BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem merupakan penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan. Analisis merupakan tahapan yang paling penting, karena kesalahan dalam tahap ini akan menyebabkan kesalahan di tahap selanjutnya.

Dari hasil analisis tersebut dapat dirancang atau diperbaiki menjadi sebuah sistem yang lebih efektif dan efisien. Pada tahap analisis sistem ini dibagi menjadi beberapa bagian, yaitu sebagai berikut :

1. Analisis Masalah.
2. Analisis Sistem yang Sedang Berjalan
3. Analisis Arsitektur Sistem
4. Analisis Alur Sistem
5. Analisis Marker
6. Analisis Metode
7. Analisis Kebutuhan Non Fungsional
8. Analisis Kebutuhan Fungsional

3.1.1 Analisis Masalah

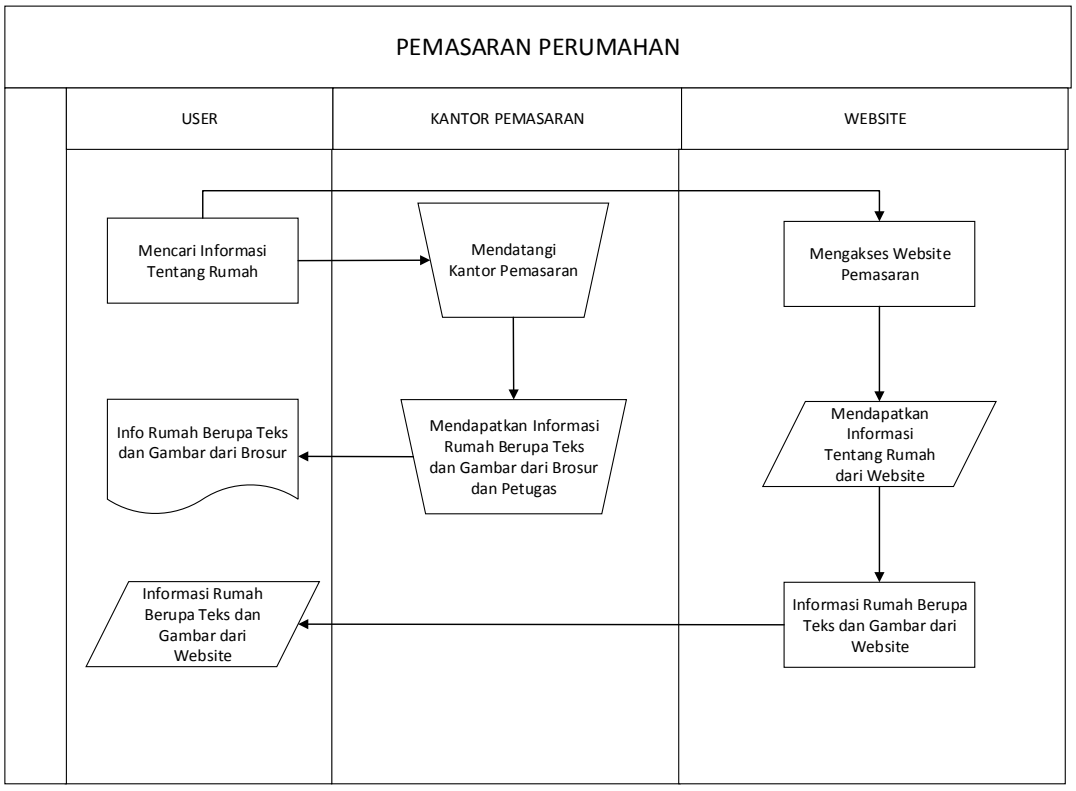
Analisis masalah adalah langkah awal dari analisis sistem. Langkah ini diperlukan untuk mengetahui pemasalahan apa saja yang terjadi didalam sistem yang telah berjalan.

Berdasarkan hasil kuesioner dan studi pustaka yang telah dilakukan dengan membaca penelitian-penelitian terdahulu, terdapat beberapa masalah yang diangkat sebagai dasar pembangunan aplikasi yang akan dibangun yaitu media brosur yang digunakan oleh pihak perusahaan tidak lebih hanya menampilkan

obyek bangunan dalam bentuk foto atau dua dimensi (2D) untuk memberikan gambaran kepada para calon pembelinya. Sehingga para calon pembeli tidak dapat melihat bentuk rumah yang ditawarkan dari berbagai sudut ruangan. Adapun masalah lainnya yaitu masih menggunakannya perangkat komputer/laptop untuk menjalankan teknologi *Augmented Reality* (AR) sebagai media penyampaian informasi di bidang properti dan masih menggunakannya *marker* khusus untuk menampilkan obyek bangunan yang ditawarkan oleh perusahaan tersebut.

3.1.2 Analisis Sistem yang Sedang Berjalan

Salah satu tahapan analisis sistem yaitu tahapan yang memberi gambaran tentang sistem yang sedang berjalan saat ini. Analisis ini bertujuan untuk memberikan gambaran bagaimana cara kerja dari sistem yang sedang berjalan. Prosedur yang sedang berjalan saat ini adalah sebagai berikut:



Gambar 3. 1 Flow Map Sistem yang sedang berjalan

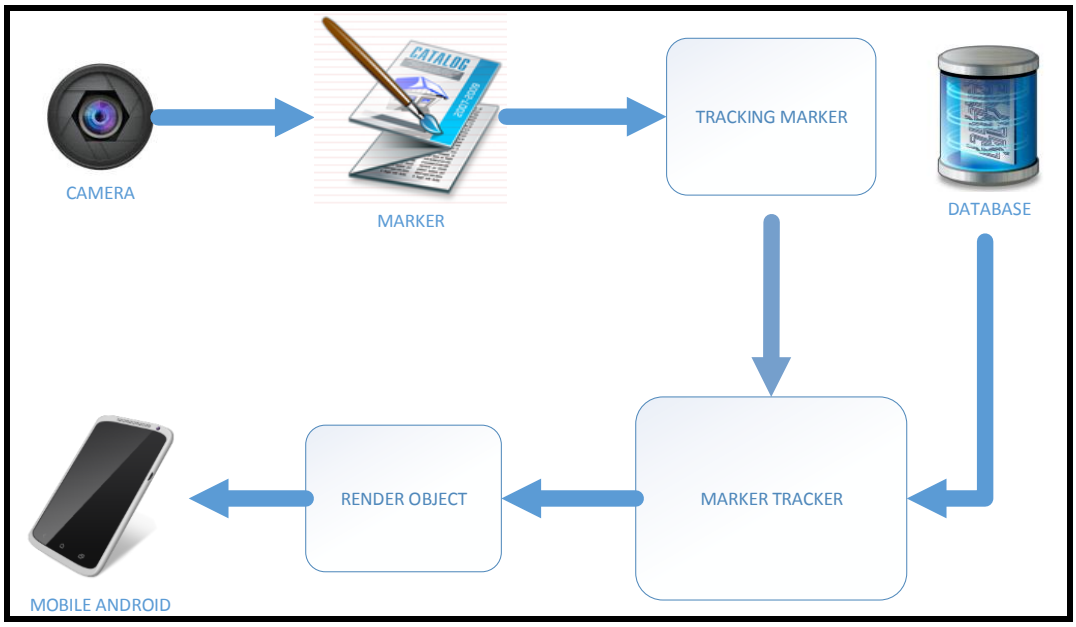
1. *User* adalah calon pembeli rumah.
2. Terdapat 2 media yang sering digunakan untuk mengetahui informasi tentang bangunan, diantaranya :
   1) Brosur : Media ini cukup mudah untuk mengetahui informasi tentang rumah-rumah yang tersedia, penyampaian informasi pada brosur ini berupa gambar 2D dan teks.
   2) Website : Media ini terkoneksi dengan internet. Pada website perusahaan biasa tersedia informasi dan gambar 2D tentang rumah yang akan dijual.

Dari gambaran prosedur media yang didapat untuk mengakses berbagai informasi pada brosur dan website, teknologi augmented reality dapat dijadikan sebagai media alternatif sehingga penyampaian informasi tentang rumah yang akan dijual menjadi lebih optimal dan efisien.

3.1.3 Analisis Arsitektur Sistem

Sistem yang akan dibangun adalah suatu aplikasi dalam bentuk prototype untuk menampilkan informasi berupa obyek tiga dimensi pada denah suatu bangunan dengan menggunakan teknologi *augmented reality*. Proses aplikasi tersebut akan dibangun dengan menggunakan teknik *markerless* yang di*support* oleh *Library* Vuforia Qualcomm.

Pada arsitektur aplikasi yang akan dibangun terdiri dari beberapa komponen, yaitu : *user* adalah pengguna yang menggunakan aplikasi tersebut, *user* mengarahkan kamera yang ada pada perangkat *mobile*, sehingga marker dapat tertangkap olah kamera. Kemudian dari gambar yang didapat dari kamera sistem melakukan *tracking* marker untuk mengidentifikasi marker yang digunakan oleh pengguna. sistem melakukan *render* obyek 3D yang digunakan dalam aplikasi. *User* dapat melihat hasil manipulasi system melalui layar perangkat *mobile* yang digunakan. Gambar 3.2 menunjukan deskripsi proses sistem yang akan dibangun.



Gambar 3. 2 Arsitektur Sistem

3.1.4 Analisis Alur Sistem

Analisis alur sistem merupakan analisis yang mendeskripsikan bagaimana proses *augmented reality* dari awal inisialisasi, *tracking* marker, sampai dengan proses *rendering* obyek 3D. Secara garis besarnya, dalam perancangan ada tiga bagian utama yaitu sebagai berikut :



Gambar 3. 3 Alur sistem

3.1.5 Analisis Marker

Tahapan dalam menganalisis bentuk dan isi pada marker yang digunakan serta proses pembuatannya dan *marker* seperti apa saja yang digunakan.

3.1.5.1 Pengolahan Marker

Dalam tahapan proses pembangunan aplikasi berbasis teknologi *Augmented Reality (AR)* di perlukan sebuah marker untuk proses

pengaplikasiannya. Adapun cara dalam dalam pembuatannya, pertama adalah dengan menentukan sampel gambar yang ingin dijadikan sebagai marker. Dalam hal ini sampel gambar yang digunakan sebagai marker adalah denah suatu bangunan dengan format .jpg seperti pada gambar 3.4.

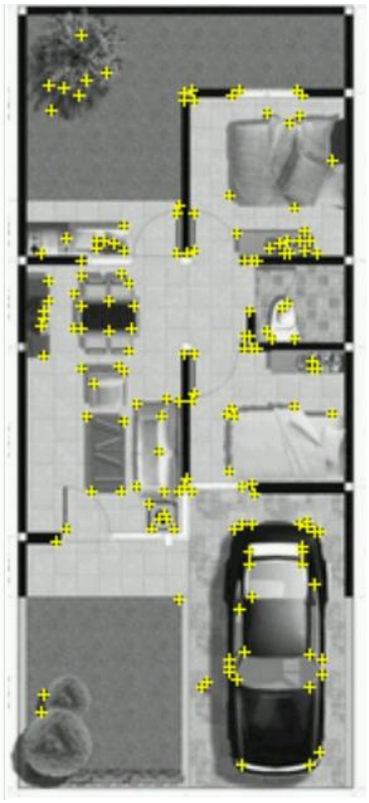


Gambar 3. 4 Sampel Marker yang Digunakan

Tahapan selanjutnya, setelah menentukan gambar yang akan digunakan sebagai marker maka perlu dilakukannya proses meng-*upload* gambar tersebut ke situs *developer* vuforia untuk kemudian diproses agar gambar tersebut dapat dikenali dan dimasukkan ke dalam sistem yang akan dibangun. Setelah selesainya proses pengenalan gambar, tahap selanjutnya adalah men-*download* gambar tersebut dengan format .*unitypackage*.

Tools pendukung yang digunakan dalam pembangunan aplikasi ini adalah menggunakan Unity3D, maka format gambar yang sebelumnya adalah .jpg, setelah di *download* kembali akan berupa .*unitypackage* agar gambar tersebut dapat dikenali oleh *tools* pendukung dan dijadikan sebagai marker pada

aplikasi yang akan dibangun. Berikut adalah gambar yang menunjukan dari hasil *tracking marker* dengan Vuforia.



Gambar 3. 5 Hasil *Tracking Marker* dengan Vuforia

3.1.6 Analisis Metode

3.1.6.1 NFT (*Natural Feature Tracking*)

Natural Feature Tracking atau yang biasa disebut NFT merupakan metode yang digunakan di dalam *Library* Vuforia Qualcomm untuk mendeteksi pola gambar. Adapun algoritma yang digunakan di dalam NFT itu sendiri adalah SIFT dan FERNS.

3.1.6.1.1 SIFT (*Scale Invariant Feature Tracking*)

Pengujian metode SIFT (*Scale Invariant Feature Transform*) pada obyek denah bangunan dilakukan untuk mencari titik-titik yang nantinya digunakan sebagai pengenalan pola dari gambar denah bangunan tersebut.

Algoritma SIFT sendiri terbagi atas beberapa bagian yaitu :

1. Pembentukan *Gaussian Scale Space* dan *Difference of Gaussian Scale Space*.

Mencari nilai Gaussian menggunakan rumus (2.1) :

$$G(x,y,k\sigma) = \frac{1}{2\pi(k\sigma)^2} e^{-(x^2+y^2)/2k^2\sigma^2}$$

Dimana,

$k = \sqrt{2}$

$\sigma = 1{,}6$

$e = 2{,}72$ (Nilai Ketetapan)

Dimisalkan dalam metode SIFT ini,

Nilai x = 2 (Nilai Titik Piksel)

Nilai y = 3 (Nilai Titik Piksel)

Maka untuk mencari nilai Gaussian (G),

$$G(x,y,k\sigma) = \frac{1}{2(3{,}14)(\sqrt{2}*1{,}6)^2} 2{,}72^{-(2^2+3^2)/2\sqrt{2}^2*1{,}6^2}$$

$$G(x,y,k\sigma) = \frac{1}{6{,}28(2{,}26)^2} 2{,}72^{-(4+9)/4*2{,}56}$$

$$G(x,y,k\sigma) = \frac{1}{6{,}28*5{,}11} 2{,}72^{-13/10{,}24}$$

$$G(x,y,k\sigma) = \frac{1}{32{,}09} 2{,}72^{-1{,}27}$$

$$G(x,y,k\sigma) = \frac{1}{32{,}09} 0{,}28$$

$$G(x,y,k\sigma) = \frac{0{,}28}{32{,}09}$$

$$G(x,y,k\sigma) = 0{,}0087$$

Setelah mendapatkan nilai dari *Gaussian Scale Space* langkah selanjutnya yaitu membentuk *Difference of Gaussian Scale (DoG)*. *DoG* dibentuk dengan menggunakan rumus (2.2) :

$$D(x,y,\sigma) = L\,(x,y,k\sigma) - L\,(x,y,\sigma)$$

Keterangan :

$L\ (x,y,\sigma)$ = $G\ (x,y,\sigma)$ pada skala σ.

$L\ (x,y,k\sigma)$ = $G\ (x,y,k\sigma)$ pada skala $k\sigma$ dengan nilai $k = \sqrt{2}$.

$$G(x,y,\sigma) = \frac{1}{2\pi(\sigma)^2} e^{-(x^2+y^2)/2\sigma^2}$$

$$G(x,y,\sigma) = \frac{1}{2 * 3{,}14 * (1{,}6)^2} 2{,}72^{-(2^2+3^2)/2*1{,}6^2}$$

$$G(x,y,\sigma) = \frac{1}{6{,}28 * 2{,}56} 2{,}72^{-(4+9)/2*2{,}56}$$

$$G(x,y,\sigma) = \frac{1}{6{,}28 * 2{,}56} 2{,}72^{-13/5{,}12}$$

$$G(x,y,\sigma) = \frac{1}{16{,}07} 2{,}72^{-2{,}54}$$

$$G(x,y,\sigma) = \frac{1}{16{,}07} 0{,}079$$

$$G(x,y,\sigma) = \frac{0{,}079}{16{,}07}$$

$$G(x,y,\sigma) = 0{,}0049$$

Jadi, diperoleh nilai *Difference of Gaussian Scale* :

$$D(x,y,\sigma) = L\ (x,y,k\sigma) - L\ (x,y,\sigma)$$

$$D(x,y,\sigma) = 0{,}0087 - 0{,}0049$$

$$D(x,y,\sigma) = 0{,}0038$$

Dari hasil yang telah diperoleh dari perhitungan DoG (*Difference of Gaussian Scale*), maka langkah selanjutnya adalah mencari nilai ekstremum yang ada pada langkah kedua.

2. Deteksi ekstremum (maksimum atau minimum) pada *Difference of Gaussian Scale Space*.

Setelah mendapatkan titik-titik ekstremum maka perlu ditingkatkan lokalisasinya dengan akurasi subpiksel dengan menggunakan ekspansi Taylor orde kedua dari fungsi ruang skala, sehingga posisi ekstremum sebenarnya didapatkan dengan rumus (2.3) :

$$z = -\left(\frac{\partial^2 D}{\partial x^2}\right)^{-1} \frac{\partial D}{\partial x}$$

Apabila sudah diperoleh nilai dari hasil perhitungan deteksi ekstremum, maka dilanjutkan pada langkah ketiga dengan menghitung nilai *keypoint*-nya.

3. Menghitung nilai *keypoint* pada ekstremum digunakan rumus (2.4), yaitu :

$$D(z) = D + \frac{1}{2}\frac{\partial D^{-1}}{\partial x}\, z$$

Jika nilai *D(z)* tidak melebihi suatu nilai *threshold* (0.03) maka *keypoint* tersebut dihilangkan dan tidak dipakai lagi. Penghapusan *keypoint* yang tidak memenuhi syarat *threshold* dilakukan juga penghapusan *keypoint* tak stabil yang berada pada daerah *edge*.

4. Penetapan orientasi *keypoint* (*orientation assignment*).

Pada penetapan orientasi ini digunakan citra Gaussian *smooth L* yang memiliki skala paling dekat dengan skala *keypoint*. Untuk setiap citra sampel *L(x,y) magnitude m(x,y)* dan orientasi *θ(x,y)* dihitung dengan menggunakan rumus (2.5) dan rumus (2.6) :

$$m(x,y) = \sqrt{\big(L(x+1,y) - L(x-1,y)\big)^2 + \big(L(x,y+1) - L(x,y-1)\big)^2}$$

$$\theta(x,y) = \arctan\left(\frac{L(x,y+1) - L(x,y+1)}{L(x+1,y) - L(x-1,y)}\right)$$

Setelah seluruh langkah terpenuhi, maka hasil yang diperoleh dari perhitungan tersebut adalah menghasilkan sebuah titik-titik yang nantinya akan menjadi keypoint atau titik bunga seperti pada gambar *Output Natural Feature Tracking* yang kemudian akan dikenali oleh sistem.

3.1.6.1.2 Ferns : *Tracking by Classification*

Untuk meningkatkan kecepatan dalam mendeteksi "feature" dan melakukan klasifikasi "marker/image target". Dapat dilakukan perhitungan dengan menggunakan teorema Bayes rumus (2.7).

$$P(C|\{Fi\}) = P(\{Fi\}|C)P(C)/P(\{Fi\})$$

Dengan begitu penggabungan algoritma SIFT dan Ferns pada metode NFT (Natural Feature Tracking) ini akan menghasilkan proses *tracking marker* yang baik dan cepat.

3.1.7 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional menggambarkan kebutuhan luar sistem yang diperlukan untuk menjalankan aplikasi yang dibangun. Adapun kebutuhan non-fungsional pada aplikasi meliputi kebutuhan perangkat keras, kebutuhan perangkat lunak dan pengguna sistem yang akan memakai aplikasi.

3.1.7.1 Analisis Kebutuhan Perangkat Keras

Analisis Perangkat keras atau *hardware* merupakan salah satu hal yang penting karena tanpa *hardware* yang memenuhi syarat, program yang akan dibuat tidak akan dapat berjalan. Spesifikasi perangkat keras yang dapat dipergunakan untuk membangun aplikasi ini adalah sebagai berikut :

Tabel 3.1 Kebutuhan Perangkat Keras Pengembang

No	Perangkat Keras	Spesifikasi
Laptop		
1	Processor	Intel(R) Core(TM) i3 CPU M330 @2.13GHz 2.13GHz

No	Perangkat Keras	Spesifikasi
2	Monitor	14” (inch)
3	VGA	1 GB
4	RAM	2 GB
5	Hard disk drive	500 GB
Mobile		
1	Layar	4” (inch)
2	RAM	1 GB
3	SD Card	2 GB
4	Kamera	5 MP

Rekomendasi minimum spesifikasi kebutuhan perangkat keras yang digunakan untuk menjalankan aplikasi teknologi *augmented reality* ini pada perangkat *mobile* adalah sebagai berikut :

Tabel 3.2 Kebutuhan Perangkat Keras Pengguna

No	Perangkat Keras	Spesifikasi
1	Layar	4” (inch)
2	RAM	1 GB
3	SD Card	2 GB
4	Kamera	3 MP

3.1.7.2 Analisis Kebutuhan Perangkat Lunak

Analisis perangkat lunak atau *software* merupakah hal yang terpenting dalam mendukung kinerja sebuah sistem. Perangkat lunak digunakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras agar dapat saling berinteraksi diantara keduannya. Perangkat lunak yang dapat digunakan untuk membangun aplikasi teknologi *augmented reality* ini adalah sebagai berikut :

Tabel 3.3 Kebutuhan Perangkat Lunak Pengembang

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows 7
2	Tool Pembangun	Unity 3D
3	Tool Desain	Autodesk 3ds Max 2010
4	Library	Vuforia Qualcomm

Rekomendasi minimum spesifikasi kebutuhan perangkat lunak yang digunakan untuk menjalankan aplikasi teknologi *augmented Realty* ini pada perangkat *mobile* adalah sebagai berikut :

Tabel 3.4 Kebutuhan Perangkat Lunak Pengguna

No	Perangkat Lunak	Spesifikasi
1	Operating System (OS)	Android
2	Versi Android	2.2 (Froyo)

3.1.7.3 Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna merupakan analisis terhadap *user* yang akan menggunakan sistem yang akan dibangun. Adapun kebutuhan pengguna yang terlibat dalam penggunaaan aplikasi teknologi Augmented Reality ini dapat dilihat pada tabel 3.5 berikut :

Tabel 3.5 Analisis Kebutuhan Pengguna

Pengguna	Calon Pembeli
Hak Akses	Menggunakan Aplikasi
User Experience	Dapat menggunakan perangkat *mobile* terutama Android.

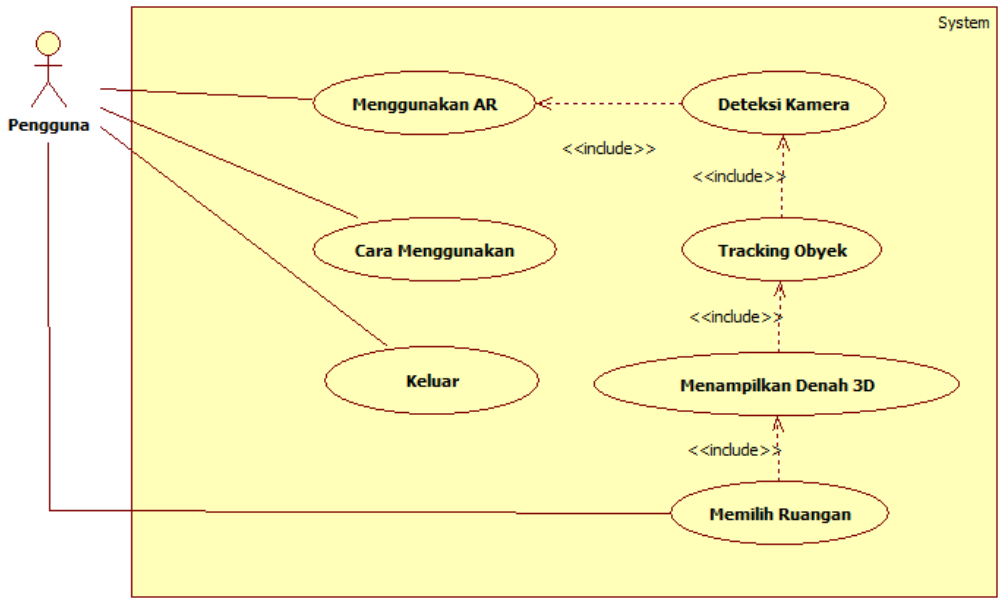
3.1.8 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan kebutuhan sistem yang akan dibangun pada aplikasi teknologi *augmented reality* ini. Adapun kebutuhan fungsional pada aplikasi yang akan dibangun ini dengan pemodelan berorientasi

objek. Perangkat lunak ini dimodelkan menggunakan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain *use case diagram, skenario, sequence diagram, activity diagram, class diagram*.

1.1.7.1 *Use Case Diagram*

Use Case merupakan gambaran skenario dari interaksi antara *user* dengan sistem. Sebuah diagram *Use Case* menggambarkan hubungan antara aktor dan kegiatan yang dapat dilakukannya terhadap aplikasi.



Gambar 3. 6 *Use Case Diagram*

1.1.7.1.1 Definisi *Actor*

Definisi *Actor* berfungsi untuk menjelaskan *Actor* yang terdapat pada *Use Case Diagram*. Definisi *Actor* diterangkan pada Tabel 3.6.

Tabel 3.6 Definisi *Actor*

No.	*Actor*	Deskripsi
1	Pengguna	Orang yang berinteraksi dengan system, yaitu pengguna yang mengakses semua fungsi yang disediakan sistem.

1.1.7.1.2 Definisi *Use Case*

Definisi *Use Case* berfungsi untuk menjelaskan fungsi *Use Case* yang terdapat pada *Use Case Diagram*. Definisi *Use Case* diterangkan pada Tabel 3.7.

Tabel 3.7 Definisi *Use Case*

NO	*Use Case*	Deskripsi
1	Menggunakan AR	Fungsionalitas untuk memulai aplikasi AR.
2	Cara Menggunakan	Fungsionalitas untuk menampilkan info cara menggunakan.
3	Keluar	Fungsionalitas untuk keluar dari aplikasi AR.
4	Deteksi Kamera	Fungsionalitas mendeteksi ketersediaan kamera.
5	*Tracking* Obyek	Fungsionalitas untuk mendeteksi keberadaan atau mengenali obyek
6	Menampilkan Denah 3D	Fungsionalitas untuk menampilkan denah berupa obyek 3D pada layar *device/* perangkat *mobile.*
7	Memilih Ruangan	Fungsionalitas untuk memilih obyek ruangan yang ingin ditampilkan.

1.1.7.2 Skenario *Use Case*

Skenario *Use Case* merupakan penggambaran dari alur penggunaan sistem dimana setiap skenario digambarkan dari sudut pandang *actor*, seseorang atau piranti yang berinteraksi dengan perangkat lunak dalam berbagai cara.

1. Skenario *Use Case* Menggunakan AR

Tabel 3.8 di bawah ini adalah *Skenario Use Case* untuk memulai aplikasi AR Bangunan.

Tabel 3.8 *Skenario Use Case* Menggunakan AR

Identifikasi	
Nomor	UCS-01
Nama	Menggunakan AR
Tujuan	Menjalankan menu Menggunakan AR
Deskripsi	Proses untuk menampilkan obyek 3D dari denah rumah.
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Pengguna berada pada menu utama
Aksi Aktor	**Reaksi Sistem**
1. Pengguna memilih menu Menggunakan AR.	
	2. Sistem mendeteksi kamera.
	3. Kamera aktif.
4. Pengguna mengarahkan kamera pada marker.	
	5. Sistem menampilkan obyek ruangan 3D.
Kondisi Akhir	Aplikasi menampilkan obyek ruangan 3D dari denah rumah.

2. Skenario *Use Case* Cara Menggunakan AR

Tabel 3.9 di bawah ini adalah *Skenario Use Case* info untuk cara menggunakan aplikasi AR Bangunan.

Tabel 3.9 Skenario *Use Case* Cara Menggunakan

Identifikasi	
Nomor	UCS-02
Nama	Cara Menggunakan
Tujuan	Menampilkan info cara menggunakan aplikasi
Deskripsi	Proses untuk menampilkan info cara menggunakan aplikasi
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Pengguna berada pada menu utama
Aksi Aktor	**Reaksi Sistem**
1. Pengguna memilih menu cara penggunaan	
	2. Menampilkan informasi cara penggunaan aplikasi
Kondisi Akhir	Aplikasi menampilkan info cara menggunakan aplikasi

3. Skenario *Use Case* Keluar

Tabel 3.10 di bawah ini adalah *Skenario Use Case* Keluar pada aplikasi AR Bangunan.

Tabel 3.10 Skenario *Use Case* Keluar

Identifikasi	
Nomor	UCS-03
Nama	Keluar
Tujuan	Keluar dari aplikasi
Deskripsi	Proses untuk menutup aplikasi
Aktor	Pengguna
Skenario utama	
Kondisi Awal	Pengguna berada pada menu utama
Aksi Aktor	**Reaksi Sistem**

Identifikasi	
1. pengguna memilih menu Keluar	
	2. Keluar dari aplikasi
Kondisi Akhir	Keluar dari aplikasi

4. Skenario *Use Case* Deteksi Kamera

Tabel 3.11 di bawah ini adalah *Skenario Use Case* deteksi kamera pada aplikasi pengenalan rangka dan bagian tumbuhan.

Tabel 3.11 Skenario *Use Case* Deteksi Kamera

Identifikasi	
Nomor	UCS-04
Nama	Mendeteksi Kamera
Tujuan	Mengaktifkan kamera
Deskripsi	Proses untuk mengaktifkan kamera
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Kamera dalam kondisi belum aktif
Aksi Aktor	**Reaksi Sistem**
1. pengguna memilih menu Menggunakan AR	
	2. Sistem mendeteksi ketersediaan kamera
	3. kamera aktif
Kondisi Akhir	Kamera dalam kondisi aktif

5. Skenario *Use Case Tracking* Obyek

Tabel 3.12 di bawah ini adalah *Skenario Use Case tracking* obyek pada aplikasi.

Tabel 3.12 Skenario *Use Case Tracking* Obyek

Identifikasi	
Nomor	UCS-05
Nama	Tracking Obyek

Identifikasi	
Tujuan	Mendeteksi keberadaan obyek
Deskripsi	Sistem mendeteksi keberadaan obyek melalui gambar yang ditangkap oleh kamera
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Kamera aktif
Aksi Aktor	**Reaksi Sistem**
1. pengguna memilih menu Menggunakan AR	
	2. Sistem memproses masukan gambar yang ditangkap oleh kamera
	3. Sistem mencocokan pola
	4. Obyek dikenali
Kondisi Akhir	Obyek dikenali

6. Skenario *Use Case* Menampilkan Obyek 3D

Tabel 3.13 di bawah ini adalah *Skenario Use Case* Menampilkan Denah 3D pada aplikasi AR.

Tabel 3.13 Skenario *Use Case* Menampilkan Obyek 3D

Identifikasi	
Nomor	UCS-06
Nama	Menampilkan obyek 3D
Tujuan	Menampilkan obyek 3D di layar *handphone*
Deskripsi	Proses untuk menampilkan obyek 3D pada layar *handphone*
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Kamera aktif
Aksi Aktor	**Reaksi Sistem**
1. Pengguna mengarahkan kamera pada obyek	

Identifikasi	
	2. Sistem memproses masukan gambar yang ditangkap oleh kamera
	3. Sistem mencocokan pola
	4. Obyek dikenali
	5. Obyek 3D tampil di layar *handphone*
Kondisi Akhir	Obyek 3D tampil di layar *handphone*

7. Skenario *Use Case* Memilih Ruangan

Tabel 3.14 di bawah ini adalah *Skenario Use Case* Memilih Ruangan pada aplikasi AR.

Tabel 3.14 Skenario *Use Case* Memilih Ruangan

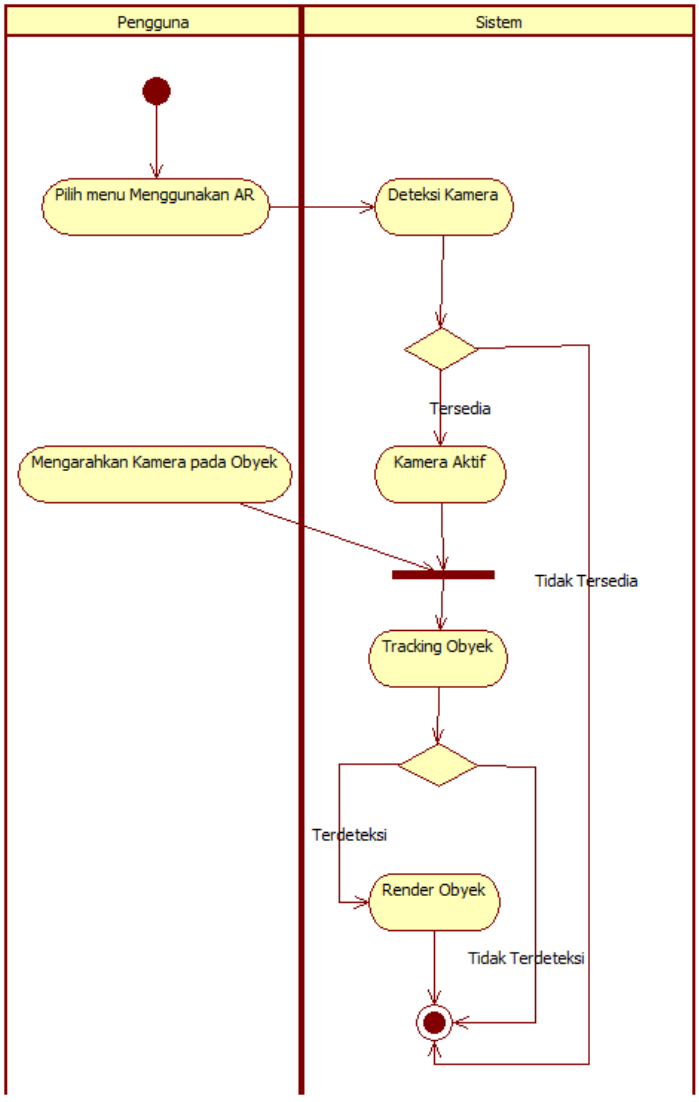
Identifikasi	
Nomor	UCS-07
Nama	Memilih ruangan
Tujuan	Menampilkan obyek 3D setiap ruangan
Deskripsi	Proses untuk menampilkan obyek 3D setiap ruangan pada layar *handphone*
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Obyek terdeteksi
Aksi Aktor	**Reaksi Sistem**
1. Pengguna mengarahkan kamera pada obyek	
	2. Sistem menampilkan obyek 3D keseluruhan dan menu pilihan setiap ruangan
3. Pengguna memilih menu ruangan yang ingin ditampilkan	
	4. Sistem menampilkan obyek ruangan 3D yang ingin ditampilkan
Kondisi Akhir	Obyek ruangan 3D yang telah dipilih tampil di layar *handphone*

1.1.7.3 *Activity Diagram*

Activity diagram menggambarkan urutan aktivitas dalam sebuah proses. Struktur diagram ini mirip *flowchart* atau *Data Flow Diagram* pada perancangan terstruktur. Berikut ini merupakan *activity diagram* pada sistem yang akan dibangun.

1.1.7.3.1 *Activity Diagram* Menggunakan AR

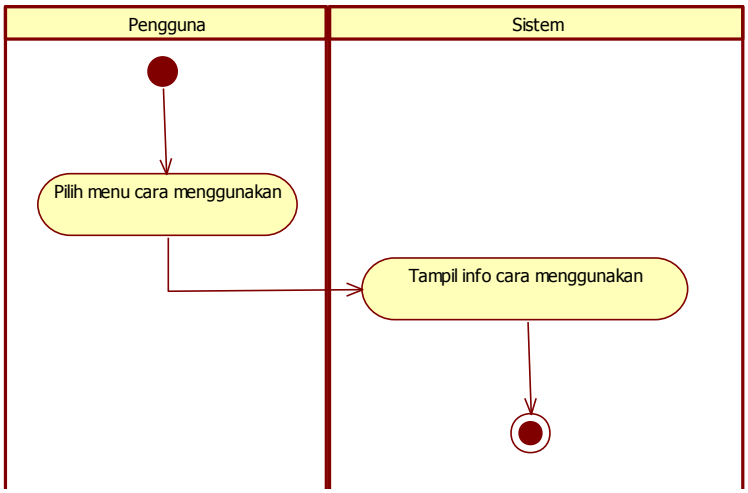
Activity Diagram terkait skenario Menggunakan AR memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3. 7 *Activity Diagram* Menggunakan AR

1.1.7.3.2 *Activity Diagram* Cara Menggunakan AR

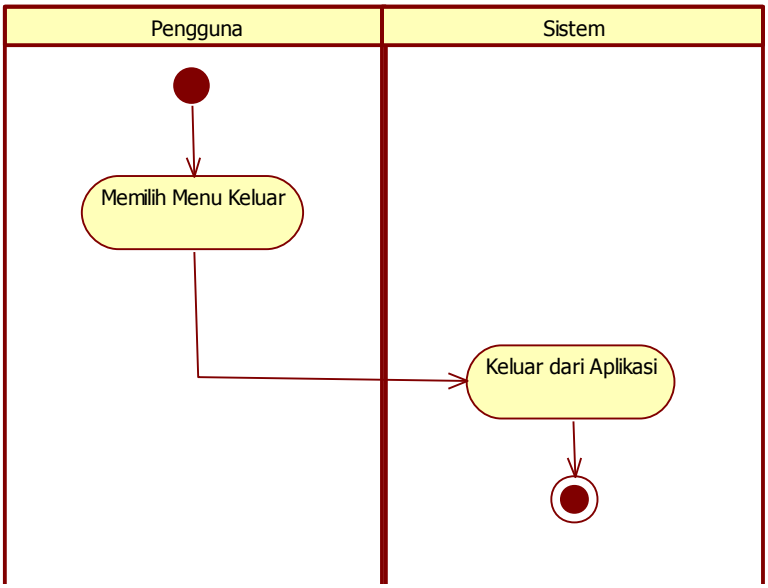
Activity Diagram terkait skenario cara menggunakan AR memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3. 8 *Activity Diagram* Cara Menggunakan

1.1.7.3.3 *Activity Diagram* Keluar

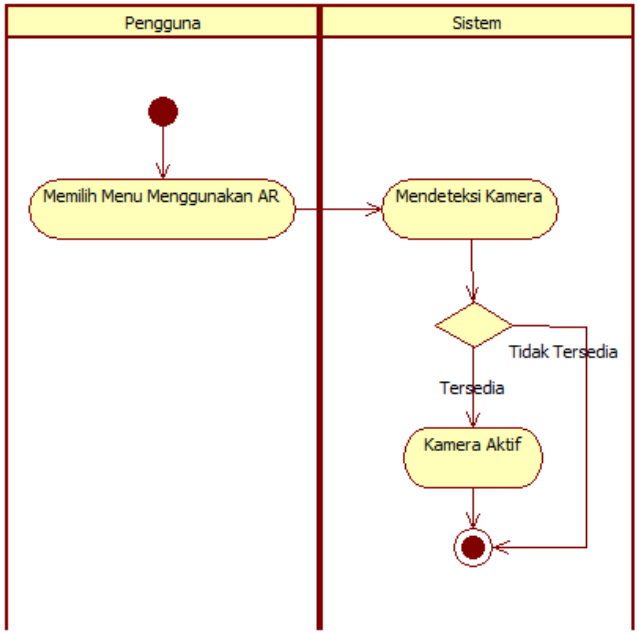
Activity Diagram terkait skenario Keluar memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3. 9 *Activity Diagram* Keluar

1.1.7.3.4 *Activity Diagram* Deteksi Kamera

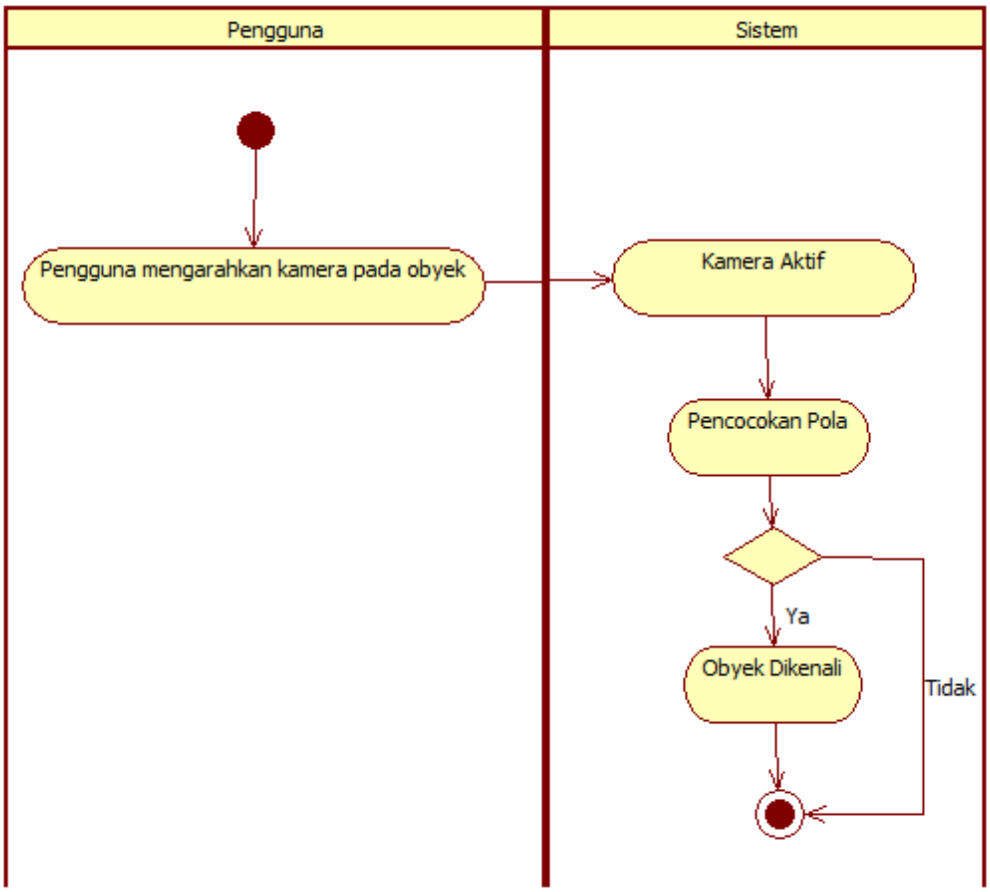
Activity Diagram terkait skenario Deteksi Kamera memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3. 10 *Activity Diagram* Deteksi Kamera

1.1.7.3.5 *Activity Diagram Tracking* Obyek

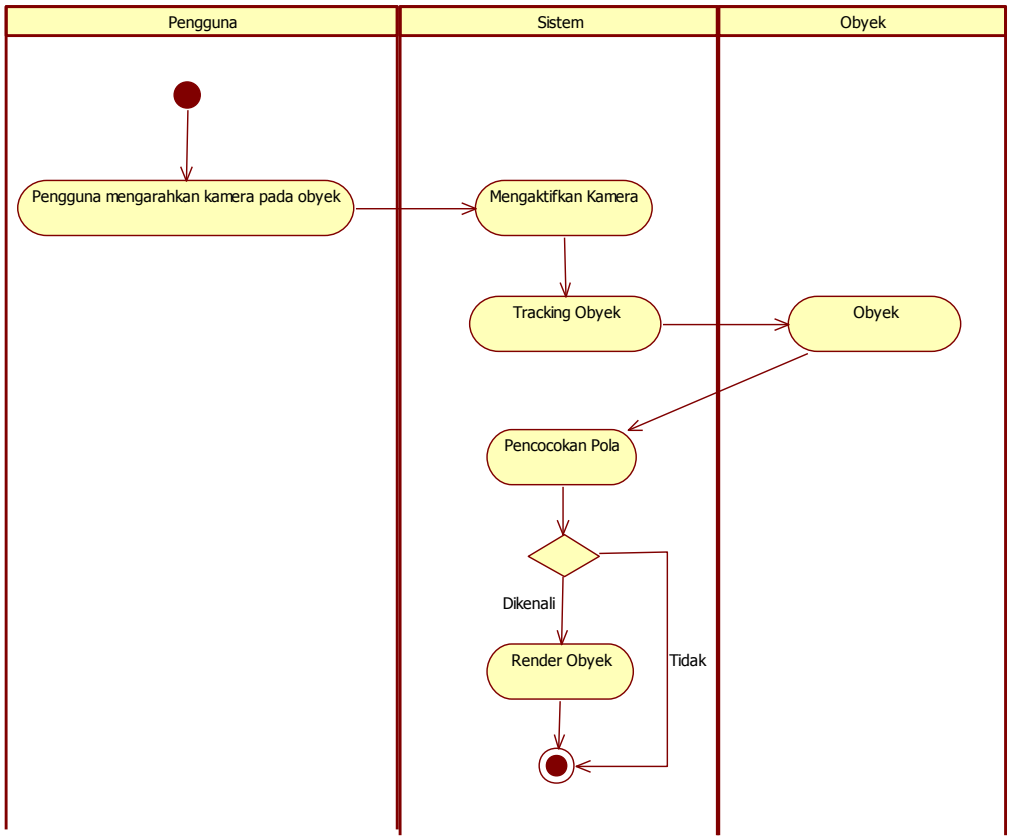
Activity Diagram terkait skenario *Tracking* Obyek memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3. 11 *Activity Diagram Tracking* Obyek

1.1.7.3.6 *Activity Diagram* Menampilkan Obyek 3D

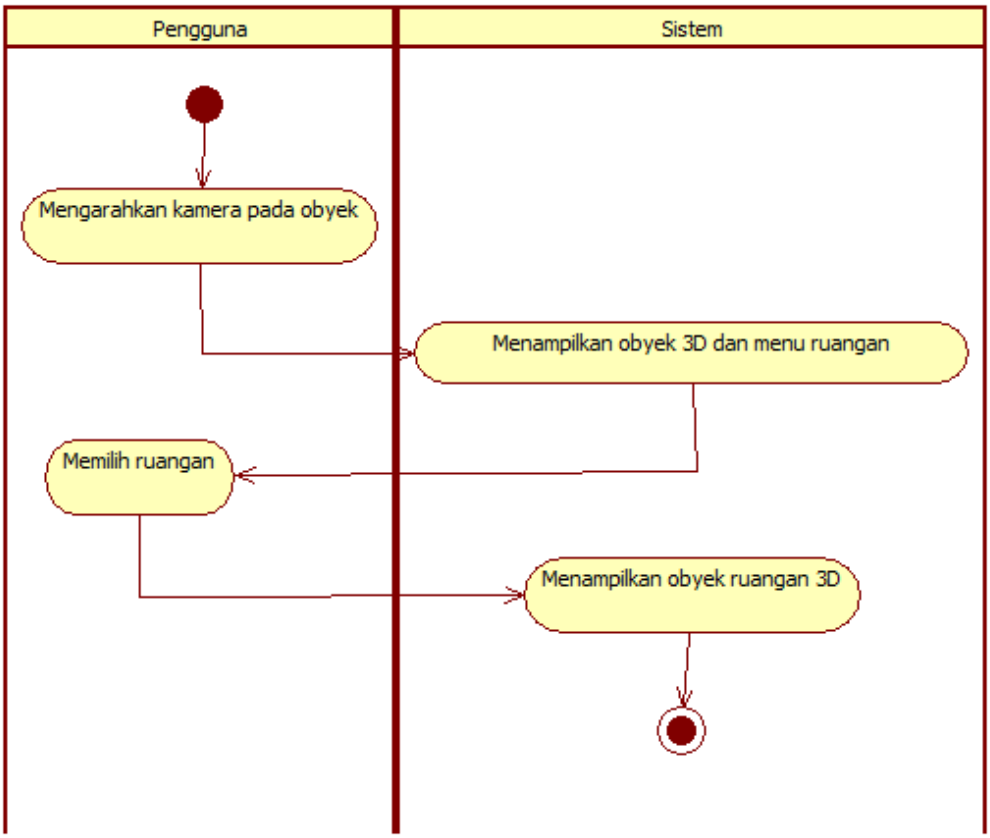
Activity Diagram terkait skenario Menampilkan Obyek 3D memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3. 12 *Activity Diagram* Menampilkan Obyek 3D

1.1.7.3.7 *Activity Diagram* Memilih Ruangan

Activity Diagram terkait skenario Memilih Ruangan memiliki alur proses seperti digambarkan sebagai berikut :

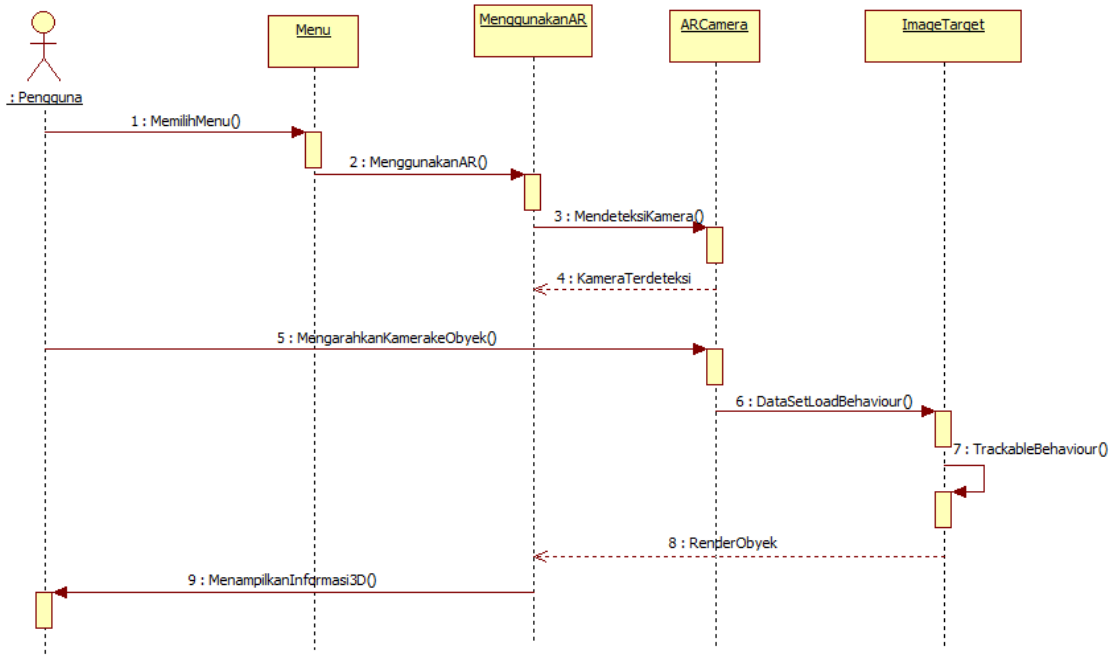


Gambar 3. 13 *Activity Diagram* Memilih Ruangan

1.1.7.4 *Sequence Diagram*

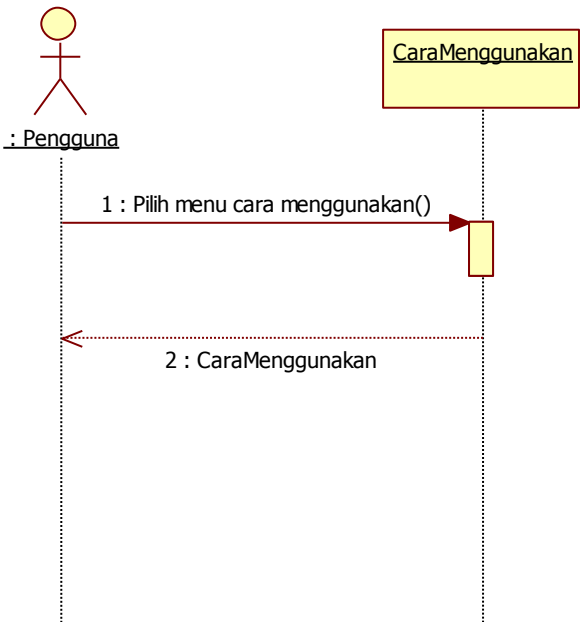
Sequence diagram menjelaskan gambaran interaksi antar obyek dalam urutan waktu. Interaksi ini berupa pengiriman serangkaian data antar obyek-obyek yang saling berinteraksi. *Sequence diagram* pada aplikasi ini terdiri dari *Sequence diagram* menu utama, mulai *game* , pengaturan dan instruksi.

1. *Sequence Diagram* Menggunakan AR



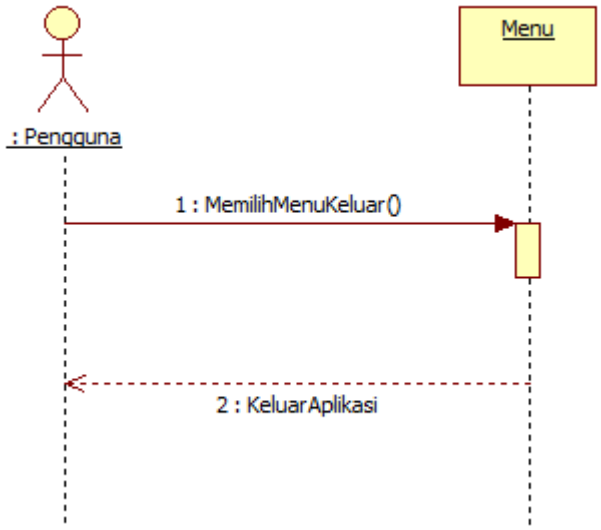
Gambar 3. 14 *Sequence Diagram* Menggunakan AR

2. *Sequence Diagram* Cara Menggunakan



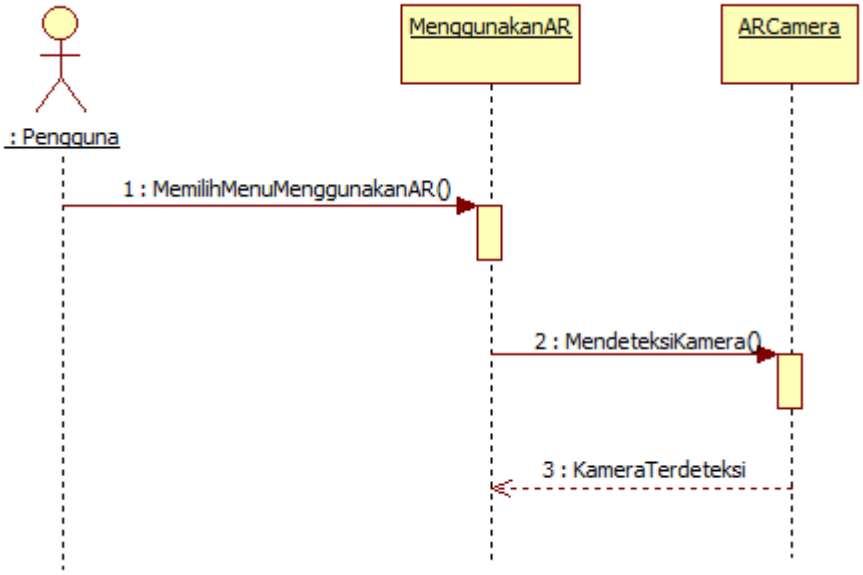
Gambar 3. 15 *Sequence Diagram* Cara Menggunakan

3. *Sequence Diagram* Keluar



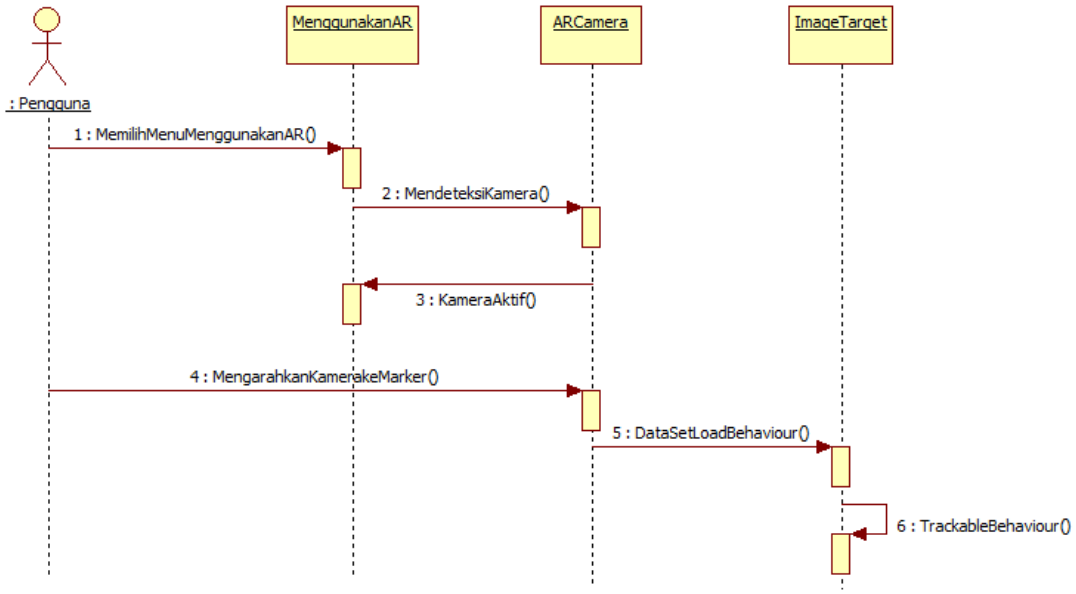
Gambar 3. 16 *Sequence Diagram* Keluar

4. *Sequence Diagram* Deteksi Kamera



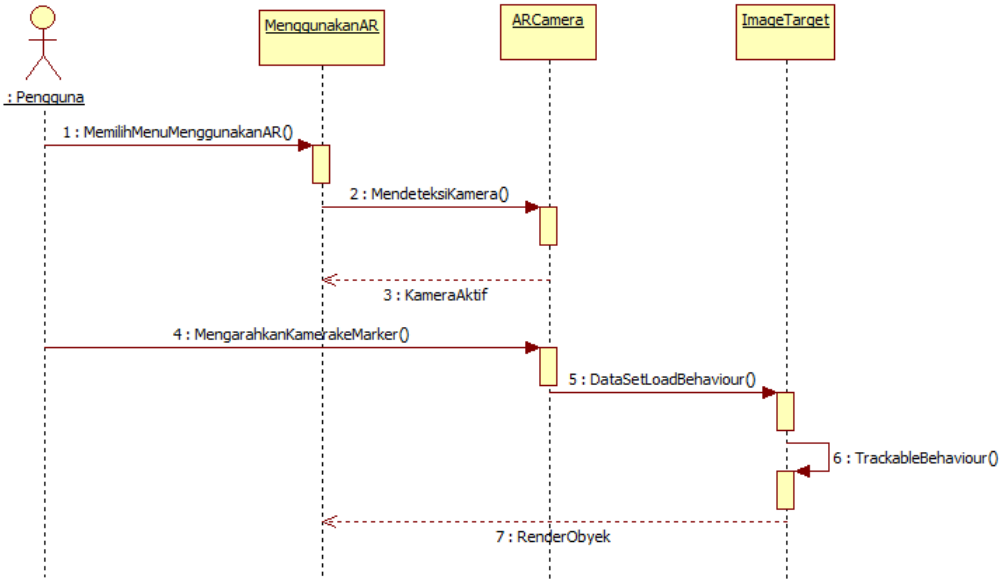
Gambar 3. 17 *Sequence Diagram* Deteksi Kamera

5. *Sequence Diagram Tracking* Obyek



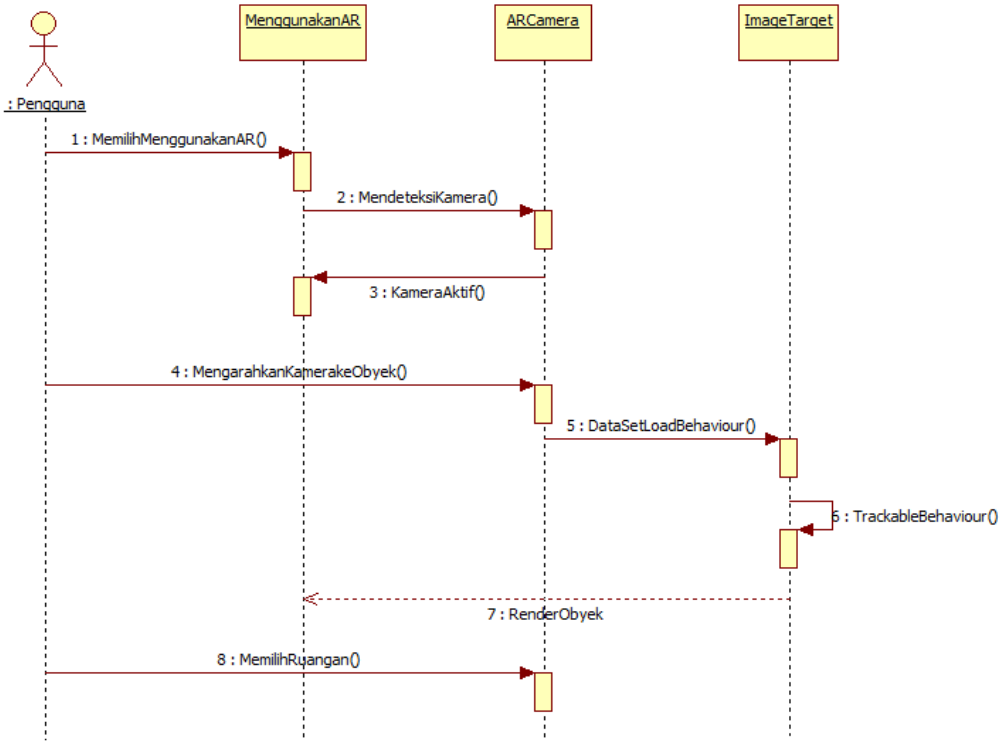
Gambar 3. 18 *Sequence Diagram Tracking* Obyek

6. *Sequence Diagram* Menampilkan Obyek 3D



Gambar 3. 19 *Sequence Diagram* Menampilkan Obyek 3D

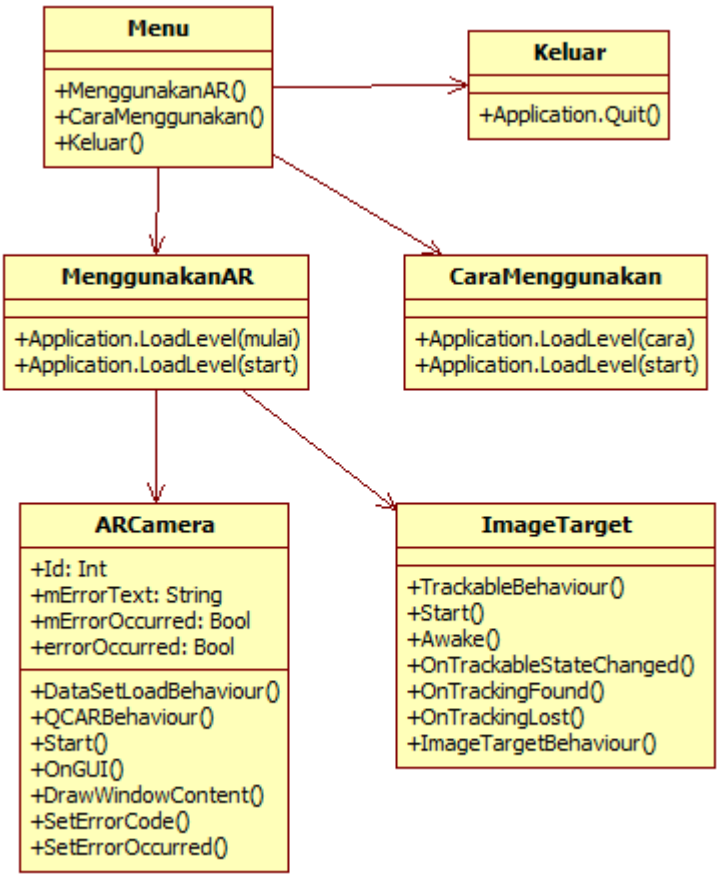
7. *Sequence Diagram* Memilih Ruangan



Gambar 3. 20 *Sequence Diagram* Memilih Ruangan

1.1.7.5 *Class Diagram*

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (*attribut* atau *property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda atau fungsi). Berikut adalah *class diagram* dari aplikasi denahAR.



Gambar 3. 21 *Class Diagram*

1.2 Perancangan Sistem

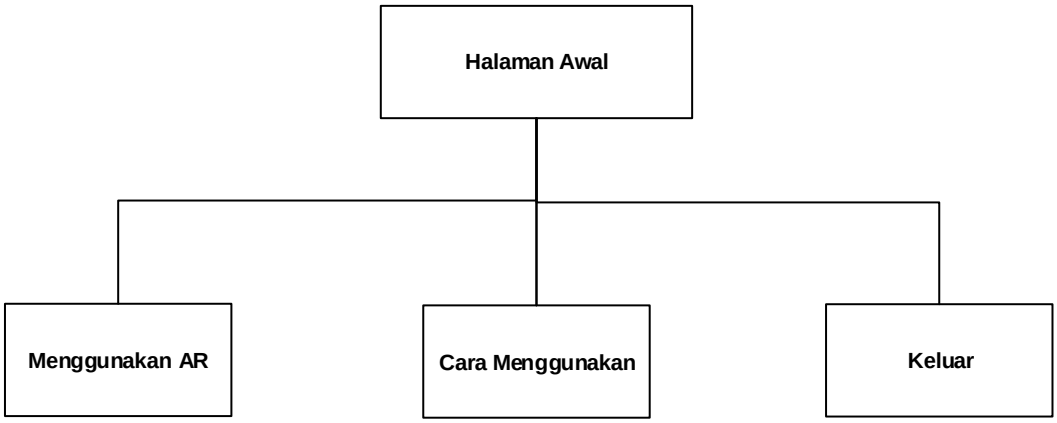
Perancangan merupakan bagian dari metodologi pembangunan suatu perangkat lunak yang harus dilakukan setelah melalui tahapan analisis. Untuk dapat melakukan perancangan sistem maka dilakukan beberapa buah langkah yaitu sebagai berikut :

1. Perancangan Struktur *Menu*
2. Perancangan Antarmuka
3. Jaringan Semantik

3.2.1 Perancangan Struktur Menu

Struktur *menu* adalah bentuk umum dari suatu rancangan program untuk memudahkan pemakai dalam menjalankan program komputer. Saat menjalankan

program, pengguna tidak mengalami kesulitan dalam memilih *menu-menu* yang diinginkan. Berikut ini perancangan struktur *menu* pada aplikasi pengenalan rangka dan bagian tumbuhan untuk siswa kelas ini yang ditunjukan pada gambar berikut ini.



Gambar 3. 22 Struktur Menu

3.2.2 Perancangan Antarmuka

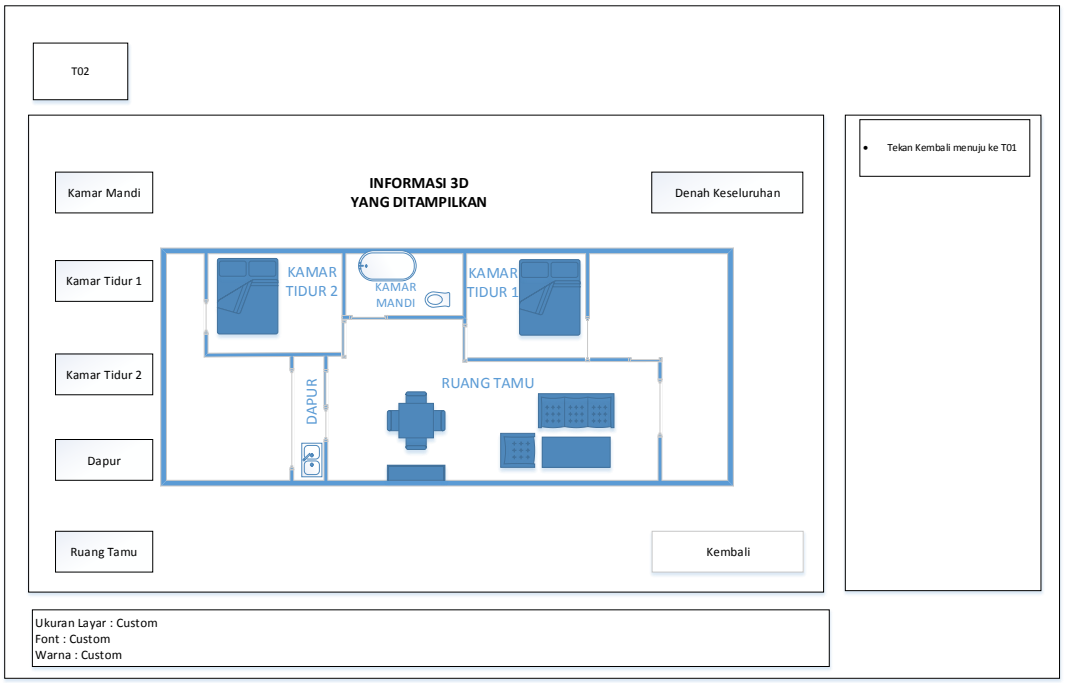
Perancangan antarmuka dibutuhkan untuk mawakili keadaan sebenarnya dari aplikasi yang akan dibangun. Berikut ini contoh perancangan antarmuka yang ada dari aplikasi yang akan dibangun yaitu :

1. Perancangan Antarmuka Menu Utama



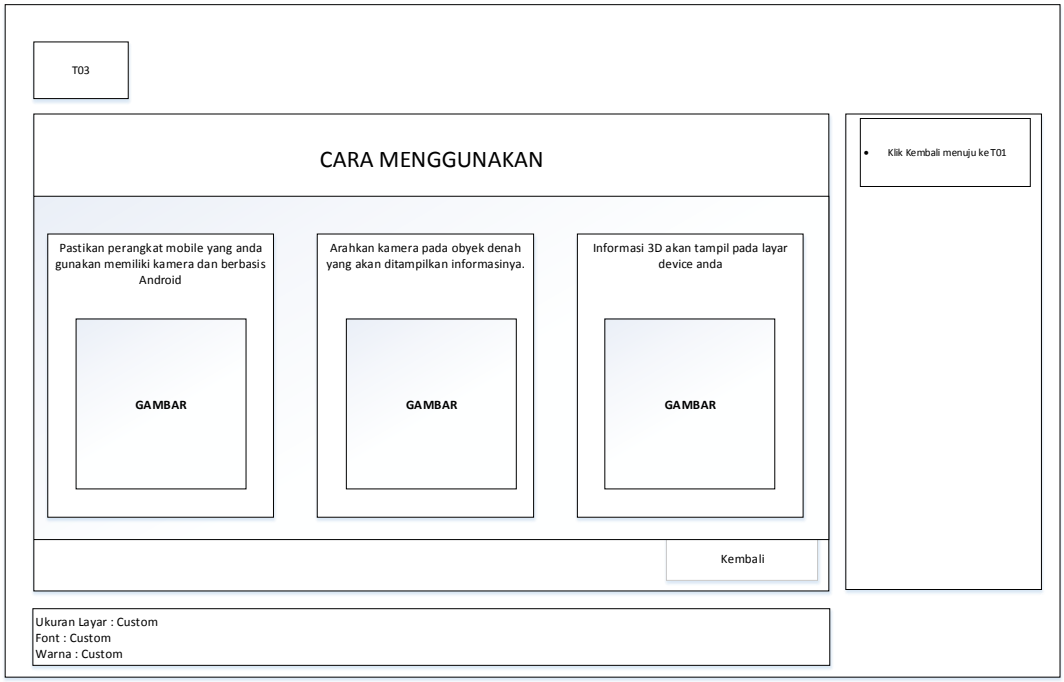
Gambar 3. 23 Antarmuka Menu Utama

2. Perancangan Antarmuka Menu Menggunakan AR



Gambar 3. 24 Antarmuka Menu Menggunakan AR

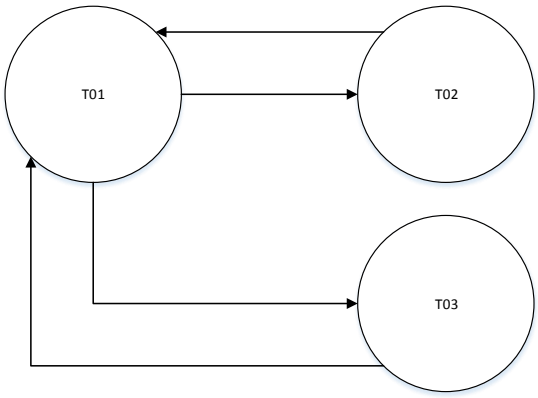
3. Perancangan Antarmuka Cara Menggunakan



Gambar 3. 25 Antarmuka Cara Menggunakan

3.2.3 Jaringan Semantik

Jaringan semantik ini menunjukan form-form yang bisa diakses oleh pengguna seperti ditunjukan pada Gambar 3.23.



Gambar 3. 26 Jaringan Semantik

Keterangan :

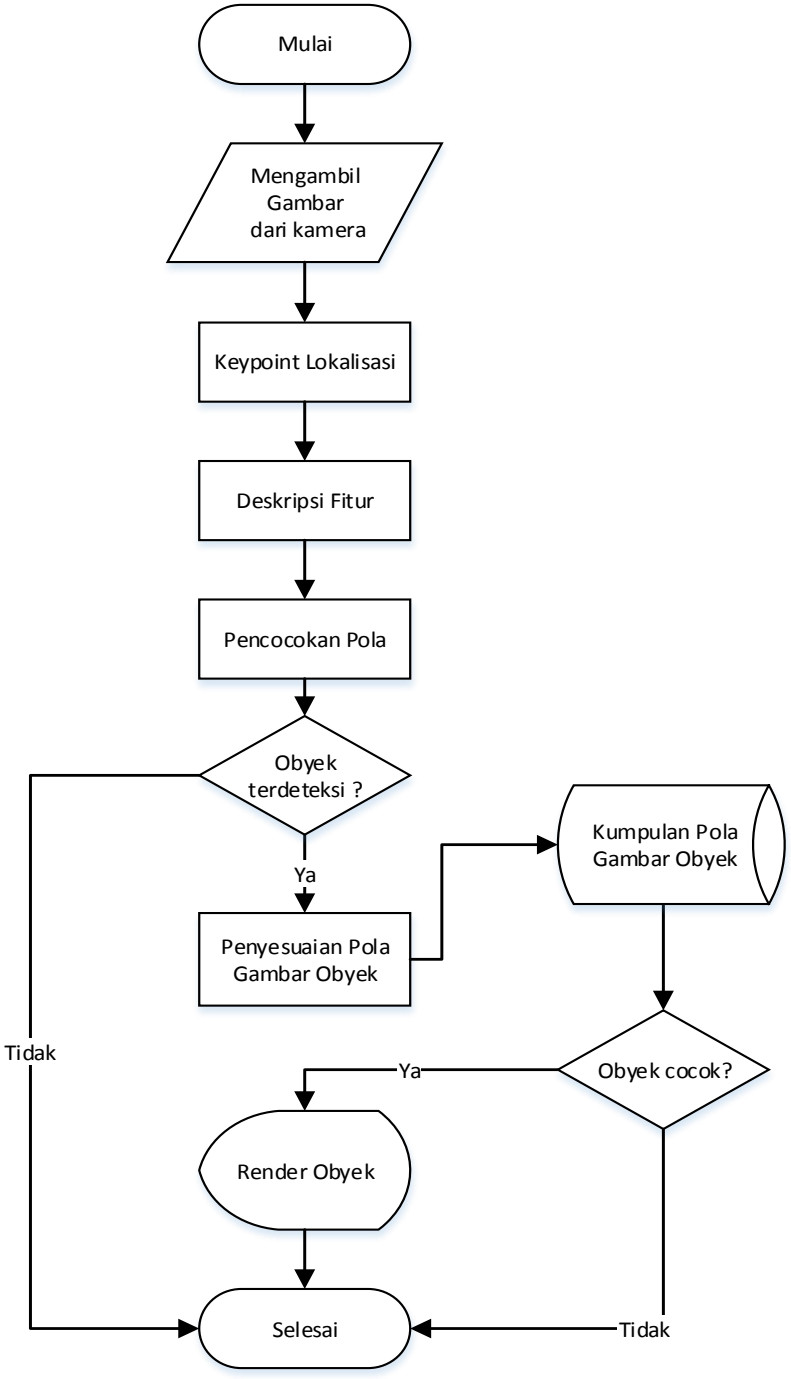
T01 : Menu Utama T03 : Cara Menggunakan

T02 : Menggunakan AR

3.2.4 Perancangan Method

Perancangan komponen method merupakan perancangan yang dibuat setelah perancangan jaringan semantik dan antarmuka. Perancangan ini berfungsi untuk mendeskripsikan method-method yang berada di dalam aplikasi pengenalan rangka dan bagian tumbuhan menggunakan teknologi augmented reality. Adapun method-method yang terdapat dalam aplikasi yang akan dibangun adalah sebagai berikut :

1. Perancangan Method Augmented Reality



Gambar 3. 27 Perancangan Method Augmented Reality