BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

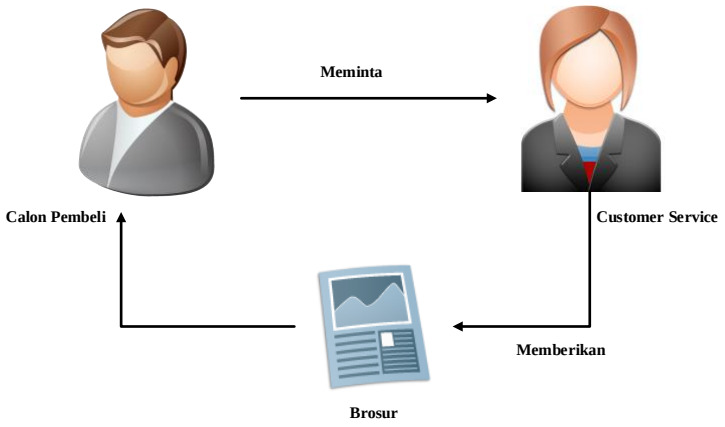
Berdasarkan analisis yang telah dilakukan maka untuk memahami sistem yang akan dibuat maka salah satu tahapan yang harus dilalui adalah melakukan analisis. Pembahasan berikut merupakan analisis masalah, analisis sistem yang sedang berjalan, analisis aplikasi sejenis, dan analisis *Augmented Reality*.

3.1.1 Analisis Masalah

Berdasarkan hasil pengamatan dan wawancara yang telah dilakukan, dalam melakukan penjelasan tipe-tipe rumah ke calon pembeli, media yang digunakan adalah brosur dan maket. Untuk maket tidak semua tersedia tipe-tipe rumah dibuatkan maketnya, sehingga tidak semua calon pembeli dapat memahami tipe-tipe rumah yang ditawarkan.

3.1.2 Analisis Sistem yang Sedang Berjalan

Prosedur yang sedang berjalan di kantor pemasaran PT Karya Sehati Utama dalam mengenalkan tipe-tipe rumah adalah dengan menjelaskan secara langsung kepada calon pembeli melalui media brosur. Calon pembeli juga dapat melihat maket rumah yang tersedia. Berikut adalah gambaran mengenai prosedur pengenalan tipe-tipe rumah yang sedang berjalan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur pengenalan tipe-tipe rumah

Penjelasan dari Gambar 3.1 dapat dilihat sebagai berikut:

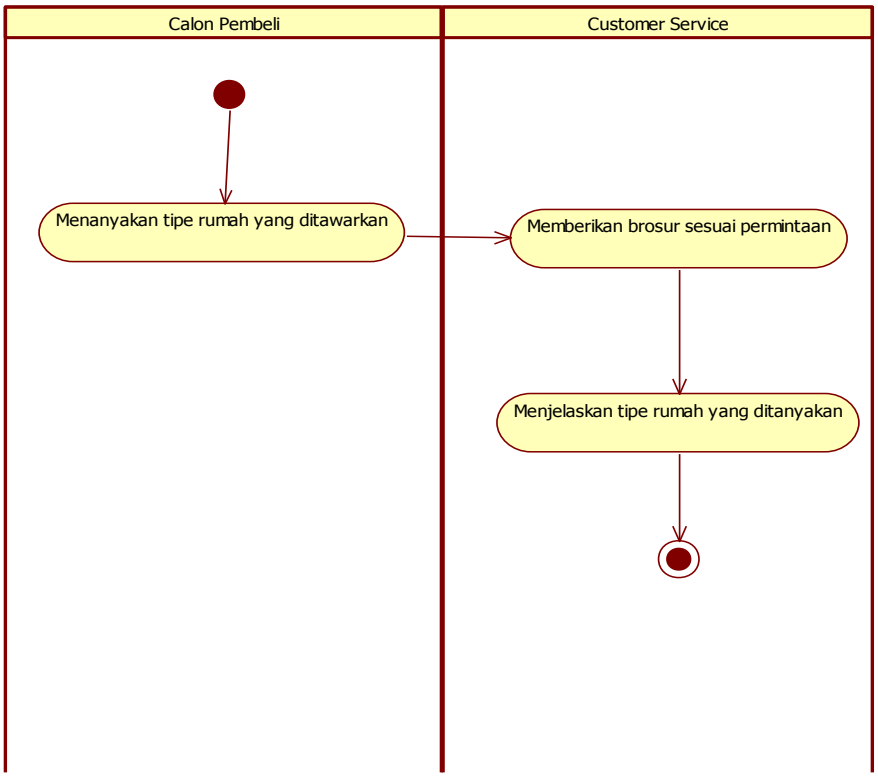
1. Calon pembeli

   Calon pembeli disini adalah sebagai orang yang memerlukan informasi mengenai tipe-tipe rumah yang ditawarkan. Calon pembeli meminta brosur kepada *customer service.*

2. Customer Service

   Customer service menjadi media yang tersedia untuk mengetahui tipe-tipe rumah apa saja yang ditawarkan. *Customer service* akan memberikan penjelasan dan brosur kepada calon pembeli.

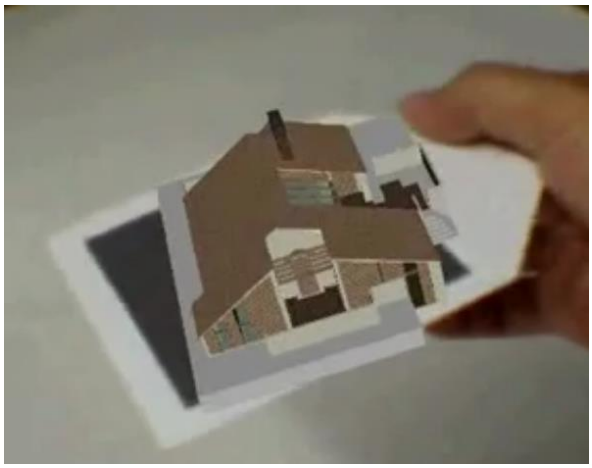
Aktifitas yang dilakukan oleh calon pembeli dan *customer service* dapat digambarkan dengan *activity diagram*. *Activity diagram* pengenalan tipe rumah dapat dilihat pada Gambar 3.2.



Gambar 3.2 Activity Diagram Pengenalan Tipe Rumah

3.1.3 Analisis Aplikasi Sejenis

Tahap analisis dilakukan untuk mengetahui kelebihan dan kekurangan serta komponen apa saja yang terlibat pada aplikasi yang dianalisis. Agar aplikasi yang akan dibangun diharapkan bisa menjadi lebih baik dari aplikasi yang dianalisis. Ada banyak aplikasi yang digunakan sebagai media promosi untuk suatu produk yang dijual. Aplikasi yang digunakan untuk dijadikan sebagai bahan referensi dan digunakan sebagai bahan analisis adalah aplikasi AR Rumah. Aplikasi ini berbasis desktop dan menggunakan kamera tambahan atau bisa juga menggunakan kamera laptop untuk membaca *marker*. *Marker* yang digunakan adalah *marker* hitam putih dimana memiliki border hitam persegi yang membingkai dari gambar atau pola unik untuk menampilkan objek 3D. Saat aplikasi ini dijalankan, kamera akan langsung membaca semua inputan yang masuk melalui kamera. Jika Kemudian program akan menyeleksi objek-objek atau *marker-marker* apa saja yang terdeteksi. Calon *marker* yang sudah sesuai dengan aturan *marker* yaitu dimana terdapat border hitam persegi akan dideteksi dan disamakan dengan data *marker* yang ada. Jika *marker* yang terdeteksi sesuai, maka aplikasi akan menempatkan objek 3D tepat diatas *marker* yang terdeteksi. Jika tidak maka tidak akan menampilkan objek 3D apapun. Tampilan dari aplikasi ini dapat dilihat pada Gambar 3.3.



Gambar 3.3 Tampilan Aplikasi AR Rumah

Komponen dasar di dalam suatu sistem AR adalah sebuah *display*, kamera untuk menangkap *marker*, dan komputer yang telah di-*install* dengan *software* yang berkaitan, *marker*, serta berbagai macam perangkat keras yang dapat digunakan untuk mendukung, seperti kamera, telepon genggam, PDA, laptop, dan lainnya. Komponen aplikasi yang terlibat adalah layar komputer, kamera laptop atau kamera tambahan, dan *marker*. Fungsionalitas dari aplikasi ini adalah hanya menampilkan objek 3D sesuai *marker* yang dideteksi. Intinya pada saat aplikasi dijalankan, aplikasi ini langsung menampilkan layar kamera untuk menangkap *marker*. Kelebihan dari aplikasi ini adalah dari *marker* yang digunakan yaitu menggunakan *marker* hitam putih dengan border hitam persegi. Dengan menggunakan *marker* seperti ini membuat deteksi dan pemrosesan untuk menampilkan objek 3D menjadi lebih cepat dan ringan.

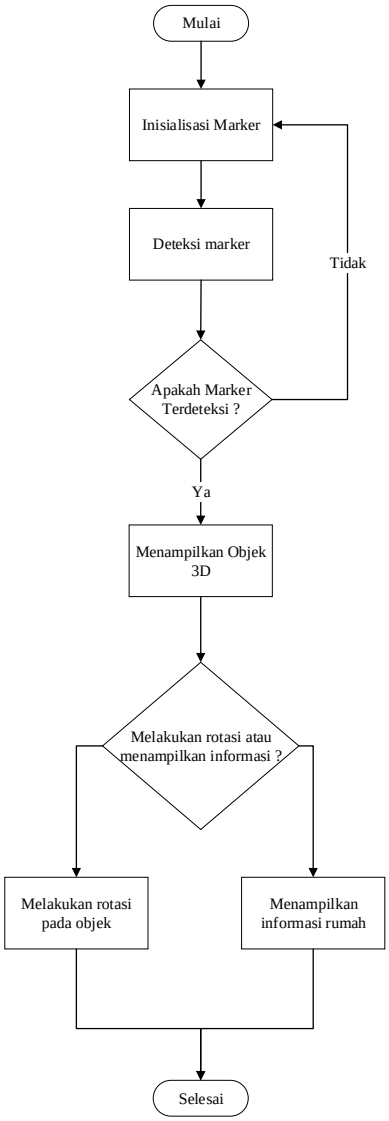
Kesimpulan dari aplikasi ini adalah aplikasi ini berbasis desktop dimana membutuhkan kemera tambahan atau kamera laptop untuk mendeteksi *marker*. Sehingga pada aplikasi yang akan dibangun yaitu aplikasi *Augmented Reality* Karya Sehati Utama (AR KSU)akan menambahkan beberapa fungsionalitas tanpa mengurangi dari aplikasi sejenis, yaitu akan menambahkan menu pilihan yang berisi, tentang yaitu berisi mengenai aplikasi yang akan dibangun, petunjuk penggunaan dari aplikasi, serta deteksi *marker* untuk melacak *marker* dan menampilkan objek 3D.

3.1.4 Analisis 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 (lebih dari atau sama dengan 1 dan kurang dari sama dengan 5) dan tidak 0 (nol).

Perancangan aplikasi *Augmented Reality*, menggabungkan objek virtual atau objek 3D dengan objek nyata atau (*marker*). Secara keseluruhan, proses dari *Augmented Reality* dimulai dari webcam atau kamera yang sudah dikalibrasi akan mendeteksi *marker* yang diberikan. Kemudian setelah mengenali dan menandai

pola *marker*, webcam atau kamera akan melakukan pengecekan apakah *marker* yang terdeteksi sesuai dengan data *marker* yang dimiliki. Bila tidak, maka objek 3D tidak akan ditampilkan, tetapi bila sesuai maka objek 3D akan ditampilkan di atas *marker*. Alur proses dari aplikai yang akan dibangun dapat dilihat pada gambar 3.4.



Gambar 3.4 Alur Proses Aplikasi

3.1.4.1 Inisialisasi Marker

Tahap ini aplikasi *Augmented Reality* akan melakukan inisialisasi *windows* atau 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.

Spesifikasi pola penanda atau *marker*:

1. Pola penanda minimum memiliki lebar 320 piksel.
2. Format dari gambar yang diunggah ke situs vuforia berformat .jpg atau .png

Contoh *marker* yang digunakan dapat dilihat pada Gambar 3.5.



Gambar 3.5 Contoh Marker

Gambar *marker* 3.5 nilai *rate* bintang bernilai 5. Sehingga sudah cukup untuk dilacak oleh sistem AR.

Selain menginisialisasi *marker*, model 3D juga dimuat dahulu pada *engine* Unity untuk dipasangkan dengan *marker* yang bersesuaian. Objek 3D yang digunakan berformat .3DS agar dapat dieksport ke Unity. Gambar dari model 3D rumah yang digunakan pada aplikasi dapat dilihat pada Gambar 3.6.

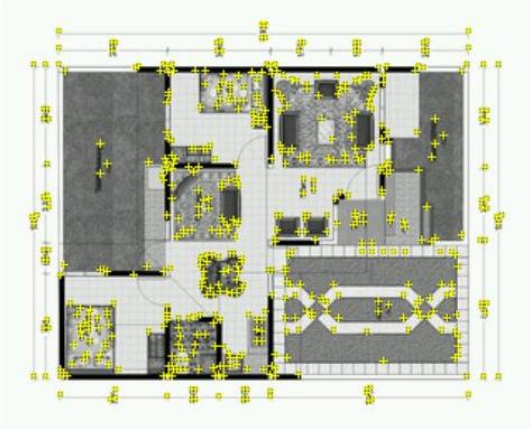


Gambar 3.6 Model 3D rumah

3.1.4.2 Deteksi Marker

NFT merupakan metode yang digunakan di dalam Library Vuforia Qualcomm untuk mendeteksi pola gambar. Sebelum dilakukannya pendeteksian fitur-fitur 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 dirubah menjadi keabu-abuan kemudian dideteksi fiturnya menggunakan metode NFT. Metode yang digunakan di dalam NFT itu sendiri adalah SIFT dan FERNS.

Hasil output gambar dari perhitungan dapat dilihat pada Gambar 3.7.



Gambar 3.7 Gambar Hasil Deteksi Fitur Lokal

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 data yang tersimpan. Untuk penulisan nama hanya diperbolehkan maksimal 64 karakter dan hanya mengandung karakter (a-z, A-Z, 0-9, [-_.]).
3. *Trackable* Status

   Pelacakan memiliki informasi status yang terkait dengannya, setiap informasi satus dilacak akan diperbaharui sebagai frame kamera yang diproses. Status yang terjadi pada saat pelacak dapat terjadi pada salah satu dibawah ini :
   a. UNKNOW : gambar *marker* pada pelacakan tidak diketahui biasanya dikembalikan sebelum *tracker initialization*
   b. UNDEFINE : gambar *marker* tidak didefinisikan
   c. NOT_FOUND: gambar *marker* tidak ditemukan pada *library* yang dituju
   d. DETECTED : gambar *marker* dapat disedeksi dalam frame
   e. TRACKED : pelacakan telah terlacak dalam frame

Beberapa fungsi yang digunakan pada vuforia adalah sebagai berikut:

CameraDevice

Fungsi kelas ini untuk menyediakan akses ke kamera dan properti.

QCARBehaviour

Merupakan kelas untuk menangani pelacakan dan memicu tampilan *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 dijalankan.

TrackableBehaviour.RegisterTrackableEventHandler

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

Status TrackableBehaviour.CurrentStatus [get]

Status pelacakan dari TrackableBehaviour.

TrackableBehaviour.UnregisterTrackableEventHandler

Berfungsi sebagai *unregisters* sebuah Tracker Event Handler mengembalikan nilai "false" jika *event handler* tidak ada.

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 antarmuka ITrackableEventHandler.

DevaultInitializationErrorHandler

Fungsi untuk menginisialisasi eror 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 3.8.



Gambar 3.8 Objek 3D Yang Ditampilkan Tepat Di Atas Marker

3.1.5 Spesifikasi Kebutuhan Perangkat Lunak

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 *marker*
SKPL-F-003	Sistem harus dapat menampilkan objek 3D
SKPL-F-004	Sistem harus dapat melakukan rotasi terhadap objek yang ditampilkan
SKPL-F-005	Sistem harus dapat menampilkan informasi dalam rumah

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.

1. Analisis Pengguna

Aplikasi ini akan digunakan oleh pihak PT Karya Sehati Utama terutama untuk bagian penjualan. Calon pembeli juga dapat menggunakan aplikasi ini terutama orang dewasa. Sehingga karakteristik pengguna untuk menggunakan aplikasi ini adalah sebagai berikut:

   a. *Customer service* dan calon pembeli harus memiliki kemampuan mengoperasikan perangkat Android.
   b. *Customer service* tidak memiliki keterbatasan fisik.
   c. *Customer service* tidak memiliki kelainan pada mata (buta warna).

2. Analisis Perangkat Keras

Spesifikasi minimum perangkat keras yang dibutuhkan untuk membangun aplikasi untuk komputer dapat dilihat pada Tabel 3.3.

Tabel 3.3 Spesifikasi Minimum Kebutuhan Perangkat Keras Komputer

No	Perangkat Keras	Spesifikasi
1	Processor	Kecepatan minimum 2.0 Ghz
2	VGA	1 GB

No	Perangkat Keras	Spesifikasi
3	Memori	RAM 1 GB DDR3
4	Keyboard	Standart Port USB
5	Mouse	Optical Mouse USB
6	Monitor	LED
7	Harddisk	250 GB

Sedangkan untuk menjalankan aplikasi spesifikasi minimum untuk perangkat *mobile* Android dapat dilihat pada Tabel 3.4.

Tabel 3.4 Spesifikasi Minimum Kebutuhan Perangkat Mobile Android

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

3. Analisis Perangkat Lunak

Spesifikasi perangkat lunak yang dibutuhkan untuk membangun aplikasi adalah yaitu, spesifikasi kebutuhan perangkat lunak untuk komputer dapat dilihat pada Tabel 3.5.

Tabel 3.5 Spesifikasi Minimum Kebutuhan Perangkat Lunak Untuk Komputer

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Microsoft Windows 7 64bit
2	Program	a. Micrsoft Office 2013 b. StarUML 5.0 c. SketchUp 8 d. 3D Max e. Vuforia f. Unity 3D

Sedangkan spesifikasi kebutuhan perangkat lunak untuk menjalankan aplikasi untuk *mobile* Android dapat dilihat pada Tabel 3.6.

Tabel 3.6 Spesifikasi Minimum Kebutuhan Perangkat Lunak Mobile Android

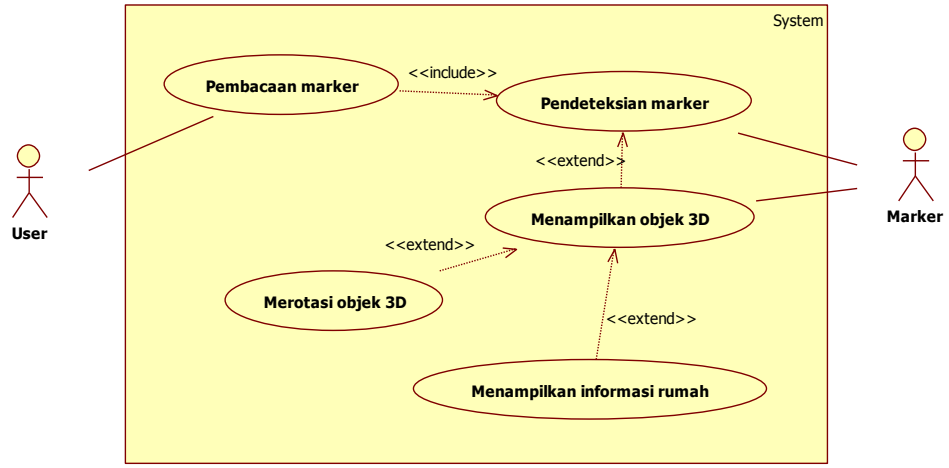
No	Perangkat Lunak	Spesifikasi
1	Operating System (OS)	Android
2	Versi Android	2.3 Gingerbread

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* StarUML.

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. Sistem ini terdapat beberapa aktor atau pengguna sistem yaitu *user* dan *marker*. Untuk use case diagram aplikasi ini dapat dilihat pada Gambar 3.9.



Gambar 3.9 Use Case Diagram Aplikasi

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 inputan 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	Pendeteksian marker	Fungsionalitas untuk mengenali *marker*
UC-03	Menampilkan objek 3D	Fungsionalitas untuk menampilkan objek 3D
UC-04	Merotasi objek 3D	Fungsionalitas untuk merotasi objek 3D yang ditampilkan
UC-05	Menampilkan informasi rumah	Fungsionalitas untuk menampilkan informasi dari rumah yang ditampilkan

3.1.7.4 Skenario Use Case

Skenario *use case* menjelaskan urutan langkah-langkah dalam proses yang dilakukan oleh sebuah *use case*.

1. Scenario 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*	
Aktor Utama	User	
Aktor Kedua	Marker	
Include	Pendeteksi marker	
Extend	-	
Pemicu	User memilih menu deteksi *marker*	
Aliran Utama	**Langkah**	**Aksi**
	1	Memilih menu deteksi *marker*
	2	Pengguna menunjukkan *marker*
	3 include::mendeteksi marker	Sistem mendeteksi *marker*

	4	Sistem me-*render* 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 Pendeteksian Marker

   Skenario *use case* pendeteksian *marker* dapat dilihat pada Tabel 3.10.

Tabel 3.10 Skenario Use Case Pendeteksian 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 mengidentifikasi *marker*
	6.1	Sistem tidak dapat mendeteksi *marker*

3. Skenario Use Case Menampilkan Objek 3D

   Skenario *use case* menampilkan objek 3D dapat dilihat pada Tabel 3.11.

Tabel 3.11 Skenario Use Case Menampilkan Objek 3D

Nama Use Case	**Menampilkan Objek 3D**
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 rumah	
Pemicu	Marker dapat dekenali dengan baik dan sesuai dengan pola *marker* yang ada	
Aliran Utama	**Langkah**	**Aksi**
	1	User mengarahkan *marker*
	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**
	5.2	**Sistem menampilkan informasi rumah**

4. Skenario Use Case Merotasi Objek 3D

   Skenario *use case* merotasi objek 3D dapat dilihat pada Tabel 3.12.

Tabel 3.12 Skenarion Use Case Merotasi Objek 3D

Nama Use Case	**Merotasi Objek 3D**	
Persyaratan Terkait	-	
Tujuan Dalam Konteks	Sistem merotasi objek 3D	
Kondisi Sebelum	Objek 3D sudah ditampilkan	
Keberhasilan Kondisi Akhir	Sistem dapat melakukan rotasi objek 3D yang ditampilkan	
Kegagalan Kondisi Akhir	Sistem tidak dapat merotasi objek 3D	
Aktor Utama	User	
Aktor Kedua	Marker	
Pemicu	Objek 3D sudah dapat ditampilkan dan pengguna menekan tombol rotasi	
Aliran Utama	**Langkah**	**Aksi**
	1	User menekan tombol rotasi
	2	Sistem melakukan rotasi objek 3D
Perluasan	**Langkah**	**Aksi Percabangan**
	2.1	Sistem tidak dapat melakukan rotasi objek 3D

5. Skenario Use Case Menampilkan Informasi Rumah

   Skenario use case menampilkan informasi rumah dapat dilihat pada Tabel 3.13.

Tabel 3.13 Skenario Use Case Menampilkan Informasi Rumah

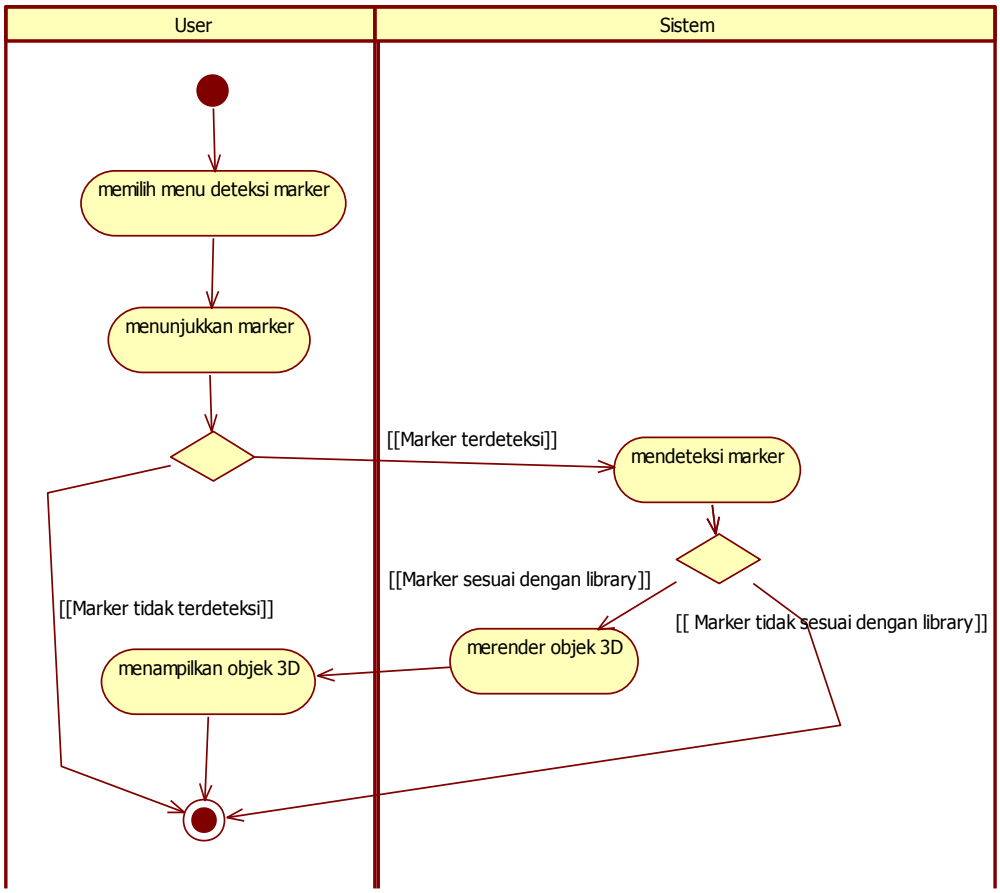
Nama Use Case	**Menampilkan informasi rumah**	
Persyaratan Terkait	-	
Tujuan Dalam Konteks	Sistem menampilkan informasi rumah	
Kondisi Sebelum	Objek 3D sudah ditampilkan	
Keberhasilan Kondisi Akhir	Sistem dapat menampilkan informasi rumah dari objek 3D yang ditampilkan	
Kegagalan Kondisi Akhir	Sistem tidak dapat menampilkan informasi rumah	
Aktor Utama	User	
Aktor Kedua	Marker	
Pemicu	Objek 3D sudah dapat ditampilkan dan pengguna menekan tombol detail	
Aliran Utama	**Langkah**	**Aksi**
	1	User menekan tombol detail
	2	Sistem menampilkan informasi rumah
Perluasan	**Langkah**	**Aksi Percabangan**
	2.1	Sistem tidak dapat menampilkan informasi rumah

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. *Activity diagram* dari aplikasi yang akan dibangun adalah sebagai berikut:

1. Activity Diagram Pembaca Marker

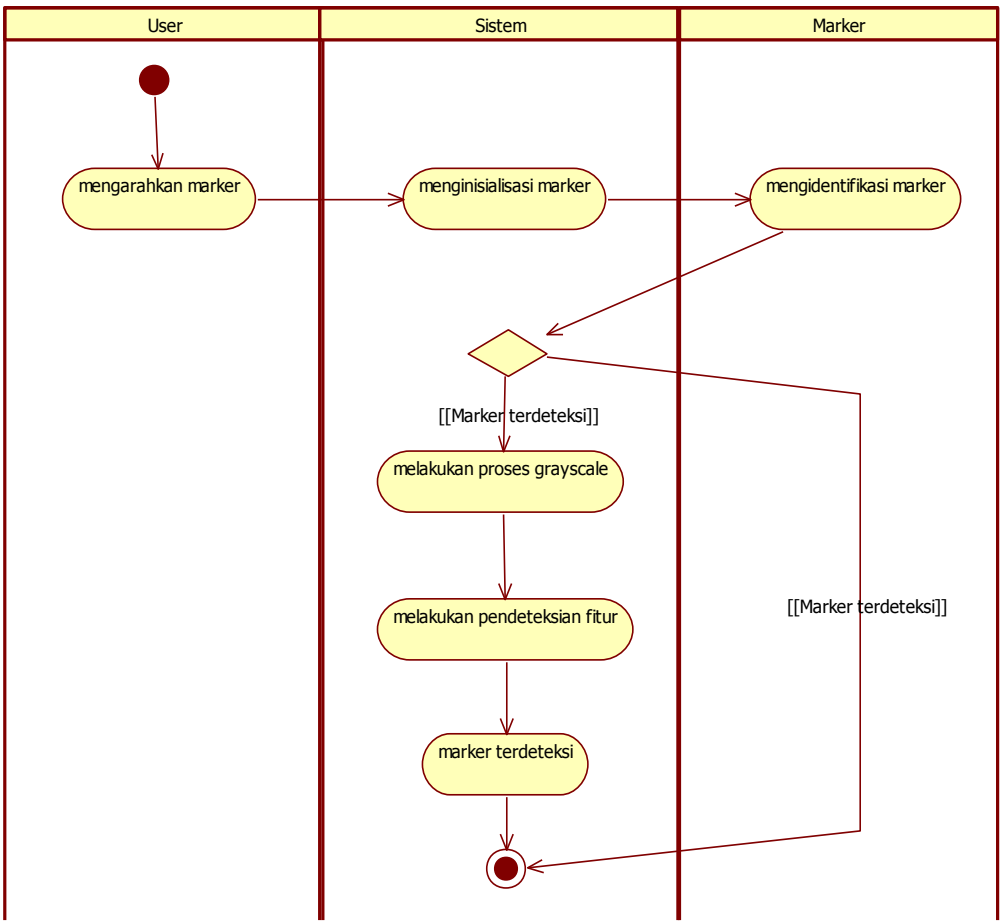
   Merupakan aktifitas dari menu pilihan deteksi *marker* untuk membaca *marker*. Activity diagram dari pembacaan *marker* dapat dilihat pada Gambar 3.10.



Gambar 3.10 Activity Diagram Pembacaan Marker

2. Activity Diagram Pendeteksian Marker

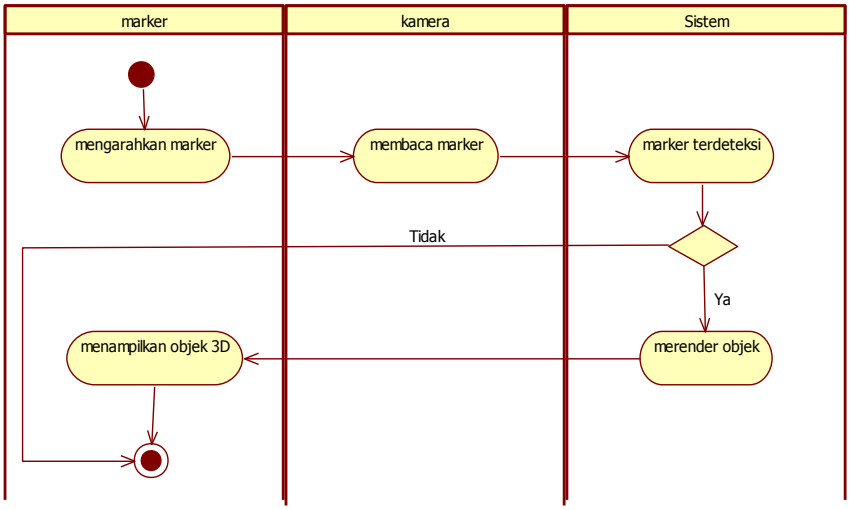
   Merupakan sub-aktifitas dari *use case* pendeteksian *marker*. *Activity diagram* dari pendeteksian *marker* dapat dilihat pada Gambar 3.11.



Gambar 3.11 Activity Diagram Pendeteksian Marker

3. Activity Diagram Menampilkan Objek 3D

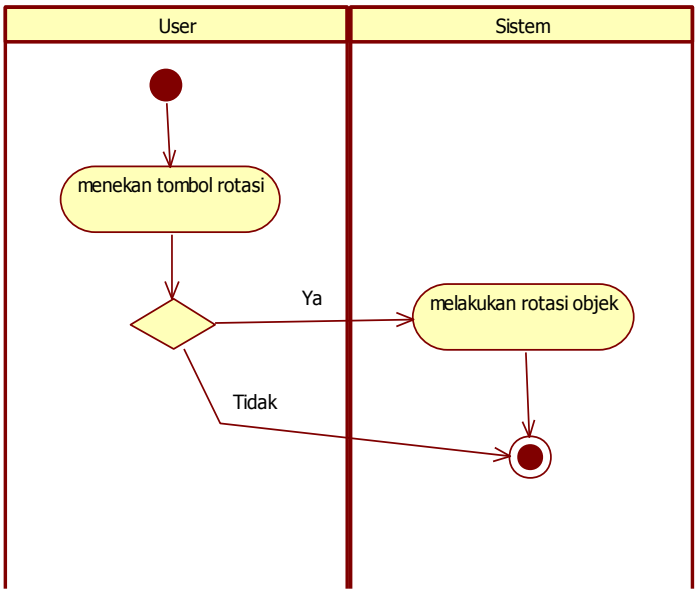
   Merupakan sub-aktifitas dari *use case* menampilkan objek 3D. *Activity diagram* dari menampilkan objek dapat dilihat pada Gambar 3.12.



Gambar 3.12 Activity Diagram Menampilkan Objek 3D

4. Activity Diagram Merotasi Objek 3D

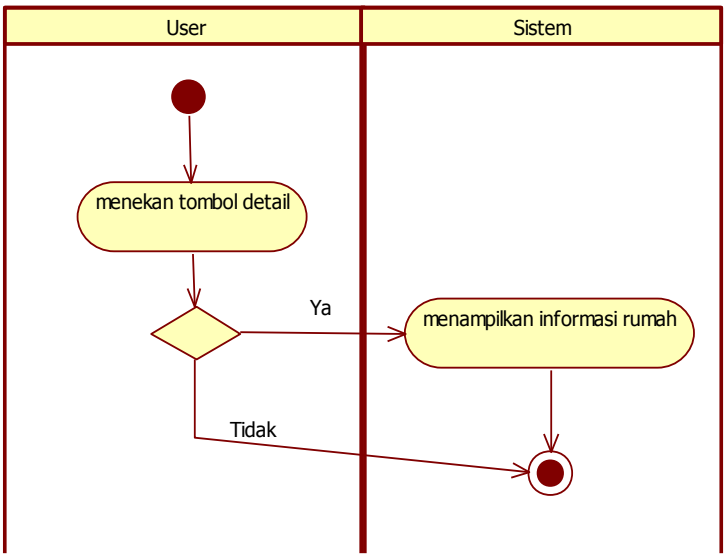
   Merupakan sub-aktifitas dari *use case* merotasi objek 3D. *Activity diagram* dari merotasi objek 3D dapat dilihat pada Gambar 3.13.



Gambar 3.13 Activity Diagram Merotasi Objek 3D

5. Activity Diagram Menampilkan Informasi Rumah

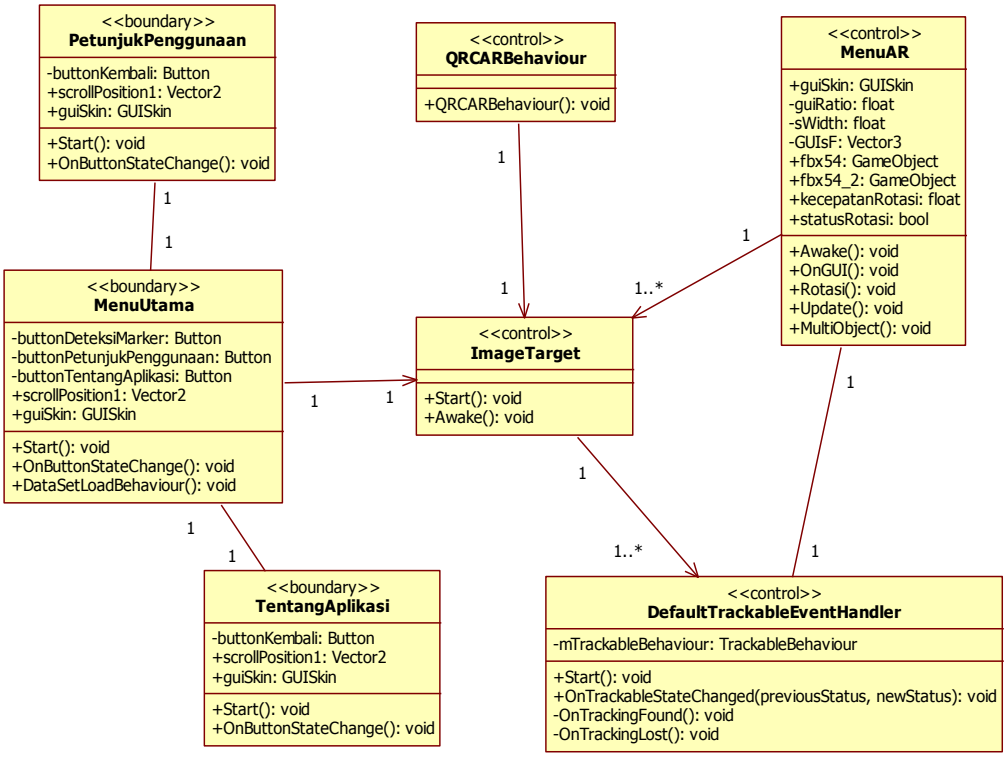
   Merupakan sub-aktifitas dari *use case* menampilkan informasi rumah. *Activity diagram* dari menampilkan informasi rumah dapat dilihat pada Gambar 3.14.



Gambar 3.14 Activity Diagram Menampilkan Informasi Rumah

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. Kelas diagram dari aplikasi AR dapat dilihat pada Gambar 3.15.



Gambar 3.15 Class Diagram Aplikasi AR KSU

Tabel deskripsi kelas pada Gambar 3.15 dapat dilihat pada Tabel 3.14.

Tabel 3.14 Deskripsi Kelas

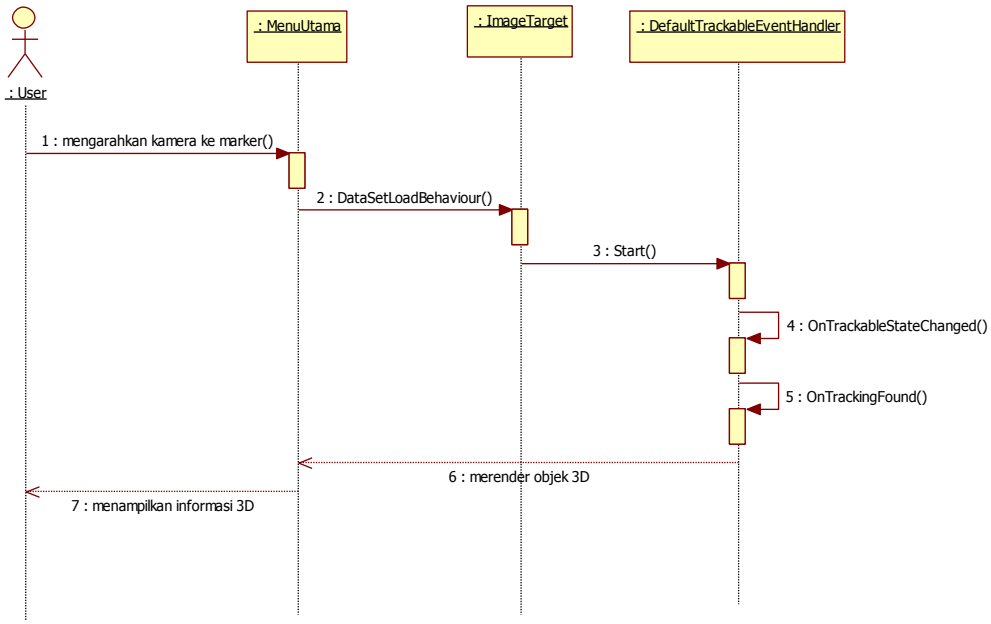
No	Nama Kelas	Tipe Kelas
1	MenuUtama	Boundary
2	TentangAplikasi	Boundary
3	PetunjukPenggunaan	Boundary
4	ImageTarget	Control
5	QRCARBehaviour	Control
6	MenuAR	Control
7	DefaultTrackableEventHandler	Control

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. *Sequence diagram* dari aplikasi yang akan dibangun adalah sebagai berikut:

1. Sequence Diagram Pembacaan Marker

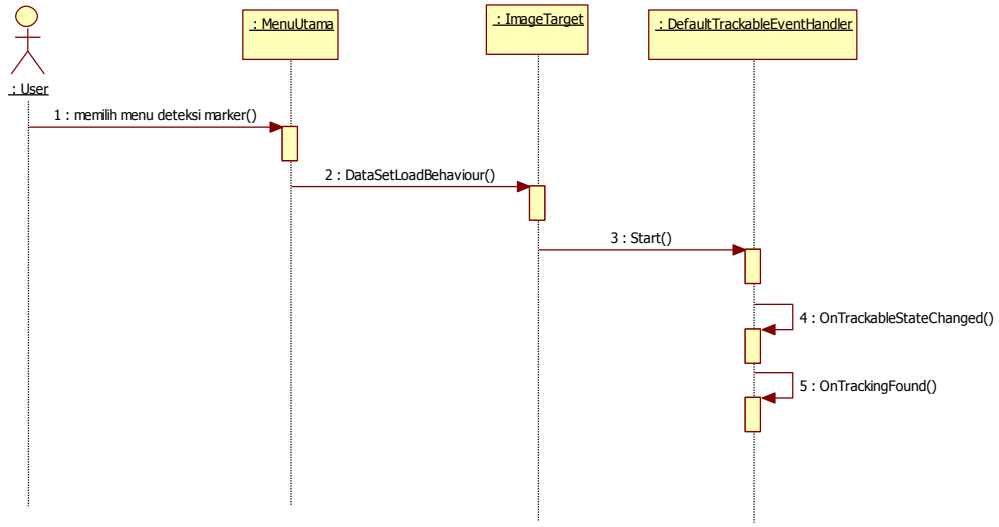
   Sequence diagram untuk pembacaan *marker* dapat dilihat pada Gambar 3.16.



Gambar 3.16 Sequence Diagram Pembacaan Marker

2. Sequence Diagram Pendeteksian Marker

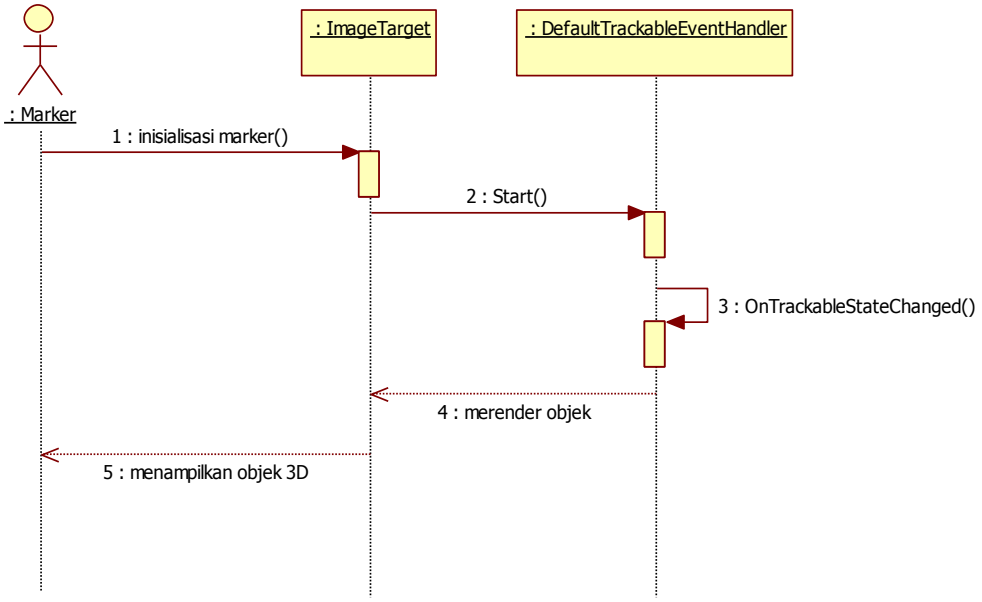
   Sequence diagram untuk pendeteksian *marker* dapat dilihat pada Gambar 3.17.



Gambar 3.17 Sequence Diagram Pendeteksian Marker

3. Sequence Diagram Menampilkan Objek

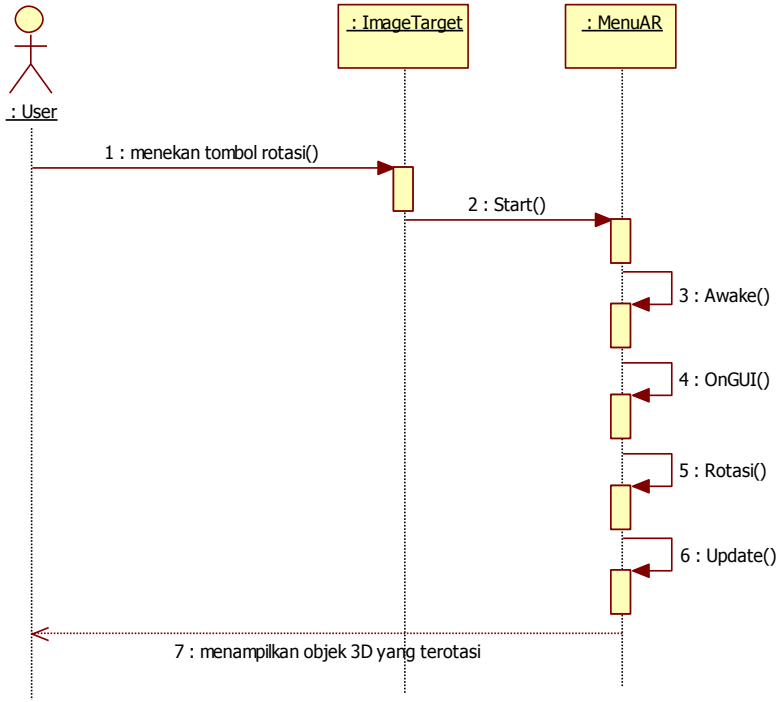
   Sequence diagram untuk menampilkan objek 3D dapat dilihat pada Gambar 3.18.



Gambar 3.18 Sequence Diagram Menampilkan Objek

4. Sequence Diagram Merotasi Objek 3D

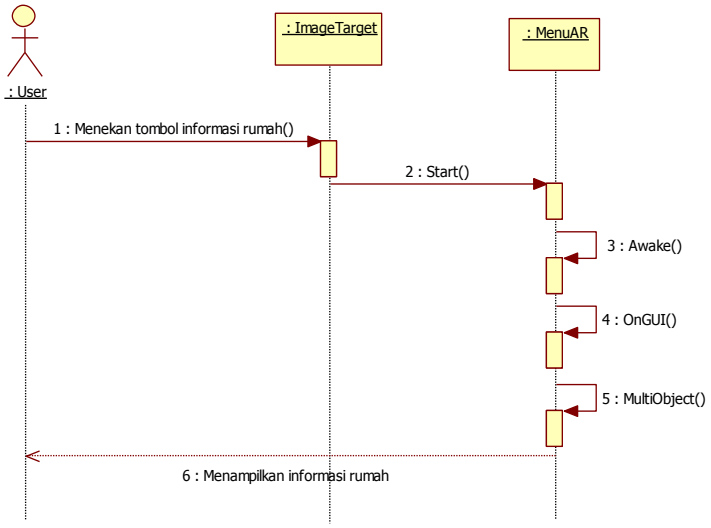
   Sequence diagram untuk merotasi objek 3D dapat dilihat pada Gambar 3.19.



Gambar 3.19 Sequence Diagram Merotasi Objek 3D

5. Sequence Diagram Menampilkan Informasi Rumah

   Sequence diagram untuk menampilkan informasi rumah dapat dilihat pada Gambar 3.20.



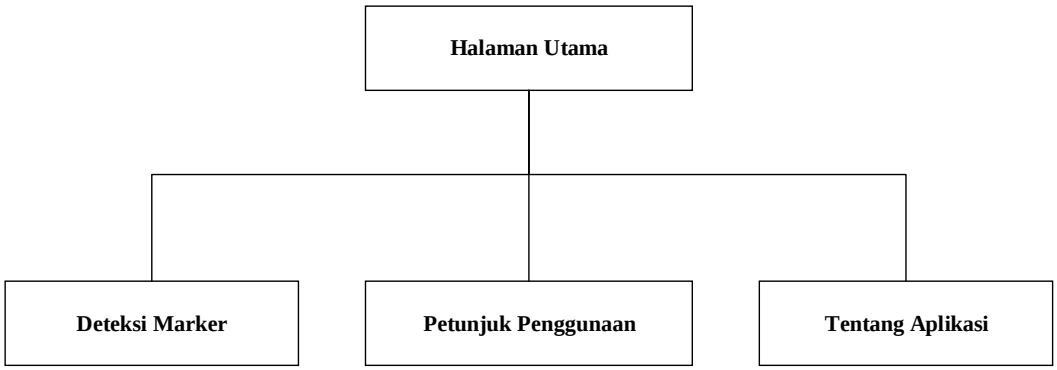
Gambar 3.20 Sequence Diagram Menampilkan Informasi Rumah

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 struktur *menu* merupakan gambaran jalur pemakaian dari aplikasi sehingga aplikasi menjadi mudah dimengerti serta mudah digunakan dan juga saat menggunakan aplikasi, pengguna tidak mengalami banyak kesulitan dalam memilih *menu-menu* yang diinginkan. Perancangan struktur menu pada aplikasi ini dapat dilihat pada Gambar 3.21.



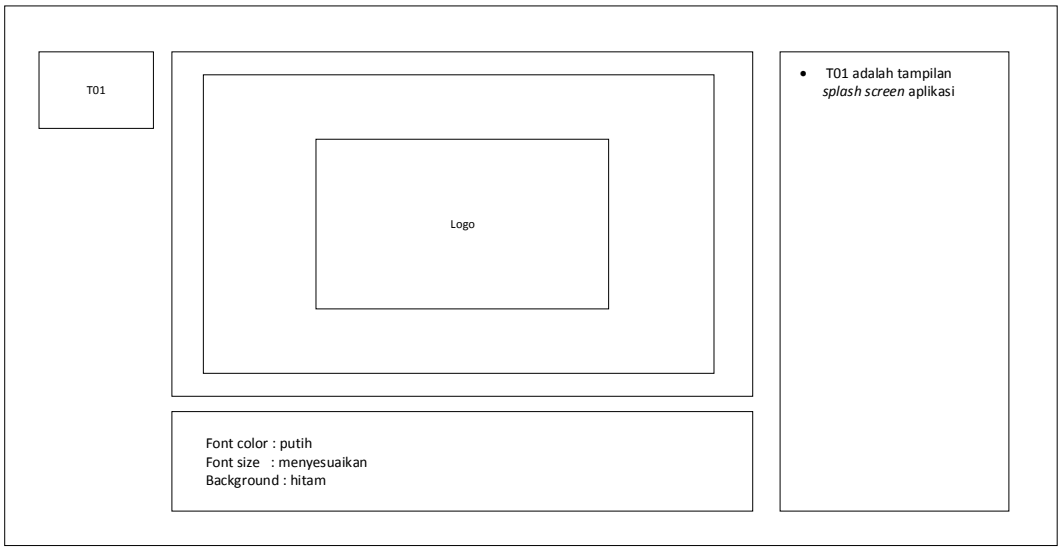
Gambar 3.21 Perancangan Struktur Menu

3.2.2 Perancangan Antarmuka

Perancangan antarmuka merupakan rancangan tampilan dari aplikasi. Perancangan antarmuka dari *splash screen*, menu utama, deteksi *marker*, petunjuk penggunaan, tentang aplikasi, dan menampilkan objek dapat dilihat sebagai berikut:

1. Perancangan Antarmuka Splash Screen

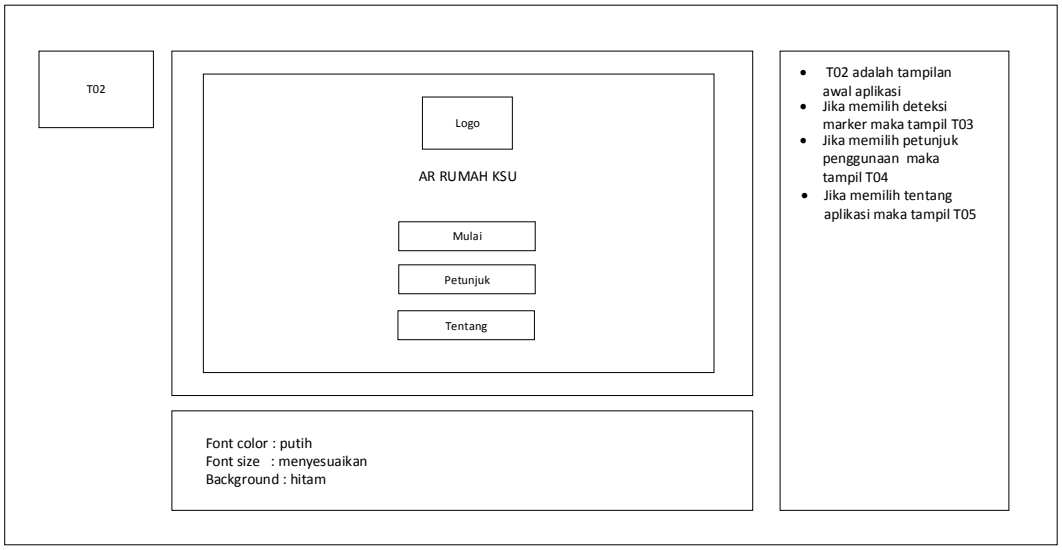
   Perancangan antarmuka splash screen aplikasi ini dapat dilihat pada Gambar 3.22.



Gambar 3.22 Perancangan Antarmuka Splash Screen

2. Perancangan Antarmuka Utama

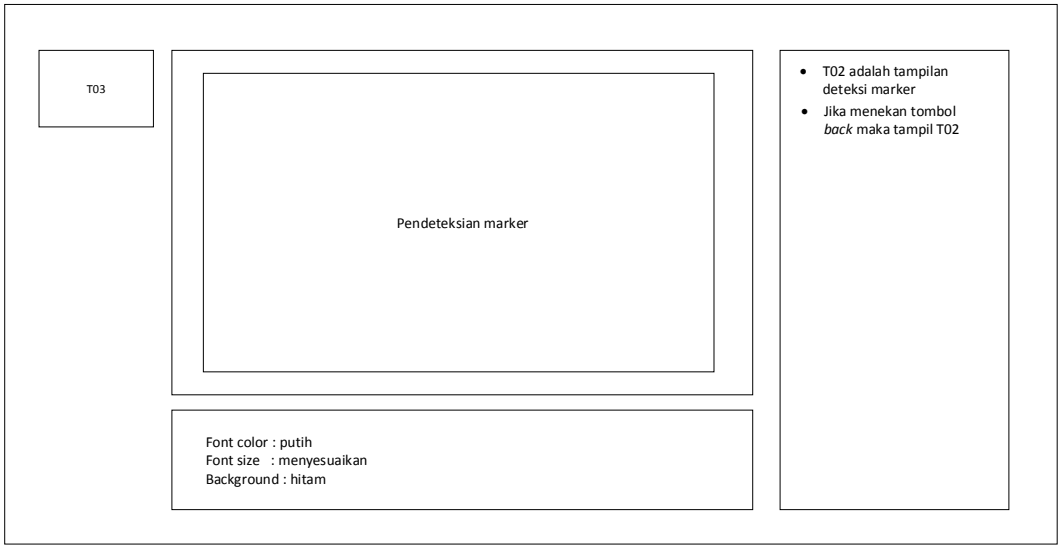
   Perancangan antarmuka utama aplikasi ini dapat dilihat pada Gambar 3.23.



Gambar 3.23 Perancangan Antarmuka Utama

3. Perancangan Antarmuka Deteksi Marker

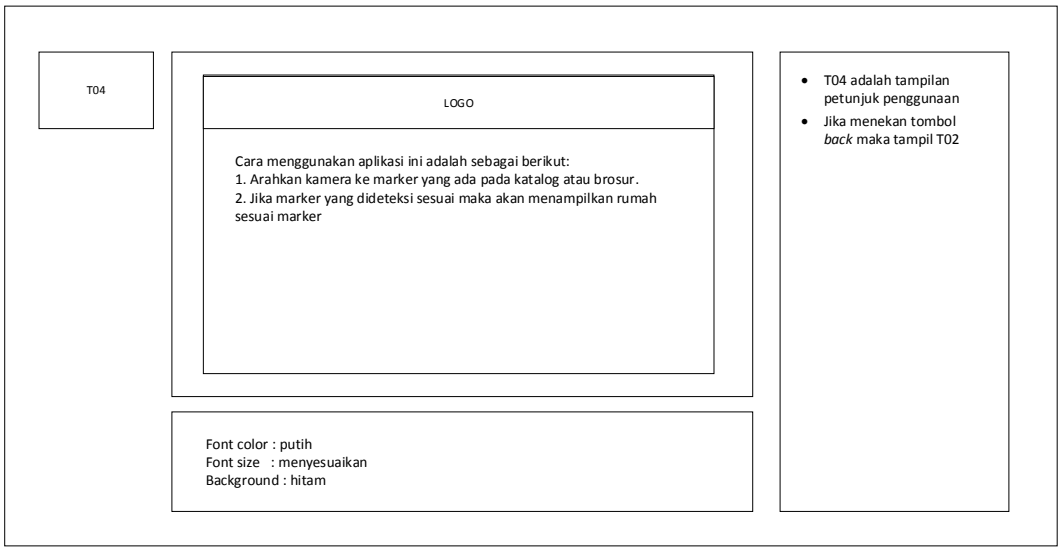
   Perancangan antarmuka deteksi *marker* aplikasi ini dapat dilihat pada Gambar 3.24.



Gambar 3.24 Perancangan Antarmuka Deteksi Marker

4. Perancangan Antarmuka Petunjuk Penggunaan

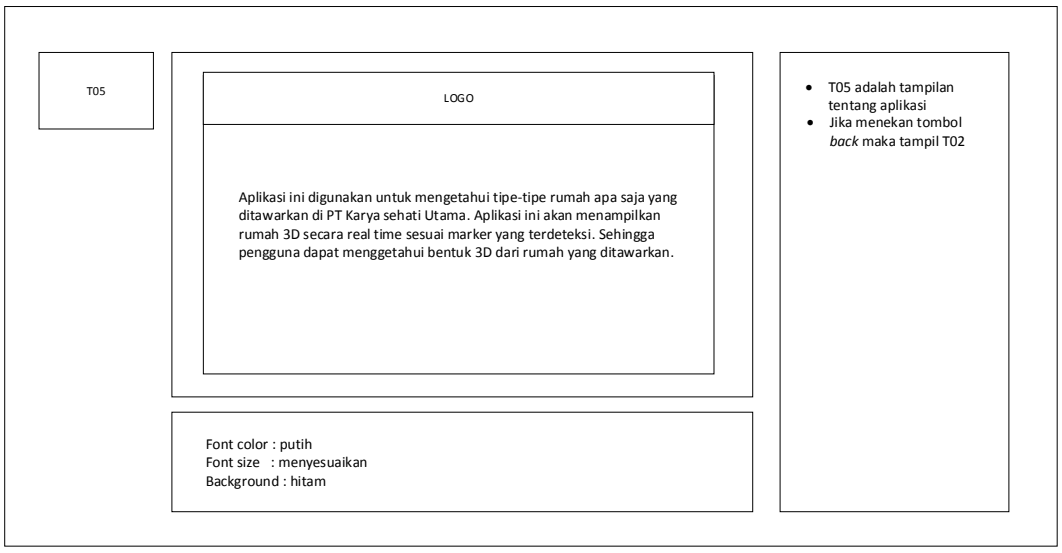
   Perancangan antarmuka petunjuk penggunaan aplikasi ini dapat dilihat pada Gambar 3.25.



Gambar 3.25 Perancangan Antarmuka Petunjuk Penggunaan

5. Perancangan Antarmuka Tentang Aplikasi

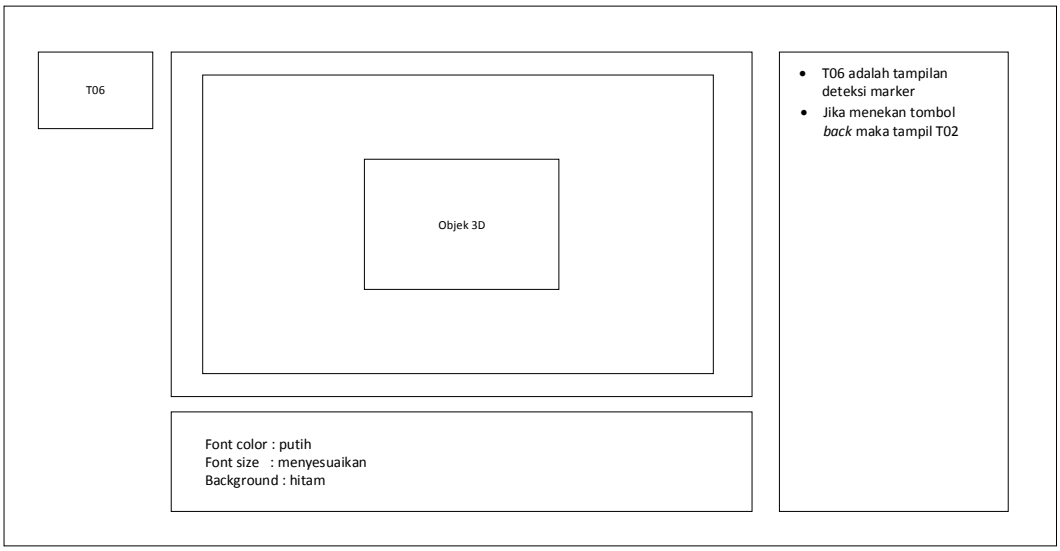
   Perancangan antarmuka tentang aplikasi ini dapat dilihat pada Gambar 3.26.



Gambar 3.26 Perancangan Antarmuka Tentang Aplikasi

6. Perancangan Antarmuka Menampilkan Objek

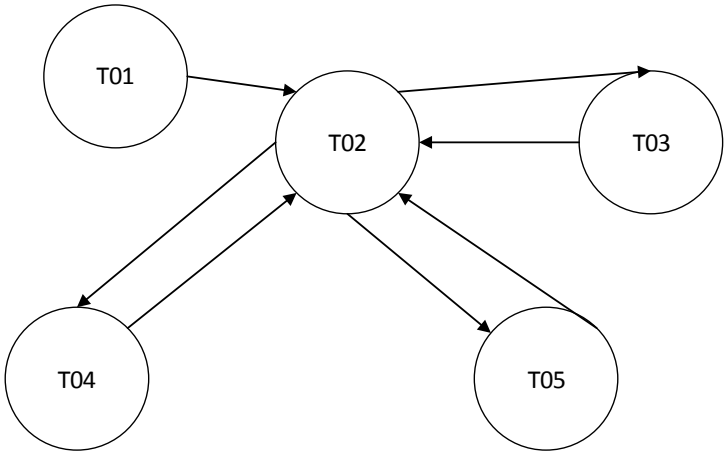
   Perancangan antarmuka menampilkan objek pada aplikasi ini dapat dilihat pada Gambar 3.27.



Gambar 3.27 Perancangan Antarmuka Menampilkan Objek

3.2.3 Jaringan Semantik

Jaringan semantik dari aplikasi dapat dilihat pada Gambar 3.28.



Gambar 3.28 Jaringan Semantik

Ketarangan:

T01 : Tampilan *splash screen*

T02 : Halaman utama

T03 : Halaman untuk mendeteksi *marker*

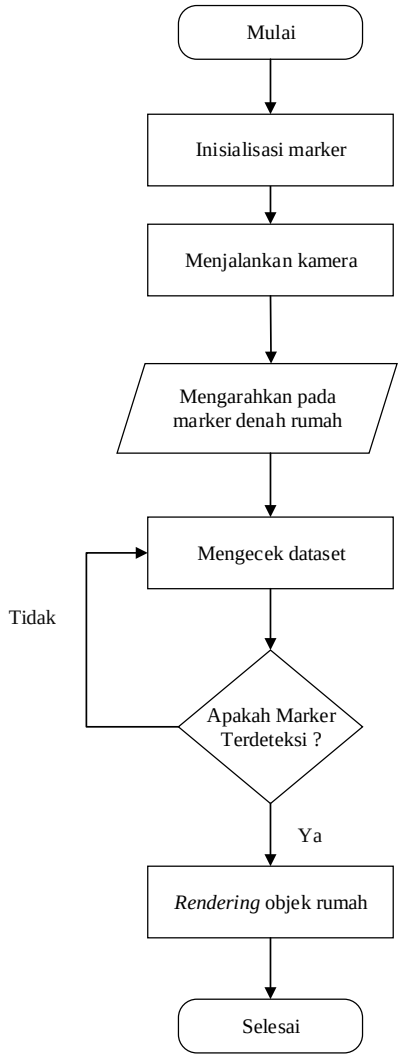
T04 : Halaman untuk menampilkan petunjuk penggunaan

T05 : Halaman untuk menampilkan informasi tentang aplikasi

3.2.4 Perancangan Method

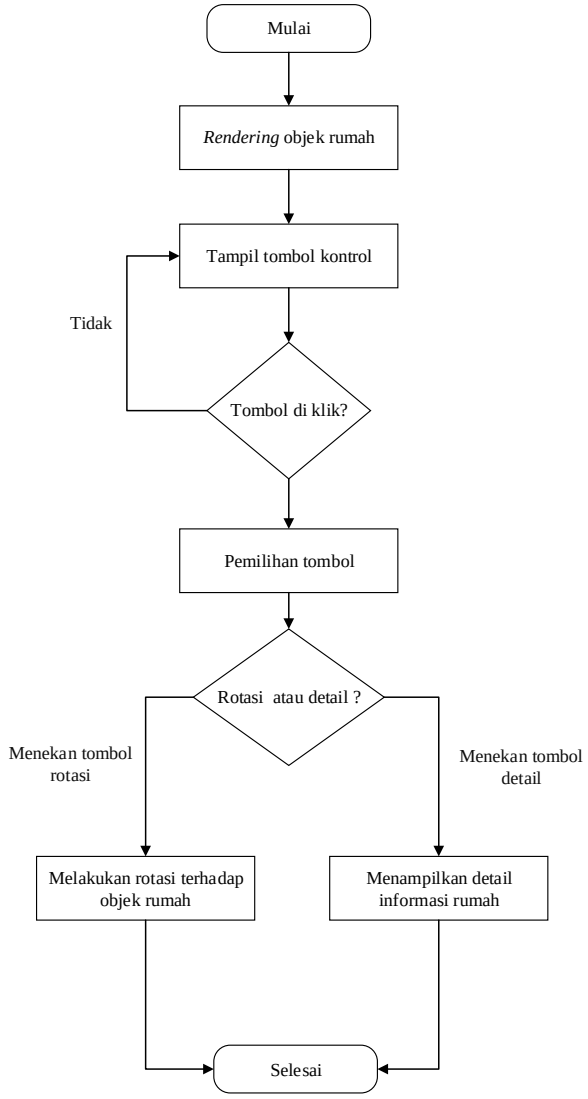
Perancangan *method* berfungsi untuk mendeskripsikan *method* yang berada dalam aplikasi. *Method* yang terdapat pada aplikasi yang akan dibangun yaitu *flowchart* menampilkan objek rumah dan *flowchart* mengontrol objek. *Method* atau *flowchart* tersebut dapat dilihat sebagai berikut:

1. Flowchart Menampilkan Objek Rumah



Gambar 3.29 Flowchart Menampilkan Objek Rumah

2. Flowchart Mengontrol Objek Rumah



Gambar 3.30 Mengontrol Objek Rumah