BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini akan menjelaskan tentang analisis, perancangan sistem menggunakan pemodelan UML(*Unified Modeling Language*), perancangan *database*, dan perancangan antarmuka (*Interface*).

3.1 Analisis Sistem

Pembuatan tugas akhir ini akan membangun aplikasi untuk mempermudah membantu dalam mengenal alat musik tradisional indonesia dalam bentuk 3D dengan menggunakan teknologi *Augmented Reality*. Dengan menggunakan objek *virtual* 3D diharapkan dapat meningkatkan pengetahuan setiap orang yang ingin mengenal dan mengetahui bagaimana bentuk dan nada suara dari alat musik tradisional indonesia.

3.1.1 Analisis Masalah

Alat musik tradisional indonesia merupakan salah satu dari kebudayaan indonesia yang perlu dilestarikan. Semakin berkembangnya jaman membuat alat musik tradisional indonesia ini terpantau cukup jauh peminatnya dibandingkan dengan alat musik modern. Penyebabnya yaitu bisa dikarenakan alat musik tradisional indonesia hanya bisa ditemukan di tempat-tempat tertentu. Dan jarak yang cukup jauh tempatnya dari keberadaan oarang yang ingin mengenalnya.

Oleh karena itu dibutuhkan suatu metode untuk mengenalkan alat musik tradisional indonesia ini untuk setiap orang. Diharapkan dengan membangun aplikasi *Augmented Reality* ini dapat memudahkan setiap orang untuk mengenal dan mempelajarinya.

3.1.2 Analisis Kebutuhan Data

Analisis kebutuhan data digunakan untuk membantu pengumpulan data dalam perancangan aplikasi, berikut kebutuhan data yang diperlukan dalam perancangan aplikasi yang akan dibuat:

Tabel 3.1 Tabel Analisis Data

Field Name	*Type*	*Description*
Marker	.Img	Merupakan gambar 2 dimensi yang digunakan untuk menampilkan objek 3 dimensi.
Alat_Musik_Tradisional	.blend	Berupa gambar 3 dimensi yang kemudian diproses Unity untuk menampikan objek *virtual*.
Vuforia	.Sdk	Merupakan *Software Development Kit*, untuk perangkat mobile dalam pembuatan AR.
Informasi_Objek 3D	Teks, Suara	Penjelasan tentang objek 3D alat musik tradisional yang muncul diatas *marker*.

3.1.3 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non-fungsional menggambarkan kebutuhan sistem seperti kebutuhan perangkat keras, perangkat lunak, serta *user* sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang diterapkan. Analisis dan kebutuhan non-fungsional memiliki tiga tahapan, yaitu:

3.1.3.1 Kebutuhan Pengguna (*user*).

Aplikasi pengenalan ini digunakan oleh user, dimana spesifikasi pengguna adalah sebagai berikut:

1. Pengguna bisa menggunakan *smartphone*.
2. Pengguna memahami tentang penggunaan aplikasi pada *smartphone*.
3. Pengguna dapat menampilkan *marker* yang dicetak dalam buku.

3.1.3.2 Kebutuhan Perangkat Lunak (*software*).

Analisis perangkat lunak terdiri dari spesifikasi minimum yang dibutuhkan untuk membangun perangkat lunak. Berikut adalah spesifikasi minimum perangkat lunak yang digunakan dalam membangun aplikasi pemasaran produk *handphone* ini:

Tabel 3.2 Kebutuhan Perangkat Lunak

PC	Smartphone
System operasi Windows 10	Android 4.0 atau lebih
Blender	
Unity 3D	
Vuforia	
JDK (Java Development Kit)	
Android SDK (Standard Development Kit)	

3.1.3.3 Kebutuhan Perangkat Keras (*hardware*).

Smartphone dan komputer terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat lunak akan memberikan perintah kepada perangkat keras untuk melakukan tugas tertentu. Pada pembangunan aplikasi pemasaran produk, perangkat keras yang digunakan untuk membangun aplikasi adalah sebagai berikut:

Tabel 3.3 Kebutuhan Perangkat Keras

PC	Smartphone
Processor Intel Core i5	Hexa-core, 1800 MHz
RAM DDR3 4 GB	RAM 3 GB
Hardisk 1 TB	*Storage* 32 GB
Graphic Card 2 GB	GPU Adreno 510
Kamera 1.3 MP	Kamera 13 MP

3.1.4 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses aktivitas yang akan diterapkan dalam sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik dan sesuai kebutuhan. Berikut adalah kebutuhan sistem yang diperlukan:

1. Aplikasi dapat menampilkan objek 3D alat musik tradisional indonesia.
2. Aplikasi dapat mengakses kamera dan memindai *marker.*
3. Aplikasi dapat menampilkan objek 3D dan informasinya berdasarkan *marker* yang dipindai.
4. Aplikasi dapat memperdengarkan nada saat objek *virtual button* disentuh.
5. Aplikasi dapat membuat Objek 3D bisa digerakan kiri, kanan, atas, bawah, *zoom in*, dan *zoom out*.
6. Admin dapat meng*update marker* dan objek 3D.

3.2 Analisis Algoritma

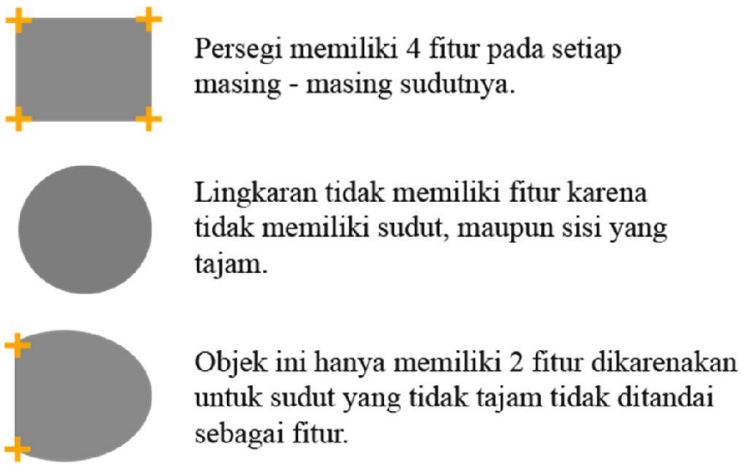
Analisis algoritma menjelaskan algoritma yang digunakan Vuforia dalam mendektesi *marker* berikut adalah alur dalam merubah gambar menjadi *marker.*

3.2.1 Algoritma FAST Corner Detection

FAST (*Features From Accelerated Segment Test*) adalah suatu jenis algoritma yang dibuat dengan tujuan untuk mempercepat waktu komputasi secara *real-time* dengan konsekuensi menurunkan tingkat akurasi pendeteksian sudut. FAST *corner detection* dimulai dengan menentukan suatu titik p pada koordinat (xp, yp) pada citra dan membandingkan intensitas titik p dengan 4 titik di sekitarnya. Titik pertama terletak pada koordinat (x, yp-3), titik kedua terletak pada koordinat (xp+3, y), titik ketiga terletak pada koordinat (x, yp+3), dan titik keempat terletak pada koordinat (xp-3, y).

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

Vuforia menggunakan algoritma FAST *Corner detection* untuk mendefinisikan seberapa baik gambar dapat dideteksi dan dilacak menggunakan Vuforia SDK. Rating *augmentable* dapat berkisar dari 0 sampai 5 untuk setiap gambar yang diberikan. Jika Sebuah rating menunjukkan 0 membuktikan bahwa target tidak terdeteksi oleh sistem *Augmented Reality*, sedangkan rating bintang 5 menunjukkan bahwa sebuah gambar dengan mudah dilacak oleh sistem *Augmented Reality*. Semakin tinggi rating *augmentable* dari target gambar, semakin kuat kemampuan deteksi dan pelacakan yang dikandungnya.

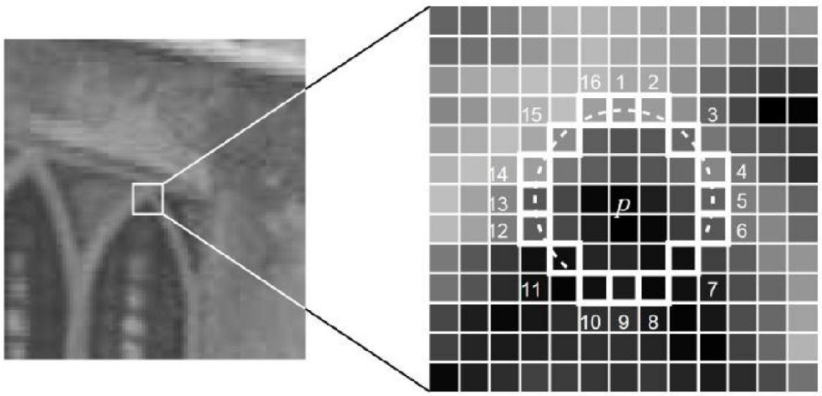


Gambar 3.1 Contoh Objek Yang Mengandung Fitur

Gambar 3.1 menunjukan bahwa algoritma hanya mendeteksi sudut pada gambar, jika gambar tidak memiliki sudut maka gambar tersebut tidak memiliki fitur dan gambar yang memiliki sudut yang tajam dan sudut yang tidak tajam maka hanya sudut yang tajam yang akan dijadikan fitur, yang nantinya sudut tesebut akan dijadikan fitur dalam menampilkan objek 3D.

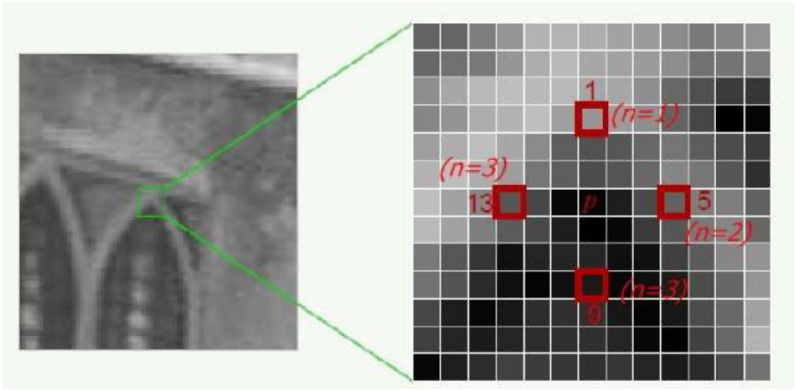
3.2.1.1 FAST corner detection bekerja pada suatu citra

1. Tentukan sebuah titik *p* pada citra dengan posisi awal (xp,yp) .



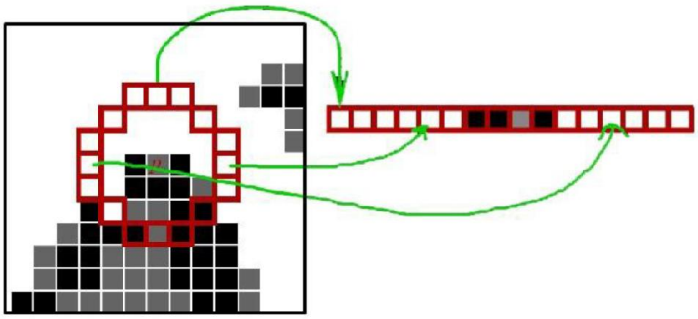
Gambar 3.2 Menampilkan Titik Awal yang Diuji

2. Tentukan keempat titik. Titik pertama (*n*=1) terletak pada koordinat (xp, yp+3), titik kedua (*n*=2) terletak pada koordinat (xp+3, yp), titik ketiga terletak pada koordinat (*n*=3) terletak pada koordinat (xp, yp-3), titik keempat (*n*=4) terletak pada koordinat (Xp- 3, yp).



Gambar 3.3 Keempat Titik Koordinat

3. Bandingkan intensitas titik pusat *p* dengan keempat titik disekitar. Jika terdapat paling sedikit 3 titik yang memenuhi syarat berikut, maka titik pusat *p* adalah titik sudut.
4. Untuk menentukan titik suatu sudut, seluruh piksel akan dibagi dengan tiga subset yaitu ; *Pixel dark , Pixel similiar* ,dan P *brighter.*
5. Ulangi proses sampai seluruh titik pada citra sudah dibandingkan intensitasnya.



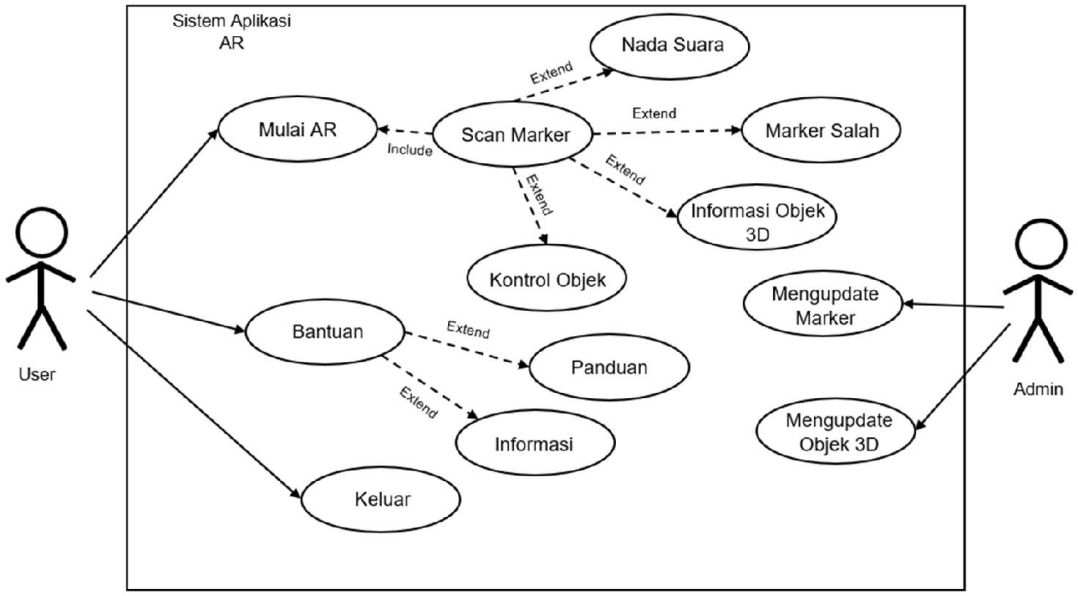
Gambar 3.4 Tiga Titik yang Memenuhi Syarat

3.3 Pemodelan Sistem

Pada analisis fungsional ini menggunakan pemodelan UML (*Unifiled Modeling Language*). Adapun diagram-diagram yang dirancang pada sistem aplikasi ini diantaranya:

3.3.1 Use Case Diagram

Merupakan deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah gambaran tentang bagaimana sistem tersebut digunakan. *Use Case* diagram terdiri dari tiga bagian yaitu definisi *actor,* definisi *use case* dan *scenario use case*. Berikut adalah *use case diagram* yang dirancang untuk aplikasi Android:



Gambar 3.5 Diagram Use Case Sistem

Gambar 3.5 ditunjukan bahwa dalam perancangan aplikasi terdapat sebelas *use case*. Pendeskripsian masing-masing *use case* ditunjukan pada tabel berikut:

Tabel 3.4 Deskripsi *Actor*

No	Aktor	Deskripsi
1.	*User*	Pengguna aplikasi AR, melakukan *scan marker*, melihat panduan aplikasi, melihat informasi aplikasi dan keluar dari aplikasi.
2	*Admin*	Dapat melakukan perubahan terhadap aplikasi, baik itu menambah, mengedit, ataupun menghapus bagian aplikasi.

Tabel 3.5 Deskripsi *Use Case*

No	*Use case*	Deskripsi
1	Halaman Utama	Tampilan utama aplikasi.
2	Mulai AR	Menu untuk memulai program AR.
3	Bantuan	Menu bantuan *user* untuk memulai aplikasi.
4	Panduan AR	Menampilkan cara menggunakan aplikasi.
5	Informasi Aplikasi	Menampilkan informasi tentang perancang aplikasi.
6	Keluar	Keluar dari aplikasi.
7	*Scan Marker*	Proses untuk melakukan scan marker dan menampilkan 3D *Animasi.*
8	*Marker* Salah	Merupakan proses *scan* terhadap *marker* secara berulang untuk menemukan *marker* yang benar.

9	Informasi Objek 3D	Menampilkan informasi objek 3D alat musik tradisional dengan tulisan atau suara.
10	Nada Suara	Suara nada alat musik yang keluar saat *user* menyentuh *virtual* button.
11	Kontrol Objek	Merupakan proses memperbesar/memperkecil dan merotasi objek 3D yang ditampilkan pada layar *display*.
12	*Update Marker*	Proses menambah, menghapus, ataupun mengubah *marker* pada sistem
13	*Update* Objek 3D	Proses menambah, menghapus, ataupun mengubah objek 3D pada sistem

Berikut merupakan skenario yang dilakukan *user* dalam menjalankan aplikasi serta reaksi sistem terhadap aksi *actor/user*.

Nama Use Case: *Halaman Utama*

Scenario:

Tabel 3.6 *Scenario* Halaman Utama

Aksi Actor	***Reaksi Sistem***
Skenario Normal	
1. User menunggu tampilan menu halaman utama	
	1. Aplikasi menampilkan Mulai AR, Bantuan, dan Keluar.

Nama Use Case : Mulai AR

Scenario :

Tabel 3.7 *Scenario* Mulai AR

Aksi Actor	***Reaksi Sistem***
Skenario Normal	
1. User memilih Mulai AR	
	1. Sistem mengaktifkan kamera, memulai scan marker.

Nama Use Case: Scan Marker

Scenario:

Tabel 3.8 *Scenario Scan Marker*

Aksi Actor	***Reaksi Sistem***
Skenario Normal	
1. User melakukan proses scan marker	
	1. Aplikasi menampilkan objek 3D alat musik tradisional dan Informasi alat musik tradisional akan muncul dengan tulisan saat tombol informasi objek 3D ditekan.
Skenario Lain	
1. User tidak melakukan proses scan marker dengan benar	

	1. Aplikasi akan melakukan proses scaning secara berulang.

Nama Use Case: Bantuan

Scenario:

Tabel 3.9 *Scenario* Bantuan

Aksi Actor	***Reaksi Sistem***
Skenario Normal	
1. User memilih Bantuan	
	1. Aplikasi menampilkan Panduan dan Informasi.

Nama Use Case: Panduan

Scenario :

Tabel 3.10 *Scenario* Panduan

Aksi Actor	***Reaksi Sistem***
Skenario Normal	
1. User memilih panduan	
	1. Aplikasi menampilkan panduan penggunaan aplikasi.

Nama Use Case: Informasi Aplikasi

Scenario:

Tabel 3.11 *Scenario* Informasi Aplikasi

Aksi Actor	***Reaksi Sistem***
Skenario Normal	
1. User memilih Informasi Aplikasi	
	1. Aplikasi menampilkan informasi tentang aplikasi.

Nama Use Case: Keluar

Scenario:

Tabel 3.12 *Scenario* Keluar

Aksi Actor	***Reaksi Sistem***
Skenario Normal	
1. User memilih Keluar	
	1. Aplikasi menghentikan semua proses dan selesai.

Nama Use Case: Informasi Objek 3D

Scenario:

Tabel 3.13 *Scenario* Informasi Objek 3D

Aksi Actor	***Reaksi Sistem***
Skenario Normal	
1. User menekan tombol informasi objek 3D tampil diatas marker	
	1. Aplikasi menampilkan informasi tentang objek 3D yang muncul atau tampil diatas marker.

Nama Use Case: Nada Suara

Scenario:

Tabel 3.14 *Scenario* Nada Suara

Aksi Actor	***Reaksi Sistem***
Skenario Normal	
1. User menyentuh tombol virtual.	
	1. Aplikasi mengeluarkan output suara nada alat musik melalui speaker.

Nama Use Case: Kontrol Objek

Scenario:

Tabel 3.15 *Scenario* Kontrol Objek

Aksi Actor	***Reaksi Sistem***
Skenario Normal	
1. User menekan tombol rotasi atas, bawah, kiri, dan kanan.	
	1. Aplikasi melakukan proses rotasi terhadap objek 3D sesuai arah tombol yang ditekan oleh user.
2. User menyentuh layar smartphone dengan 2 jari.	
	2. Proses memperbesar dan memperkecil objek 3D.

Nama Use Case: update marker

Scenario:

Tabel 3.16 *Scenario Update Marker*

Aksi Actor	***Reaksi Sistem***
Skenario Normal	
1. Admin mengupdate marker yang sudah ada pada database vuforia	
	1. Aplikasi akan mengupdate marker yang terbaru.

Nama Use Case: update objek 3D

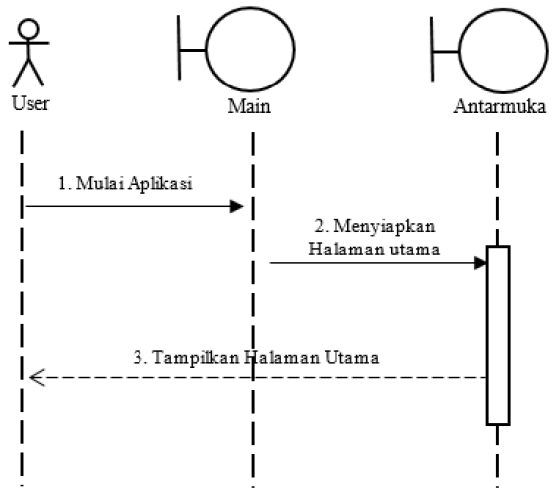
Scenario:

Tabel 3.17 *Scenario Update* Objek 3D

Aksi Actor	***Reaksi Sistem***
Skenario Normal	
1. Admin mengupdate objek 3D terbaru pada aplikasi.	
	1. Aplikasi akan mengupdate objek 3D yang terbaru.

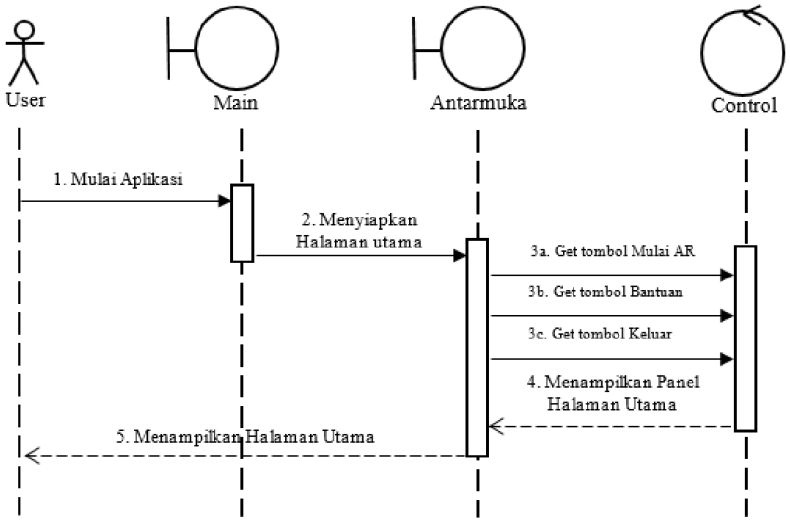
3.3.2 Sequence Diagram

Sequence diagram (diagram urutan) adalah suatu diagram yang memperlihatkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, display, dan sebagainya berupa pesan/message.



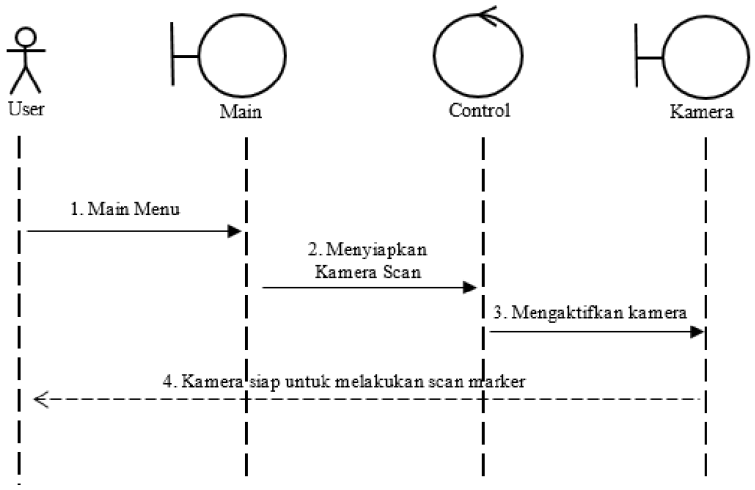
Gambar 3.6 Sequence Diagram Mulai Aplikasi

Gambar 3.6 menunjukan interaksi sistem ketika pengguna memilih mulai aplikasi, dimana sistem akan menampilkan *splash screen*, lalu sistem akan menampilkan halaman utama pada aplikasi.



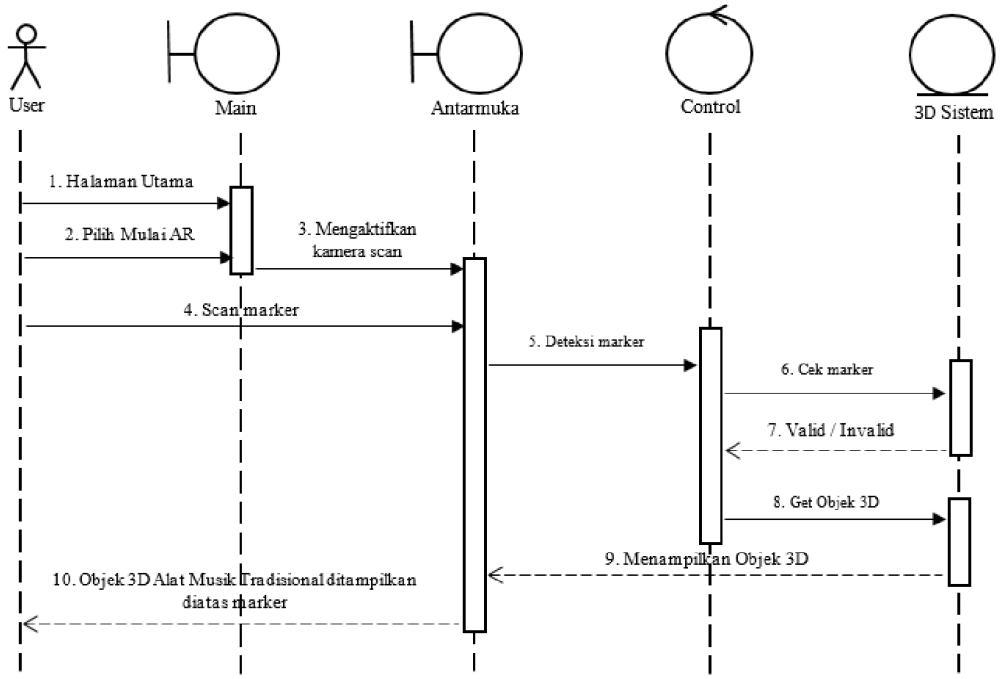
Gambar 3.7 Sequence Diagram Halaman Utama

Gambar 3.7 menunjukan interaksi sistem ketika pengguna mendapatkan halaman utama aplikasi, dimana sistem akan menampilkan panel halaman aplikasi, pada halaman aplikasi terdapat beberapa tombol yang dapat di akses oleh pengguna, dintaranya, tombol menu mulai AR, tombol menu bantuan, dan tombol keluar.



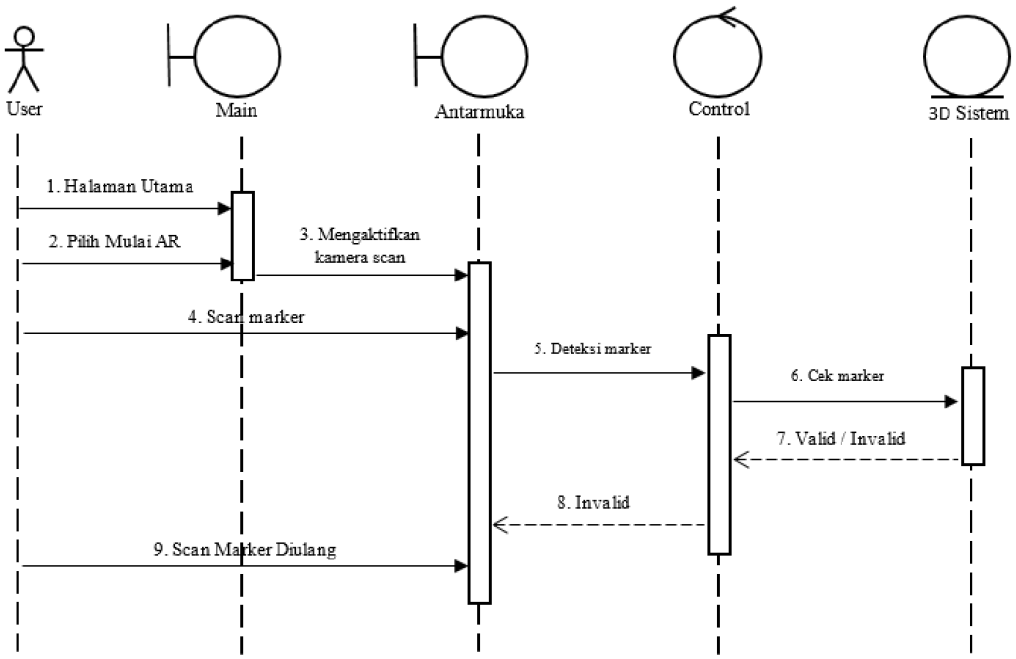
Gambar 3.8 Sequence Diagram Mulai AR

Gambar 3.8 menunjukkan interaksi sistem ketika user memilih Mulai AR dan sistem akan menyiapkan kamera lalu akan menampilkannya untuk melakukan *scan marker.*



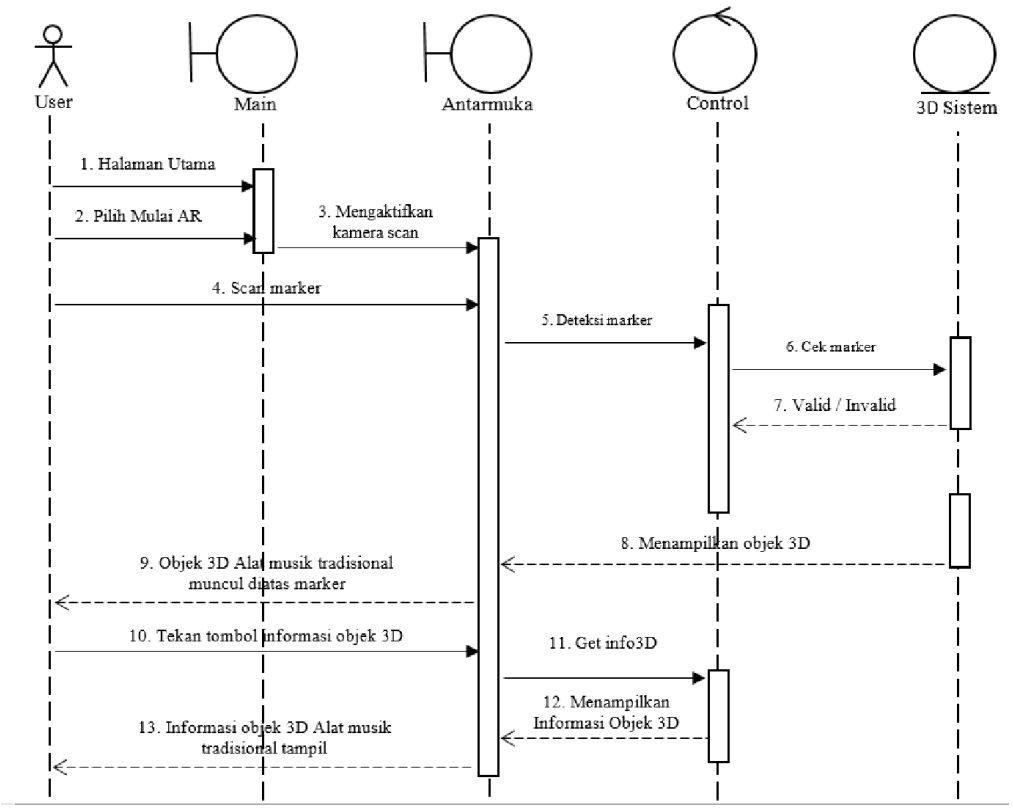
Gambar 3.9 Sequence Diagram Memulai Scan marker AR

Gambar 3.9 menunjukkan interaksi sistem ketika pengguna atau *user* melakukan *scan marker*, sistem akan mengatifkan kamera, lalu sistem akan melakukan *scan marker*, selanjutnya akan ada proses pendeteksian *marker* dan akan ada pengecekan sesuai atau tidaknya dengan *marker* yang terdapat dalam di *database 3D engine*. Apabila *marker* cocok maka sistem akan menampilkan objek 3D diatas *marker* yang di *scan* oleh pengguna.



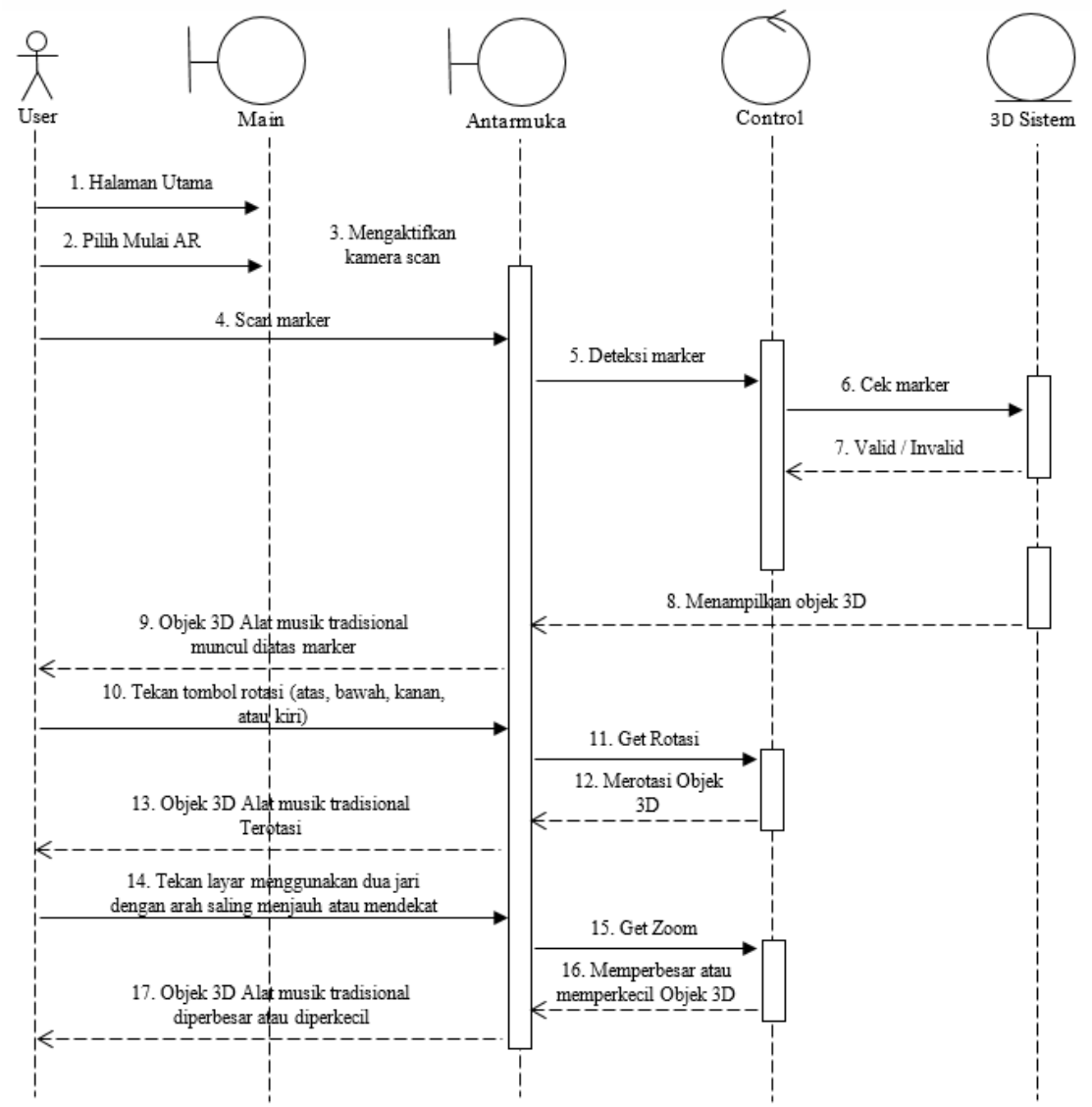
Gambar 3.10 Sequence Diagram Scan Marker Salah.

Gambar 3.10 menunjukan interaksi sistem dengan pengguna ketika pengguna melakukan *scan* AR salah, sistem akan mengaktifkan kamera, sistem akan melakukan proses *scan* AR, meminta *marker* kepada pengguna aplikasi, pengguna melakukan *scan marker* salah, sistem akan melakukan cek dan deteksi terhadap *marker,*apakah *marker* sesuai dengan *marker* yang tersimpan pada database sistem, *marker* tidak ditemukan, sistem meminta ulang pengguna untuk melakukan *scanmarker*, pengguna melakukan *scan marker* ulang.



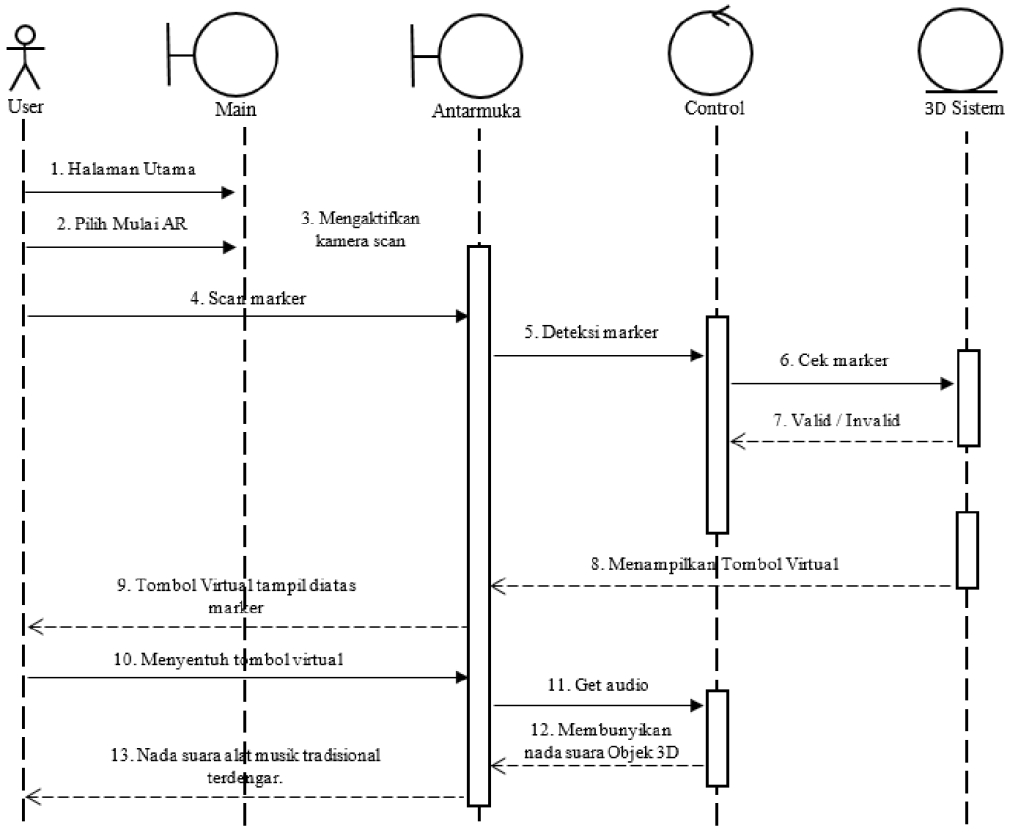
Gambar 3.11 Sequence Diagram Tombol Informasi Objek 3D

Gambar 3.11 menunjukan interaksi sistem dengan pengguna ketika pengguna menekan tombol informasi objek 3D, dimana tombol informasi ini akan muncul ketika pengguna melakukan proses *scan marker* dengan benar sehingga objek 3D akan muncul diatas *marker* bersamaan dengan tombol informasi objek 3D. Ketika pengguna menekan tombol tersebut, sistem akan menampilkan informasi tentang objek 3D yang muncul diatas *marker* baik itu berupa tulisan text ataupun suara, sehingga pengguna dapat mengetahui objek 3D apa yang muncul diatas *marker*.



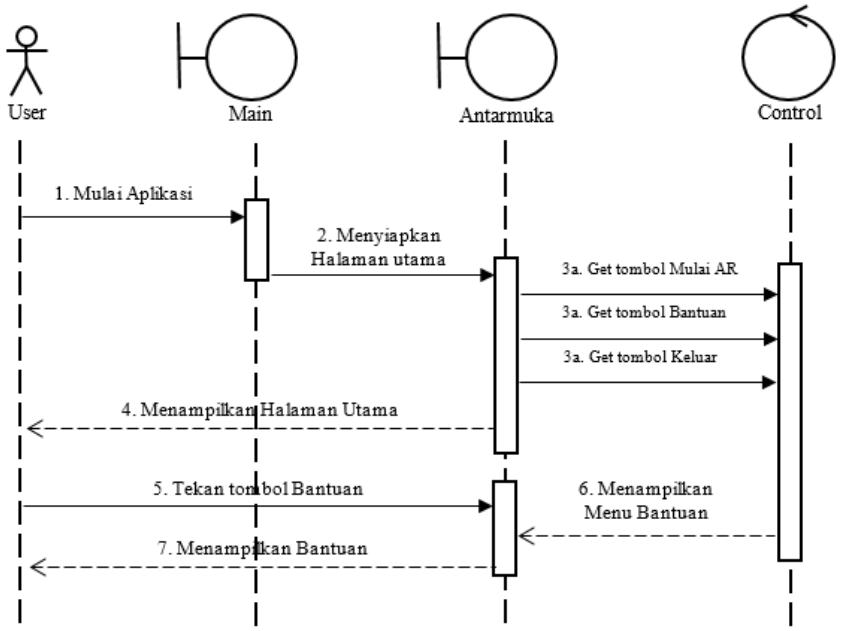
Gambar 3.12 Sequence Diagram Kontrol Objek

Gambar 3.12 menunjukan interaksi sistem dengan pengguna, ketika pengguna ingin mengontrol objek. Sistem akan merubah posisi , ukuran, dan merotasi objek berdasarkan permintaan *user.* Untuk mengontrol objek 3D, *user* harus menampilkan objek 3D diatas marker terlebih dahulu kemudian tombol untuk mengontrol objek baru akan keluar.



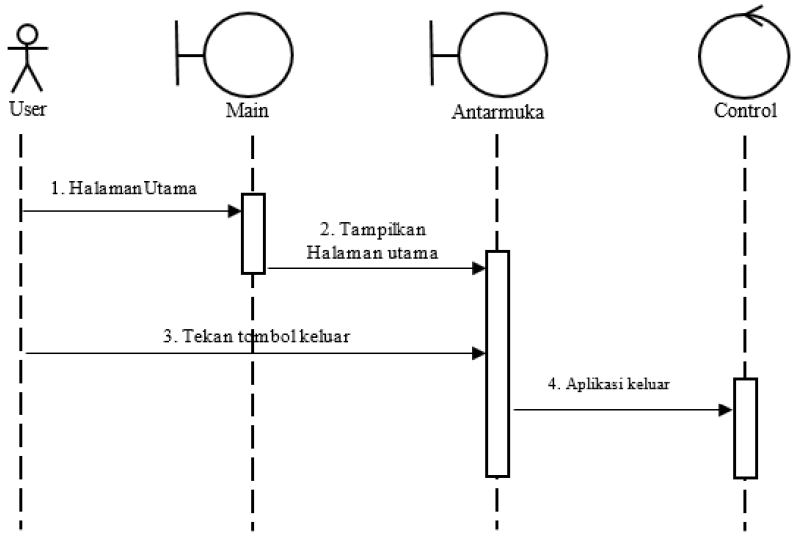
Gambar 3.13 Sequence Diagram Nada Suara

Gambar 3.13 menunjukan interaksi sistem dengan pengguna ketika pengguna menyentuh tombol *virtual*, sistem akan memperdengarkan nada suara alat musik tradisional kepada user. User harus menampilkan tombol *virtual* diatas marker dengan benar, bila tidak sempurna maka nada suara akan terdengar tidak jelas ataupun suara tidak akan muncul.



Gambar 3.14 Sequence Diagram Menu Bantuan

Gambar 3.14 menunjukan interaksi sistem dengan pengguna ketika pengguna menekan tombol bantuan, sistem akan menampilkan menu bantuan, yang di dalamnya terdapat tombol menu panduan bagaimana caranya melakukan *scan marker* AR dan juga terdapat tombol menu informasi tentang aplikasi.

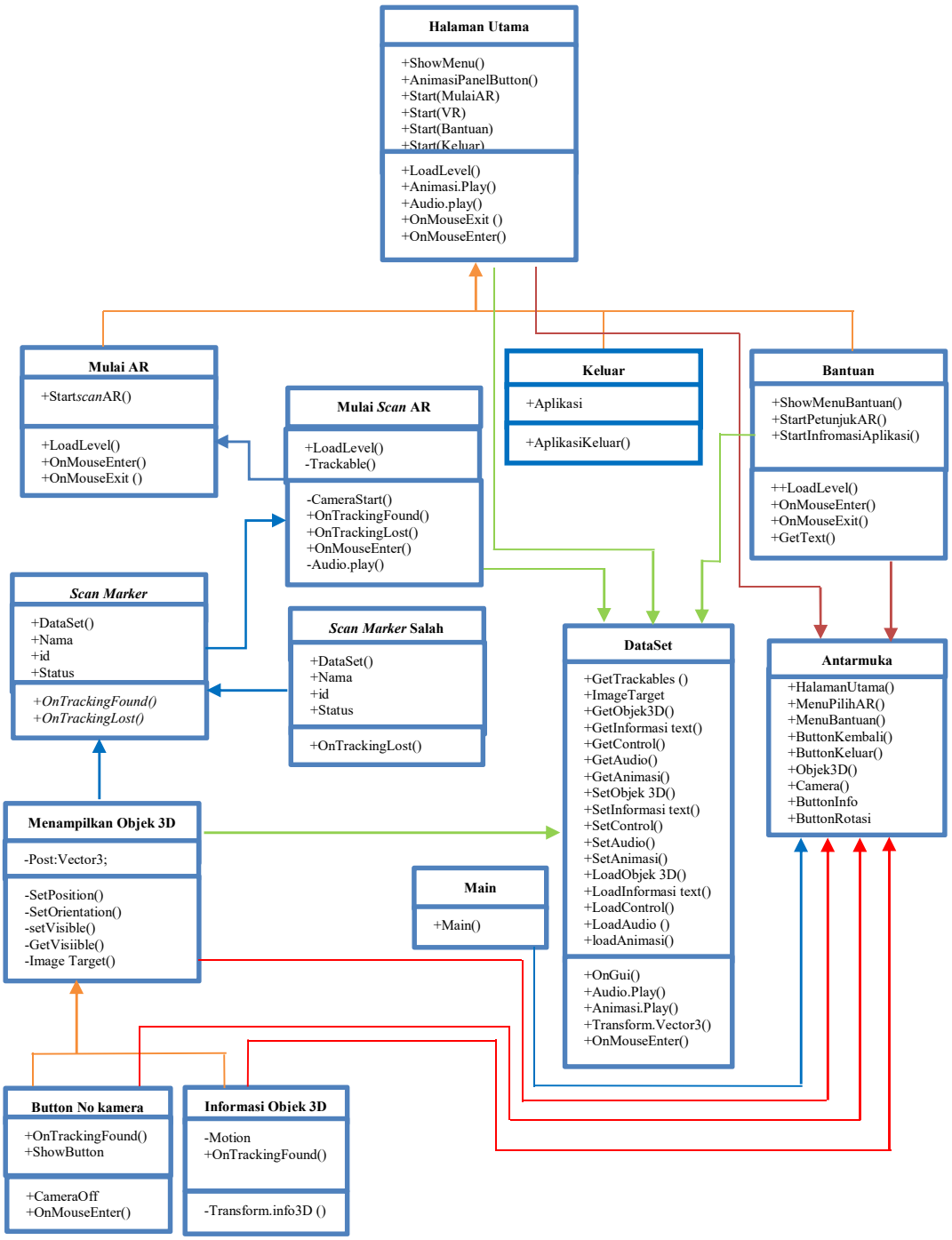


Gambar 3.15 Sequence Diagram Keluar

Gambar 3.15 menunjukan interaksi sistem dengan pengguna ketika pengguna menekan tombol keluar, sistem akan menutup aplikasi.

3.3.3 Class Diagram

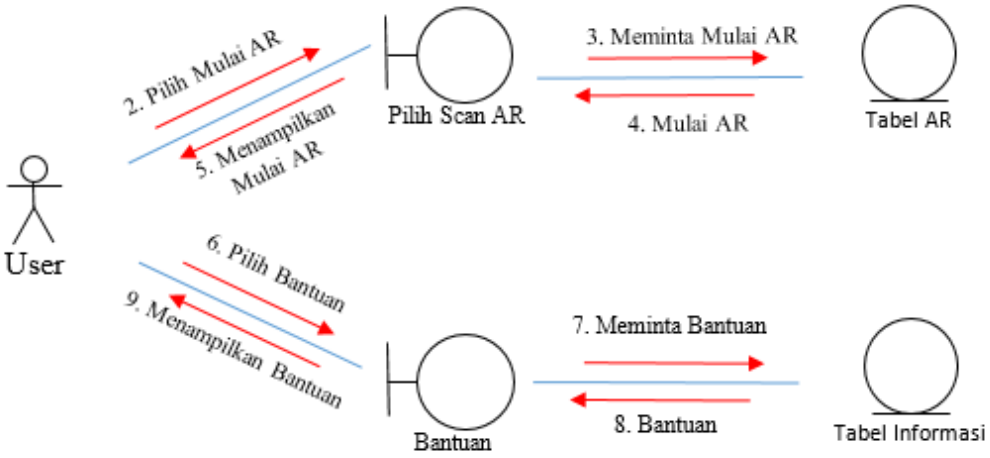
Gambar 3.13 merupakan diagram yang akan menampilkan subset dari kelas-kelas dan relasiny untuk mendapatkan gambaran lengkap terhadap sistem yang dibangun.



Gambar 3.16 Class Diagram

3.3.4 Collaboration Diagram

Menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus bersosialisasi dengan *use case*. *Sequence diagram*, memperlihatkan tahap demi tahap yang seharusnya terjadi untuk menghasilkan sesuatu dalam *use case*.



Gambar 3.17 Colaboration Diagram Sistem

3.4 Perancangan Sistem

Perancangan sistem dibuat untuk mempermudah pengerjaan aplikasi, dimana dalam perancangan sistem terdapat perancangan data, perancangan arsitektur dan perancangan *marker*. Berikut akan dibahas mengenai dua tahapan tersebut.

3.4.1 Perancangan Data

Dalam perancangan aplikasi, data yang akan diolah adalah merupakan gambar beserta informasi. Data ini memiliki peran yang sangat penting dalam pembuatan aplikasi ini, karena data ini merupakan hal yang seharusnya menjadi daya tarik bagi *user*. Dalam membuat data penulis menggunakan unity.

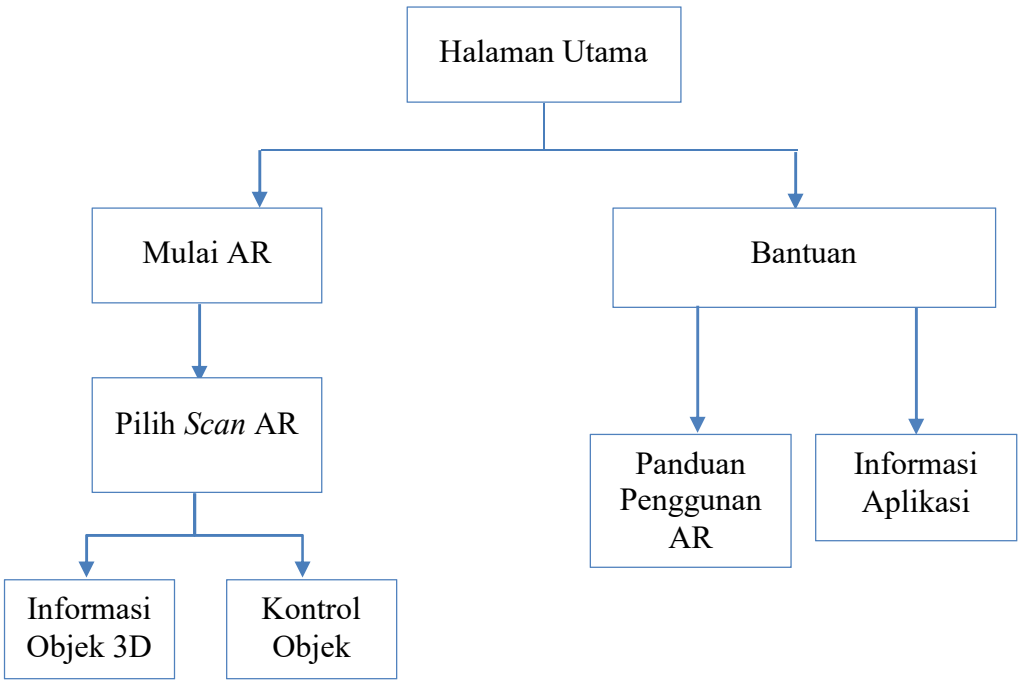
Berikut merupakan tabel perancangan data :

Tabel 3.18 Perancangan Data

Field Name	Type	Size	Description
Marker	*Img*	-	Gambar 2D yang digunakan sebagi *marker*.
Alat Musik Tradisional Indonesia	*.blend*	-	Berupa objek animasi 3D yang digunakan untuk pengenalan.
Informasi	*Text, Audio*	-	*Text* atau Audio yang digunakan sebagai informasi untuk pengenalan komponen.
Target ID	*Char*	255	ID yang digunakan untuk *scan marker.*
App_licenceykey	*Char*	255	Sebagai *lisensi* program agar AR pada dapat digunakan pada Unity.

3.4.2 Perancagan Arsitektur

Proses perancangan arsitektur berhubungan dengan penetapan kerangka kerja struktur dasar untuk suatu sistem, hal ini dapat diilustrasikan melalui struktur chart.

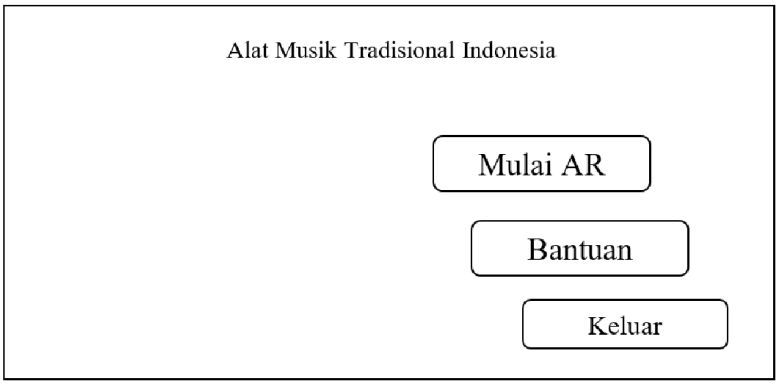


Gambar 3.18 Stuktur Chart Aplikasi

Gambar 3.18 menunjukan struktur chart diagnosa pada aplikasi yang akan dibangun. Fungsi dari Struktur Chart digunakan untuk mendefinisikan dan mengilustrasikan sistem secara berjenjang dalam bentuk modul dan sub modul.

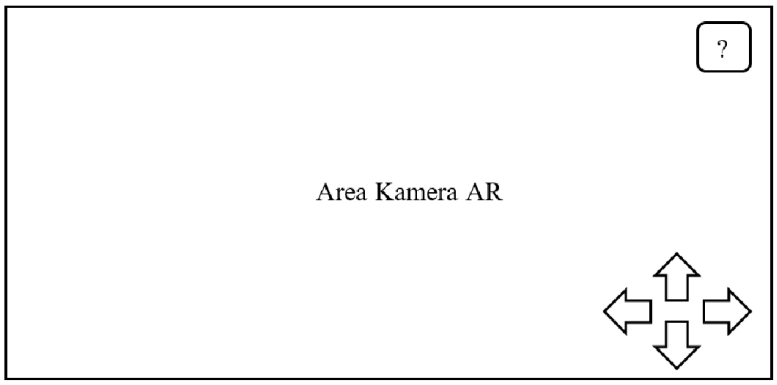
3.4.3 Perancangan Antarmuka Sistem

Aplikasi pengenalan alat musik tradisional indonesia ini merupakan perangkat lunak yang berfungsi untuk menampilkan informasi dengan cara yang efektif, serta membantu pengguna untuk memahami informasi yang ditampilkan serta memperkaya tampilan informasi bagi pengguna. Oleh karena itu penulis merancang antarmuka untuk mencapai hal tersebut.



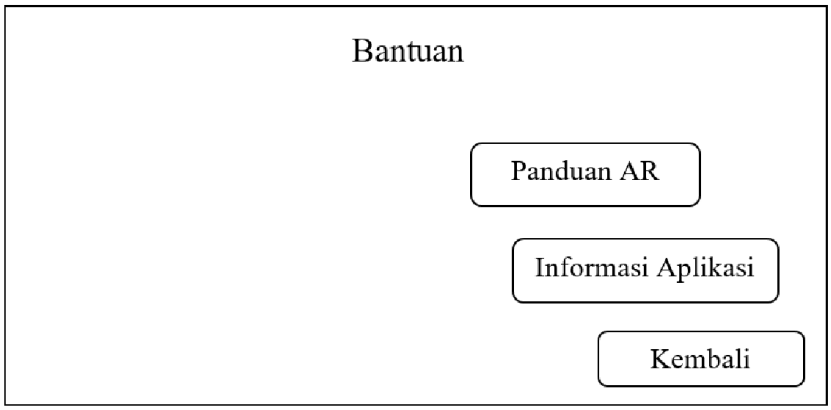
Gambar 3.19 Perancangan Antarmuka Halaman Utama

Pada gambar 3.19 memperlihatkan perancangan tampilan halaman dari aplikasi. Disini pengguna dapat memilih 3 pilihan tombol pada layar halaman utama.



Gambar 3.20 Perancangan Antarmuka Mulai AR

Pada gambar 3.20 ditunjukan tampilan Mulai *Augmented Reality* pengenalan alat musik tradisional Indonesia. Dimana pada menu ini kamera belakang akan aktif, lalu pada bagian kanan atas terdapat satu tombol '?', saat pengguna menekan tombol tersebut maka objek 3D yang muncul diatas *marker* akan ditampilkan informasinya. Dibawah tombol informasi tersebut terdapat tombol arah untuk merotasi objek 3D yang muncul diatas marker. Bila pengguna menekan layar dengan 2 jari kemudian jari itu diarahkan saling menjauh atau mendekat maka objek 3D yang muncul di layar akan diperbesar atau diperkecil.



Gambar 3.21 Perancangan Antarmuka Menu Bantuan

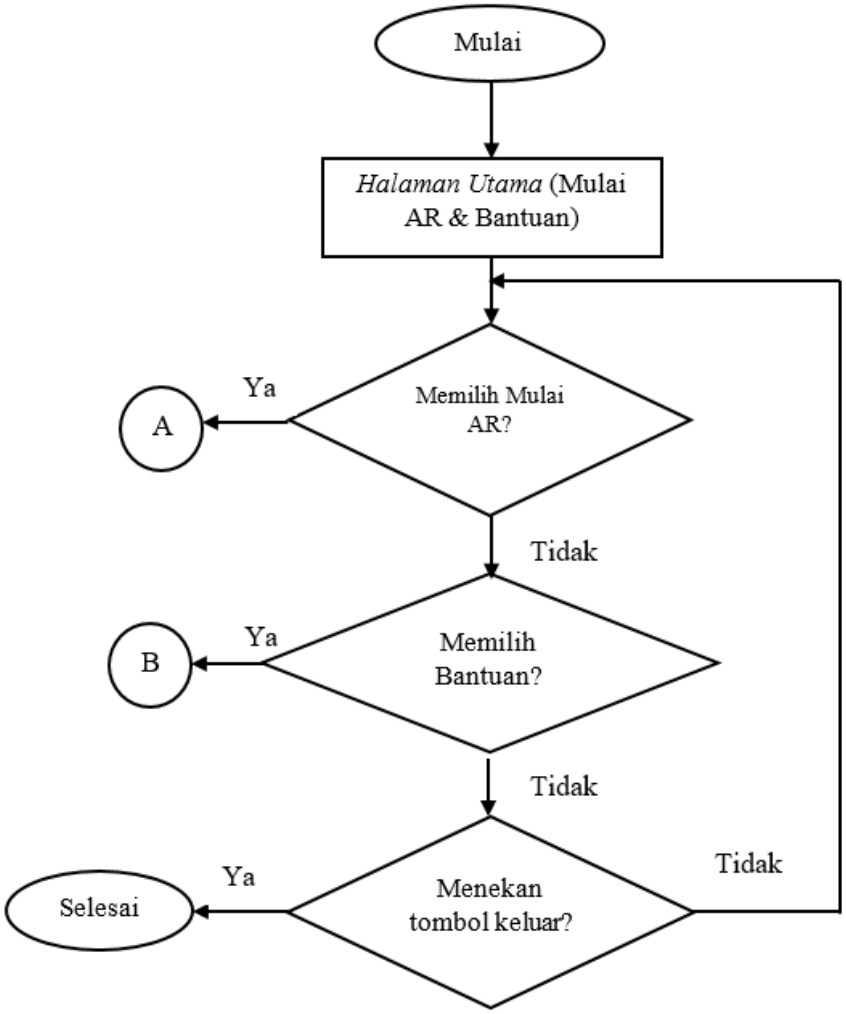
Pada gambar 3.21 terdapat perancangan tampilan menu bantuan, dimana terdapat 2 pilihan menu yang diperuntukan untuk membantu pengguna dalam menjalankan aplikasi. Panduan AR yaitu berisi tentang petunjuk menggunakan aplikasi AR seperti *scan marker*, sedangkan informasi aplikasi yaitu tentang versi aplikasi dan informasi dari pembuat.

3.4.4 Perancangan Prosedural

Perancangan prosedural merupakan perancangan yang dibuat untuk menetapkan algoritma yang akan diterapkan di dalam sistem. Perancangan prosedural sendiri ditunjukan dalam bentuk *flowchart diagram*,

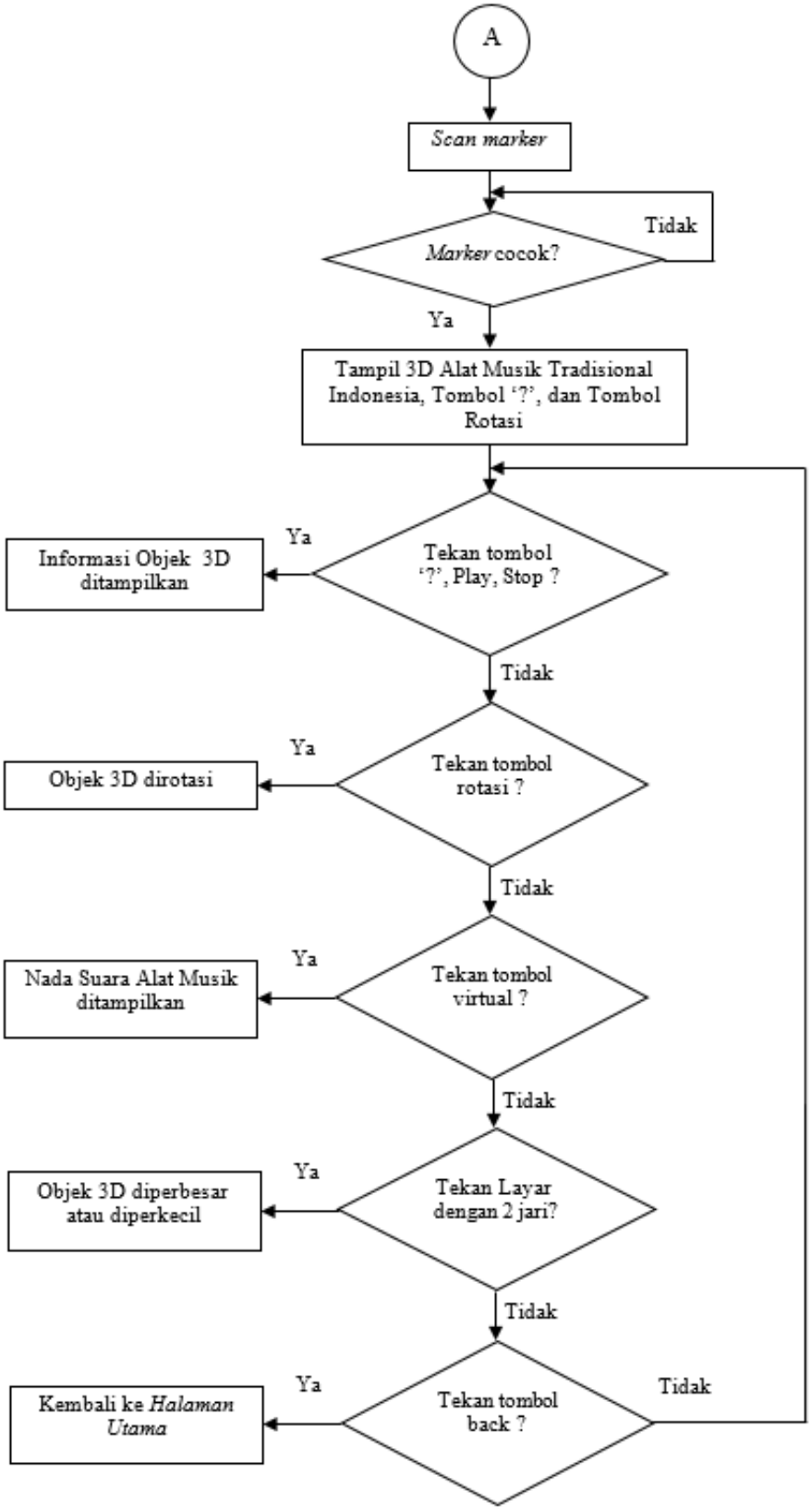
- Struktur data yang terbentuk pada perancangan data
- Struktur modul kendali pada struktur *chart* hasil perancangan arsitektur
- Struktur, rancangan menu dan format layar hasil rancangan antarmuka

Alat bantu yang digunakan dalam perancangan prosedural adalah *flowchart* dan algoritma atau *pseudocode* atau program *design language*.



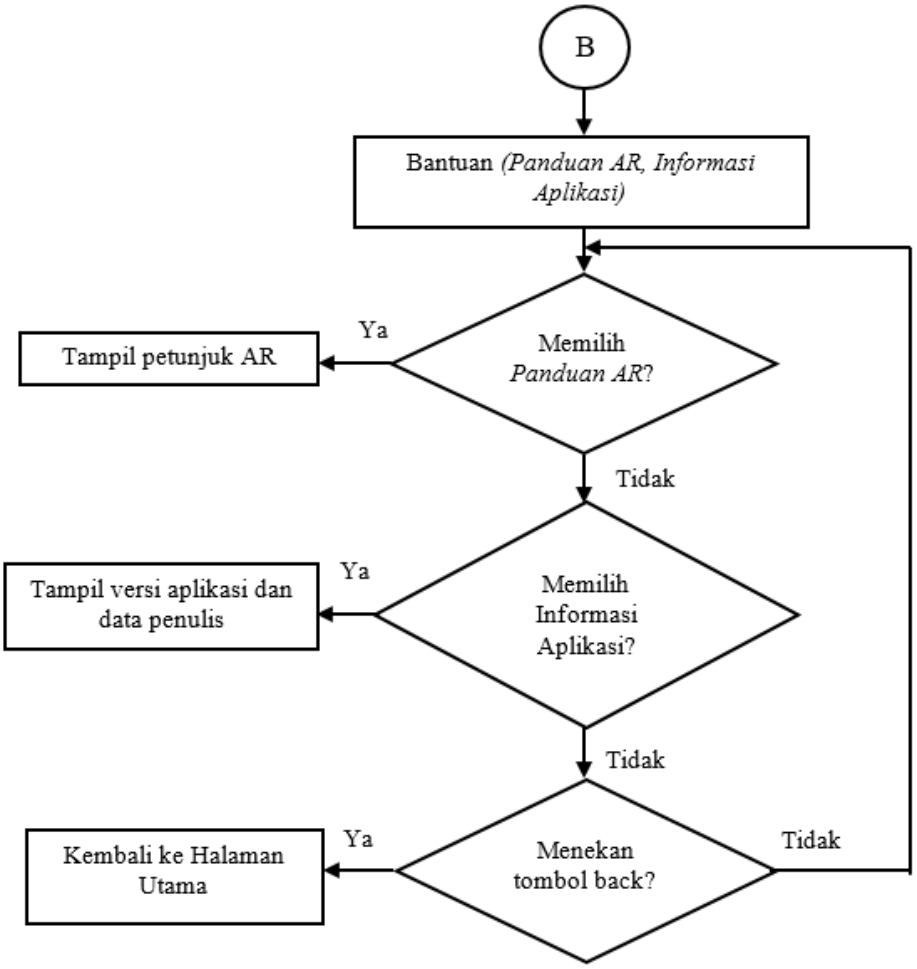
Gambar 3.22 Flowchart Halaman Utama

Pada gambar 3.22 menunjukan diagram alur yang dilakukan pengguna saat menggunakan aplikasi pertama kali. Ketika memulai aplikasi pengguna memungkinkan memilih 3 pilihan menu yang tersedia.



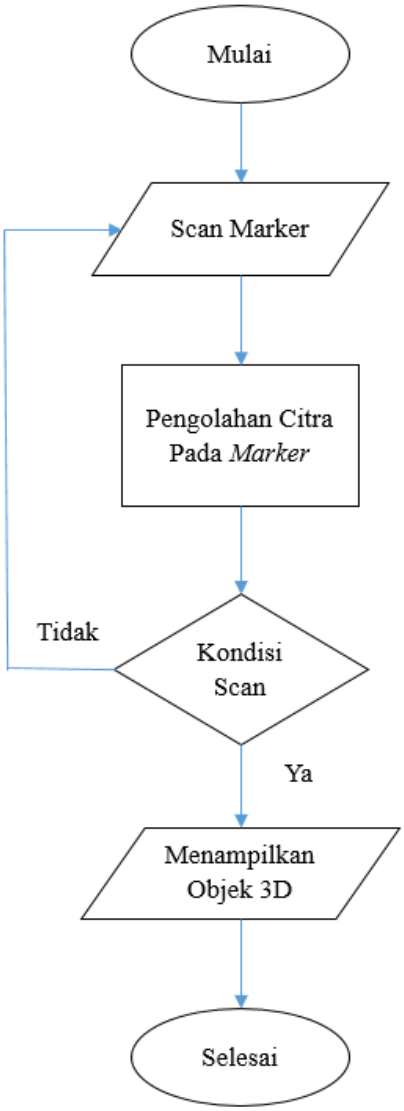
Gambar 3.23 *Flowchart Mulai AR*

Pada gambar 3.23 menunjukan diagram alur Mulai AR, dimana pada menu ini aplikasi akan otomatis mengaktifkan kamera belakang *smartphone,* lalu pengguna diharuskan untuk mengarahkan kamera pada *marker.* Setelah itu pengguna dapat memilih beberapa opsi seperti '?'(Informasi Objek 3D), rotasi, tombol *virtual* dan perbesar atau perkecil objek 3D.



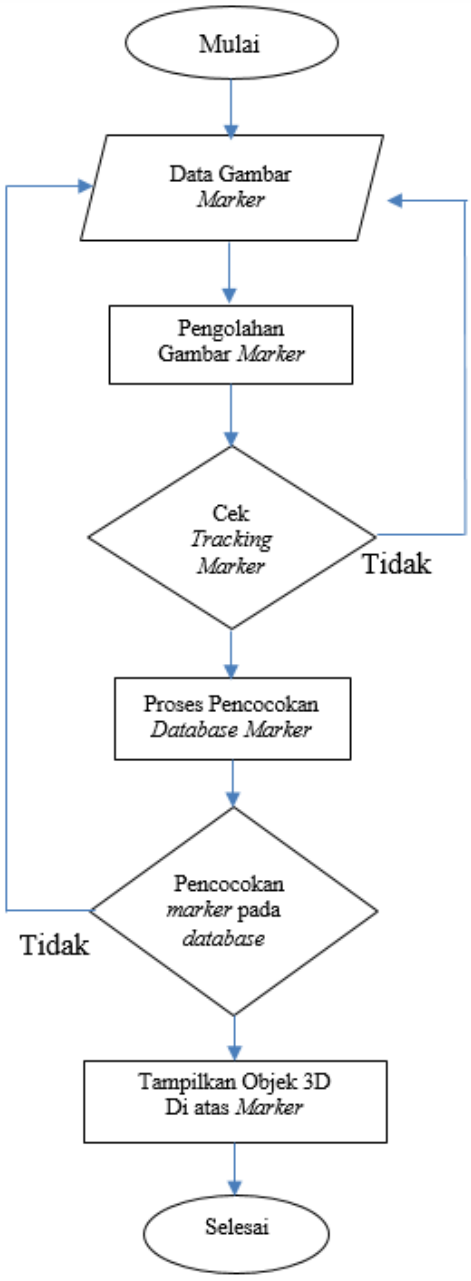
Gambar 3.24 Flowchart Bantuan

Pada gambar 3.24 menunjukan diagram alur petunjuk penggunaan, dimana pada menu ini berisi bantuan untuk pengguna yang memakai aplikasi ini, pada menu ini pengguna dimungkinkan memilih 2 pilihan menu. Pada Panduan *AR,* pengguna akan diberikan informasi mengenai cara menggunakan AR dan *marker* yang dapat dipakai. Pada informasi aplikasi akan menampilkan informasi versi aplikasi dan informasi tentang penulis.



Gambar 3.25 *Flowchart Sistem*

Gambar 3.25 Menjelaskan tentang sistem *scan marker*, pada kondisi ini kamera *scan marker* akan aktif untuk melakukan *scan* terhadap *marker*, jika *marker* terdeteksi maka akan menampilkan objek 3D di atas *marker*, jika *marker* tidak terdeteksi sistem akan meminta pengguna melakukan *scan marker* ulang.



Gambar 3.26 Flowchart Proses Tracking Marker

Gambar 3.26 Menjelaskan tentang proses *tracking marker*, pada kondisi ini sistem akan mendeteksi titik- titik pada *marker*, jika *marker* terdeteksi maka akan dilakukan pencocokan *marker* pada *database*, diakhiri dengan penempatan objek 3D di atas *marker*.