BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem bertujuan untuk memahami sistem yang akan dibangun. Adapun tahapan yang harus dilakukan adalah dengan melakukan beberapa analisis. Pembahasan berikut merupakan analisis masalah, analisis sistem yang sedang berjalan, analisis aplikasi sejenis, analisis Augmented Reality, analisis kebutuhan non-fungsional dan analisis kebutuhan fungsional. Dari hasil analisis tersebut dapat dirancang atau diperbaiki menjadi sebuah sistem yang lebih efektif dan efesien.

3.1.1 Analisis Masalah

Analisis masalah merupakan langkah awal dari suatu analisis sistem. Langkah ini diperlukan karena untuk mengetahui permasalahan apa saja yang terjadi di dalam sistem yang sedang berjalan. Setelah melakukan penelitian dan wawancara dengan Bapak Iwan Setiawan selaku staf PGA *(Personalia General Affair)* di *dealer* Toyota Auto 2000 Bandung maka dapat disimpulkan bahwa sulitnya konsumen dalam mendapatkan informasi seputar mobil dikarenakan kurangnya sarana dan media yang ada.

Dari permasalahan tersebut, maka perlu diterapkan sebuah aplikasi yang dapat menampilkan gambar 3 dimensi secara detail yang memudahkan konsumen dalam mendapatkan informasi seputar mobil. Aplikasi ini dibangun pada *platform* Android dengan teknologi Augmented Reality yang menggunakan *markerless* pada sebuah brosur sebagai objek yang akan dideteksi oleh kamera

3.1.2 Analisis Sistem Yang Sedang Berjalan

Prosedur yang sedang berjalan di *dealer* Toyota Auto 2000 Bandung dalam memberikan informasi seputar mobil adalah dengan melalui sales dan brosur. Konsumen dapat langsung bertanya kepada sales ataupun konsumen dapat melihat

melalui brosur yang ada. Alur sistem yang sedang berjalan dapat dilihat pada gambar 3.1.



Gambar 3.1 Alur Sistem Yang Sedang Berjalan

Penjelasan dari gambar 3.1 dapat dilihat sebagai berikut :

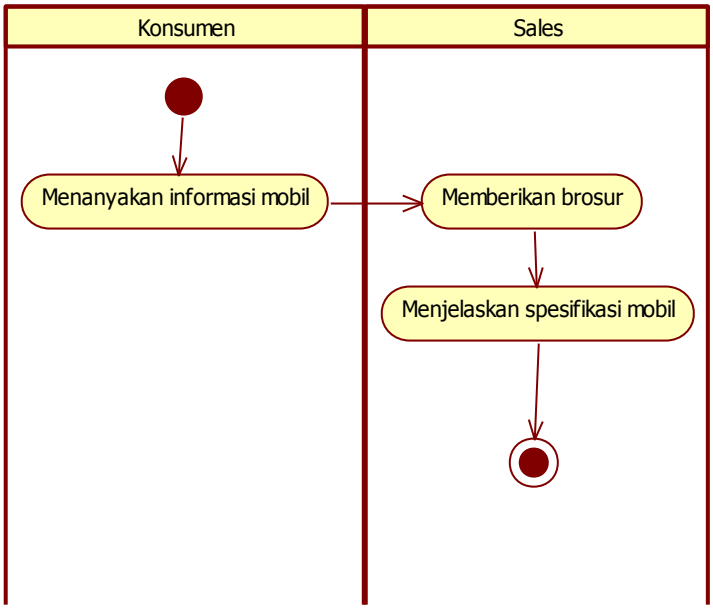
1. Konsumen

   Konseumen berperan sebagai orang yang memerlukan informasi seputar mobil yang ada di *dealer*. Konsumen akan meminta brosur kepada sales.

2. Sales

   Sales menjadi media yang memberikan informasi spesifikasi mobil secara detail melalui brosur yang tersedia kepada konsumen.

Aktifitas sistem yang berjalan yang dilakukan oleh konsumen dan sales dapat digambarkan dengan Activity diagram. Gambaran sistem yang sedang berjalan dapat dilihat pada gambar 3.2.



Gambar 3.2 Sistem Yang Sedang Berjalan

3.1.3 Analisis Aplikasi Sejenis

Di dalam tahapan analisis ini bertujuan untuk mengetahui kelebihan dan kekurangan aplikasi sehingga dapat diambil kesimpulan agar aplikasi yang akan dibangun bisa lebih efektif dan efesien. Ada banyak aplikasi AR yang digunakan sebagai media promosi untuk suatu produk yang dijual. Aplikasi yang digunakan untuk dijadikan sebagai bahan referensi adalah aplikasi AR Mercedes-Benz C-Class. Yaitu aplikasi virtual aksesori mobil yang memungkinkan pengguna dapat menambah aksesori mobil dengan sentuhan pada layar *smartphone* atau *tablet* dengan *platform* Android. Adapun tampilan dari aplikasi AR Mercedes-Benz C-Class dapat dilihat pada gambar 3.3.



Gambar 3.3 Tampilan Aplikasi AR Mercedes-Benz C- Class

Adapun komponen yang terdapat pada aplikasi AR Mercedes-Benz C-Class adalah sebagai berikut :P

1. Objek 3 dimensi

   Merupakan hasil akhir yang menampilkan objek 3 dimensi secara detail pada bagian *exterior* salah satu tipe mobil Mercedes Benz.

2. Virtual Button

   Terdapat beberapa menu *virtual button* untuk menambah aksesori tambahan pada beberapa bagian *exterior* mobil.

3. Markerless

   Aplikasi yang dibangun menggunakan metode *markerless* dengan media kertas dan gambar berwarna sebagai target pendeteksian marker.

4. Berbasis Mobile

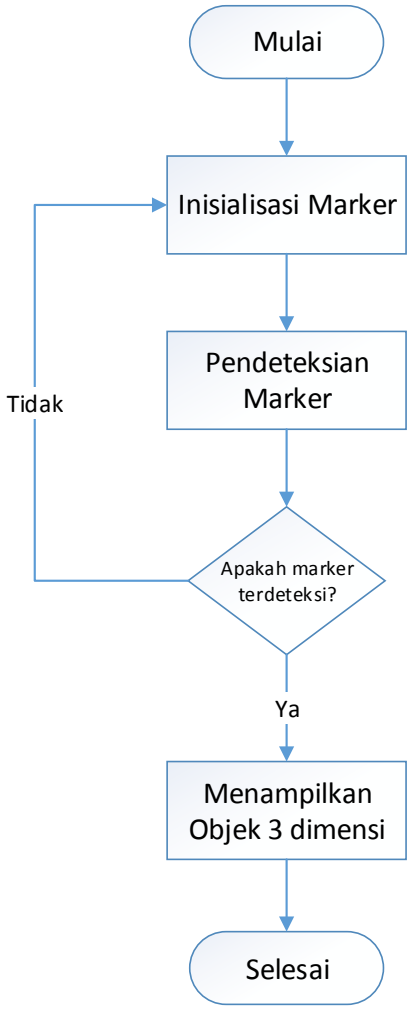
   Aplikasi Mercedes-Benz C-Class berjalan pada *platform* Android.

Kesimpulan aplikasi ini adalah berbasis *mobile* dengan *platform* Android yang menggunakan metode *markerless*, aplikasi ini memiliki menu berupa *virtual button* yang dapat memudahkan pengguna dalam menyimulasikan aksesori mobil. Namun terdapat kekurangan karena aplikasi ini tidak memiliki fitur petunjuk penggunaan. Selain itu, aplikasi ini juga belum bisa melakukan rotasi pada objek yang ditampilkan. Sehingga tentunya menjadi salah satu kekurangan yang dapat dianalisis agar lebih baik lagi.

3.1.4 Analisis Augmented Reality

Dalam Augmented Reality, marker merupakan salah satu komponen yang penting. Penentuan jenis marker berpengaruh kepada cara kerja program Augmented Reality untuk membaca marker dan menampilkan objek 3D. Marker yang digunakan adalah marker dengan nilai *rate* bintang yang cukup dan tidak 0.

Analisis Augmented Reality ini merupakan analisis yang mendeskripsikan bagaimana proses Augmented Reality dari awal inisialisasi, *tracking marker*, sampai dengan proses menampilkan objek 3D dengan metode *markerless*. Alur proses dari aplikasi yang akan dibangun dapat dilihat pada gambar 3.4.



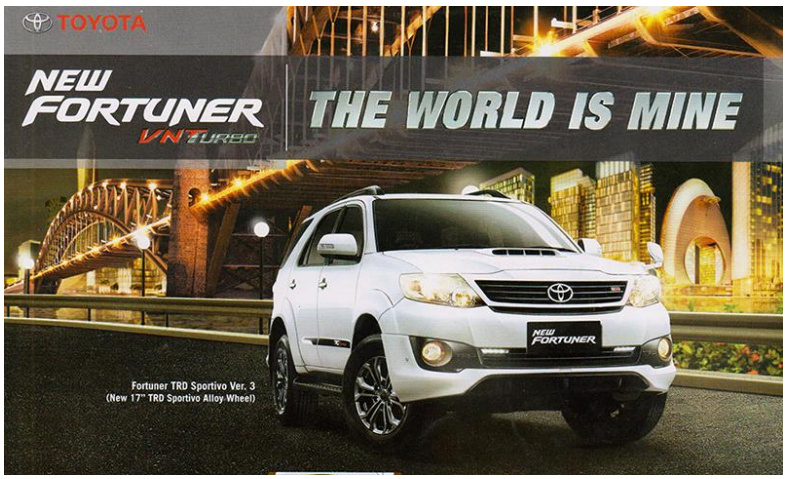
Gambar 3.4 Alur Proses Aplikasi

Proses pengenalan deteksi *markerless* mempunyai beberapa tahapan. Secara garis besarnya ada tiga bagian utama yaitu :

1. Inisialisasi Marker
2. Deteksi Marker
3. Menampilkan objek 3D

3.1.4.1 Inisialisasi Marker

Pada tahap ini aplikasi AR akan melakukan inisialisasi *windows* / layar berupa warna transparan. Setelah itu aplikasi akan dihubungkan dengan tampilan kamera untuk memulai proses pendeteksian marker. Kemudian aplikasi akan melakukan inisialisasi variabel awal yang dipakai untuk menampung sementara marker, status pengecekan, serta variabel informasi yang akan ditampilkan. Contoh marker yang digunakan dapat dilihat pada gambar 3.5.



Gambar 3.5 Contoh Marker

Pada gambar 3.4 gambar marker memiliki kontras dan ketajaman gambar yang sangat tinggi sehingga nilai rate bintang yang didapat bernilai 5 pada *image target* di situs www.vuforia.com. Tentunya marker yang dibuat dengan gambar ini akan lebih mudah untuk dideteksi oleh sistem AR.

3.1.4.2 Deteksi Marker

NFT *(Natural feature Tracking)* merupakan metode yang digunakan di dalam *library* Vuforia Qualcomm untuk mendeteksi pola gambar. Sebelum

melakukan pendeteksian fitur-fitur yang terdapat pada gambar, terlebih dahulu dilakukan proses identifikasi marker. Identifikasi marker adalah proses untuk menangkap gambar yang ditangkap oleh kamera. Gambar yang diperoleh itu dianggap marker terlepas ada atau tidaknya marker yang sesungguhnya pada gambar. Setelah marker diperoleh maka proses selanjutnya adalah mengubah marker menjadi keabu-abuan atau proses *grayscale*. Proses ini dilakukan untuk memudahkan dalam pendeteksian fitur pada marker. Marker yang sudah diubah menjadi keabu-abuan kemudian dideteksi fiturnya menggunakan metode NFT. Adapun metode yang digunakan di dalam NFT itu sendiri adalah SIFT dan FERNS. Contoh proses deteksi fitur dapat dilihat pada gambar 3.6.



Gambar 3.6 Proses Deteksi Fitur Lokal

Adapun komponen-komponen dalam proses deteksi fitur adalah sebagai berikut :

1. *Trackable* Tipe
   a. UNKNOWN_TYPE : Pelacakan yang tidak diketahui
   b. IMAGE_TARGET : Pelacakan berdasarkan gambar
   c. MULTI_TARGET : Pelacakan gabungan
   d. MARKER : Pelacakan marker
2. *Trackable* Nama

   Sebuah kalimat yang unik yang digunakan untuk mengidentifikasi pelacakan dari database. Untuk penulisan nama hanya diperbolehkan maksimal 64 karakter dan hanya mengandung karakter (a-z, A-Z, 0-9, [-_.]).

3. *Trackable* Status

   Dalam pelacakan memiliki informasi status yang terkait dengannya, setiap informasi satus dilacak akan diperbaharui sebagai *frame* kamera diproses. Status yang berbeda dilacak dapat menjadi salah satu dibawah ini :

   a. UNKNOWN : tempat atau lokasi pelacakan tidak diketahui biasanya dikembalikan sebelum tracker initialization
   b. UNDEFINE : tempat atau lokasi pelacakan tidak didefinisikan
   c. NOT_FOUND : tempat atau lokasi tidak ditemukan pada database yang dituju
   d. DETECTED : tempat atau lokasi pelacakan diseteksi dalam frame
   e. TRACKED : pelacakan telah terlacak dalam frame

Adapun beberapa fungsi yang digunakan pada vuforia, adalah sebagai sebagai berikut :

CameraDevice

Fungsi class ini untuk menyediakan akses ke kamera dan properti.

QCARBehaviour

Merupakan class untuk menangani pelacakan dan memicu asli *background rendering* video pada kamera.

DataSetLoadBehaviour

Fungsi ini memungkinkan untuk secara otomatis memuat dan mengaktifkan satu atau lebih data set ketika startup pada *image target*.

TrackbleBehaviour.Trackable [get]

Pelacakan pada saat runtime.

TrackableBehaviour.RegisterTrackableEventHandler

Berfungsi sebagai register baru `Tracker Event Handler` pada *tracker*. Penanganan ini dilakukan segera setelah semua Trackables telah diperbarui.

TrackableBehaviour.UnregisterTrackableEventHandler

Berfungsi sebagai unregisters sebuah `Tracker Event Handler`. Returns “false” jika event handler tidak ada.

Status TrackableBehaviour.CurrentStatus [get]

Status pelacakan dari `TrackableBehaviour`.

```
String TrackableBehaviour.TrackableName [get]
```

Pelacakan nama yang ada pada `TrackableBehaviour`.

```
void TrackableBehaviour.OnTrackerUpdate (Status newStatus)
[inline, virtual]
```

Dipicu oleh **`TrackerBehaviour`** setelah itu diperbarui.

```
DevaultTrackableEventHandler.OnTrackableStateChanged
```

Sebuah *handler custom* yang mengimplementasikan `ITrackableEventHandler` antarmuka.

```
DevaultInitializationErrorHandler
```

Fungsi untuk menginisialisasi error pada **`DevaultTrackableEventHandler`**

3.1.4.3 Menampilkan Objek 3D

Hasil pencocokkan pola yang dilakukan sebelumnya akan berpengaruh pada objek 3D yang dihasilkan. Jika tidak ada yang sesuai maka tidak akan menampilkan objek apapun. Jika marker sesuai dengan marker yang ada maka aplikasi akan menampilkan objek 3D tepat diatas marker atau dapat dilihat pada Gambar 3.7.



Gambar 3.7 Objek 3D Yang Ditampilkan Tepat Di Atas Marker

3.1.5 Spesifikasi Kebutuhan Perangkat Lunak

Pada tahap ini, menjelaskan analisis spesifikasi dari kebutuhan perangkat lunak yang akan dibangun yaitu meliputi spesifikasi kebutuhan perangkat lunak fungsional (SKPL-F) dan spesifikasi kebutuhan perangkat lunak non-fungsional (SKPL-NF).

1. Spesifikasi Kebutuhan Fungsional

   Spesifikasi kebuthan funsional berisi mengenai fungsi atau layanan apa yang harus disediakan atau bias dilakukan oleh sistem, bagaimana sistem berperilaku terhadap suatu kondisi tertentu, dan apa yang harus dilakukan oleh sistem jika terdapat suatu inputan tertentu. Detail dari spesifikasi kebutuhan fungsional dapat dilihat pada Tabel 3.1.

Tabel 3.1 Spesifikasi Kebutuhan Fungsional

Nomor	Spesifikasi Kebutuhan Fungsional
SKPL-F-001	Sistem harus dapat membaca marker
SKPL-F-002	Sistem harus dapat mendeteksi kamera
SKPL-F-003	Sistem harus dapat mendeteksi marker
SKPL-F-004	Sistem harus dapat menampilkan objek 3D
SKPL-F-005	Sistem harus dapat menampilkan menu pilihan petunjuk pengguna
SKPL-F-006	Sistem harus dapat menampilkan menu pilihan tentang aplikasi

2. Spesifikasi Kebutuhan Non-Fungsional

   Spesifikasi kebutuhan non-fungsional merupakan penjelasan mengenai batasan-batasan fungsi dari sistem yang akan dibangun. Detail mengenai spesifikasi kebutuhan non-fungsional dapat dilihat pada Tabel 3.2.

Tabel 3.2 Spesifikasi Kebutuhan Non-Fungsional

Nomor	Spesifikasi Kebutuhan Non-Fungsional
SKPL-NF-001	Sistem memerlukan kamera dengan resolusi yang cukup untuk mengenali marker
SKPL-NF-002	Sistem memerlukan cahaya yang cukup agar kamera dapat mendeteksi marker
SKPL-NF-003	Sistem memelukan sudut yang cukup untuk membaca marker dari arah kamera

3.1.6 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional dilakukan untuk mengetahui spesifikasi kebutuhan untuk aplikasi. Spesifikasi kebutuhan melibatkan analisis pengguna, perangkat keras, dan analisis perangkat lunak. Adapun penjelasan analisis kebutuhan non-fungsional adalah sebagai berikut :

1. Analisis Pengguna

   Aplikasi ini akan digunakan oleh pihak *dealer* Toyota Auto 2000 Bandung terutama untuk bagian marketing. Selain itu, konsumen juga dapat

menggunakan aplikasi ini terutama orang dewasa. Sehingga karakteristik pengguna untuk menggunakan aplikasi ini adalah sebagai berikut :

a. Sales dan konsumen harus memiliki kemampuan mengoperasikan perangkat Android.
b. Sales tidak memiliki keterbatasan fisik.
c. Sales tidak memiliki kelainan pada mata (buta warna).

2. Analisis Perangkat Keras

Spesifikasi kebutuhan perangkat keras komputer yang dibutuhkan untuk membangun aplikasi AR Toyota dapat dilihat pada Tabel 3.3.

Tabel 3.3 Spesifikasi Kebutuhan Perangkat Keras Komputer

No	Perangkat Keras	Spesifikasi
1	Processor	Kecepatan minimum 2.5 GHz
2	VGA	1 GB
3	Memori	RAM 2 GB DDR3
4	Keyboard	Standart Port USB
5	Mouse	Optical Mouse USB
6	Monitor	LED
7	Harddisk	500 GB
8	Koneksi Internet	Ada

Sedangkan untuk perangkat Android dapat dilihat pada Tabel 3.4.

Tabel 3.4 Spesifikasi Kebutuhan Perangkat Android

No	Perangkat Keras	Spesifikasi
1	Layar	4,6’ (inch)
2	RAM	1 GB
3	Prosesor	ARmV7 / Dual core
4	Kamera	5 MP

3. Analisis Perangkat Lunak

Spesifikasi perangkat lunak komputer yang dibutuhkan untuk membangun aplikasi AR Toyota dapat dilihat pada tabel 3.5.

Tabel 3.5 Spesifikasi Kebutuhan Perangkat Lunak Komputer

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Microsoft Windows 8 64bit
2	Program	a. Autodesk 3Ds Max 2012 b. StarUML 5.0 c. Vuforia SDK 3.0.9 d. Unity 3D 4.3.4 e. Google Chrome

Sedangkan untuk perangkat lunak *platform* Android dapat dilihat pada Tabel 3.6.

Tabel 3.6 Spesifikasi Kebutuhan Perangkat Lunak *platform* Android

No	Perangkat Lunak	Spesifikasi
1	Operating System (OS)	Android
2	Versi Android	1. Ice Cream Sandwich 4.0.4 2. Jelly bean 4.1.2

3.1.7 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional bertujuan untuk menganalisis proses yang akan diterapkan dalam sistem dan menjelaskan kebutuhan yang diperlukan agar aplikasi dapat berjalan dengan baik. Pemodelan yang akan digunakan adalah pemodelan berorientasi objek menggunakan *tools* Star UML.

3.1.7.1 Use Case Diagram

Use case diagram merupakan model untuk mendeskripsikan hubungan-hubungan yang terjadi antar aktor dengan aktivitas yang terdapat pada sistem. Pada sistem ini terdapat beberapa aktor atau pengguna sistem yaitu pengguna dan marker. Untuk Use Case diagram aplikasi ini dapat dilihat pada gambar 3.8.



Gambar 3.8 Use Case Diagram

3.1.7.2 Definisi Aktor

Aktor disini adalah pihak mana saja yang mengakses Use Case. Daftar aktor yang terlibat beserta deskripsinya dapat dilihat pada Tabel 3.7.

Tabel 3.7 Daftar Aktor

No	Nama Aktor	Deskripsi tugas
1	*User*	Mengarahkan kamera pada marker
2	Marker	Sebagai *input* untuk menampilkan objek 3D

3.1.7.3 Definisi Use Case

Use case diagram merupakan konstruksi untuk mendeskripsikan hubungan-hubungan yang terjadi antar aktor dengan aktivitas yang terdapat pada sistem. Tujuan pemodelan Use Case diantaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem. Definisi dari setiap Use Case yang ada pada aplikasi dapat dilihat pada Tabel 3.8.

Tabel 3.8 Definisi Use Case

Nomor	Nama Use Case	Deskripsi
UC-01	Pembacaan marker	Fungsionalitas untuk membaca marker
UC-02	Menampilkan petunjuk penggunaan	Fungsionalitas untuk menampilkan petunjuk penggunaan aplikasi
UC-03	Menampilkan tentang aplikasi	Fungsionalitas untuk menampilkan informasi mengenai aplikasi
UC-04	Pendeteksian marker	Fungsionalitas untuk mengenali marker
UC-05	Merender objek 3D	Fungsionalitas untuk menampilkan objek 3D

3.1.7.4 Skenario Use Case

Skenario Use case menjelaskan urutan langkah-langkah dalam proses yang dilakukan oleh sebuah Use Case. Adapun Skenario Use Case dari AR Toyota adalah sebagai berikut :

1. Skenario Use Case Pembacaan Marker

   Skenario Use Case Pembacaan Marker dapat dilihat pada Tabel 3.9

Tabel 3.9 Skenario Use Case Pembacaan Marker

Nama Use Case	Pembacaan marker
Persyaratan Terkait	-
Tujuan Dalam Konteks	Sistem dapat menampilkan objek 3D
Kondisi Sebelum	Kamera belum terdeteksi
Keberhasilan Kondisi Akhir	Sistem dapat mengenali marker dan dapat menampilkan objek 3D
Kegagalan Kondisi Akhir	Sistem tidak dapat menginisialisasi marker

Nama Use Case	**Pembacaan marker**	
Aktor Utama	*User*	
Aktor Kedua	Marker	
Include	Mendeteksi marker, merender objek 3D	
Extend	-	
Pemicu	*User* memilih menu mendeteksi AR Toyota	
Aliran Utama	**Langkah**	**Aksi**
	1	*User* memilih menu mendeteksi marker
	2	*User* menunjukkan marker
	3 include::mendeteksi marker	Sistem mendeteksi marker
	4 *include*::merender objek 3D	Sistem merender objek 3D
	5	Sistem menampilkan objek 3D sesuai marker yang terdeteksi
Perluasan	**Langkah**	**Aksi Percabangan**
	4.1	Sistem tidak dapat mengenali marker yang terdeteksi
	5.1	Sistem tidak dapat merender objek 3D sesuai dengan marker

2. Skenario Use Case Menampilkan Petunjuk Penggunaan

   Skenario use case menampilkan petunjuk penggunaan dapat dilihat pada Tabel 3.10.

Tabel 3.10 Menampilkan Petunjuk Penggunaan

Nama Use Case	**Menampilkan Petunjuk Penggunaan**	
Persyaratan Terkait	-	
Tujuan Dalam Konteks	Sistem dapat menampilkan petunjuk penggunaan	
Kondisi Sebelum	*User* berada di tampilan utama aplikasi	
Keberhasilan Kondisi Akhir	Sistem dapat menampilkan petunjuk penggunaan	
Kegagalan Kondisi Akhir	Sistem tidak dapat menampilkan petunjuk penggunaan	
Aktor Utama	*User*	
Aktor Kedua	Marker	
Extend	-	
Pemicu	*User* memilih menu pilihan petunjuk penggunaan	
Aliran Utama	**Langkah**	**Aksi**
	1	*User* memilih menu petunjuk penggunaan
	2	Sistem menampilkan petunjuk penggunaan
Perluasan	**Langkah**	**Aksi Percabangan**
	2.1	Sistem tidak dapat menampilkan petunjuk penggunaan

3. Skenario Use Case Menampilkan Tentang Aplikasi

   Skenario use case menampilkan informasi tentang aplikasi dapat dilihat pada Tabel 3.11.

Tabel 3.11 Skenasio Use Case Menampilkan Tentang Aplikasi

Nama Use Case	**Menampilkan Informasi Tentang Aplikasi**	
Persyaratan Terkait	-	
Tujuan Dalam Konteks	Sistem dapat menampilkan informasi tentang aplikasi	
Kondisi Sebelum	*User* berada di tampilan utama aplikasi	
Keberhasilan Kondisi Akhir	Sistem dapat menampilkan informasi tentang apliaksi	
Kegagalan Kondisi Akhir	Sistem tidak dapat menampilkan informasi tentang apliaksi	
Aktor Utama	*User*	
Aktor Kedua	Marker	
Extend	-	
Pemicu	*User* memilih menu pilihan tentang aplikasi	
Aliran Utama	**Langkah**	**Aksi**
	1	Pengguna memilih menu pilihan tentang aplikasi
	2	Sistem menampilkan tentang aplikasi
Perluasan	**Langkah**	**Aksi Percabangan**
	2.1	Sistem tidak dapat menampilkan tentang aplikasi

4. Skenario Use Case Pendeteksian Marker

   Skenario use case pendeteksian marker dapat dilihat pada Tabel 3.12.

Tabel 3.12 Skenario Use Case Mendeteksi Marker

Nama Use Case	**Pendeteksian marker**	
Persyaratan Terkait	-	
Tujuan Dalam Konteks	Sistem dapat mengenali marker yang terdeteksi	
Kondisi Sebelum	Marker belum terdeteksi	
Keberhasilan Kondisi Akhir	Sistem dapat mengenali marker	
Kegagalan Kondisi Akhir	Sistem tidak dapat mengenali marker	
Aktor Utama	*User*	
Aktor Kedua	Marker	
Include	-	
Pemicu	Kamera dapat mendeteksi keberadaan marker	
Aliran Utama	**Langkah**	**Aksi**
	1	*User* mengarahkan marker
	2	Sistem menginisialisasi marker
	3	Sistem mengidentifikasi marker
	4	Sistem dapat melakukan proses *grayscale*
	5	Sistem dapat melakukan proses pendeteksian fitur
	6	Sistem mampu mendeteksi marker sesuai dengan *library*

Perluasan	Langkah	Aksi Percabangan
	3.1	Sistem tidak dapat mendeteksi kamera
	6.1	Sistem tidak dapat mengidentifikasi marker

5. Skenario Use Case Merender Objek

Skenario use case merender objek dapat dilihat pada Tabel 3.13.

Tabel 3.13 Skenario Use Case Merender Objek

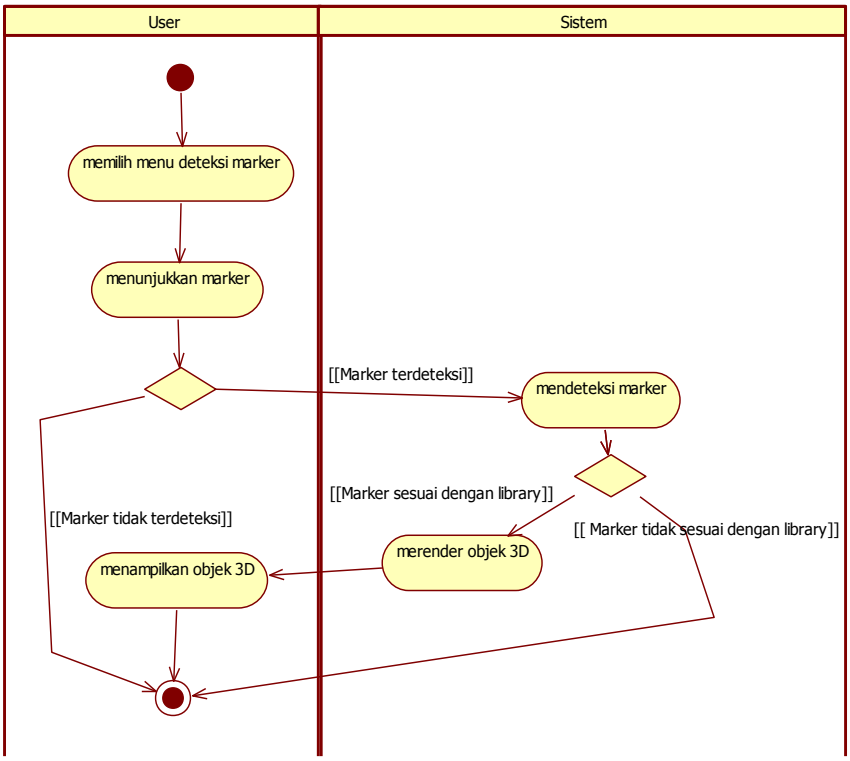
Nama Use Case	Merender Objek	
Persyaratan Terkait	-	
Tujuan Dalam Konteks	Sistem menampilkan objek 3D	
Kondisi Sebelum	Marker yang terdeteksi belum dapat dikenali	
Keberhasilan Kondisi Akhir	Sistem menampilkan objek 3D sesuai marker yang dideteksi tepat di atas marker	
Kegagalan Kondisi Akhir	Sistem tidak dapat menampilkan objek 3D	
Aktor Utama	*User*	
Aktor Kedua	Marker	
Extend	Merotasi objek 3D, menampilkan informasi nama mobil	
Pemicu	Marker dapat dekenali dengan baik dan sesuai dengan pola marker yang ada	
Aliran Utama	**Langkah**	**Aksi**
	1	*User* mengarahkan marker
Aliran Utama	**Langkah**	**Aksi**
	2	Kamera membaca marker
	3	Sistem mengenali marker
	4	Sistem merender objek 3D
	5	Objek 3D ditampilkan tetap di atas marker
Perluasan	**Langkah**	**Aksi Percabangan**
	3.1	Sistem tidak dapat menemukan pola marker yang sesuai
	3.2	Sistem tidak merender objek 3D
	3.3	Sistem tidak menampilkan objek 3D
	5.1	Sistem melakukan rotasi objek 3D

3.1.7.5 Activity Diagram

Activity diagram menggambarkan berbagai aliran aktivitas dalam sistem yang dirancang, bagaimana masing-masing fungsionalitas bekerja, dan bagaimana suatu fungsional berakhir. Activity Diagram memodelkan *event-event* yang terjadi pada use case. Adapun Activity Diagram dari aplikasi yang akan dibangun adalah sebagai berikut :

1. Activity Diagram Pembacaan Marker

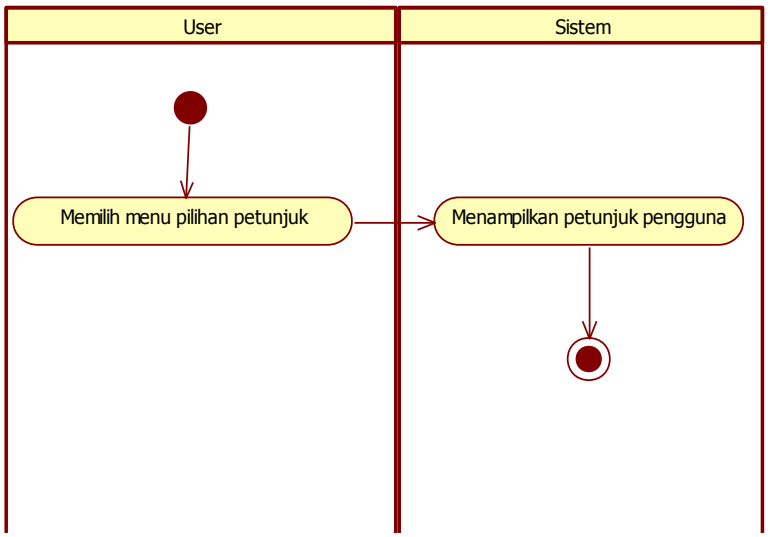
   Merupakan aktifitas dari menu pilihan deteksi marker untuk membaca marker. Activity diagram pembacaan marker dapat dilihat pada gambar 3.9.



Gambar 3.9 Activity Diagram Membaca Marker

2. Activity Diagram Petunjuk Penggunaan

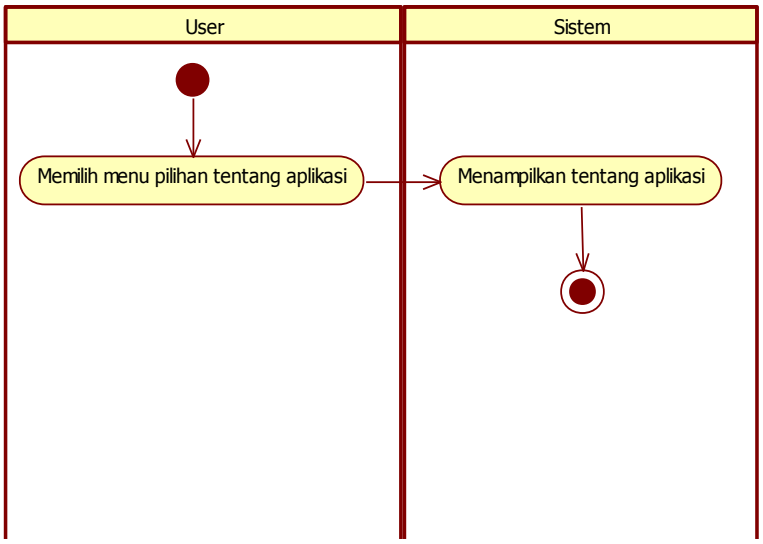
   Merupakan aktifitas dari menu pilihan petunjuk penggunaan sehingga menampilkan cara penggunaan aplikasi AR Toyota. Activity diagram dari petunjuk penggunaan dapat dilihat pada Gambar 3.10.



Gambar 3.10 Activity Diagram Petunjuk Penggunaan

3. Activity Diagram Menampilkan Tentang Aplikasi

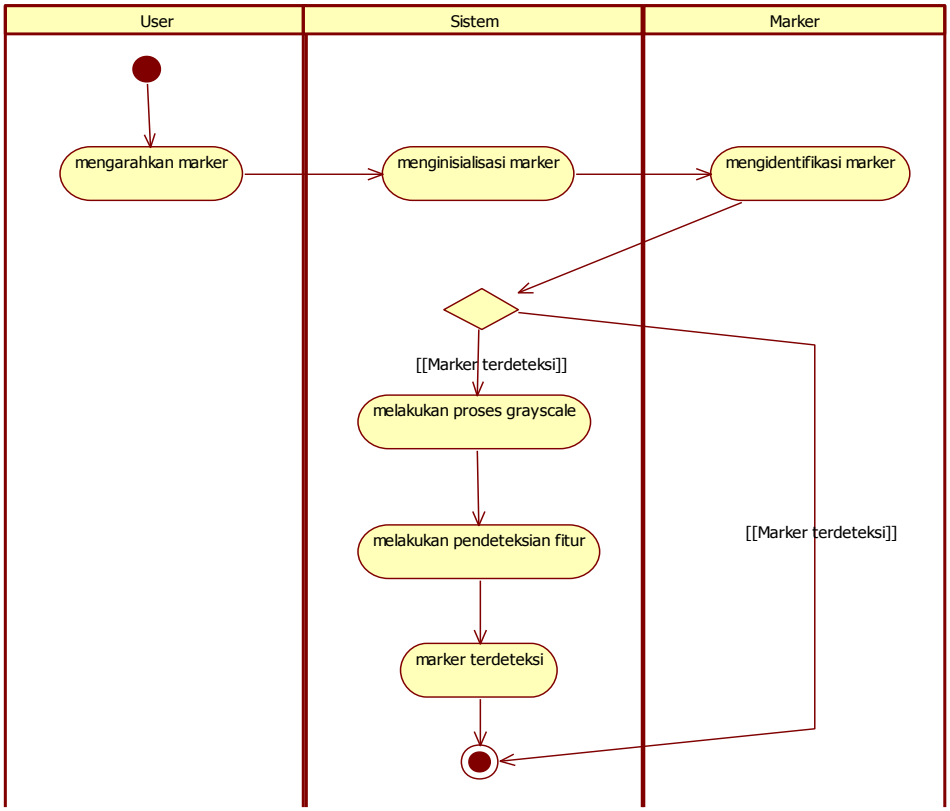
   Merupakan aktifitas dari menu pilihan tentang aplikasi sehingga menampilkan informasi tentang aplikasi AR Toyota. Activity diagram dari tentang aplikasi dapat dilihat pada Gambar 3.11.



Gambar 3.11 Activity Diagram Menampilkan Tentang Aplikasi

4. Activity Diagram Pendeteksian Marker

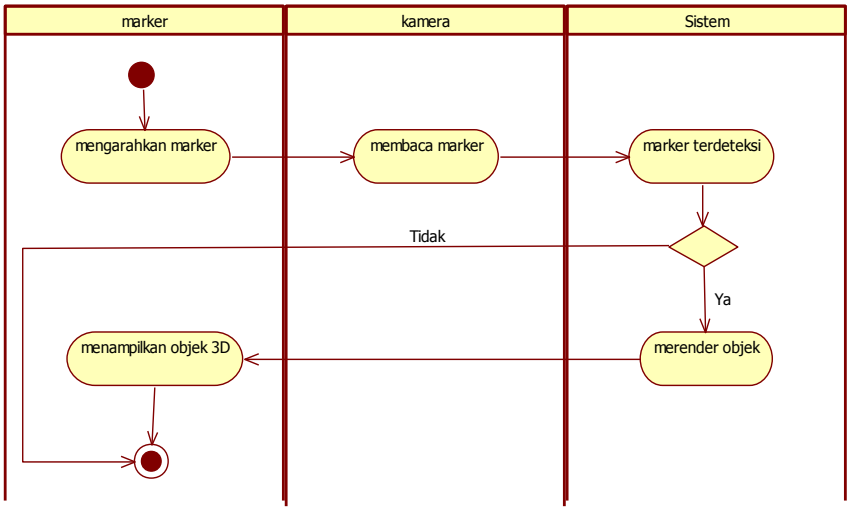
   Merupakan sub-aktifitas dari Use Case pendeteksian marker. Activity diagram dari pendeteksian marker dapat dilihat pada gambar 3.12.



Gambar 3.12 Activity Diagram Mendeteksi Marker

5. Activity Diagram Merender objek 3D

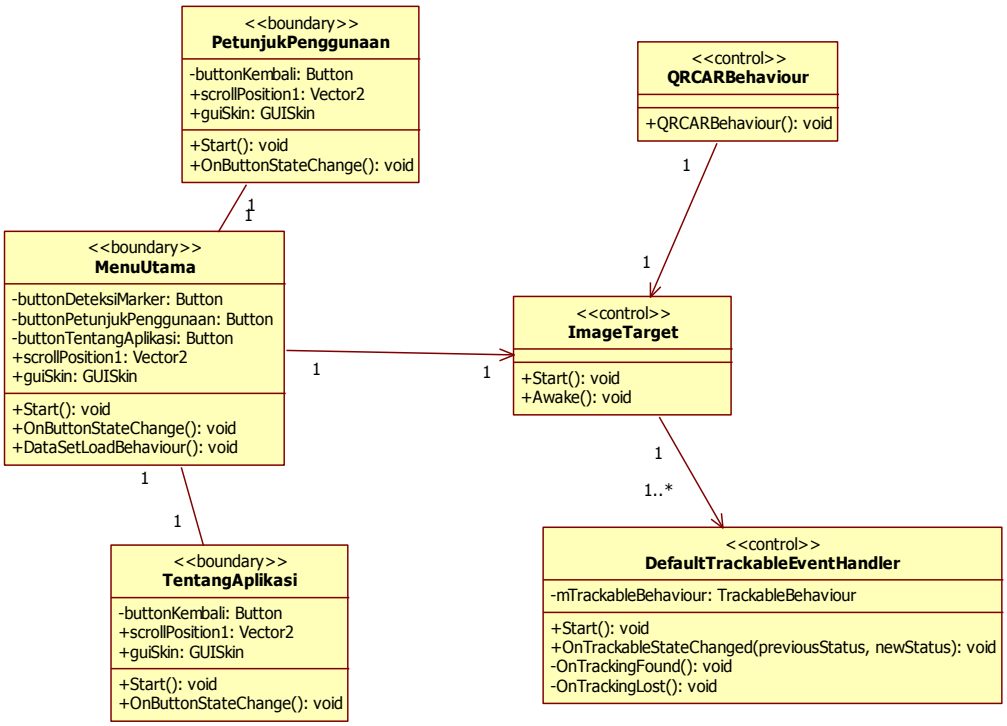
   Merupakan sub-aktifitas dari Use Case merender objek. Activity diagram dari mendeteksi marker dapat dilihat pada Gambar 3.13.



Gambar 3.13 Activity Diagram Merender Objek

3.1.7.6 Class Diagram

Class Diagram menggambarkan struktur dan hubungan antar objek-objek yang ada pada sistem. Struktur itu meliputi atribut-atribut dan metode-metode yang ada pada masing-masing kelas. Adapun kelas diagram dari aplikasi AR Toyota dapat dilihat pada Gambar 3.14.



Gambar 3.14 Class Diagram AR Toyota

Berikut ini merupakan deskripsi dari Class Diagram AR Toyota yang dapat dilihat pada tabel 3.14.

Tabel 3.14 Deskripsi Class Diagram AR Toyota

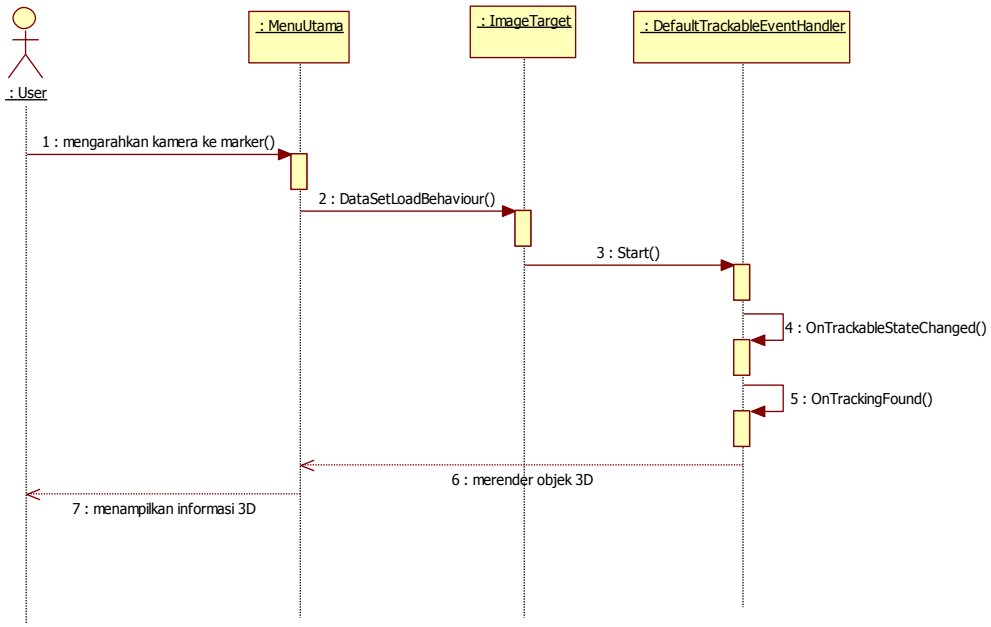
No	Nama Kelas	Tipe Kelas
1	MenuUtama	Boundary
2	TentangAplikasi	Boundary
3	PetunjukPenggunaan	Boundary
4	ImageTarget	Control
5	QRCARBehaviour	Control
6	DefaultTrackableEventHandler	Control

3.1.7.7 Sequence Diagram

Sequence diagram menggambarkan interaksi antara masing-masing objek pada setiap use case dalam urutan waktu. Interaksi berupa pengiriman serangkaian data antar objek-objek yang saling berinteraksi. Adapun Sequence Diagram dari aplikasi yang akan dibangun adalah sebagai berikut :

1. Sequence Diagram Mendeteksi Marker

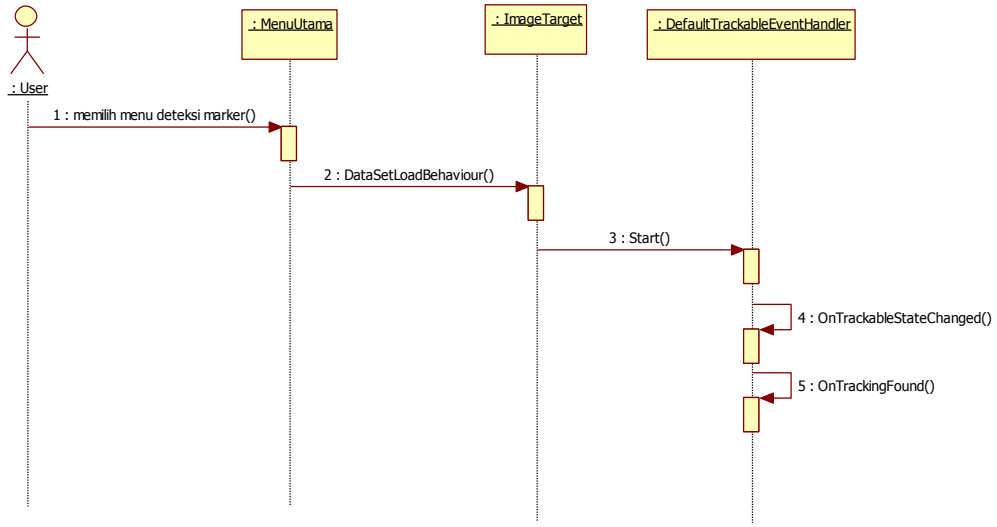
   Sequence Diagram mendeteksi marker dapat dilihat pada gambar 3.15.



Gambar 3.15 Sequence Diagram Mendeteksi Marker

2. Sequence Diagram Membaca Marker

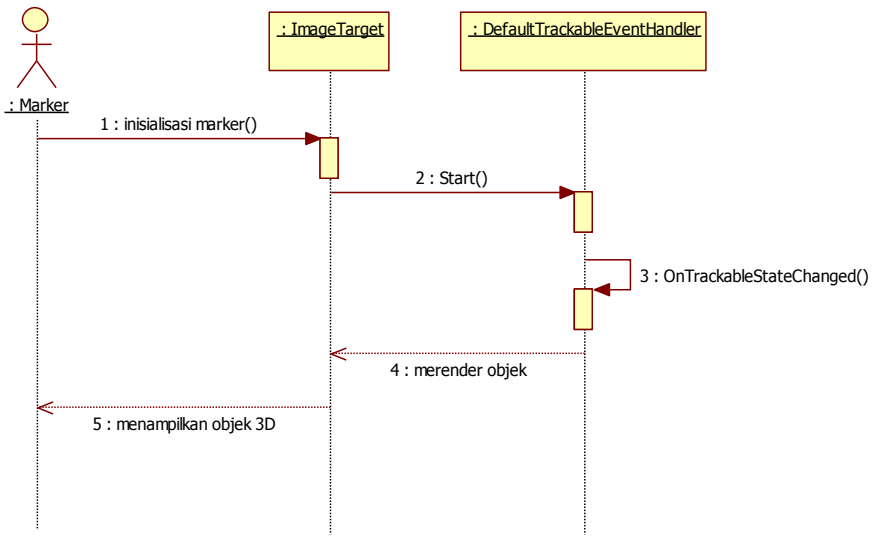
   Sequence Diagram membaca marker dapat dilihat pada gambar 3.16



Gambar 3.16 Sequence Diagram Membaca Marker

3. Sequence Diagram Merender Objek

   Sequence Diagram merender objek dapat dilihat pada gambar 3.17



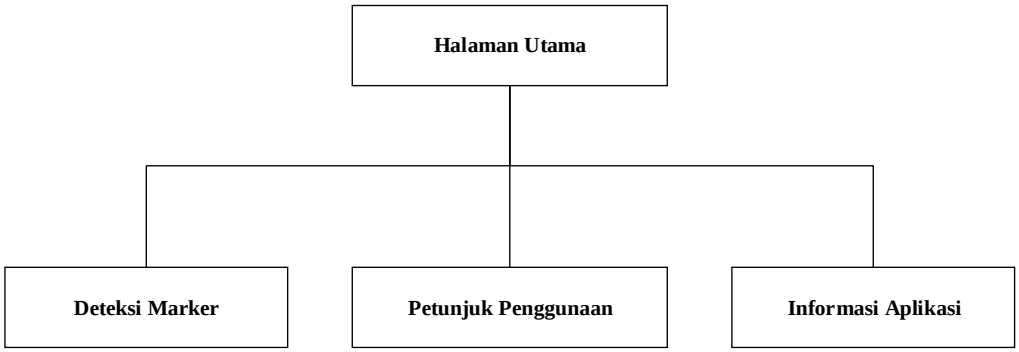
Gambar 3.17 Sequence Diagram Merender Objek

3.2 Perancangan Sistem

Perancangan sistem bertujuan untuk memberikan gambaran sistem yang dibuat. Perancangan disini terdiri dari perancangan struktur menu, perancangan antarmuka, jaringan semantik, serta perancangan method.

3.2.1 Perancangan Struktur Menu

Perancangan *method* berfungsi untuk mendeskripsikan *method* yang berada dalam aplikasi. Adapun method yang terdapat pada aplikasi yang akan dibangun yaitu mendeteksi marker, rotasi objek 3D. Gambar perancangan struktur menu dapat dilihat pada gambar 3.18.



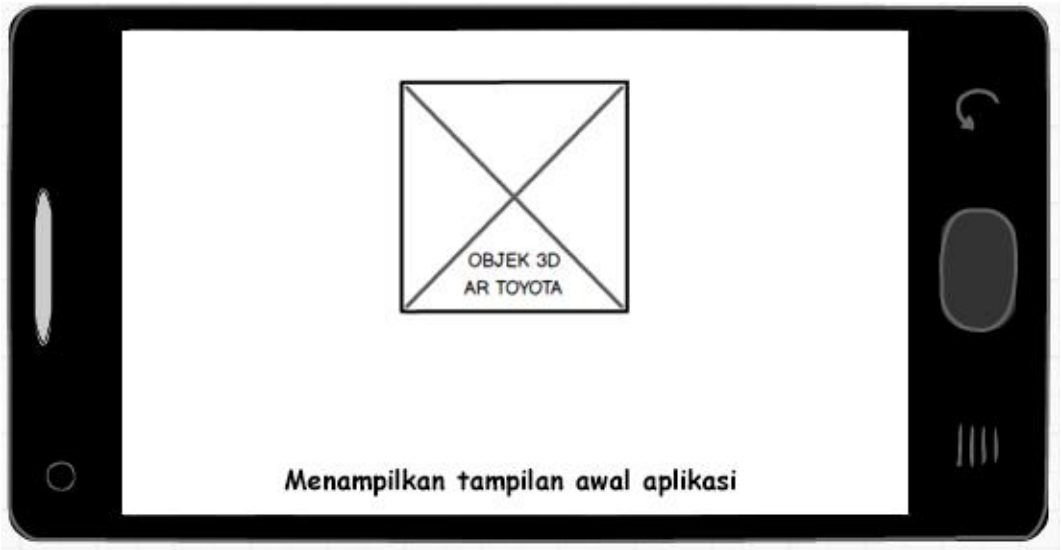
Gambar 3.18 Perancangan Struktur Menu

3.2.2 Perancangan Antarmuka

Perancangan antarmuka merupakan rancangan tampilan dari aplikasi. Perancangan antarmuka dapat dilihat sebagai berikut :

1. Perancangan Antarmuka *Spalshscreen*

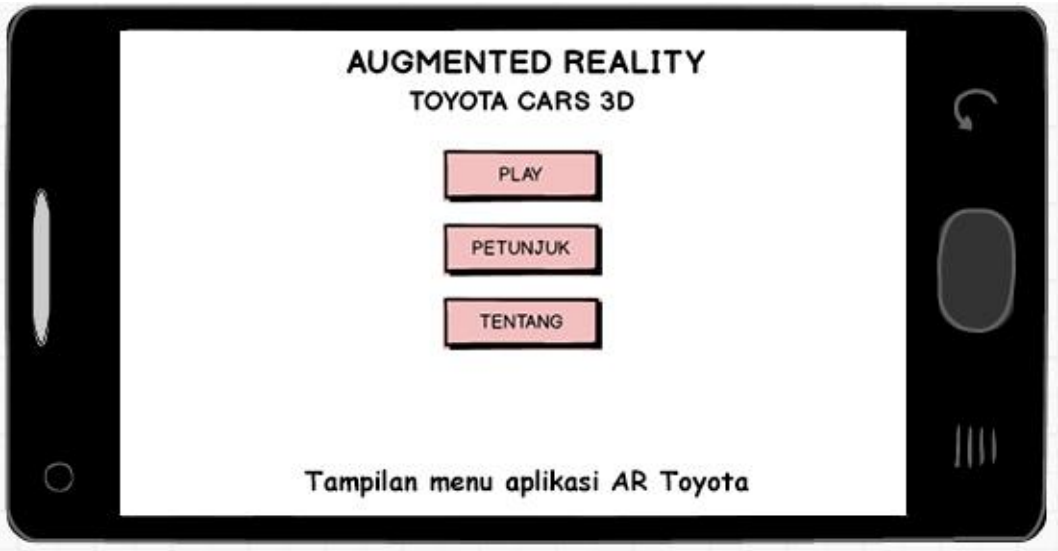
   Perancangan antarmuka *splashscreen* ini dapat dilihat pada gambar 3.19.



Gambar 3.19 Perancangan Antarmuka *Splashscreen*

2. Perancangan Antarmuka Utama

   Perancangan antarmuka utama aplikasi ini dapat dilihat pada Gambar 3.20.



Gambar 3.20 Perancangan Antarmuka Utama

3. Perancangan Antarmuka Deteksi Marker

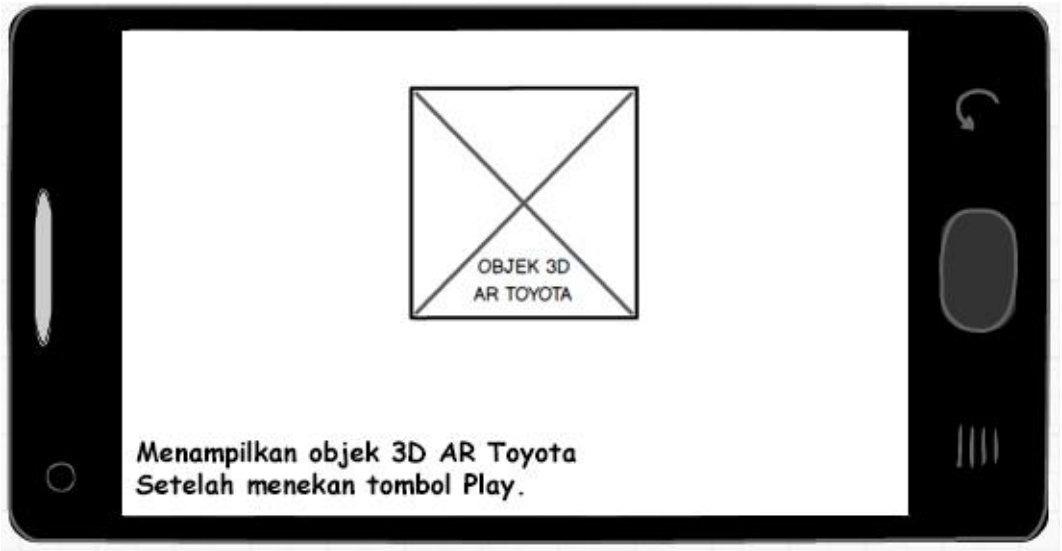
   Perancangan antarmuka deteksi marker aplikasi ini dapat dilihat pada Gambar 3.21.



Gambar 3.21 Perancangan Antarmuka Proses Pendeteksian Marker

4. Perancangan Antarmuka Menampilkan Objek

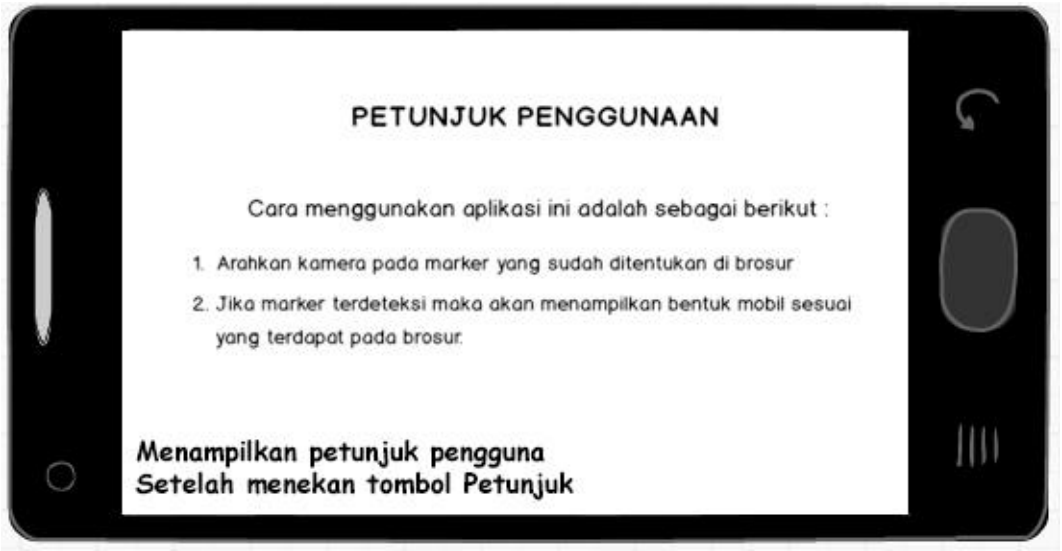
   Perancangan antarmuka petunjuk penggunaan aplikasi ini dapat dilihat pada Gambar 3.22.



Gambar 3.22 Perancangan Antarmuka Menampilkan Objek

5. Perancangan Antarmuka Petunjuk Penggunaan

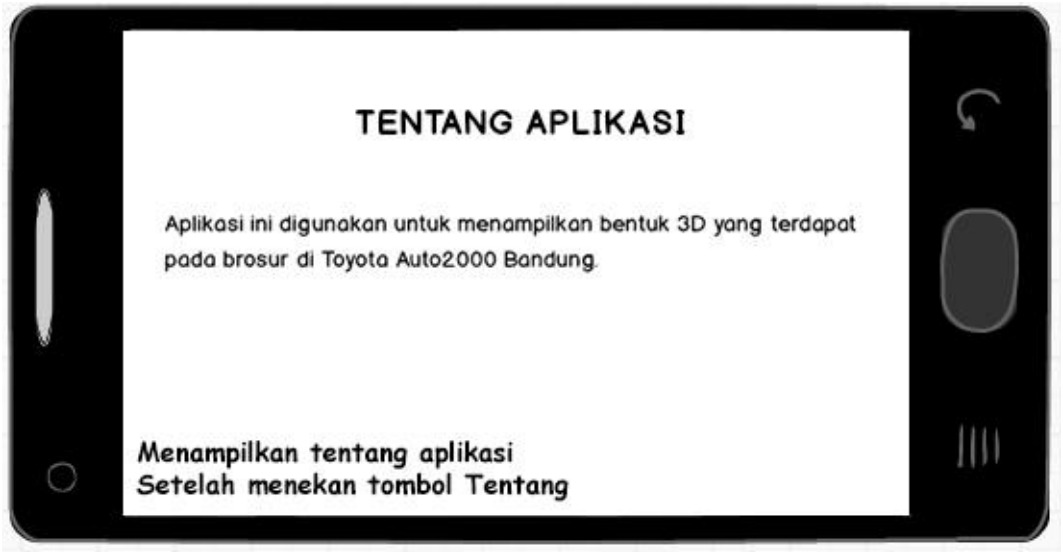
   Perancangan antarmuka tentang aplikasi ini dapat dilihat pada Gambar 3.23.



Gambar 3.23 Perancangan Antarmuka Petunjuk Penggunaan

6. Perancangan Antarmuka Tentang Aplikasi

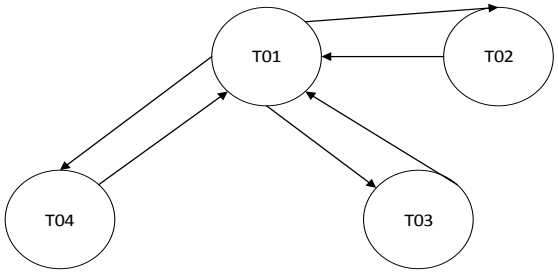
   Perancangan Antarmuka Tentang Aplikasi dapat dilihat pada gambar 3.24.



Gambar 3.24 Perancangan Antarmuka Tentang Aplikasi

3.2.3 Jaringan Semantik

Jaringan semantik dari aplikasi dapat dilihat pada Gambar 3.25.



Gambar 3.25 Jaringan Semantik

Ketarangan:

T01 : Halaman utama AR Toyota

T02 : Halaman untuk mendeteksi marker

T03 : Halaman untuk menampilkan petunjuk penggunaan

T04 : Halaman untuk menampilkan tentang aplikasi

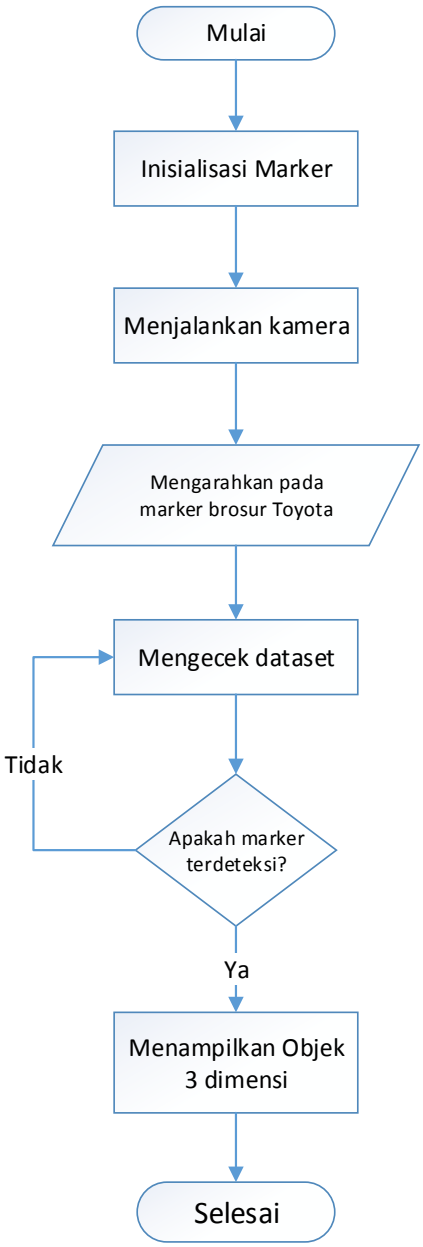
3.2.4 Perancangan Method

Perancangan method berfungsi untuk mendeskripsikan method yang berada dalam aplikasi. Adapun *method* yang terdapat pada aplikasi yang akan dibangun

yaitu mendeteksi marker, rotasi objek 3D. Method tersebut dapat dilihat sebagai berikut :

1. Perancangan *Method* Pendeteksian Marker

   Perancangan *Method* pendeteksian marker dapat dilihat pada gambar 3.26



Gambar 3.26 Perancangan Method Pendeteksian Marker