BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis

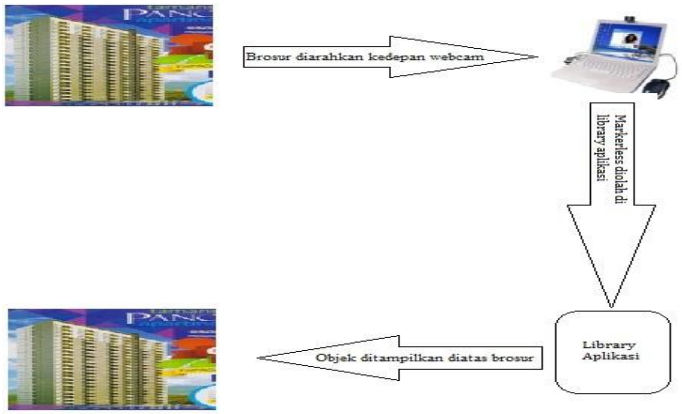
Analisis bertujuan untuk mengidentifikasi permasalahan-permasalahan yang ada pada sistem serta menentukan kebutuhan dari sistem yang dibangun.Analisis tersebut meliputi analisis masalah, analisis kebutuhan data dan analisis sistem. Analisis merupakan penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponen dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, sehingga diperoleh solusi. Analisis merupakan tahapan yang paling penting, karena kesalahan dalam tahap ini akan menyebabkan kesalahan di tahap selanjutnya. Aplikasi yang dibuat merupakan program untuk mendeteksi sebuah *markerless* sekaligus menampilkan objek tiga dimensi. Objek yang dibuat merupakan bangunan apartemen secara keseluruhan dan setiap kamar, supaya pengguna dapat berinteraksi langsung dengan objek virtual dalam dunia nyata digunakan Marker brosur pemasaran apartemen sebagai media interaksinya.

3.1.1 Analisis Masalah

Saat ini para sales apartemen dalam melakukan promosi apartemen hanya menggunakan brosur apartemen. Dimana para konsumen hanya mengetahui informasi mengenai apartemen, gambar apartemen, dan tipe apartemen yang ditawarkan.Ukuran maket yang cukup besar tidak memungkinkan untuk dibawa serta harganya yang mahal menjadi masalah yang selalu muncul dalam melakukan promosi yang dilakukan para sales dalam melakukan interaksi kepada konsumen. Para konsumen kesulitan mengetahui secara detail bentuk apartemen yang ditawarkan baik tipe unit maupun interior karena dalam melakukan promosi hanya membawa brosur apartemen.

3.1.2 Analisis Arsitektur Aplikasi

Pada pembuatan aplikasi ini *webcam* berperan penting bagi perangkat lunak untuk menangkap *markerless.* Komputer akan mendeteksi pola *markerless* yang sesuai kemudian *markerless* dapat ditentukan. Dengan informasi yang didapat, objek virtual digabungkan dengan *markerless* yang didapat dari *webcam* dan merendernya sesuai dengan informasi posisi yang diperoleh dari *markerless detector* tersebut. Proses tersebut terjadi secara *real-time* sehingga model virtual yang tampil di media *display* akan mengikuti pergerakan *markerless*. Berikut merupakan gambaran arsitektur aplikasi yang akan dibuat dapat dilihat pada gambar 3.1



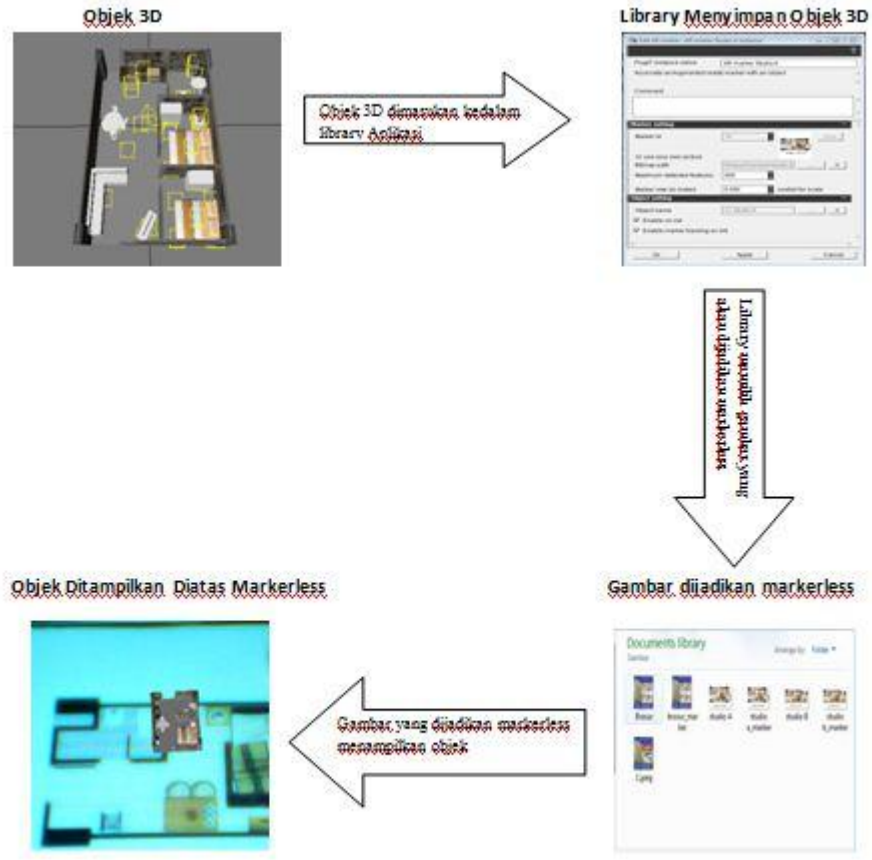
Gambar 3.1 Arsitektur Aplikasi AR Media Promosi Apartemen

3.1.3 Analisis Metode

Markerless Augmented Reality merupakan metode yang biasa digunakan pada *augmented reality*, metode ini tidak menggunakan marker untuk menambahkan objek virtual ke lingkungan nyata. *Pattern Matching* merupakan teknik pelacakan pada metode *Markerless Augmented Reality* yang *mengganti* marker dengan gambar biasa, cara kerja teknik *Pattern Matching* adalah dengan mengamati lingkungan nyata melalui pendeteksian pola dan orientasi gambar teknik ini dapat mengenali pola apa saja selain marker seperti cover buku, lukisan, wajah manusia dan sebagainya.

3.1.4 Analisis *Augmented Reality* Objek 3D

Analisis *Augmented Reality* Objek 3D merupakan analisis yang mendeskripsikan bagaimana proses menampilkan sebuah objek 3D dari awal pembuatan objek 3Dsampai dengan proses menampilkan objek 3D. Dalam proses menampilkan objek 3D ini menggunakan sebuah *markerless* yang berupa brosur apartemen yang kemudian di tracking oleh kamera, Setelah brosur dapat dikenali oleh kamera maka aplikasi akan menampilkan objek 3D sesuai dengan brosur yang digunakan. alur dari proses menampilkan Objek 3D dapat dilihat pada gambar 3.2



Gambar 3.2 Analisis *Augmented Reality* Objek 3D

3.1.5 Analisis Penentuan *Markerless*

Analisis penentuan *markerless* merupakan analisis metode yang mendeskripsikan bagaimana proses dari awal *inisialisasi*, sampai proses memunculkan objeck 3D dengan metode *markerless augmented rality*. Proses pengenalan deteksi *markerless* ini melalui beberapa tahapan secara garis besarnya, dalam perancangan ada dua bagian utama yaitu :

1. *Inisialisasi* Gambar
2. Memunculkan Objek 3D

3.1.5.1 *Inisialisasi* Gambar

Pada tahapan ini ditentukan proses untuk mengambil gambar sebagai datamasukan yang akan di proses

1. Pilih File Gambar

Proses *inisialisasi* berawal dari memilih file gambar yang akan dijadikan *markerless*. Berikut ini merupakan script untuk memilih file gambar.

Script Pilih Gambar

```
fun cbBtnPickFile(ctrlbtn, p)=
     let p -> [inst winstr ctrlpath ctrlurl ctrlisurl] in
     let getEdCtrlTextLineValue ctrlpath -> path in
     let getPathFile path "" -> [fpath fname] in
       (
     setEdCtrlButtonEnable ctrlbtn 1;
      _DLGrflopen   _DLGOpenFile   _channel   winstr.EDW_win
      fpath                                          fname
      "Bitmaps\0*.bmp;*.jpg;*.png;*.tga;\0Bitmap\0*.bmp\0Jp
      eg\0*.jpg\0Png\0*.png\0Targa\0*.tga\0All\0*.*\0\0"
      @cbDlgGetPath [inst ctrlbtn winstr ctrlpath ctrlurl
      ctrlisurl];
       );
0;;
```

2. Resize Bitmap

Gambar yang dijadikan *markerless* akan di resize terlebih dahulu. Berikut ini merupakan script proses resize file Bitmap.

Script Resize File Bitmap

```
fun G2DstrechBitmap(chan, bmp, width, height, bcolor)=
      let _GETbitmapSize bmp -> [bw bh] in
      if (bw == width) && (bh == height) then bmp else
```

```
        let _FILLbitmap _CRbitmap chan width height bcolor ->
        newbmp in
        let if bh >= bw then ftoi ((itof height) /. ((itof
        bh) /. (itof bw))) else width -> fw in
        let if bh >= bw then height else ftoi ((itof width)
        /. ((itof bw) /. (itof bh))) ->fh in
        _SCPbitmap newbmp ((width / 2) - (fw / 2)) ((height /
        2) - (fh / 2)) ((width / 2) + (fw / 2)) ((height / 2)
        + (fh / 2)) bmp 0 0 bw bh nil
 ;;
```

3. Convert Bitmap To AlphaBitmap

Merupakan proses mengubah gambar Bitmap menjadi AlphaBitmap. Berikut Ini merupakan script proses merubah file Bitmap menjadi AlphaBitmap.

```
Convert Bitmap To AlphaBitmap
fun G2DconvertBmpToAlphaBmp (chan, bmp)=
     if bmp != nil then
     (
     let _GETbitmapSize bmp -> [w h] in
     let _FILLbitmap8 _CRbitmap8 chan w h 0xff -> bmp8 in
     let _CRalphaBitmap chan bmp bmp8 0 0 -> newalpha in
     (
     _DSbitmap bmp;
     _DSbitmap8 bmp8;
     newalpha;
     );
     )
else nil;;
```

4. Create Layer AlphaBitmap

Merupakan proses pembuatan Layer untuk AlphaBitmap. Berikut ini merupakan script proses create layer AlphaBitmap.

```
Create layer AlphaBitmap
fun G2DcreateLayer (chan, w, h, border, bcolor, color,
opacity)=
     let mktab 768 0 -> greypal in
     (
     let 0 -> i in
     let 0 -> c in
     while (i < 768) do
     (
     set greypal.(i) = c;
     set greypal.(i+1) = c;
     set greypal.(i+2) = c;
     set i = i+3;
     set c = c+1;
      );
     let (255 * opacity) / 100 -> opcoef in
     let if !border then
     _FILLbitmap _CRbitmap chan w h color
     else
```

```
    (_DRAWrectangle
    (_FILLbitmap (_CRbitmap chan w h) color)
    0 0 w h DRAW_SOLID border bcolor DRAW_INVISIBLE
    color)
     ->nodebmp in
    let if !border then
    _FILLbitmap8 (_SETbitmapPalette _CRbitmap8 chan w h
    greypal) (make_rgb opcoef opcoef opcoef)
    else
    (_DRAWrectangle8
    (_FILLbitmap8 (_SETbitmapPalette _CRbitmap8 chan w h
    greypal) 0)
    0 0 w h DRAW_SOLID border 0xffffff DRAW_SOLID
    (make_rgb opcoef opcoef opcoef))
    ->nodebmpalpha in
    let _CRalphaBitmap chan nodebmp nodebmpalpha nil nil
    -> alphabmp in
     (
      _DSbitmap nodebmp;
      _DSbitmap8 nodebmpalpha;
  alphabmp;
     );
);;
```

3.1.5.2Memunculkan Objek 3D

Apabila inputan gambar sesuai dengan yang ada didatabase maka objek 3D akan ditampilkan sesuai data-data yang ada di database berikut ini merupakan script untuk memunculkan objek 3D.

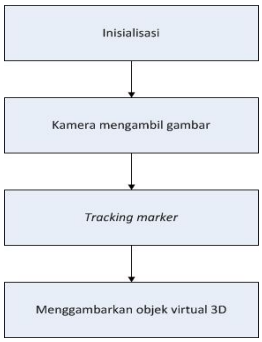
```
fun newOb(inst)=
let (getPluginInstanceParam inst "Apartement") -> objname in
let (getPluginInstanceParam inst "StudioB") -> objname in
let (getPluginInstanceParam inst "StudioB") -> objname in
let  (getPluginInstanceParam  inst  "Brosur_marker.jpg")  ->
path in
let  (getPluginInstanceParam  inst  "StudioA_marker.jpg")  ->
path in
let  (getPluginInstanceParam  inst  "StudioB_marker.jpg")  ->
path in
let atoi (getPluginInstanceParam inst "enable") -> enable in
(if show "Apartement" then hide "StudioA","StudioB"
Then
disable ->disable in
else
hide"Apartement","StudioA","StudioB"
if show "StudioA" then hide "Apartement","StudioB"
then
disable ->disable in
else
hide"StudioA","Apartement","StudioB"
if show "StudioB" then hide "Apartement","StudioA"
then
disable ->disable in
```

```
else
hide"StudioB","Apartement","StudioA"
       );
 );;
```

3.1.6 Analisis Perancangan Aplikasi

Analisis perancangan aplikasi merupakan analisis yang mendeskripsikan bagaimana proses perancangan aplikasi dari awal inisialisasi, mengambil gambar, sampai dengan proses *rendering* objek 3D dengan metode *markerless augmented realitiy.*

Dalam perancangan aplikasi, seolah-olah menggabungkan objek virtual dengan objek nyata dalam hal ini objek virtual berupa objek 3D dan objek nyatanya berupa gambar dengan pola tertentu (*markerless*) alur perancangan aplikasi dapat dilihat pada gambar 3.3



Gambar 3.3 *Diagram* Alur Sistem *Augmented Reality*

Keterangan:

a. Inisialisasi

   Pada tahap ini aplikasi akan mendeteksi ketersediaan kamera pada perangkat keras user.

b. Kamera mengambil gambar

   Pada tahapan ini, kamera akan menangkap langsung gambar dari dunia nyata.

c. *Tracking markerless*

   Tahapan ini sistem melakukan tracking terhadap *markerless* yang digunakan kemudian dicocokan dengan *markerless* yang ada di dalam *database*.

d. Menggambarkan objek virtual 3D

Dengan menggunakan kamera dan *markerless* objek virtual dapat muncul diatas brosur seperti berada di dunia nyata

3.1.7 Analisis Bagian-Bagian Apartemen Dalam Aplikasi Media Promosi Penjualan Apartemen berbasis AR

Bagian-bagian apartemen yang akan ditampilkan dalam aplikasi ini diantaranya meliputi:

1. Bangunan Apartemen
2. Tipe-Tipe Unit Apartemen

3.1.8 Analisis Markerless

Markerless merupakan salah satu teknologi tracking. Tracking jenis ini menggunakan gambar sebagai markernya. Proses tracking ini menggunakan tekstur gambar (berkas jenis gambar) yang disimpan dalam database sebagai sumber referensinya, dan membandingkan tekstur yang tertangkap oleh kamera dengan tekstur gambar yang ada di database markernya.

Markerless yang digunakan di dalam pembuatan aplikasi AR media promosi penjualan apartemen dapat dilihat pada gambar 3.4.



(a) (b) (c)

Gambar 3.4 *Markerless* Aplikasi AR Media Promosi Apartemen

Keterangan:

Gambar a :*markerless* untuk gambar apartemen 3D

Gambar b :*markerless* untuk tipe unit apartemen studio A

Gambar c :*markerless* untuk tipe unit apartemen studio B

3.1.9 *Real-Time Markerless Tracking Augmented Reality*

Real-Time Markerless Tracking merupakan teknik tracking yang memungkinkan obyek 3D dapat mengikuti pergerakan *tracking object* yang telah disediakan, biasanya teknik ini digunakan untuk fungsi *zoom-in, zoom-out* dan rotasi obyek 3D.Metode ini biasanya digunakan untuk menggantikan fungsi *zoom-in, zoom-out* serta fungsi rotasi yang biasanya menggunakan *markerless*.

Pada tahap ini, akan dicari model *indicator* dengan pengaturan *parameter* berupa:

a. Permukaan 2D
b. Koordinat *x, y, z*
c. Jarak antar *markerless*

3.1.10 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 media promosi apartemen berbasis *Augmented Reality* ini meliputi kebutuhan perangkat keras, kebutuhan perangkat lunak dan pengguna sistem yang akan memakai aplikasi.

3.1.10.1 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 keduanya. Analisis kebutuhan perangkat lunak ini dibagi menjadi tiga bagian, diantaranya seperti :

A. Analisi kebutuhan perangkat lunak pengembang

Kebutuhan perangkat lunak yang digunakan untuk membangun aplikasi media promosi penjualan apartemen dapat dilihat pada tabel 3.1.

Tabel 3.1 Kebutuhan Perangkat Lunak Pengembang

No	Perangkat Lunak
1	*Microsoft windows* 7 Ultimate 32 bit
2	*Autodesk 3D Studio Max 2010*
3	Ogre3D untuk *export* file 3D dari Autodesk 3D *Studio Max*
4	Scol_plugin untuk OpenSpace3D
5	OpenSpace3D Editor
6	*Adobe Photoshop 7*
7	Adobe Flash CS 3
8	StarUML

B. Analisis kebutuhan perangkat lunak pengguna

Kebutuhan perangkat lunak yang dibutuhkanoleh pengguna untuk menjalankan aplikasi media promosi penjualan apartemen dapat dilihat pada tabel 3.2.

Tabel 3.2 Kebutuhan Perangkat Lunak Pengguna

No	Perangkat Lunak
1	*Microsoft windows* 7 Ultimate 32 bit

C. Evaluasi kebutuhan perangkat lunak

Untuk dapat menjalankan aplikasi ini, perangkat lunak yang dibutuhkan oleh pengguna yaitu Sistem Operasi *Microsoft windows 7* Ultimate 32

3.1.10.2 Analisis Kebutuhan Perangkat Keras

Analisis Perangkat keras atau *hardware* merupakan salah satu hal yang penting karena tanpa *hardware* yang memenuhi syarat, aplikasi yang akan dibuat tidak akan dapat berjalan. Analisis kebutuhan perangkat keras ini dibagi menjadi tiga bagian, diantaranya seperti :

A. Analisis kebutuhan perangkat keras pengembang

Kebutuhan perangkat lunak yang digunakan untuk membangun aplikasi media promosi penjualan apartemen dapat dilihat pada tabel 3.3.

Tabel 3.3 Kebutuhan Perangkat Keras Pengembang

No	Perangkat Keras
1	Processor Intel® Core™ i5-2367M CPU@2.60GHz
2	*Random Access Memory* (RAM) 4 GB
3	*Harddisk* dengan space 320 GB
4	VGA Ati Radeon HD 6770 2GB
5	*Webcam* 30 fps 1,3 Mega Pixel

6	Printer Epson T11
7	*Markerless* berupa Brosur Apartemen

B. Analisis kebutuhan perangkat keras pengguna

Kebutuhan perangkat keras yang dibutuhkan untuk untuk menjalankanaplikasi media promosi penjualan apartemen dapat dilihat pada tabel 3.4.

Tabel 3.4 Kebutuhan Perangkat Keras Pengguna

No	Perangkat Keras
1	Processor Intel® Core™ i5-2367M CPU@2.60GHz
2	*Random Access Memory* (RAM) 4 GB
3	*Harddisk* dengan space 320 GB
4	VGA Ati Radeon HD 6770 2GB
5	*Webcam* 30 fps 5,0 Mega Pixel

C. Evaluasi kebutuhan perangkat keras

Untuk dapat menjalankan aplikasi ini, perangkat keras yang dibutuhkan oleh pengguna sebaiknya memiliki spesifikasi yang lebih baik sehingga aplikasi dapat dijalankan secara maksimal.

3.1.11 Analisis Kebutuhan Pengguna

Analisis pengguna dimaksudkan untuk mengetahui siapa saja pengguna aplikasi yang akan di bangun ini, maka dari itu analisis terhadap pengguna perlu dilakukan analisis .

Pengguna aplikasi dibagi menjadi dua bagian, yaitu :

1. Sales Apartemen yang mengerti dan memahami komputer sehingga dapat menjalankan aplikasi dengan baik.
2. Pengguna aplikasi yaitu para konsumen yang ingin mengetahui aplikasi media promosi apartemen.

Berikut ini karakteristik pengguna aplikasi yang ditunjukan oleh tabel 3.5

Tabel 3.5 Karakteristik Pengguna Aplikasi

Pengguna	Hak Akses	Tingkat Pendidikan	Pengalaman
Sales Apartemen	Menjalankan aplikasi dan fitur – fitur yang ada di aplikasi	Minimal D3	Min dapat menjalankan computer dan dapat menjalankan aplikasi serta fungsi – fungsi yang ada di aplikasi
Konsumen	Melihat-lihat fitur yang ada di aplikasi	Minimal SMA atau Sederajat	Dapat menjalankan fitur-fitur yang ada di aplikasi

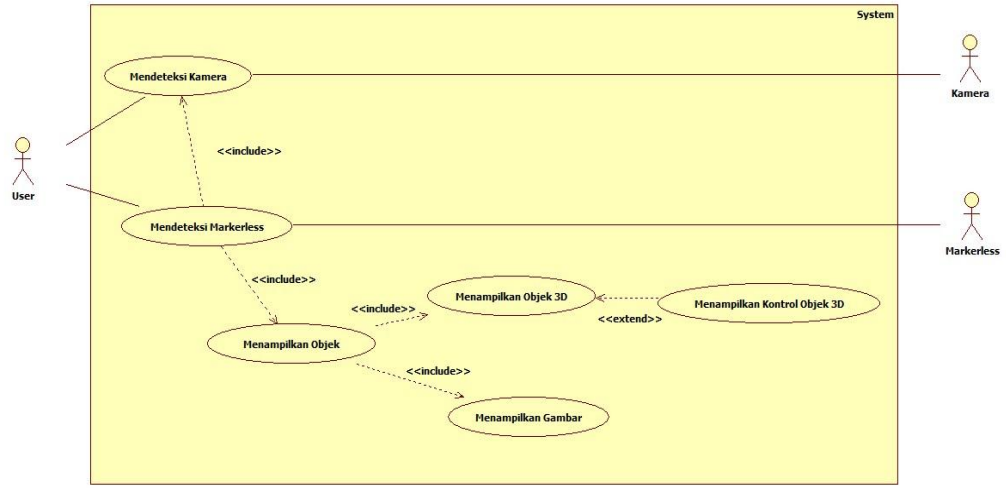
3.1.12 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional ini dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Dimana tahap-tahap perancangan yang dilakukan dalam membangun aplikasi media promosi apartemen berbasis *augmented reality*lain *Use Case Diagram, Class Diagram, Sequence Diagram*, dan *Activity Diagram.*

3.1.12.1 *Use Case Diagram*

Use case Diagram merupakan konstruksi untuk mendeskripsikan hububungan-hubungan yang terjadi antar aktor dengan aktivitas yang terdapat pada sistem. Sasaran pemodelan *use case* diantaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan scenario penggunaan sistem yang akan dibangun.

Berikut ini adalah perancangan proses-proses yang terdapat pada aplikasi media promosi apartemen berbasis *augmented reality*, yang digambarkan dengan Use Case Diagram yang dapat dilihat pada gambar 3.5



Gambar 3.5 *Use Case Diagram* Aplikasi AR Media Promosi Apartemen

1. Identifikasi *Actor*

Actor adalah abstraksi dari orang dan sistem yang lain yang mengaktifkan fungsi dari target sistem. Berikut adalah aktor yang berperan dalam menjalankan sistem yang dibangun. Dapat dilihat pada tabel 3.6

Tabel 3.6 *Actor* dala*m Use Case*

No	Actor	Deskripsi
A-01	User	Merupakan *actor* atau pengguna dari aplikasi yang dibangun
A-02	Kamera	Merupakan *actor* yang berperan sebagai alat bantu yang digunakan sebagai pendeteksi keberadaan markerless
A-03	Markerless	Merupakan *actor* yang berperan sebagai library yang digunakan untuk menampilkan objek 3D

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	Kode *Use Case*	Use Case	Deskripsi
1	UC-01	Mendeteksi Kamera	Proses untuk mendeteksi keberadaan kamera
2	UC-02	Mendeteksi Markerless	Proses untuk mendeteksi keberadaan *markerless* melalui gambar yang ditangkap oleh kamera

3	UC-03	Menampilkan Objek 3D	Proses untuk menampilkan objek dari markerless yang dideteksi oleh kamera
4	UC-04	Menampilkan Kontrol Objek 3D	Proses untuk mengkontol objek dari objek 3D yang ditampilkan

3. Skenario *Use Case*

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

1. Skenario Mendeteksi Kamera dapat dilihat pada tabel 3.8 berikut ini

Tabel 3.8 Skenario Mendeteksi Kamera

Identifikasi		
Nama *Use Case*	Mendeteksi Kamera	
Nomor	UC-01	
Aktor	User, kamera	
Kondisi Awal	Komputer atau laptop memiliki kamera	
Skenario Utama		
Aksi Aktor		**Reaksi Sistem**
1. User memasang kamera		2. Sistem mendeteksi ketersediaan kamera.
Skenario Gagal Mendeteksi Kamera		
Aksi Aktor		**Reaksi Sistem**
Pengecualian	1. Kamera tidak terdeteksi cek ulang pemasangan kamera	
Kondisi Akhir	Kamera terdeteksidan siap mendeteksi markerless	

2. Skenario Deteksi *Markerless* dapat dilihat pada tabel 3.9 berikut ini

Tabel 3.9 Skenario Deteksi *Markerless*

Identifikasi		
Nama *Use Case*	Mendeteksi Markerless	
Nomor	UC-02	
Aktor	User, kamera, markerless	
Kondisi Awal	Kamera terdeteksi dan aplikasi siap untuk mendeteksi markerless	
Skenario Utama		
Aksi Aktor		**Reaksi Sistem**
1. User mengarahkan *markerless* ke kamera.		2. Sistem mendeteksi ketersediaan markerless. 3. Markerless terdeteksi dan siap digunakan
Skenario Gagal Mendeteksi Markerless		
Aksi Aktor		**Reaksi Sistem**
Pengecualian	1. Markerless tidak terdeteksi cek ulang markerless	
Kondisi Akhir	Markerless terdeteksi	

3. Skenario Deteksi Menampilkan Objek 3D dapat dilihat pada tabel 3.10 berikut ini

Tabel 3.10 Skenario Menampilkan Objek 3D

Identifikasi	
Nama *Use Case*	Menampilkan Kontrol Objek 3D
Nomor	UC-03
Aktor	User, kamera, markerless
Kondisi Awal	Markerless terdeteksi dan siap menampilkan objek 3D
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. User mengarahkan *markerless* ke kamera.	2. Sistem mendeteksi ketersediaan markerless. 3. Kamera menangkap pattern dari *markerless* 4. Sistem menampilkan objek-objek 3D
Skenario Gagal Mendeteksi Markerless	
Aksi Aktor	**Reaksi Sistem**
Pengecualian	1. Library tidak menyimpan objek sehingga objek 3D tidak muncul
Kondisi Akhir	Menampilkan objek 3D

4. Skenario Deteksi Menampilkan Kontrol Objek 3D dapat dilihat pada tabel 3.11 berikut ini

Tabel 3.11 Skenario Menampilkan Kontrol Objek 3D

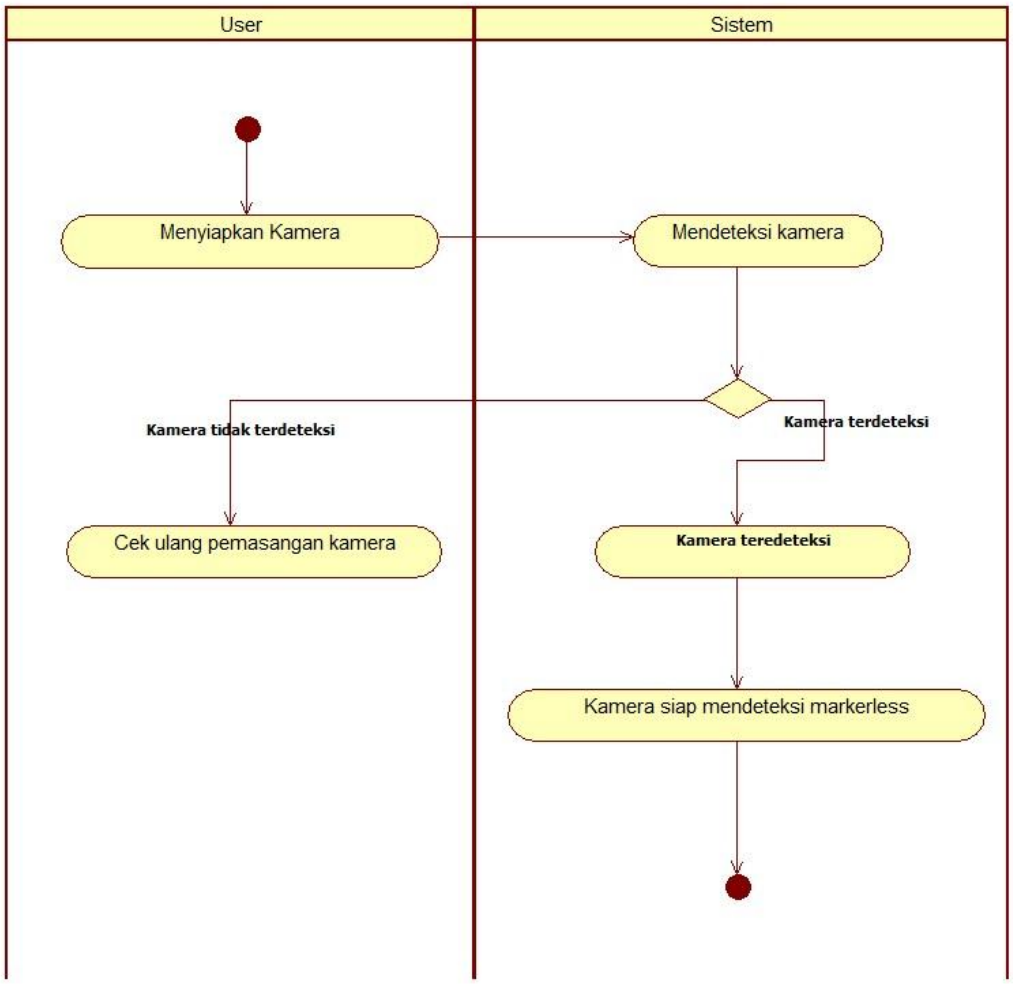
Identifikasi	
Nama *Use Case*	Menampilkan Kontrol Objek 3D
Nomor	UC-04
Aktor	User, kamera, markerless
Kondisi Awal	Objek 3D ditampilkan oleh sistem
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. User siap melakukan kontrol objek 3D	2. Sistem menampilkan objek 3D. 3. Sistem menampilkan objek-objek 3D yang dikontrol oleh user
Skenario Gagal Mendeteksi Markerless	
Aksi Aktor	**Reaksi Sistem**
Pengecualian	1. Objek 3D tidak muncul
Kondisi Akhir	Objek siap dikontrol oleh user

3.1.12.2 *Activity Diagram*

Diagram activity menggambarkan berbagai alir aktivitas dalam sistem yang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. Penggambaran *activity* diagram memiliki kemiripan dengan *flowchart diagram. Activity diagram* memodelkan *event-event*

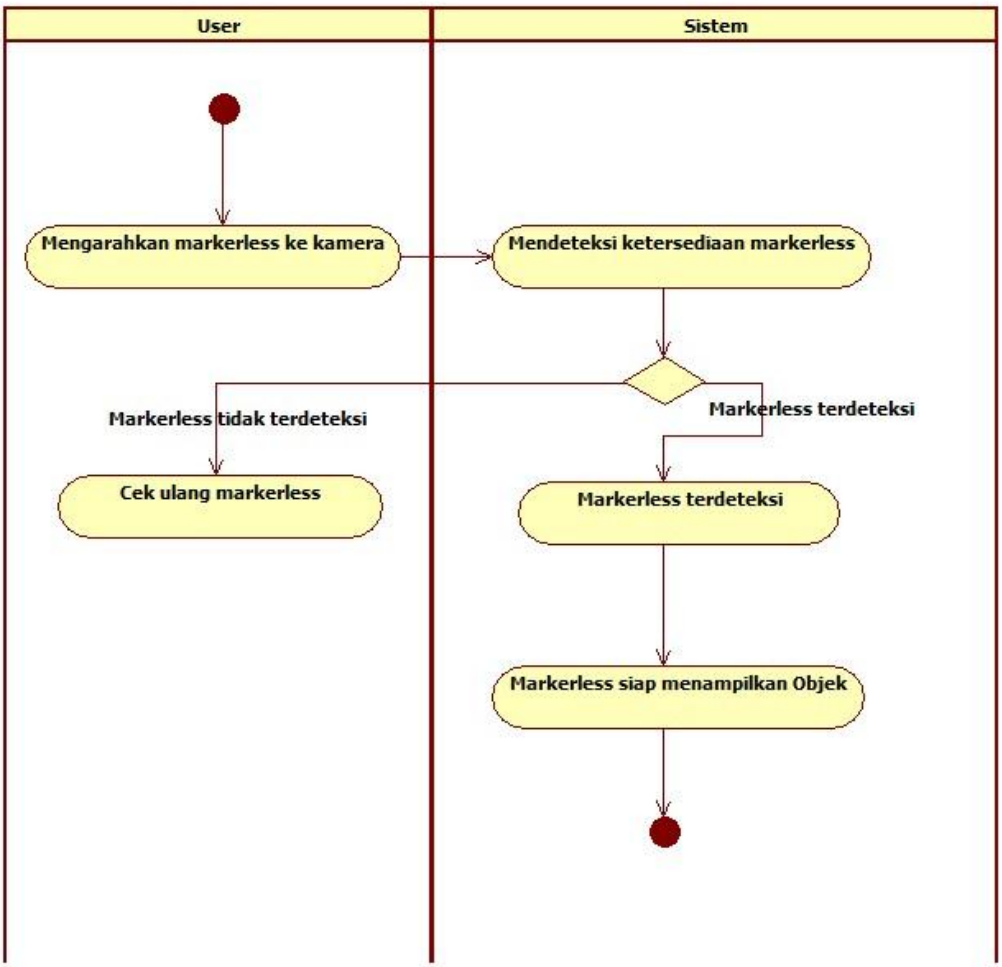
yang terjadi pada *Use Case* dan digunakan untuk pemodelan aspek dinamis dari sistem.

1. *Activity diagram* terkait dengan skenario Mendeteksi Kamera dapat dilihat pada gambar 3.6.



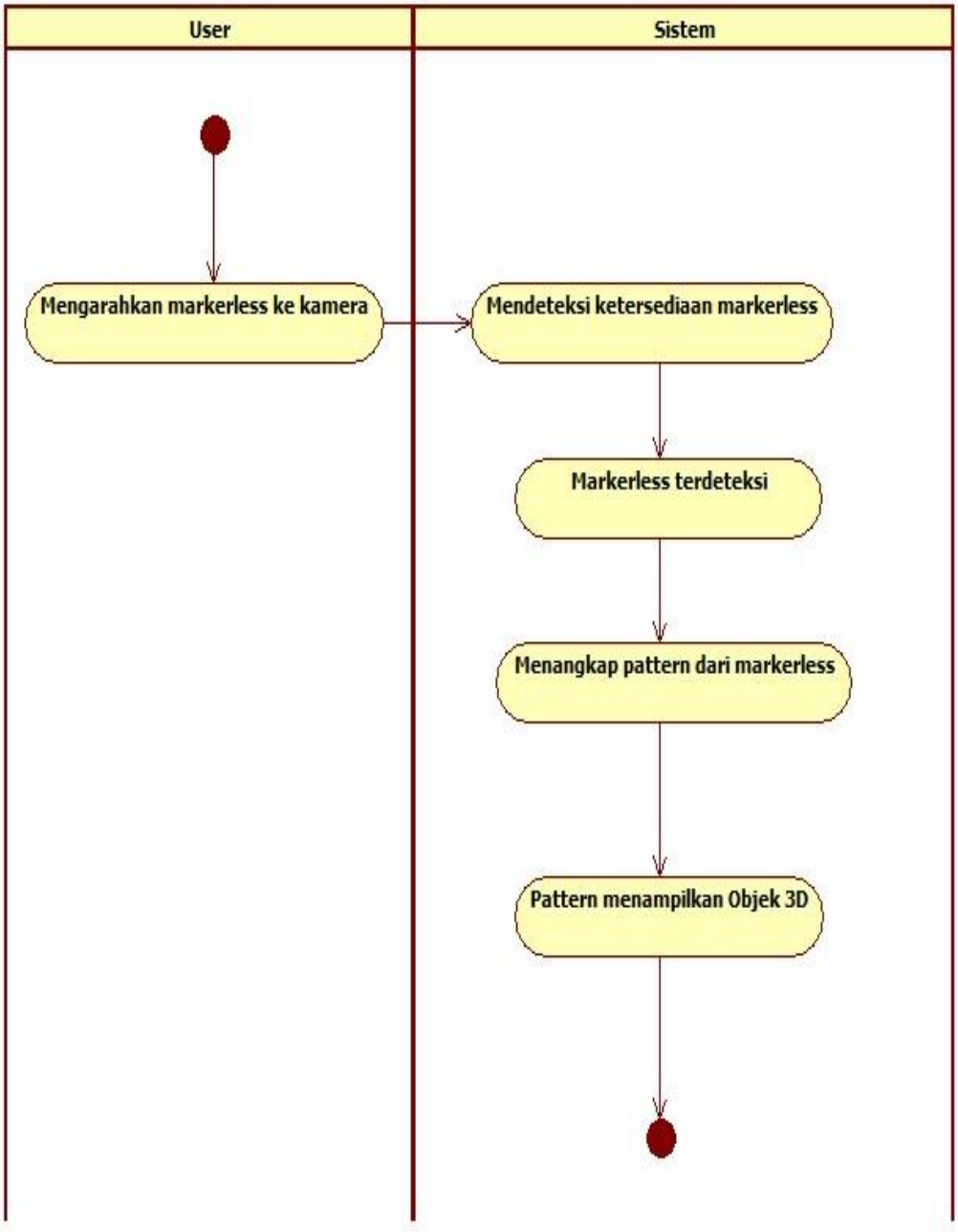
Gambar 3.6 *Activity Diagram* Mendeteksi Kamera

2. *Activity diagram* terkait dengan skenario Deteksi Markerless dapat dilihat pada gambar 3.7.



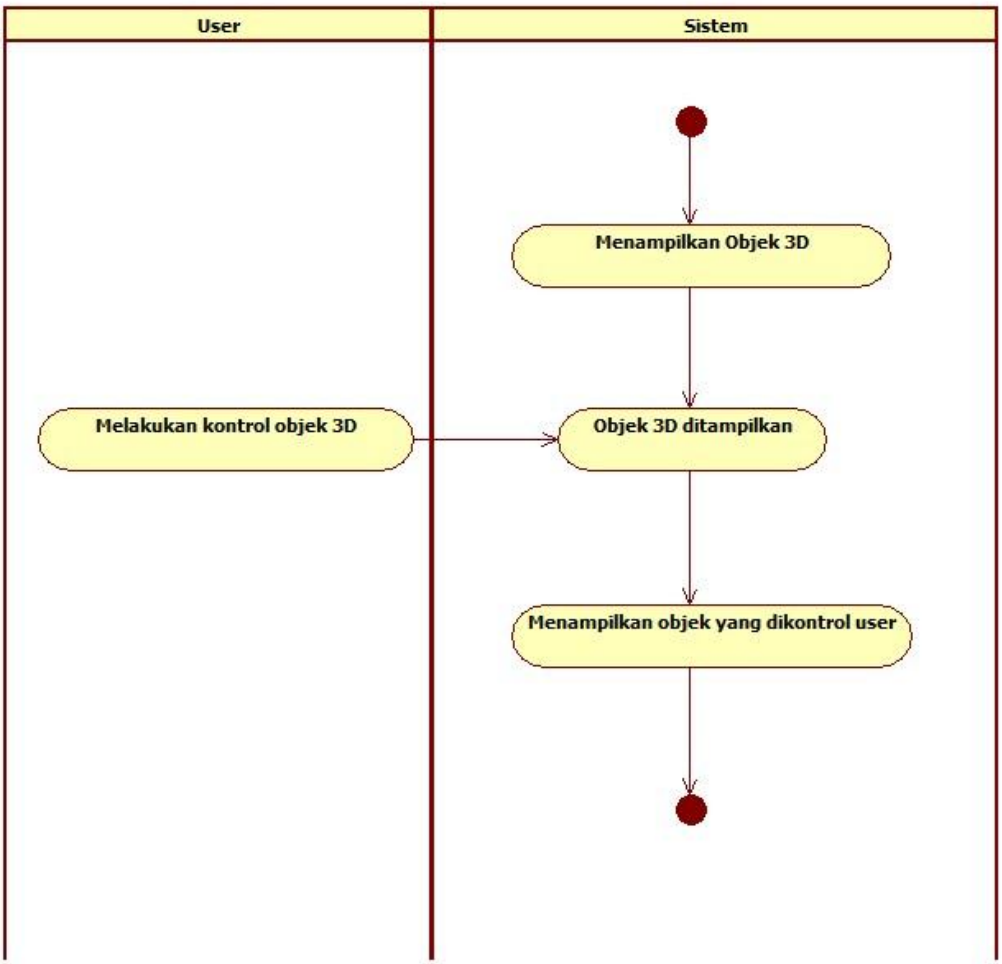
Gambar 3.7 *Activity Diagram* Deteksi *Markerless*

3. *Activity diagram* terkait dengan skenario Menampilkan Objek dapat dilihat pada gambar 3.8.



Gambar 3.8 *Activity Diagram* Menampilkan Objek 3D

4. *Activity diagram terkait* dengan skenario Mengontrol Objek 3D dapat dilihat pada gambar 3.9.

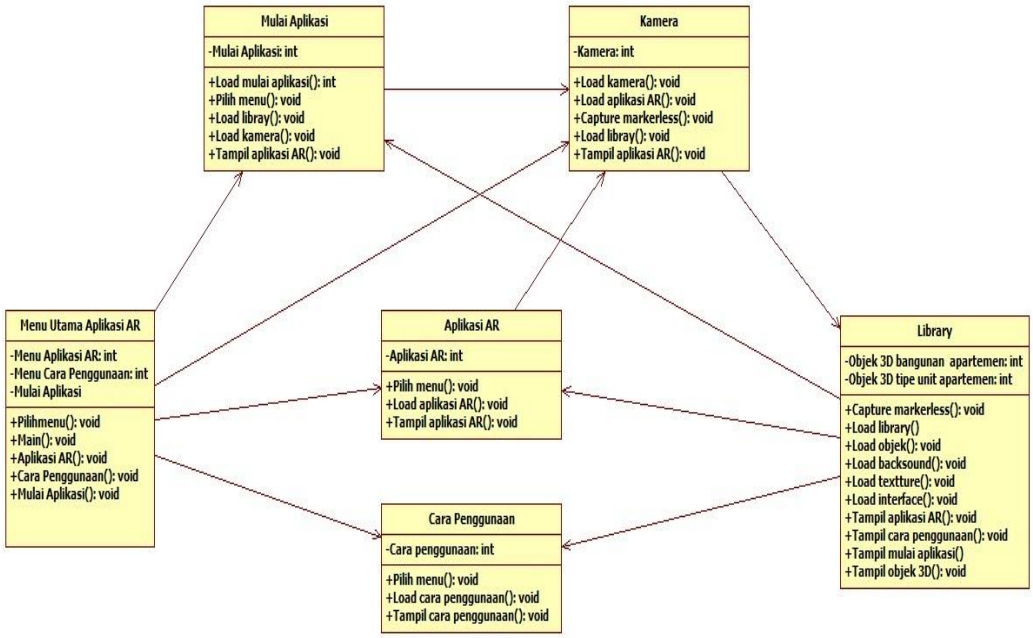


Gambar 3.9 *Activity Diagram* Mengontrol Objek 3D

3.1.12.3 *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).

Perancangan struktur sistem yang terdapat pada aplikasi AR media promosi Apartemen Taman Sari Panoramic yang digambarkan dengan *Class Diagram* yang dapat dilihat pada gambar 3.10.



Gambar 3.10 *Class Diagram* Aplikasi AR Media Promosi Apartemen

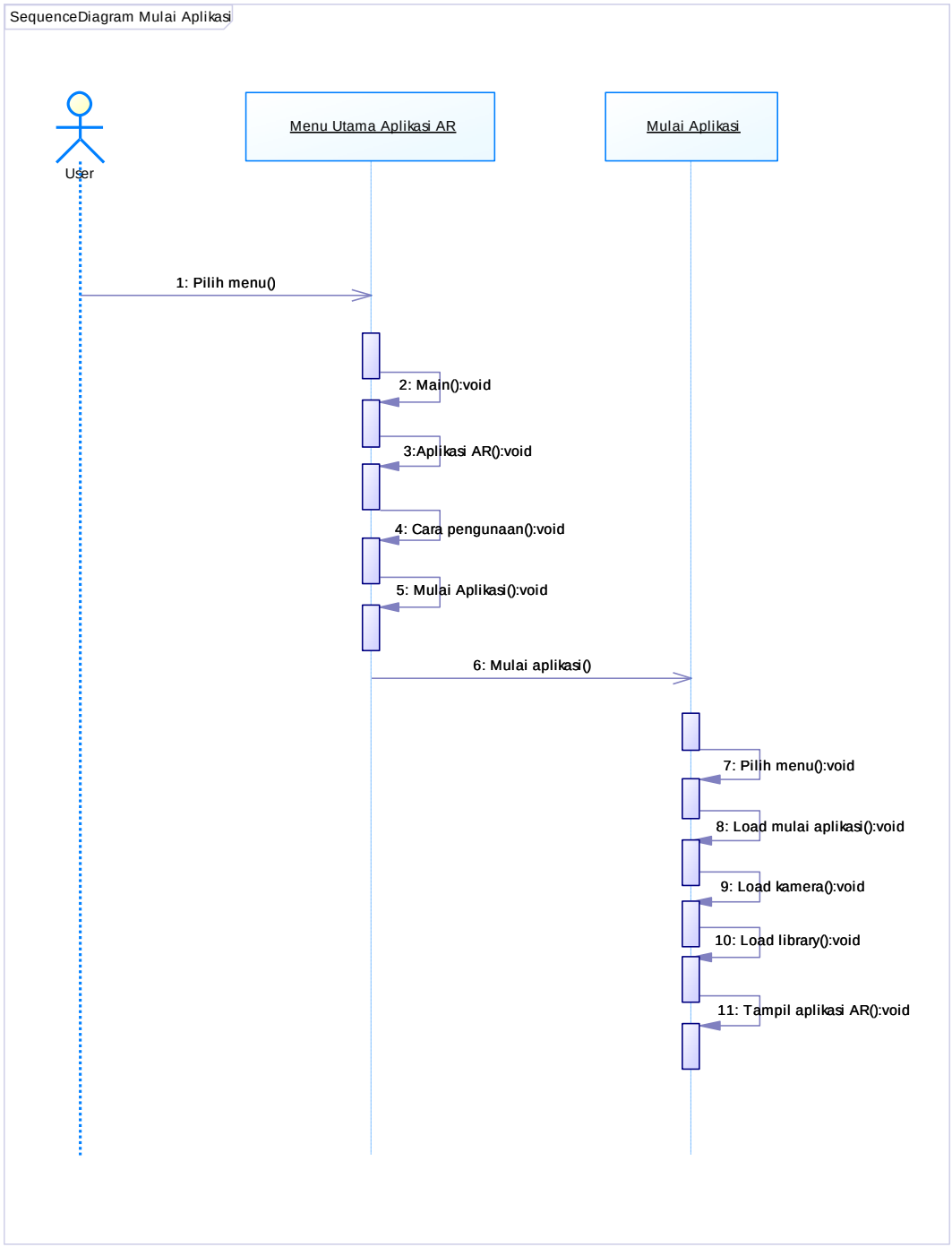
***Class Diagram* Aplikasi AR**		
Class	**Jenis Class**	**Deskripsi**
Menu Utama Aplikasi AR	Interface	Class yang berisikan metode-metode yang berfungsi menampilkan halaman pada tiap menu
Aplikasi AR	Interface	Class yang berisikan metode-metode untuk menampilkan informasi mengenai aplikasi AR
Cara Penggunaan	Interface	Class yang berisikan metode-metode untuk menampilkan informasi mengenai cara penggunaan
Mulai Aplikasi	Control	Class yang berisikan method-method untuk menjalankan fungsionalitas menjalankan Aplikasi AR
Kamera	Control	Class yang berisikan method
Library	Entity	Class yang berisikan method-method ke database serta proses pencocokkan gambar

3.1.12.4 *Sequence Diagram*

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

1. *Sequence Diagram* Mulai Aplikasi

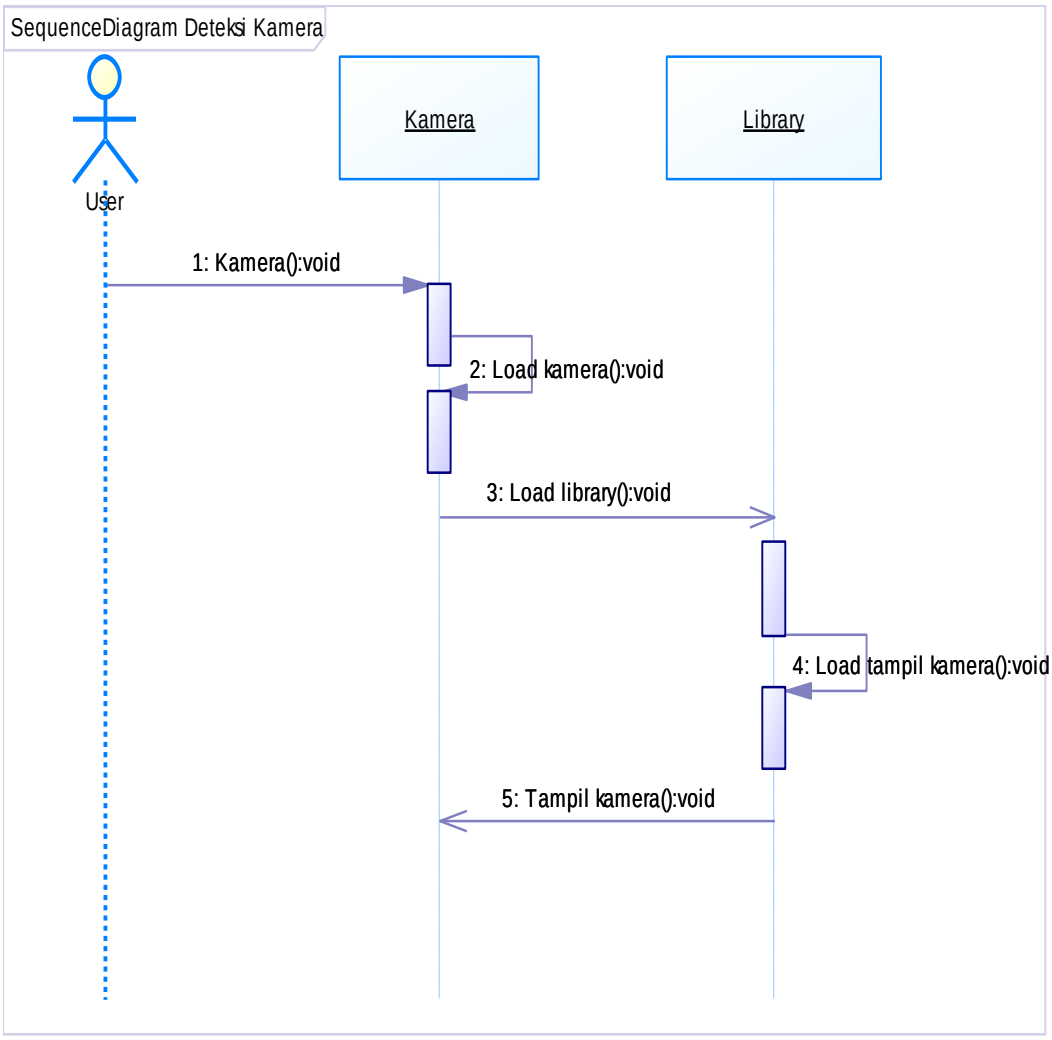
Sequence Diagram Mulai Aplikasi menunjukan rangkaian pesan yang dikirim oleh user dalam memulai aplikasi mulai dari menu utama, aplikasi AR, cara penggunaan dan mulai aplikasi. *Sequence Diagram* Mulai Aplikasi dapat dilihat pada gambar 3.11.



Gambar 3.11 *Sequence Diagram* Mulai Aplikasi

2. *Sequence Diagram* Deteksi Kamera

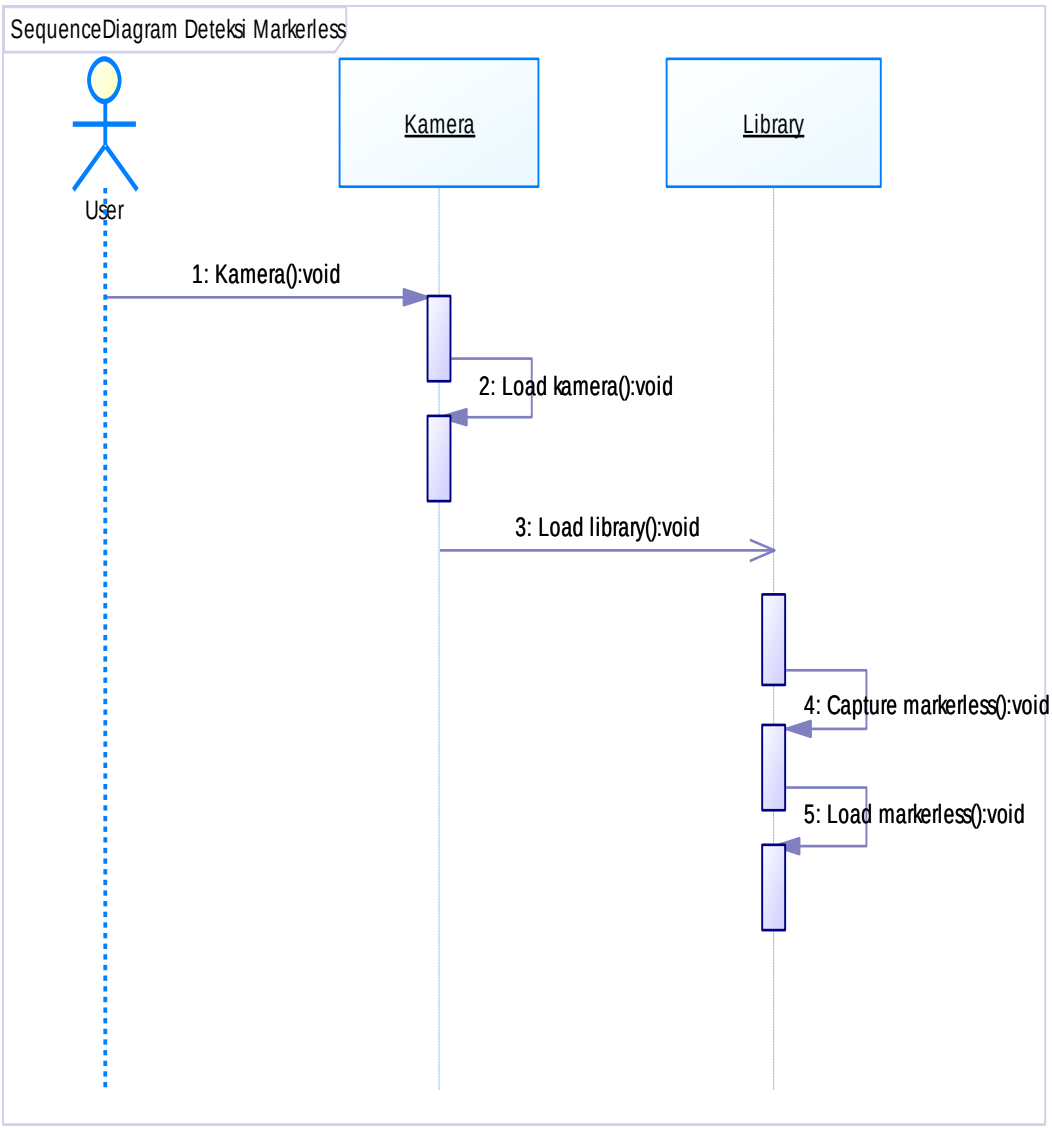
Sequence Diagram Deteksi kamera menunjukan rangkaian pesan yang dikirim oleh user dalam mendeteksi kamera. *Sequence Diagram* Deteksi Kamera dapat dilihat pada gambar 3.12.



Gambar 3.12 *Sequence Diagram* Deteksi Kamera

3. *Sequence Diagram* Deteksi *Markerless*

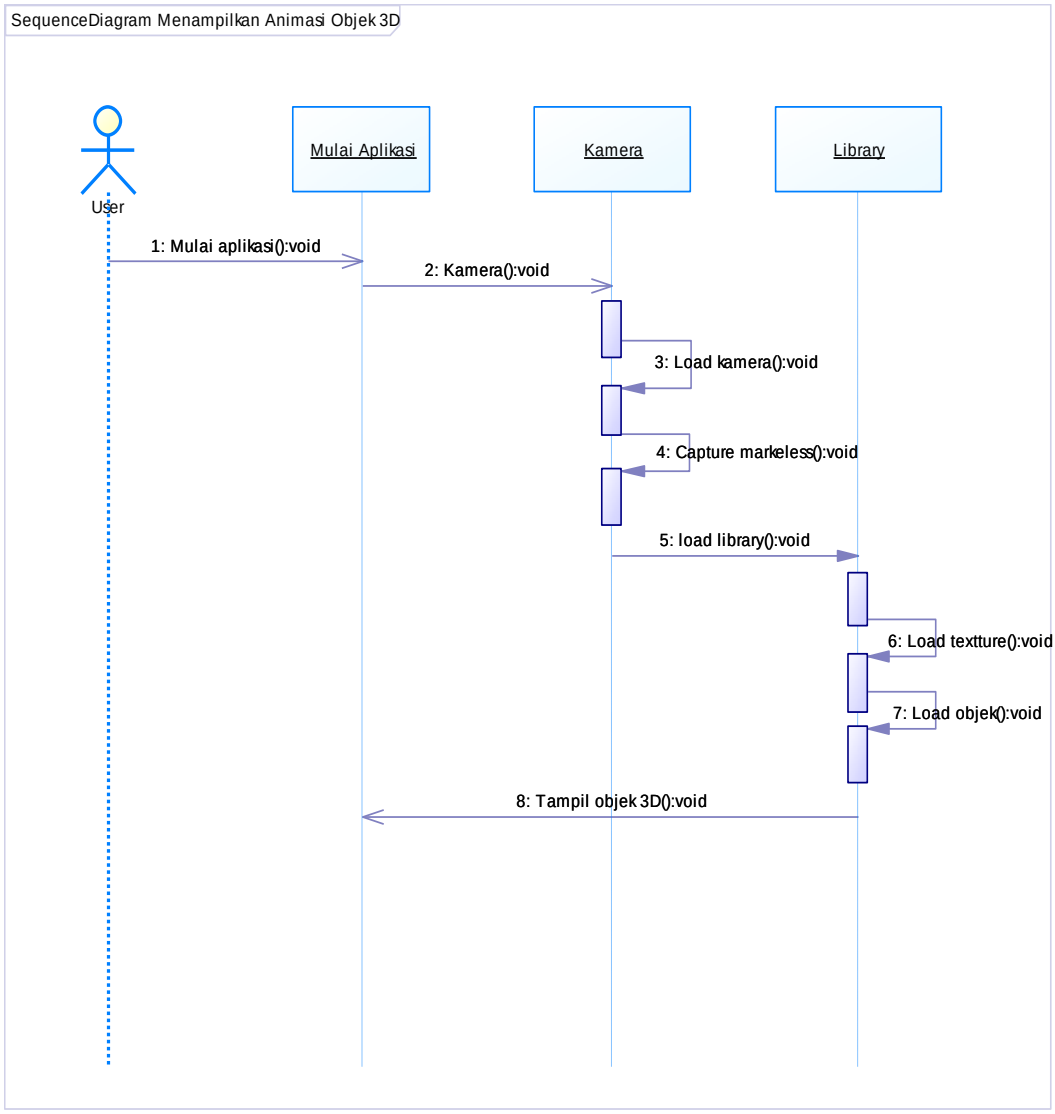
Sequence Diagram Deteksi *Markerless* menunjukan rangkaian pesan yang dikirim oleh user dalam mendeteksi markerless. *Sequence Diagram* Deteksi Markerless dapat dilihat pada gambar 3.13.



Gambar 3.13 *Sequence Diagram* Deteksi *Markerless*

4. *Sequence Diagram* Menampilkan Objek 3D

Sequence Diagram Menampilkan Objek 3D menunjukan rangkaian pesan yang dikirim oleh user dalam menampilkan objek. *Sequence Diagram* Menampilkan Objek 3D dapat dilihat pada gambar 3.14



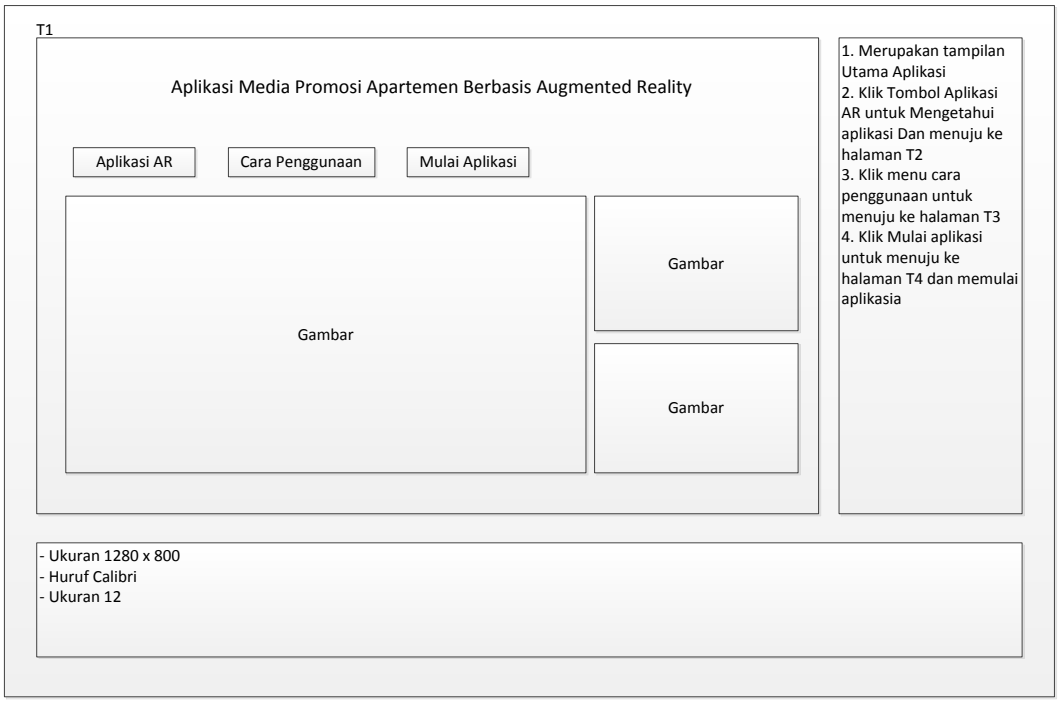
Gambar 3.14 *Sequence Diagram* Menampilkan Objek 3D

3.2 Perancangan

Perancangan antarmuka (*interface*) menggambarkan bagaimana perangkat lunak berkomunikasi dengan dirinya sendiri, dengan sistem yang berinteroperasi dengannnya, dan dengan manusia yang menggunakannya.

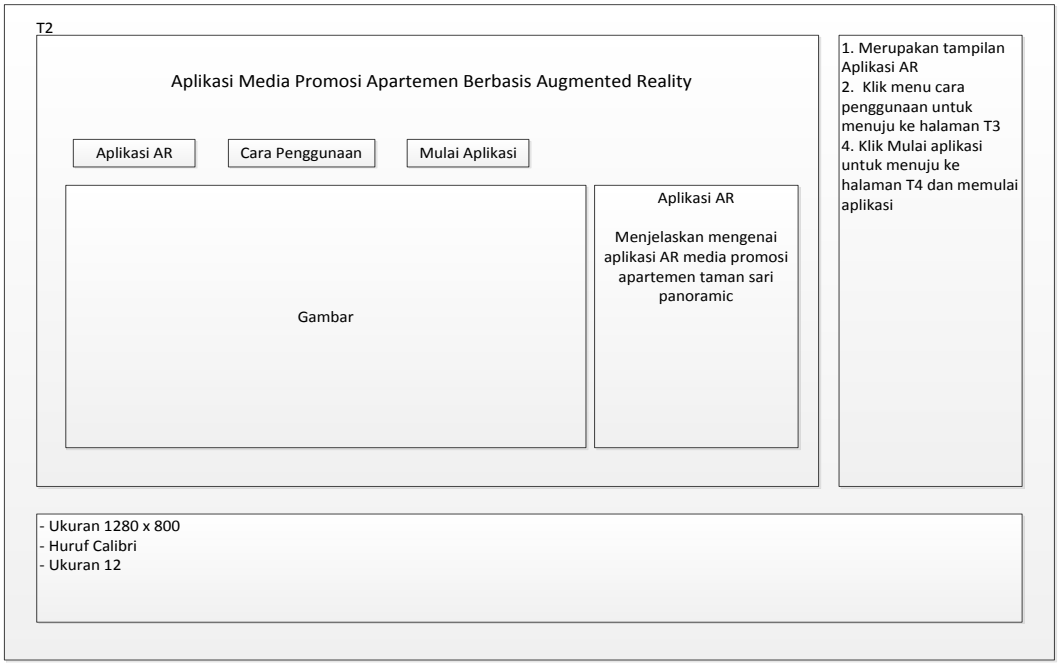
3.2.1 Perancangan Antarmuka

Desain yang dibuat berupa aplikasi berbasis desktop yang mudah untuk digunakan oleh pengguna, serta informasi yang dihasilkan aplikasi tersebut dapat dimengerti oleh pengguna.Perancangan antarmuka bertujuan untuk memberikan gambaran aplikasi yang dibuat. Objek tiga dimensi merupakan objek yang akan ditampilkan pada *markerless* yang dapat dilihat oleh pengguna dengan bantuan *webcam*. Gambaran umum dari perancangan antarmuka Aplikasi AR Media Promosi Apartemen dapat dilihat pada gambar 3.15.



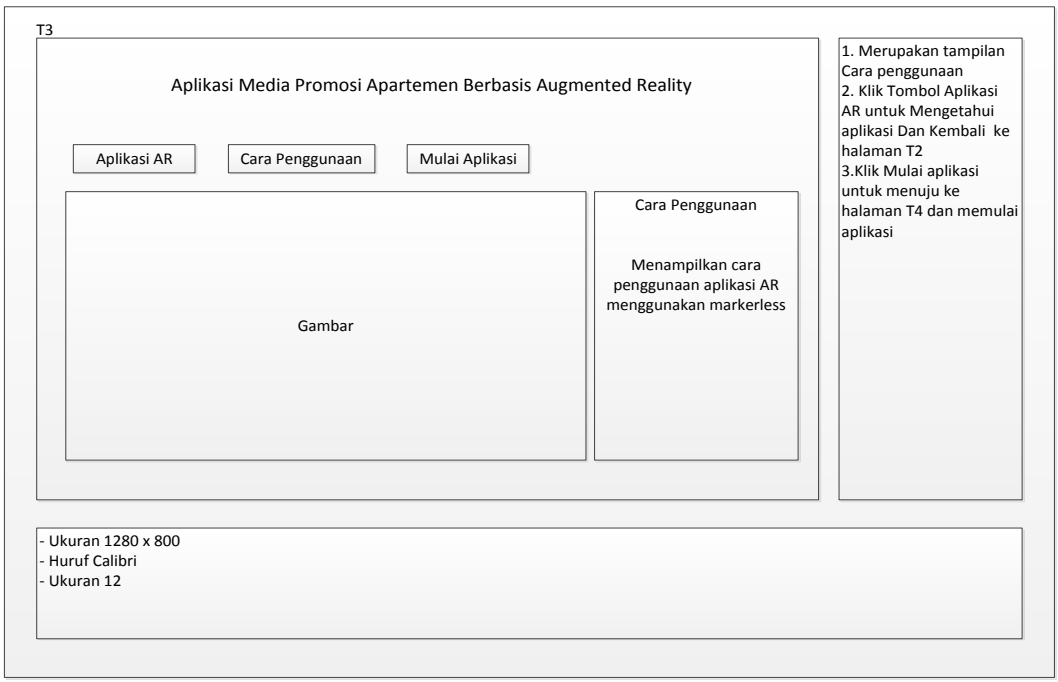
Gambar 3.15 Antarmuka Aplikasi AR Media Promosi Apartemen

Gambaran umun dari perancangan antarmuka menu Aplikasi AR dapat dilihat pada gambar 3.16



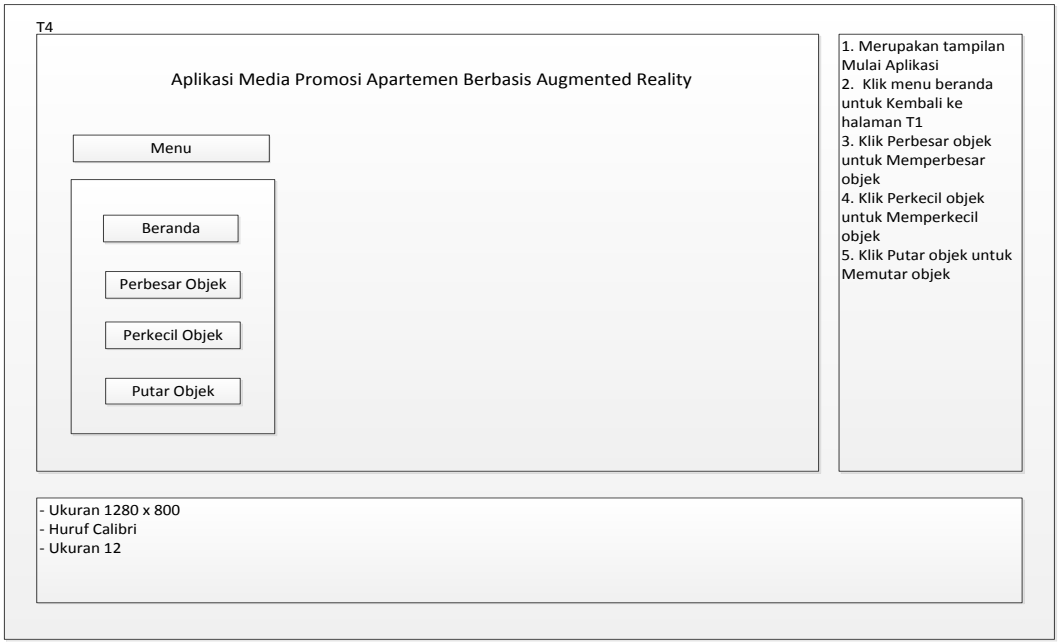
Gambar 3.16 Tampilan Menu Aplikasi AR

Gambaran umun dari perancangan antarmuka menu Cara Penggunaan dapat dilihat pada gambar 3.17



Gambar 3.17 Tampilan Menu Cara Penggunaan

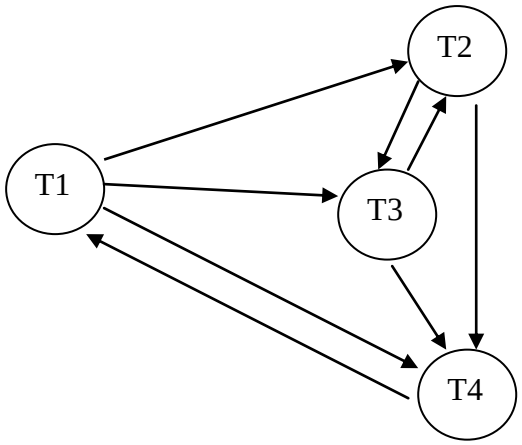
Gambaran umun dari perancangan antarmuka menu Mulai Aplikasi AR dapat dilihat pada gambar 3.18



Gambar 3.18 Tampilan Menu Mulai Aplikasi AR

3.2.2 Jaringan Semantik

Jaringan semantik adalah diagram yang menggambarkan aliran-aliran menu dan pesan dalam sebuah program. Berikut ini adalah jaringan semantik pada sistem yang dibangun dapat dilihat pada gambar 3.19



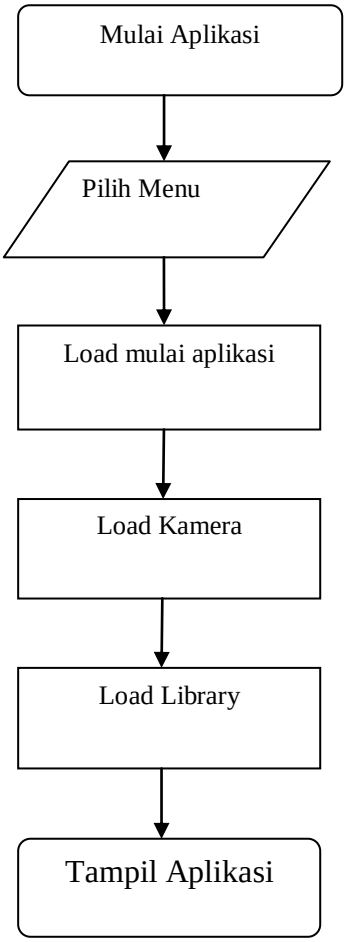
Gambar 3.19 Jaringan Semantik Aplikasi AR Media Promosi Apartemen

3.2.3 Perancangan Method

Perancangan komponen method merupakan perancangan yang dibuat setelah perancangan antarmuka dan jaringan semantik. Perancangan ini berfungsi untuk mendeskripsikan method-method yang berada di dalam aplikasi AR media promosi apartemen. Adapun method-method yang terdapat dalam AR media promosi apartmen yang dibangun adalah sebagai berikut :

A. Method Mulai_Aplikasi.mulai_aplikasi()

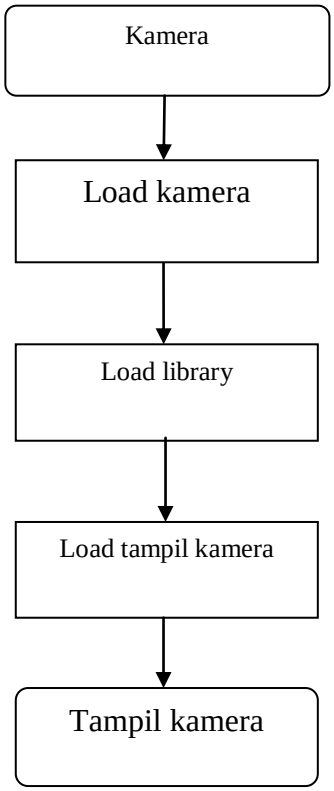
Method Mulai_Aplikasi.mulai_aplikasi() akan dipanggil ketika user akan memulai Aplikasi, seperti terlihat pada gambar 3.20.



Gambar 3.20 *Method* Mulai_Aplikasi.mulai_aplikasi()

B. Method Kamera.kamera()

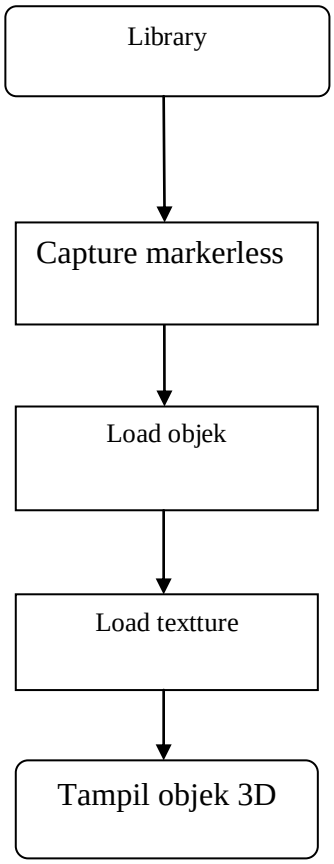
Method Kamera.kamera() dipanggil untuk mendeteksi kamera, seperti pada gambar 3.21.



Gambar 3.21 *Method* Kamera.kamera()

C. Method Library.objek()

Method Library.objek() dipanggil untuk menampilkan objek 3D, seperti pada gambar 3.22.



Gambar 3.22 *Method* Library.objek()