pengguna dapat berinteraksi langsung dengan obyek-obyek 3 dimensi yang dibuat.

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

3.1.1 Analisis Masalah

Analisis masalah adalah suatu gambaran masalah yang diangkat dalam penulisan skripsi tentang mengimplentasikan teknologi *augmented reality* pada aplikasi yang dapat melatih anak agar dapat menyayangi dan melindungi binatang. Salah satu aplikasi yang dapat memberikan informasi tentang binatang bagi anak usia 5 tahun keatas adalah pengenalan binatang menggunakan *magic book*, maka aplikasi yang akan diimplementasikan kedalam teknologi *augmented reality* adalah pengenalan binatang. Seperti hal yang sama saat anak melihat binatang secara langsung di dunia nyata, aplikasi pengenalan binatang ini didesain agar binatang yang di tampilkan semirip mungkin dengan binatang aslinya,sehingga anak pun dapat mengetahui bentuk asli binatang tersebut.

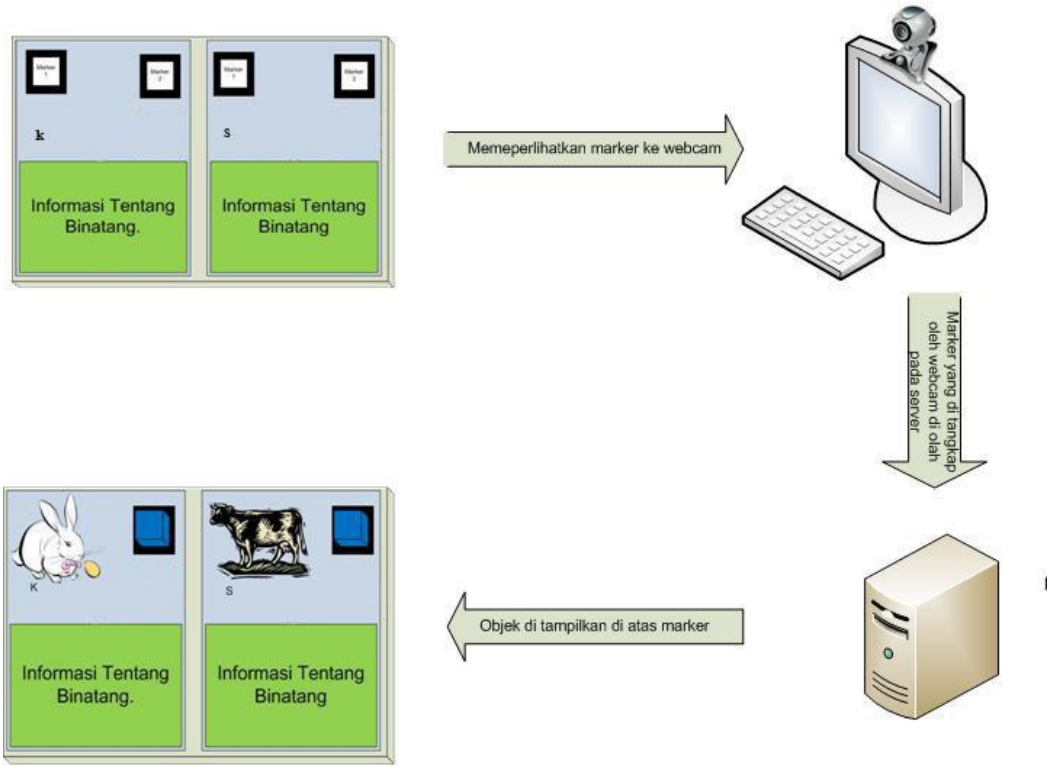
Aplikasi ini menampilkan objek hewan dalam bentuk 3 dimensi dan dapat mengeluarkan suara dari hewan tersebut. Untuk menampilkan binatang yang

terdapat di dalam apilkasi *user* mengarahkan marker yang telah disediakan ke webcam kemudian menutup virtual button unuk menampilkan binatang dan suara dari binatang tersebut.

3.1.2 Analisis Arsitektur Sistem

Pada arsitektur aplikasi yang akan dibangun terdiri dari beberapa komponen, yaitu ; *user* yang menggunakan aplikasi pengenalan binatang berbasis *augmented reality*, *user* mengarahkan *marker* / penanda sehingga *marker* dapat tertangkap olah kamera. Kemudian dari gambar yang didapat dari kamera sistem komputer melakukan *tracking marker* untuk mengidentifikasi *marker* yang digunakan oleh user. Komputer melakukan render obyek-obyek 3D yang digunakan dalam aplikasi.

User dapat melihat hasil manipulasi system melalui layar komputer/ monitor. Gambaran arsitektur sistem dapat dilihat pada gambar 3.1.



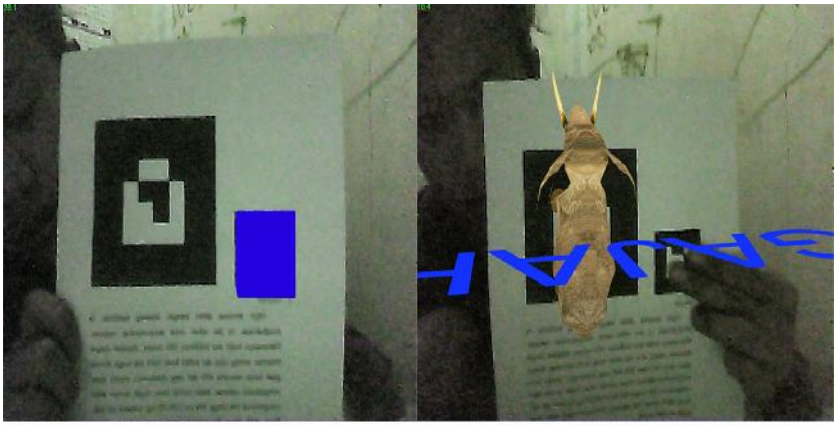
Gambar 3.1 Arsitektur Aplikasi

3.1.3 Analisis metode

Occlusion adalah hubungan antara suatu benda dengan benda lain jika kita lihat dari suatu sudut pandang. Hal ini tentunya mengurangi informasi antar

objek dalam lingkungan 3D, karena jika dilihat dari satu sudut pandang maka lingkungan 3D akan diproyeksikan kepada suatu bidang sehingga seolah- olah menjadi lingkungan 2D. Pengurangan dimensi ini menyebabkan informasi interaksi antar objek seperti keadaan bersinggungan atau beririsan. *Occlusion Based Detection* juga berguna untuk mendeteksi adanya suatu objek yang saling bertabrakan atau menghalangi, perancangan aplikasi ini menggunakan obyek *kubus* yang berfungsi sebagai virtual button. Teknik *occlusion based detection* pada aplikasi yang diterapkan para marker dan objek virtual button.

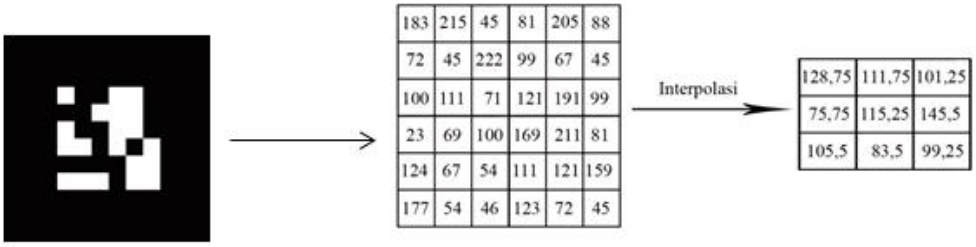
Secara sederhana *occlusion bassed detection* hanya mendefinisikan keadaan dimana suatu marker tidak terdeteksi karena tertutup oleh benda lain. Adapun simulasi dapat dilihat pada gambar 3.2



Gambar 3.2 (A) marker belum terhalang (B) marker telah terhalang

Untuk mendeteksi adanya suatu tabrakan atau penghalang yang mengakibatkan suatu marker tidak terdeteksi dapat dilakukan dengan cara menghitung jumlah tiap piksel, yang dilakukan secara iteratif mulai dari ujung kiri atas citra hingga ujung kanan bawah dengan pergeseran sebesar Δx dan Δy. Semakin kecil nilai Δx dan Δy, maka semakin akurat pula proses deteksi.

Sebagai contoh perhitungan *matriks image* dengan angka-angka piksel *sampel* dapat dilihat di dalam gambar tabel matrik di bawah ini :



Nilai piksel pada koordinat pada citra hasil interpolasi diperoleh dengan menghitung nilai rata – rata dari 4 nilai piksel pada citra asli, yaitu :

Perhitungan Nilai Piksel Hasil Interpolas

Nilai Piksel Citra Asli	**Nilai Piksel Citra Interpolasi**
(183+215+72+45) / 4	128,75
(100+111+23+69) / 4	75,75
(124+67+177+54) / 4	105,5
(45+81+222+99) / 4	111,75
(71+121+100+169) / 4	115,25
(54+111+46+123) / 4	83,5
(205+88+67+45) / 4	101,25
(191+99+211+81) / 4	145,5
(121+159+72+45) / 4	99,24

Dari perhitungan *matriks image* diatas, maka diperolah nilai matriks, maka nilai pada marker tersebut menggunakan persamaan 2.1 adalah :

f(x) = (128.75 +111.75 + 101.25) + (75.75 + 115.25 + 145.5) + (105.5 + 83.5 + 99.25) = 996.5

Apabila jumlah perhitungan suatu nilai dari *matriks image* tersebut kurang dari **996.5** maka sistem akan mendeteksi adanya suatu tabrakan atau suatu penghalang yang mengakibatkan tidak terbacanya marker.

3.1.4 Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan system. Spesifikasi ini juga melipiti elemen ataupun komponen-komponen apa saja yang dibutuhkan system yang akan dibangun sampai system tersebut diimplementasikan. Analisis kebutuhan ini juga

menentukan spesifikasi masukan yang dibutuhkan system. Pada analisis kebutuhan system non-fungsional ini dijelaskan mengenai analisis perangkat keras (*hardware*), perangkat lunak (*software*) dan pengguna (*user*) .

Aplikasi pengenalan binatang ini merupakan aplikasi berbasis *Augmented reality*. *Aplikasi* penganalan binatang berbasis *Augmented reality* dapat dijadikan alternatif cara mengenalkan anak kepada binatang untuk melatih anak agar dapat melestarikan, melindungi, dan menyayangi binatang. Karena a*plikasi* penganalan binatang adalah produk berbasis AR yang menggunakan *marker* sebagai pola pelacakan atau *tracking marker,* maka dibutuhkan komputer dan perangkat lunak sebagai pengolah citra dan *marker*. Berikut ini adalah perangkat lunak yang dibutuhkan oleh *aplikasi* penganalan binatang.

3.1.5 Analisis Kebutuhan Perangkat Keras

Aplikasi penganalan binatang adalah program yang membutuhkan grafis video dengan kemampuan relative tinggi. Hal ini karena model-model yang ditampilkan menggunakan teknik *rendering.*

Berdasarkan studi literatur terhadap proyek-proyek pengembangan teknologi AR, maka diperolah spesifikasi minimum perangkat keras yang dibutuhkan oleh pihak pengembang untuk teknologi AR seperti berikut ini:

1. *Processor* Intel core 2 duo atau setara.
2. Sistem Operasi Windows 7
3. RAM minimal 2 GB
4. Ruang sisa *hardisk* minimal 4 GB
5. *Motherboard* dengan *chipset* yang kompatibel dengan VGA *card* yang dipakai.
6. Optimum menggunakan VGA *card* dengan kemampuan me-*render* grafis 3D, seperti GeForce 6xxx atau ATI 1xxx series.
7. *Webcam*

   Webcam dipakai untuk menangkap citra yang kemudian diproses oleh *tracker library.* Hasil dari proses ini akan menghasilkan matriks transformasi *marker* relative terhadap *webcam.*

8. *Speaker*

   Speaker dibutuhkan untuk mengetahui apakah sistem yang di bangun telah dapat mengeluarkan suara atau belum.

Sedangkan spesifikasi minimum perangkat keras yang di butuhkan oleh pihak pengguna untuk teknologi AR seperti berikut ini:

1. *Processor* Intel core 2 duo atau setara.
2. Sistem Operasi Windows 7
3. RAM minimal 2 GB
4. Ruang sisa *hardisk* minimal 70 MB
5. *Motherboard* dengan *chipset* yang kompatibel dengan VGA *card* yang dipakai.
6. Optimum menggunakan VGA *card* dengan kemampuan me-*render* grafis 3D, seperti GeForce 6xxx atau ATI 1xxx series.
7. *Webcam*

   Webcam dipakai untuk menangkap pola yang berada pada marker.
8. *Speaker*

   Speaker dibutuhkan untuk dapat mendengarkan suara dari binatang yang terdapat didalam aplikasi.

3.1.6 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan kumpulan perintah-perintah yang diberikan kepada perangkat keras agar saling berinteraksi untuk melakukan suatu tugas. Perangkat lunak yang dibutuhkan oleh pihak pengembang untuk membangun aplikasi adalah sebagai berikut :

1) Sistem operasi Windows 7
2) Adobe Flash Builder 4.6 Premium
3) Adobe Flash CS5
4) Photoshop CS3
5) Autodesk 3D max
6) Adobe flash player / Gom player
7) Mozilla firefox

Sedangkan Perangkat lunak yang dibutuhkan oleh pihak pengguna untuk aplikasi ini adalah sebagai berikut :

1) Sistem operasi Windows 7
2) Adobe flash player / Gom player

3.1.7 Analisis Pengguna

Analisis pengguna dimaksudkan untuk mengetahui siapa saja pengguna yang terlibat system. Pengguna yaitu *public* yang dapat mengerti dan memahami komputer sehingga dapat menggunakan aplikasi yang akan dibangun. Karena aplikasi ini ditujukan agar anak dapat lebih mengenal dunia binatang, dan usia anak yang menggunakan aplikasi in adalah 5 tahun ke atas, oleh sebab itu anak harus didampingi orang tua. Aplikasi ini juga tidak hanya dapat digunakan dikalang anak –anak akan tetapi dapat digunakan oleh umum yang memahami komputer.

3.1.8 Analisis Kebutuhan Fungsional

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

3.2 Pemodelan Sistem

Pemodelan sistem dilakukan dengan metode berorientasi Obyek dengan menggunakan standarisasi UML, dengan tahap – tahap penentuan siapa saja actor yang terlibat dan perancangan *use case specification, sequence diagram*.

3.2.1 *Use Case Diagram*

Use case diagram ini digunakan untuk menggambarkan hubungan sejumlah external actor dengan use case yang terdapat dalam system. *Use case* diagram ini hanya menggambarkan keadaan lingkungan system yang dapat dilihat dari luar oleh actor. Pada use case ini terdapat dua actor, yaitu user dan system. Penjelasan mengenai dua actor dapat dilihat pada tabel 3.1

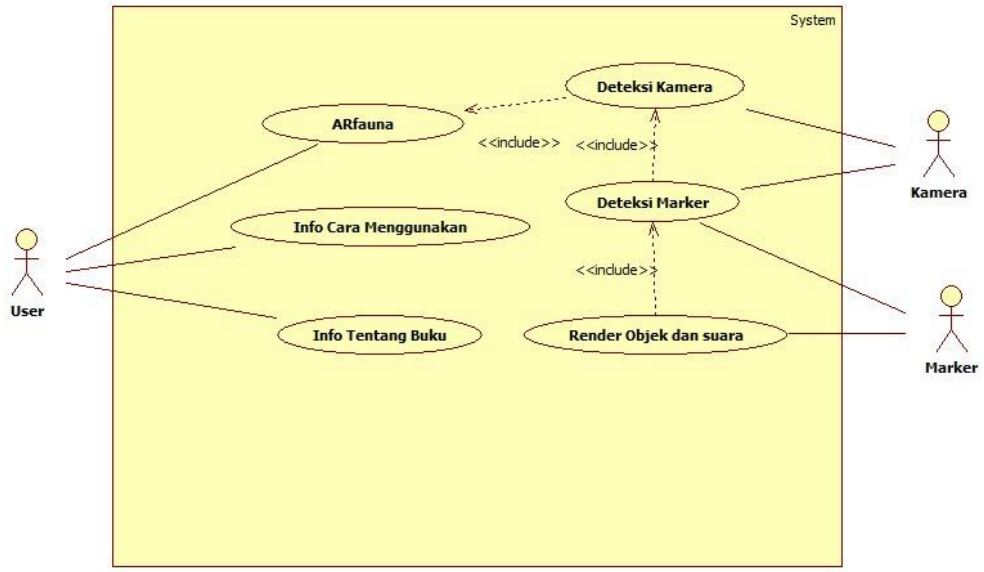
Tabel 3.1 Aktor dalam *Use Case*

Nama Aktor	Definisi
User	Orang yang berinteraksi dengan system, yaitu pengguna yang mengakses semua fungsi yang disediakan system.
Marker	Marker adalah alat bantu yang digunakan sebagai penanda dimana obyek virtual 3D akan ditampilkan.
Kamera	Kamera adalah alat bantu yang digunakan sebagai pendeteksi keberadaan marker.

3.2.2 *Skenario Use Diagram*

Use case pada aplikasi berhubungan dengan aplikasi dan penggunaan *augmented reality* pada permaianan. Dimana *augmented reality* digunakan dalam memanipulasi obyek-obyek maya dengan berinteraksi dengan obyek fisik di lingkungan nyata. Agar dapat melihat detail *use case* dapat lihat pada tabel 3.2

Diagram *use case* pada gambar 3.3 dilakukan oleh actor dengan User dimana diagram *use case* ini lebih bagaimana actor dapat menjalankan fungsi-fungsi yang terdapat pada aplikasi.



Gambar 3.3 *Use case Diagram*

Bagian ini menjelaskan masing-masing deskripsi dari *Use Case* diagram yang dilakukan oleh pengguna. Mulai dari proses memiih masuk kedalam aplikasi, cara menggunakan aplikasi, dan tentang buku.

Tabel 3.2 Deskripsi *Use Case Diagram*

NO	Kode *Use Case*	Nama *Use Case*	Keterangan
1	UC-P01	ARfauna	Proses untuk memulai aplikasi
2	UC-P02	Mendeteksi Kamera	Aplikasi Mendeteksi ketersediaan Kamera
3	UC-P03	Deteksi Marker	Sistem mendeteksi keberadaan marker melalui gambar yang
4	UC-P04	Render Obyek dan suara	Sistem akan melakukan render obyek pada awal aplikasi
5	UC-P05	Info Cara menggunakan aplikasi	Menampilkan petunjuk cara menggunakan aplikasi
6	UC-P06	Info Tentang buku	Menampilkan tentang buku *Magic book*

Sekenario aplikasi AR fauna terkait dengan *use case* AR fauna dapat dilihat pada tabel 3.3

Tabel 3.3 Spesifikasi Skenario : ARfauna

Nama *Use Case*	ARfauna	
Nomor	UC-P01	
Aktor	User	
Kondisi Awal	User berada di tampilan awal aplikasi	
Reaksi Aktor		**Reaksi Sistem**
1. User memilih menu ARfauna. 1. User menunujukan marker pada kamera.		2. Sistem mendeteksi kamera. 2. Sistem merender objek dan menampilkan diatas marker.
Kondisi Akhir	ARfauna menrender objek dan suara	
Pengecualian	User memilih menu yang lain.	

Sekenario mendeteksi kamera terkait dengan *use case* mendeteksi kamera dapat dilihat pada tabel 3.4

Tabel 3.4 Spesifikasi Skenario : Mendeteksi Kamera

Nama *Use Case*	Mendeteksi Kamera
Nomor	UC-P02
Aktor	User, kamera

Kondisi Awal	User memilih menu ARfauna	
Reaksi Aktor		**Reaksi Sistem**
1. Memilih menu ARfauna		2. Sistem mendeteksi ketersediaan kamera
Kondisi Akhir	Kamera Terdeteksi	
Pengecualian	Kamera tidak terdeteksi	

Sekenario deteksi kamera terkait dengan *use case* deteksi marker dapat dilihat pada tabel 3.5

Tabel 3.5 Spesifikasi Skenario: Deteksi *Marker*

Nama *Use Case*	Deteksi *Marker*	
Nomor	UC-P03	
Aktor	User, Kamera, Marker	
Kondisi Awal	Kamera telah terdeteksi	
Reaksi Aktor		**Reaksi Sistem**
User menunjukan marker ke Kamera		Inisiasi DirectShowCapture Inisiasi FLAR TOOLKIT *Marker* Tracker
Kondisi Akhir	Aplikasi ARfauna siap mendeteksi *marker*	
Pengecualian	1.Perangkat kamera tidak ditemukan 2. File kalibrasi tidak ditemukan	

Sekenario render obyek terkait dengan *use case* render obyek dapat dilihat pada tabel 3.6

Tabel 3.6 Spesifikasi Skenario: Render Obyek

Nama *Use Case*	Render Obyek	
Nomor	UC-P04	
Aktor	User, Kamera, Marker	
Kondisi Awal	*Marker* telah terdeteksi	
Reaksi Aktor		**Reaksi Sistem**

Menunjukan *marker*	Render objek binatang. Memanggil suara binatang
Kondisi Akhir	Semua konten berhasil direnderuntuk digunakan dalam Aplikasi
Pengecualian	1. Filepart yang didefinisikan untuk melakukan render tekstur tidak ditemukan 2. Filepart yang didefinisikan untuk melakukan render

Sekenario cara menggunakan aplikasi terkait dengan *use case* cara menggunakan aplikasi dapat dilihat pada tabel 3.7

Tabel 3.7 Spesifikasi Skenario : Cara menggunakan aplikasi

Nama *Use Case*	Info cara menggunakan aplikasi
Nomor	UC-P05
Aktor	User
Kondisi Awal	User berada di menu tampilan utama
Reaksi Aktor	**Reaksi Sistem**
User memilih menu cara menggunakan	2. Sistem menampilkan petunjuk
Kondisi Akhir	User memilih cara menggunakan
Pengecualian	User memilih menu lain

Sekenario tentang terkait dengan *use case* tentang buku dapat dilihat pada tabel 3.8

Tabel 3.8 Spesifikasi Skenario : Tentang buku

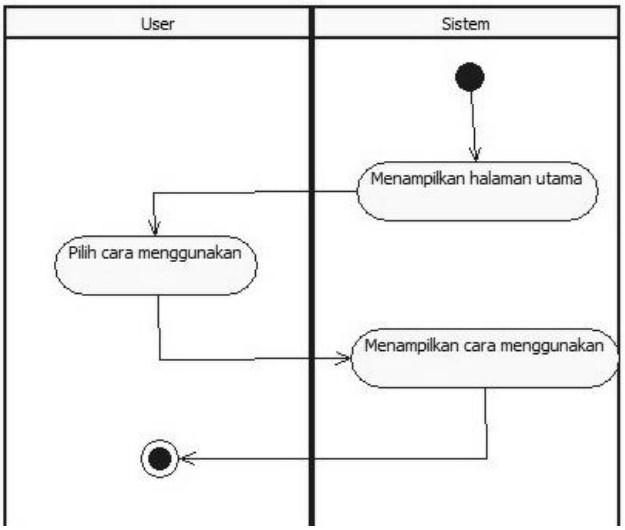
Nama *Use Case*	Info tentang buku
Nomor	UC-P06
Aktor	User
Kondisi Awal	User berada di menu tentang buku
Reaksi Aktor	**Reaksi Sistem**
1. User memilih tentang buku	Sistem menampilkan informasi tentang buku

Kondisi Akhir	User berada di menu tentang buku
Pengecualian	1. User memilih menu lain

3.2.3 *Activity Diagram*

Diagram activity menggambarkan berbagai alir aktivitas dalam sistem yang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. Penggambaran *activity* diagram memiliki kemiripan dengan *flowchart diagram. Activity diagram* memodelkan *event-event* yang terjadi pada *Use Case* dan digunakan untuk pemodelan aspek dinamis dari sistem.

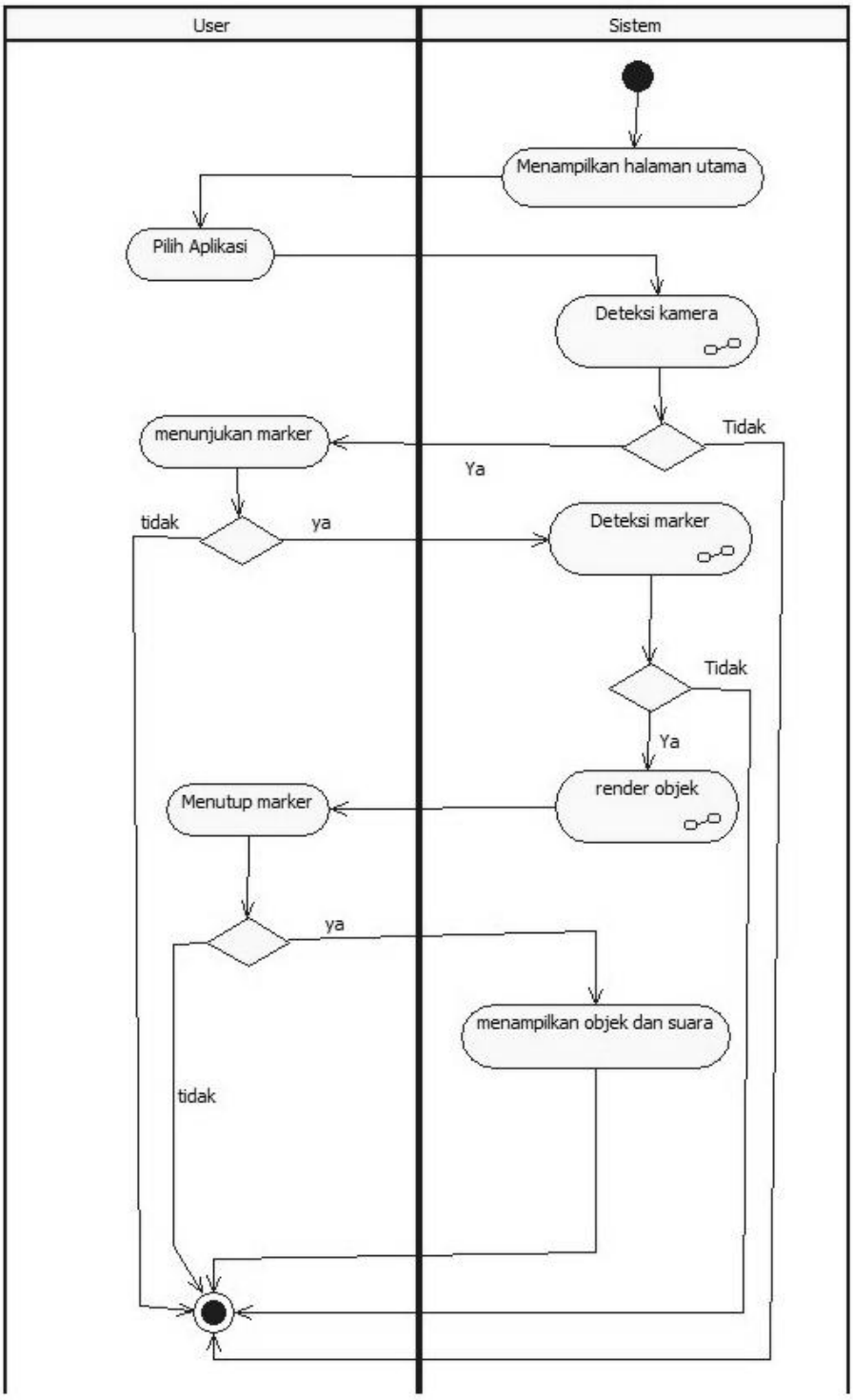
1. *Activity Diagram* terkait dengan *use case* cara menggunakan dapat dilihat pada gambar 3.4.



Gambar 3.4 *Activity Diagram* cara menggunakan aplikasi

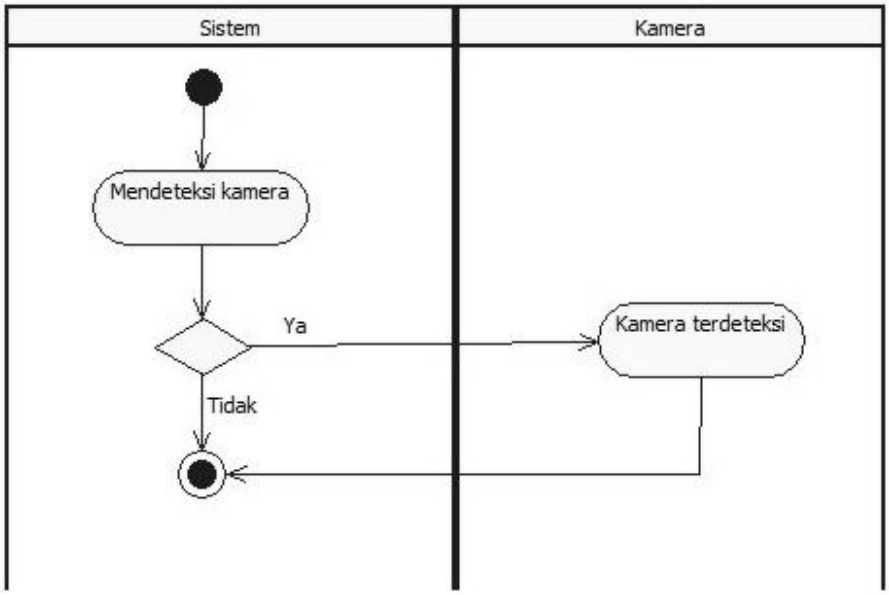
2.

3. *Activity diagram* terkait dengan Skenario aplikasi dapat dilihat pada gambar 3.5.



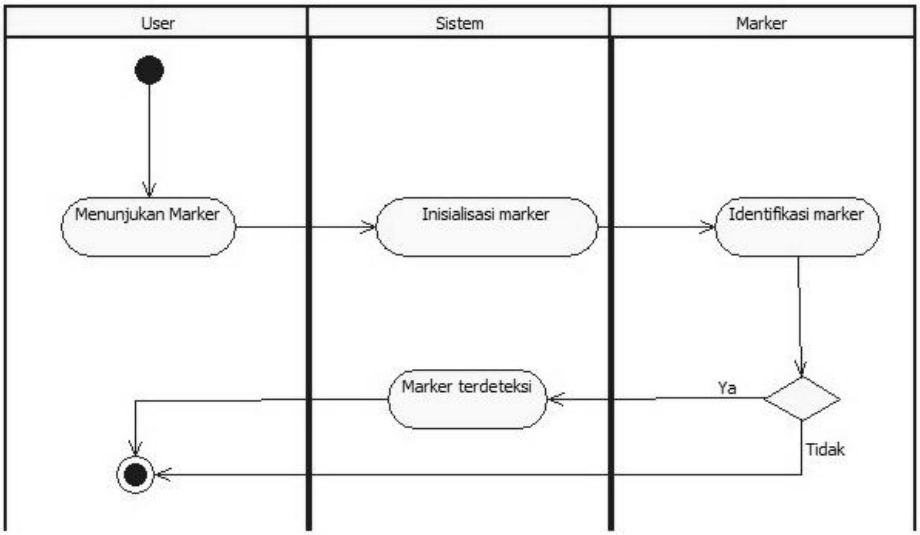
Gambar 3.5 *Activity Diagram* ARfauna

4. Suba*ctivity Diagram* terkait dengan *use case* deteksi kamera dapat dilihat pada gambar 3.6



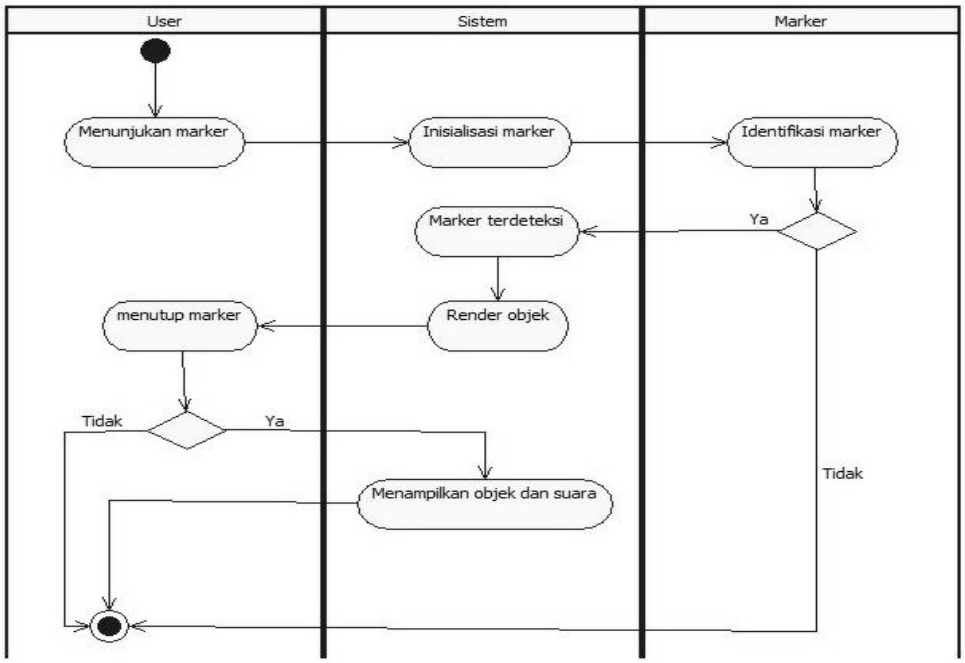
Gambar 3.6 Suba*ctivity Deteksi kamera*

5. Suba*ctivity Diagram* terkait dengan *use case* deteksi marker dapat dilihat pada gambar 3.7



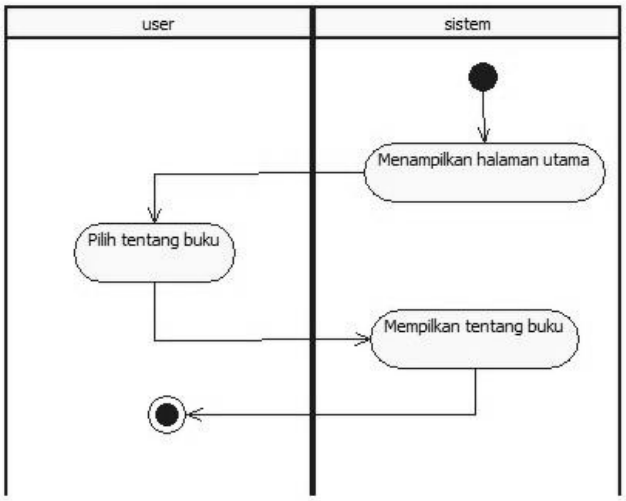
Gambar 3.7 Suba*ctivity Diagram* deteksi marker

6. Suba*ctivity Diagram* terkait dengan *use case* render objek dapat dilihat pada gambar 3.8



Gambar 3.8 Suba*ctivity Diagram* render objek dan suara

7. *Activity Diagram* terkait dengan *use case* tentang buku dapat dilihat pada gambar 3.9

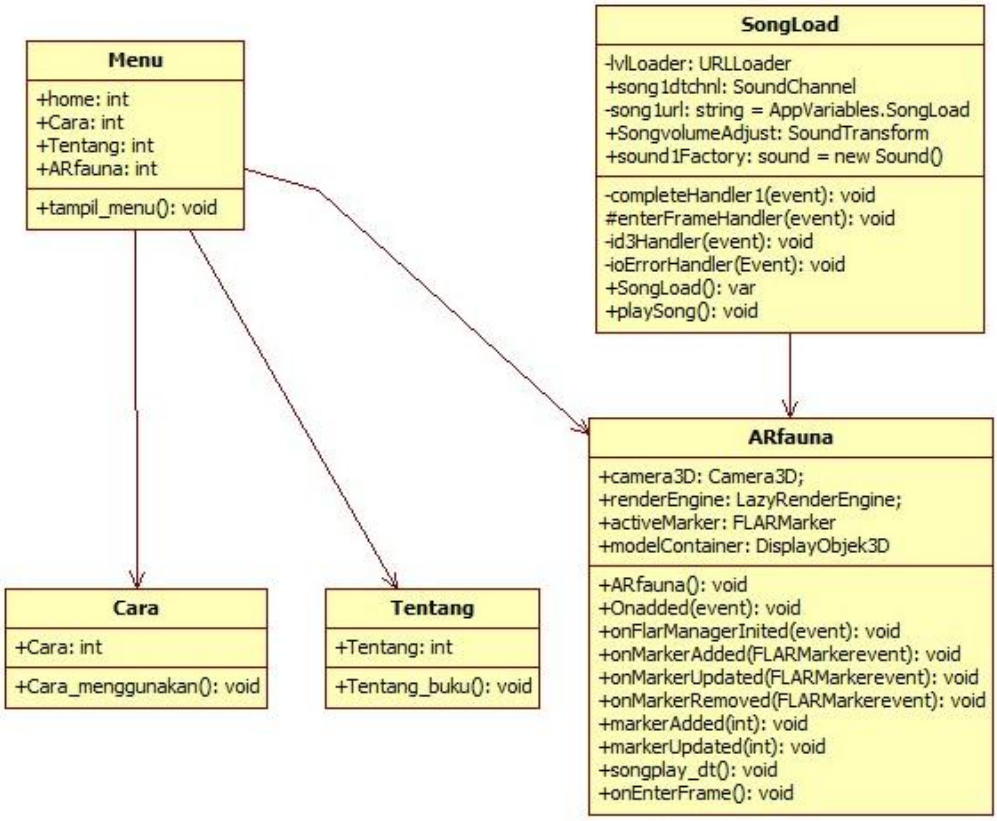


Gambar 3.9 *Activity Diagram* tentang buku

3.2.4 *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). Gambar 3.11 adalah kelas diagram dari Implementasi Teknologi Augmented Reality Pada Aplikasi *Pengenalan binatang untuk anak usia dini*.



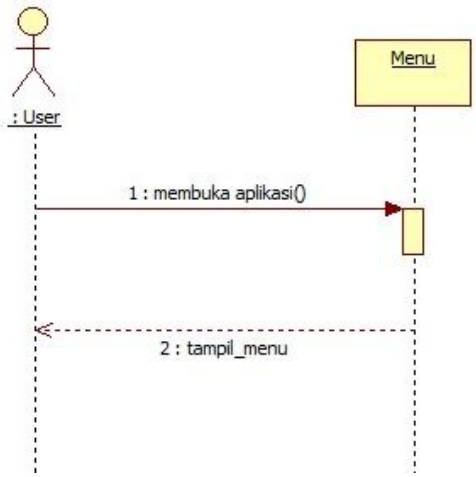
Gambar 3.10 *Class Diagram* Aplikasi

3.2.5 *Sequence Diagram*

Sequence diagram menggambarkan perilaku pada sebuah scenario. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek- obyek ini di dalam use case. Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progres vertikal.

1. *Sequence Diagram* menu

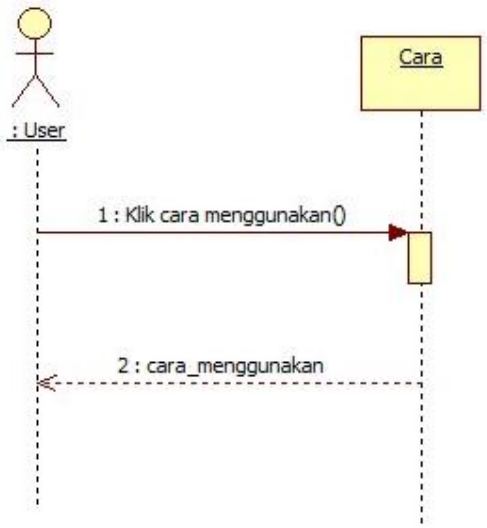
   Gambar *Sequence Diagram* menu dapat dilihat pada gambar 3.12



Gambar 3.11 *Sequence Diagram* Menu

2. *Sequence Diagram* Cara Menggunakan Aplikasi

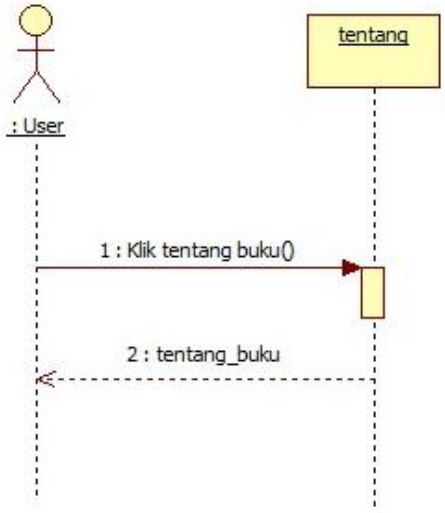
Gambar *Sequence Diagram* cara menggunakan aplikasi dapat dilihat pada gambar 3.13



Gambar 3.12 *Sequence Diagram* Cara Menggunakan Aplikasi

3. *Sequence Diagram* Tentang buku

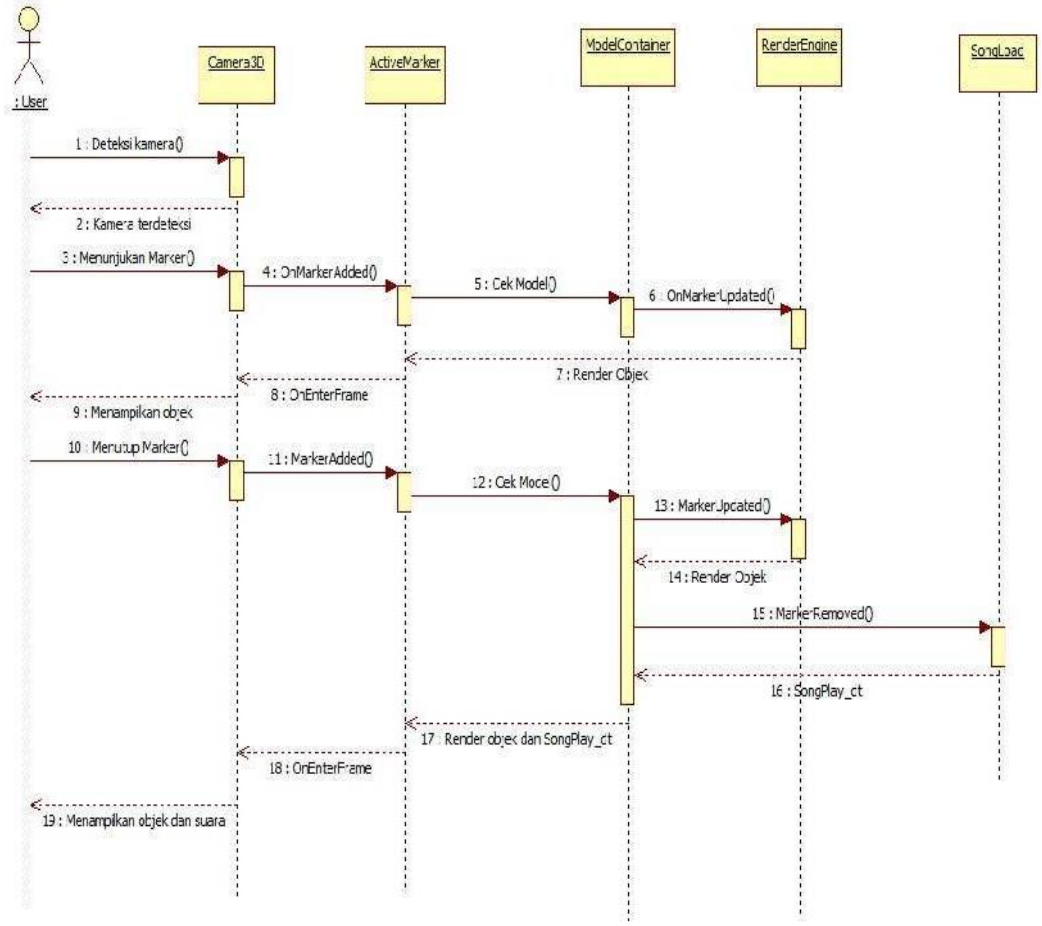
Gambar *Sequence Diagram* tentang buku dapat dilihat pada gambar 3.14



Gambar 3.13 *Sequence Diagram* tentang buku

4. *Sequence Diagram* ARfauna

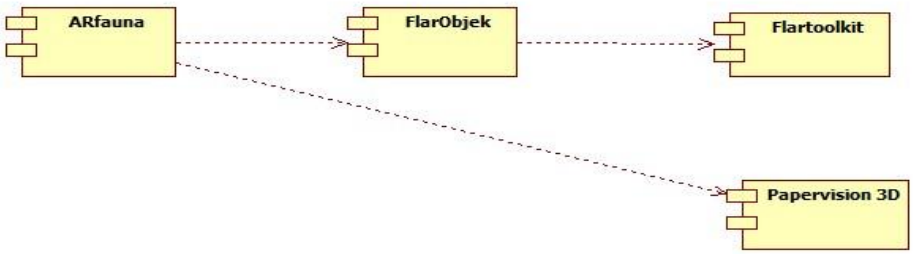
   Gambar *Sequence Diagram* ARfauna dapat dilihat pada gambar 3.15



Gambar 3.14 *Sequence Diagram* ARfauna

3.2.6 *Component Diagram*

Component diagram adalah bagian fisik dari sebuah system, karena menetap dikomputer, bukan dibenak para analis. Komponen bisa berupa table, file data, file exe dan lain-lain(dynamic link library), dokumen dan dan lain-lain. *Component* adalah implementasi dari sebuah class. *Component diagram* rancangan aplikasi *game* yang akan dibangun dapat dilihat pada gambar 3.17.

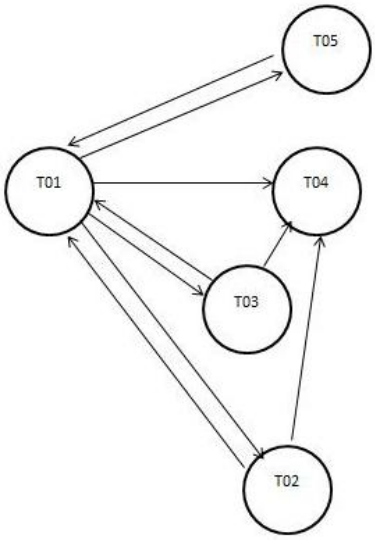


Gambar 3.15 *Component Diagram* Rancangan Aplikasi

Hal terpenting dari *component* adalah *component* memewakili potongan-potongan yang independen yang bisa dipesan dan diperbaharui sewaktu-waktu []12].

3.2.7 Jaringan Semantik

Jaringan semantik merupakan jaringan data dan informasi, yang menunjukan hubungan antar berbagai objek. dapat dilihat pada gambar 3.18



Gambar 3.16 Jaringan Semantik

Keterangan :

T01= Home

T02= Cara menggunakan

T03= Tentang buku

T04= Aplikasi AR fauna

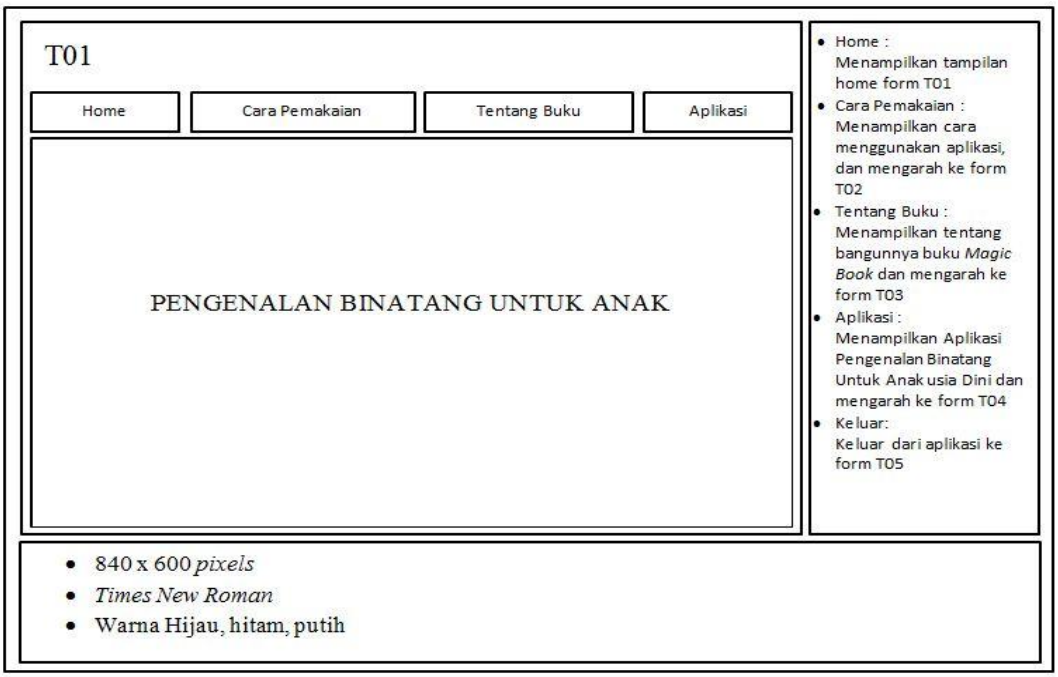
T05= Keluar

3.2.8 Perancangan Sistem

Perancangan sistem akan dimulai setelah tahap analisis terhadap sistem selesai dilakukan. Perancangan dapat di definisikan sebagai proses aplikasi berbagai teknik dan prinsip bagi tujuan pendefinisian suatu perangkat, suatu proses atau sistem dalam detail yang memadai untuk memungkinkan merealisasikan objek. Perancangan digambarkan sebagai proses multi-langkah dimana representasi menu antarmuka dan aplikasi.

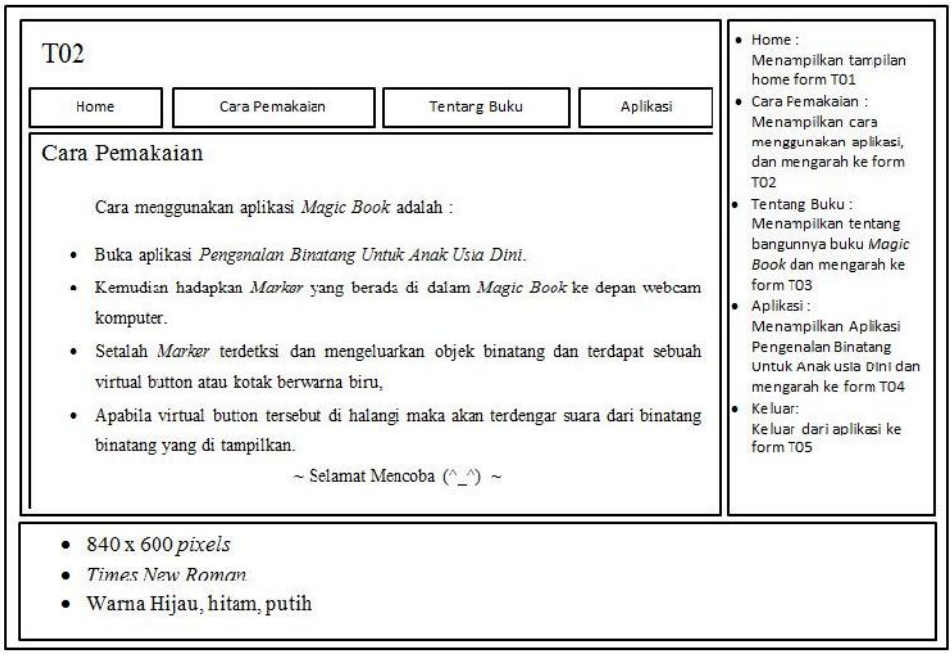
3.2.9 Perancangan Antarmuka

Desain perancangan antarmuka aplikasi dapat dilihat pada gambar 3.19 sampai gambar 3.23



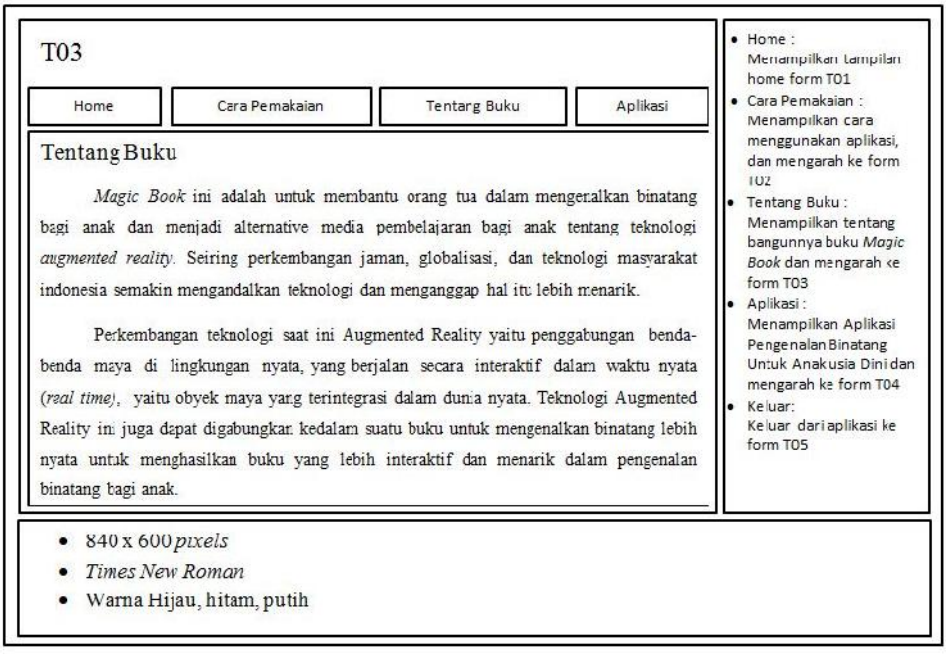
Gambar 3.17 Perancangan Antarmuka Home

Perancangan antarmuka cara menggunakan aplikasi dapat dilihat pada gambar 3.20



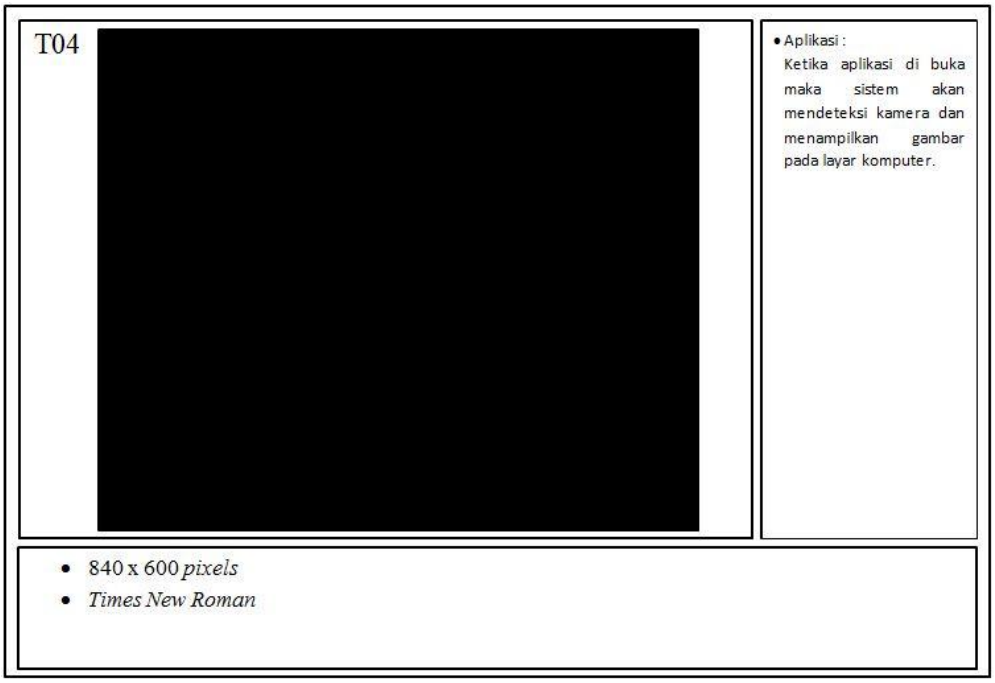
Gambar 3.18 Perancangan Antarmuka Cara Menggunakan Aplikasi

Perancangan antarmuka tentang buku dapat dilihat pada gambar 3.21



Gambar 3.19 Perancangan Antarmuka Tentang Buku

Perancangan antarmuka aplikasi ARfauna dapat dilihat pada gambar 3.22



Gambar 3.20 Perancangan Antarmuka Aplikasi ARfauna

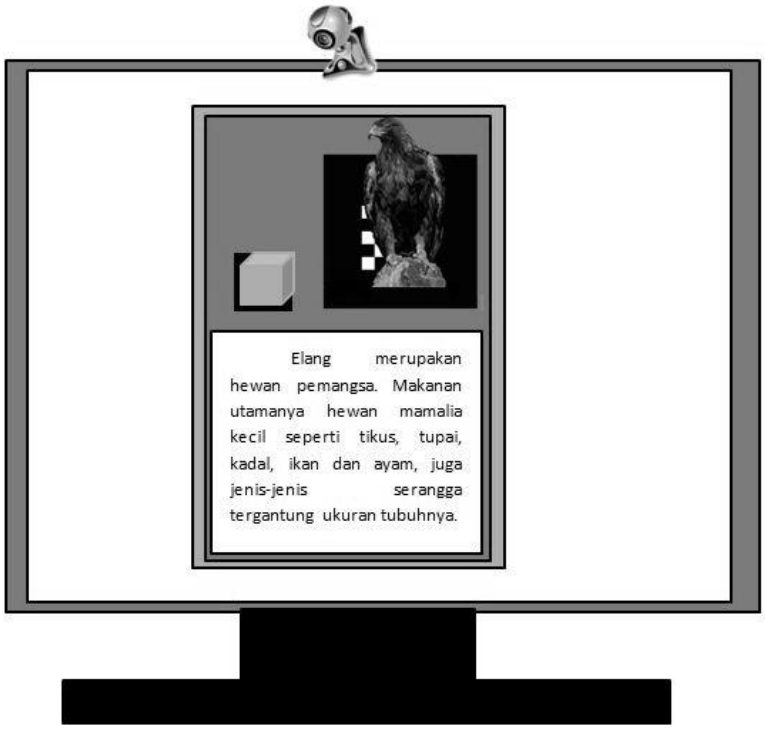
3.2.10 Perancangan *Aplikasi*

Daerah area aplikasi berada dalam *marker* yang telah dibuat. Dalam area aplikasi ini terdapat obyek-obyek binatang dalam model 3 dimensi dan dapat mengeluarkan suara apabila marker dihadapkan pada kamera. Aplikasi ini memiliki tujuan yaitu, agar anak dapat balajar dan mengetahui berbagi macam jenis binatang berdasarkan jenis makanan. Untuk mencapai tujuan itu user bertindak sebagai pihak yang mengarahkan *marker* ke webcam.

Area aplikasi ini berbentuk sebuah kotak, dimana setiap marker tersebut terdapat suatu obyek binatang. Semakin besar marker maka akan semakin besar pula obyek yang akan di tampilkan begitu pun dengan semakin jauh jarak marker dengan webcam maka semakin kecil pula obejk yang akan di tampilkan. Desain perancangan antarmuka aplikasi dapat dilihat pada gambar 3.24 dan gambar 3.25



Gambar 3.21 Tampilan pada buku



Gambar 3.22 Tampilan pada layar komputer

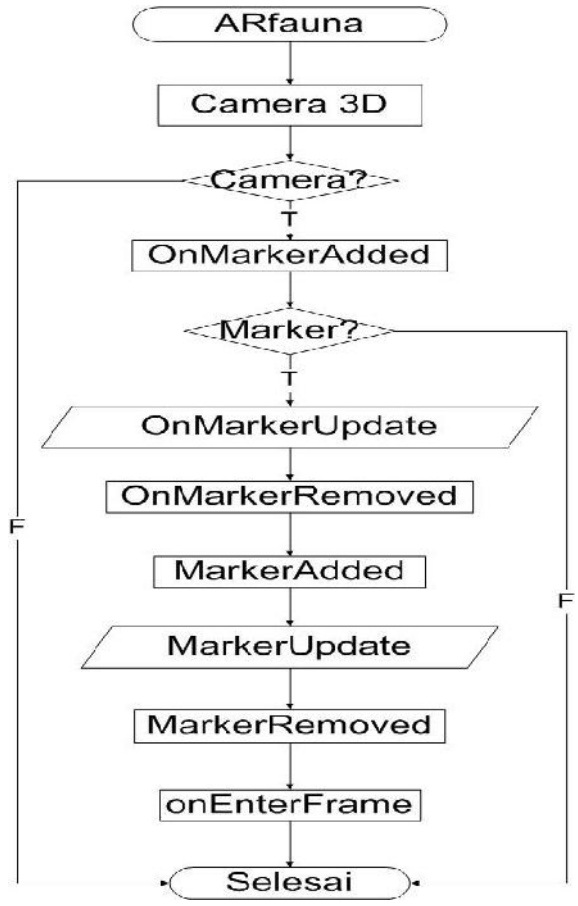
3.2.11 Perancangan method

Perancangan komponen method merupakan perancangan yang dibuat setelah perancangan jaringan semantik dan antarmuka. Perancangan ini berfungsi

untuk mendeskripsikan method-method yang berada di dalam aplikasi pengenalan binatang untuk anak usia dini menggunakan teknologi augmented reality. Adapun method-method yang terdapat dalam aplikasi yang akan dibangun adalah sebagai berikut :

1 Perancangan methode ARfauna

Method cara menggunakan seperti terlihat pada gambar 3.26



Gambar 3.23 perancangan methode ARfauna