BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis bertujuan untuk mengidentifikasi permasalahan-permasalahan yang terdapat pada sistem serta menentukan kebutuhan-kebutuhan dari sistem yang dibangun. Analisis tersebut meliputi analisis masalah, analisis kebutuhan data, analisis arsitektur sistem, analisis metode, analisis data fungsional dan non fungsional. Analisis sistem merupakan penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan solusi atau perbaikan. Dari hasil analisis tersebut dapat dirancang atau diperbaiki menjadi sebuah sistem yang lebih mudah, praktis dan interaktif. Sistem yang dibuat merupakan aplikasi untuk mendeteksi *marker* berupa objek monumen yang diambil menjadi sebuah gambar (foto). Objek yang dibuat merupakan bangunan monumen bersejarah di kota Bandung, dimana pengguna seolah-olah berinteraksi langsung dengan objek virtual dalam dunia nyata yang disajikan dalam bentuk foto atau gambar.

3.1.1 Analisis Masalah

Analisis masalah merupakan langkah awal dari analisis sistem. Langkah ini diperlukan untuk mengetahui pemasalahan apa saja yang terjadi didalam sistem yang telah berjalan. Penelitian ini dilakukan dengan memberikan kuisioner secara langsung kepada masyarakat. Berdasarkan hasil dari kuisioner yang telah diberikan bahwa terdapat 30 orang yang mengisi kuisioner, dimana sebanyak 16 orang menyatakan kurang mengetahui dengan monumen bersejarah yang berada di kota Bandung, 3 orang ragu-ragu dan 11 orang yang cukup mengetahui dengan adanya monumen bersejarah. Penyampaian informasi yang minim dan kurang informatif pada monumen tersebut menjadi kendala bagi pengunjung yang datang dan ingin mengetahui informasi secara langsung. Sehingga untuk dapat mengetahuinya masyarakat harus menanyakan kebagian informasi atau

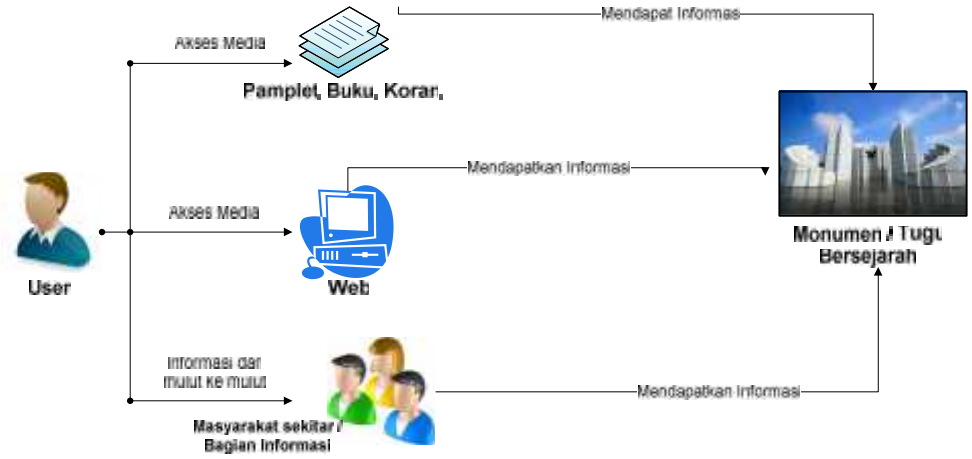
mengakses internet terlebih dahulu. Sebagian masyarakat yang datang pun kadang membuat mereka tidak mengetahui akan bangunan monumen bersejarah sehingga hanya dianggap sebagai objek foto atau bangunan hiasan saja, bukanlah sebagai monumen bersejarah.

Maka dengan adanya teknologi yang berkembang sekarang ini, khususnya teknologi *mobile* yang unggul dengan sisi ruang geraknya (mobilitas), *user friendly*, serta praktis sehingga dapat menutupi kekurangan yang terdapat pada media saat ini. Berdasarkan permasalahan yang ada maka akan dibangun aplikasi *mobile* sebagai media pengenalan menggunakan teknologi *Augmented Reality* (AR). Aplikasi ini bertujuan menampilkan informasi secara *real time* berupa text maupun video dilayar telepon.

Identifikasi pengenalan gambar bangunan bersejarah menggunakan *Library Qualqomm Vuforia* dimana *library* ini menggunakan metode *Natural Feature Tracking* sebagai pengenalan pola gambar sehingga gambar yang akan ditangkap oleh kamera telepon seluler dapat dikenali sebagai *markerless*. *Markerless* AR inilah yang berfungsi untuk menampilkan informasi tersebut secara *real-time* dilayar telepon seluler, khususnya android.

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. Analisa prosedur pada proses media yang sedang berjalan sekarang pada bangunan monumen bersejarah kota Bandung adalah sebagai berikut:



Gambar 3.1 Alur Sistem Yang Sedang Berjalan

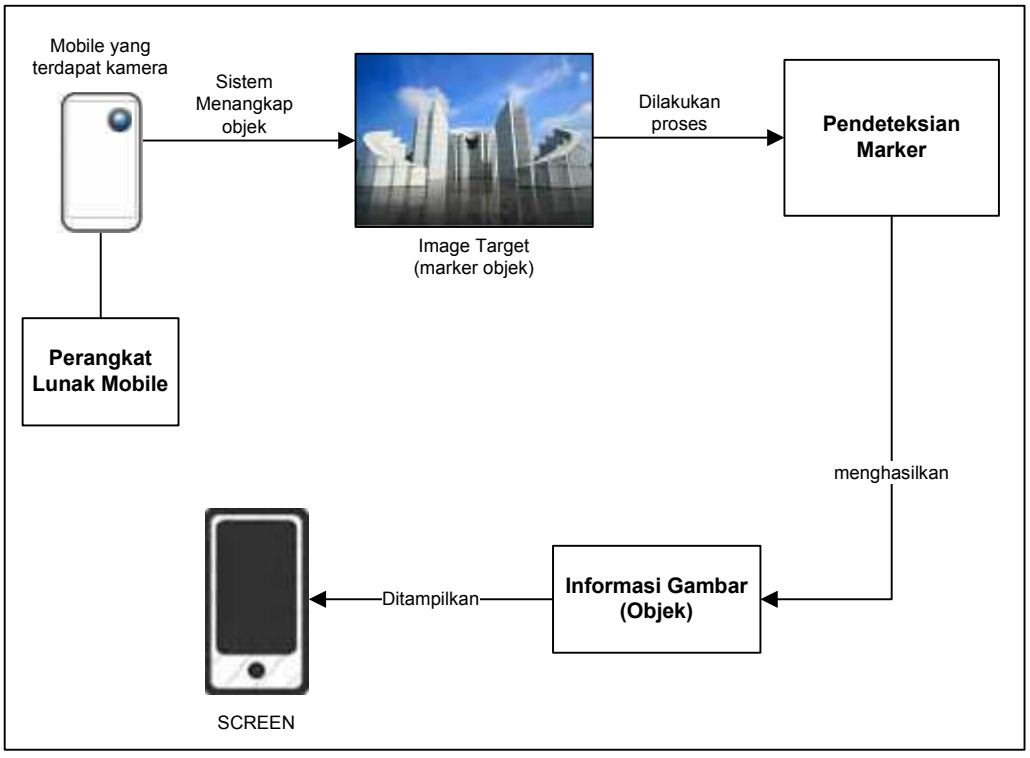
Keterangan alur prosedur pada sistem yang sedang berjalan sebagai berikut:

1. User merupakan masyarakat atau pengunjung yang ingin mengetahui informasi mengenai bangunan monumen bersejarah tersebut.
2. Pamplet, koran, buku merupakan media cetak yang berisi informasi baik berupa text maupun gambar pada media tersebut. Media ini cukup mudah untuk mengetahui informasi sejarah bangunan tersebut, namun terkadang tidak semua orang bisa mendapatkan pamplet atau kurang menariknya tampilan lembaran media cetak yang ada sehingga masyarakat terkadang mengabaikan info yang ada dalam selembaran info tersebut.
3. Web merupakan media yang terkoneksi dengan internet dan untuk mengaksesnya diperlukan perangkat hardware yang cukup nyaman atau layar akses cukup besar seperti komputer atau laptop, serta diperlukan alamat web yang singkat dan mudah diingat oleh banyak orang agar mempermudah akses web tersebut, sehingga untuk mengakses media ini pengunjung setidaknya mempunyai perangkat hardware yang memenuhi kriteria tersebut dan mengetahui alamat situs web yang menyediakan informasi monumen/tugu bersejarah tersebut.
4. Masyarakat/Bagian informasi, informasi yang disampaikan dari mulut kemulut membuat sebagian pengunjung hanya mengetahui dan mengenal nama bangunan bersejarah tersebut namun tidak tahu secara detail akan sejarah bangunan bersejarah tersebut.

Dari gambaran prosedur diatas terlihat telah terdapat beberapa media yang dapat memperkenalkan monumen bersejarah namun masih terdapat kekurangan diantaranya informasi yang kurang diketahui dan cara mendapatkan informasi yang kurang menarik bagi pengunjung, serta akses web yang terkadang bermasalah dijaringan koneksi dan alamat web yang kurang diketahui oleh user.

3.1.3 Analisis Alur Sistem

Sistem yang akan dibangun merupakan sebuah aplikasi *mobile* sebagai media pengenalan bangunan sejarah menggunakan platform Android dengan teknologi *Augmented Reality*. Teknologi ini dapat dengan mudah diakses dimana saja dan sangat berkembang pada masa kini sehingga memudahkan user untuk mengakses aplikasi ini. Keunggulan dari teknologi *augmented reality* juga sangat menonjol dalam segi menampilkan suatu informasi secara *real-time* sehingga dapat diterapkan kedalam aplikasi, untuk dapat memberikan solusi dari permasalahan yang ada. Tujuan yang ingin dicapai dari perancangan aplikasi ini adalah dapat menampilkan informasi mengenai monumen bersejarah secara *real-time* melalui pengenalan pola (*pattern recognition*).



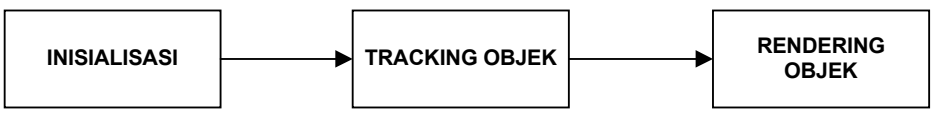
Gambar 3.2 Alur Sistem Yang Akan Dibangun

Berikut penjelasan arsitektur sistem pada aplikasi ini, yaitu:

1. Perangkat Lunak Mobile, perangkat lunak tersebut diinstall pada *mobile* android yang memiliki kamera.
2. *Mobile (handphone),* sebagai alat untuk mendeteksi gambar pada kamera ketika diarahkan ke marker
3. Image Target (marker objek), merupakan target gambar atau objek yang akan dideteksi oleh sistem.
4. Pendeteksian Marker, proses mengenali keberadaan pola dari hasil gambar yang telah ditangkap oleh kamera pada gambar atau foto yang nantinya akan dijadikan *object tracker*. Setelah itu dilakukan pengenalan pola dengan mengenalkan pola gambar yang nantinya akan dijadikan *markerless* dengan data yang ada pada sistem, kemudian dilakukan pendeteksian dan penyesuaian pola untuk menentukan objek marker AR dapat mengeluarkan informasi atau tidak.
5. Informasi pada gambar (objek), informasi yang akan muncul pada layar *handphone* ketika terdeteksi oleh sistem.
6. *Screen,* informasi yang muncul akan ditampilkan dilayar *handphone* atau monitor laptop.

3.1.4 Analisis Alur Sistem Pada *Augmented Reality* (AR)

Analisis alur sistem merupakan analisis yang mendeskripsikan bagaimana proses alur sistem pada *augmented reality* mulai dari inisialisasi, *tracking objek* hingga proses rendering objek. Secara umumnya tahapan perancangan ini ada tiga bagian utama, diantaranya:



Gambar 3.3 Alur Sistem AR

3.1.4.1 Inisialisasi Marker

Pada tahap ini ditentukan *marker* yang akan digunakan, sumber input videonya, dan objek yang akan digunakan baik berupa objek teks, suara maupun video. Pada bagian inisialisasi ini, aplikasi ini akan dihubungkan dengan tampilan kamera untuk memulai proses pendeteksian marker. Kemudian sistem akan melakukan inisialisasi variabel awal yang dipakai untuk menampung sementara marker, status pengecekan, serta variabel informasi yang akan ditampilkan.

3.1.4.1.1 Proses Pembentukan Marker

Marker digunakan sebagai penanda sebuah pola yang terekam dalam sebuah kamera *real-time*. Marker biasanya identik dengan ilustrasi hitam dan putih dalam sebuah persegi dengan batas hitam tebal dan latar belakang putih. Penanda dilakukan dengan cara markerless, dimana *tracking* jenis ini menggunakan objek gambar sebagai markernya. Proses *tracking* ini menggunakan tekstur gambar (berkas jenis gambar) yang disimpan dalam *library vuforia* sebagai sumber referensinya, dan membandingkan tekstur yang tertangkap oleh kamera dengan tekstur gambar yang ada disistem markernya. Dalam perancangan aplikasi dengan teknologi AR, menggabungkan objek virtual dengan objek nyata, dalam hal ini objek virtual berupa objek text dan video serta objek nyatanya berupa gambar dengan pola tertentu (markerless).



Gambar 3.4 Rancang Bangun Markerless

Adapun proses pembentukan *image target* adalah sebagai berikut :



Gambar 3.5 Proses Pembentukan Marker

Pada sistem *markerless* ini tidak perlu adanya marker yang identik dengan daerah kotak hitam putih namun target marker dapat berupa objek langsung atau gambar. *Image target* dari sebuah gambar yang akan dijadikan target marker sebagai pengenalan pola pada *vuforia* akan dikenali dan dideteksi keberadaan polanya. Proses pembuatan *image target* dilakukan dengan mengupload gambar pada web vuforia (www.developer.vuforia.com) sebagai penyedia fasilitas untuk menghasilkan *target.package*. Setelah gambar di-*convert* menghasilkan file dengan format .package. File ini nantinya akan dijadikan masukan pada sistem untuk mendeteksi gambar yang akan dijadikan *marker*. Pola target minimal harus memiliki lebar 550 pixel dan format gambar yang dikirimkan berupa .jpg.



Gambar 3.6 Contoh Marker Gambar Monumen

3.1.4.1.2 Input Data Marker Objek

Data yang akan dimasukan ke dalam sistem *library vuforia* berupa gambar, teks, dan video. Vuforia menyediakan fasilitas untuk meng-*convert* gambar menjadi marker (*www.developer.vuforia.com*). Hasil objek yang telah di-*convert* ke *vuforia* berupa file .unitypackage. File ini nantinya akan dimasukan ke sistem sebagai inisialisasi image target terhadap objek.

3.1.4.2 *Tracking* Marker

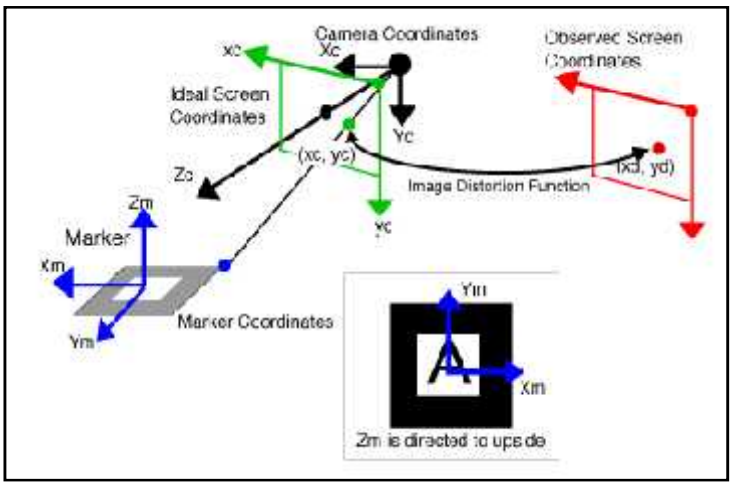
Proses *tracking* dilakukan dengan mengenali pola pada gambar yang ditangkap (diterima) oleh sensor kamera. *Library Vuforia* sebagai *tool* pendukung UNITY memiliki kemampuan untuk mendeteksi gambar menggunakan kamera standar. Informasi posisi yang didapatkan akan dipergunakan untuk menempatkan objek kedalam posisi gambar (*markerless*). Setelah pola gambar dikenali dan terdeteksi maka akan tampil informasi dilayar telepon seluler android.

3.1.4.2.1 Proses Pengenalan Pola *(Pattern Recognition)*

Berikut proses pengenalan pola pada *feature* gambar ketika dilakukan *tracking* marker :

1. Mengambil gambar dari kamera (pemerolehan data)

   Mendapatkan masukan gambar dari sebuah kamera *handphone* adalah langkah awal yang harus dilakukan, seperti yang ditunjukan gambar dibawah ini. Sistem mengolah dan menganalisis frame per frame video yang di-streaming secara *real-time* dan hasilnya berupa citra digital yang akan digunakan untuk tahap berikutnya.



Gambar 3.7 Sistem Koordinat Lingkungan AR



Gambar 3.8 Mengambil gambar dari kamera

Pengambilan gambar yang diterima dari pembacaan kamera selanjutnya akan dilakukan proses binarisasi (*grayscale*), kemudian nilai *threshold* (binarisasi citra masukan) ditentukan sehingga gambar tersebut akan menghasilkan gambar hitam putih. Nilai *threshold* berada pada angka 0 - 255 dan secara *default threshold* bernilai 100. Fungsi dari proses ini adalah untuk memudahkan sistem agar dapat mengenali pola pada gambar yang diterima.



Gambar 3.9 Binarisasi Citra Masukan (*threshold*)

Pseudo code untuk algoritma *grayscale*:

```
function grayscale (input panjang_pixel, lebar_pixel : integer) → real
{I.S : nilai matriks gambar belum terdefinisi}
{F.S : menghasilkan nilai matriks gambar yang baru yang sudah diproses grayscale}
Kamus:
 red, green, blue, alpha, i : integer
 gray : real
```

```
Algoritma:
for  i ← 1 to panjang_pixel do
 red ← pixel.r
 green ← pixel.g
 blue ← pixel.b
 alpha ← pixel.a
 gray ← (red + green + blue) / 3;
 pixel.r ← gray
 pixel.g ← gray
 pixel.b ←  gray
endfor
return pixel.r, pixel.g, pixel.b
```

Gambar 3.10 *Pseudo Code* untuk algoritma *grayscale*

Proses ini berfungsi sebagai proses untuk membantu sistem agar dapat mengenali bentuk *feature* dan pola *marker* pada video yang diterima. Nilai *threshold* dapat dirubah dan disesuaikan dengan kondisi cahaya disekitar marker, karena ketika cahaya disekitar marker berkurang (redup) ataupun berlebih (terlalu terang) pada saat proses *thresholding*, sistem akan kesulitan dan tidak dapat mendeteksi *marker*. Hal ini penting karena aplikasi ini bekerja dengan cara mengenali pola pada *marker*. *Pseudo Code* untuk algoritma proses *thresholding* (ambang batas) adalah sebagai berikut:

```
function threshold (input panjang_pixel, lebar_pixel : integer) → integer
{I.S : nilai matriks gambar belum terdefinisi}
{F.S : menghasilkan nilai matriks gambar yang baru yang sudah diproses
threshold}
Kamus:
k, level, Tmean : integer
Algoritma:
Tmean ← 0 /*deklarasi nilai awal t*/
for k ← 0 to level do
for x ← 0 to panjang_pixel-1 do
for y ← 0 to lebar_pixel-1 do
 N ← image[i], [y]
 If  n <= T then
  new_x = 0
  new_y = 0
```

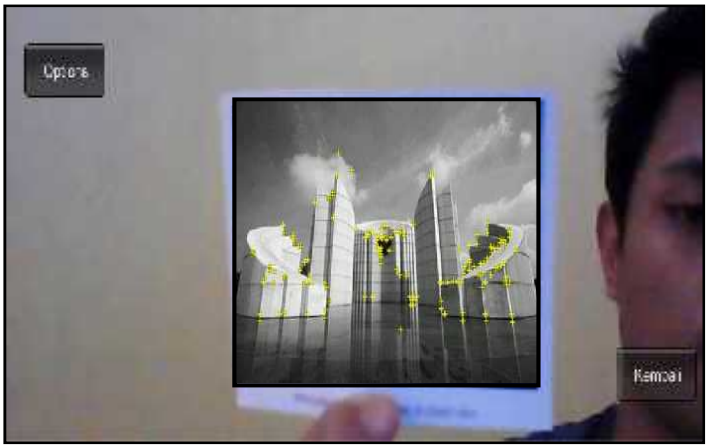
```
  t = k;
 else
  new_x  ← 255
  new_y  ← 255
 end if
endfor
endfor
return panjang_pixel, lebar_pixel
```

Gambar 3.11 *Pseudo Code* untuk algoritma *thresholding*

Setelah dilakukan proses *tresholding* maka dilanjutkan dengan pendeteksian marker, dimana sistem akan mengenali bentuk dan pola yang ada pada marker objek. Sistem akan mencari *feature* pada pola gambar dan menandainya berupa titik-titik keypoint sebagai pengenalan polanya.

2. Mengenali dan mendeteksi pola pada *feature* gambar (ekstraksi ciri)

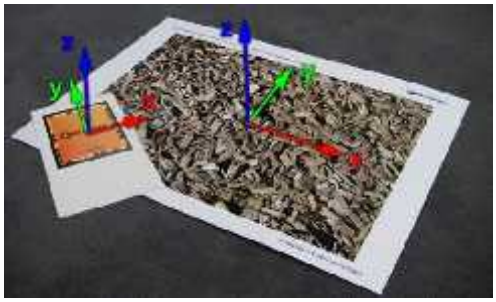
   Setelah dilakukan penangkapan gambar sebagai inputan pengenalan pola pada marker kemudian sistem akan menganalisa citra yang berada pada gambar dan akan mengenali fitur pada sudut-sudut tepi yang dijadikan sebagai titik-titik keypoint untuk dijadikan pengenalan pola. Berikut gambar yang menggambarkan pendeteksian marker dengan melakukan pencarian *feature* dapat dilihat pada Gambar 3.12.



Gambar 3.12 Pendeteksian *feature* pada pola gambar

Deteksi marker (gambar) merupakan tahapan dimana marker akan diidentifikasi oleh sensor kamera sebagai image target untuk penempatan objek yang akan dirender. Pendeteksian marker digunakan metode NFT untuk mencari

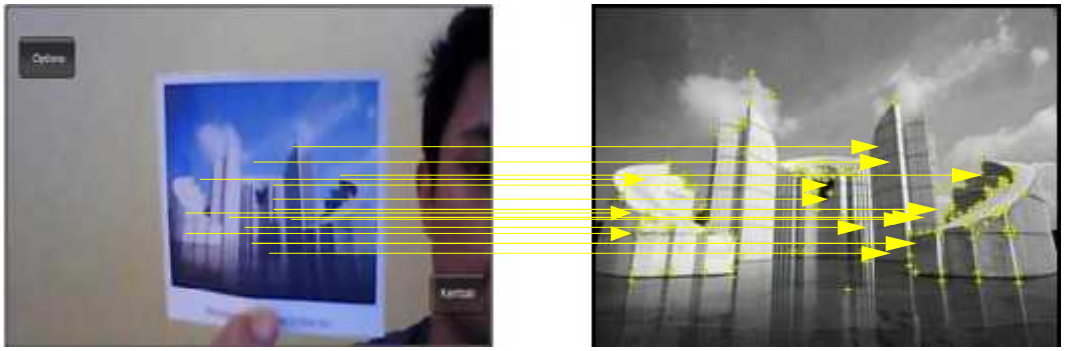
feature pada pola gambar. Sebelum disesuaikan dengan pola yang ada pada library, *marker* yang telah terdeteksi dan koordinat 3D dalam ruang lingkup obyek diberikan dalam (xi, yi, 0) dan koordinat dari sudut *marker* tersebut (Xi,Yi). Koordinat z diatur menjadi 0 karena marker bersifat *planar* / sebidang. Setiap image target mendefinisikan sistem koordinat lokal dengan (0,0,0) sebagai titik pusatnya (tengah). Untuk mendapatkan koordinat 3D dan koordinat 2D dapat dilihat pada Gambar 3.13.



Gambar 3.13 Hasil pelacakan berupa penentuan koordinat

3. Penyesuaian Pola (pengenalan data / klasifikasi)

   Pola marker yang telah terdeteksi berupa titik-titik *keypoint* akan disesuaikan dengan pola yang ada pada sistem dengan cara menyesuaikan *feature* dan penempatan titik *keypoint* pada pola gambar.



Gambar 3.14 Penyesuaian pola pada *feature* gambar

Dalam metode ini informasi yang diperlukan untuk tujuan pelacakan dapat diperoleh dengan cara *optical-flow* berbasis korenspodensi fitur. Dimana akan dicocokan dengan marker yang berada di-*library* dari hasil pengolahan marker yang telah di-*generate* dari target management sistem milik vuforia.

3.1.4.3 Rendering Text dan Video

Proses rendering objek dilakukan setelah melalui proses *tracking* dan penyesuaian pola yang nantinya akan menampilkan objek berupa teks, dan video.

1. Rendering Text, akan menampilkan hasil objek informasi berupa teks melalui proses inisialisasi dan *tracking* dimana pada saat mendeteksi pola marker dan sesuai dengan pola yang ada pada sistem akan memunculkan informasi berupa teks. Teks yang akan ditampilkan berupa tulisan deskripsi singkat mengenai sejarah monumen tersebut. Berikut tampilan informasi dari hasil objek teks.



Gambar 3.15 Tampilan hasil objek text

2. Rendering Video, akan menampilkan hasil objek informasi berupa video. Proses ini tidak berbeda jauh dengan render text namun bedanya hanya memasukan *video.package* kedalam sistem setelah itu dilakukan proses *tracking* dan pencocokan pola yang nantinya akan menampilkan video sejarah monumen.



Gambar 3.16 Tampilan Objek Video

3.1.5 Analisis Pelacakan AR

Natural Feature Tracking (NFT) yang terdapat pada *library vuforia* berfungsi untuk menentukan apakah suatu gambar dapat dikenali atau tidak berdasarkan pengenalan pola dengan mendeteksi titik-titik sudut pola pada gambar. Tujuannya adalah agar pendeteksian *feature* pola lebih mudah. Pada dasarnya *library Vuforia* menggunakan NFT (*Natural Feature Tracking)* yang didalamnya terdapat pendekatan pelacakan fitur alami yaitu SIFT *(Scale Invariant Feature Transform).* SIFT digunakan untuk mendeteksi "*feature point*" dan menentukan skala dari marker (*image target*) dengan memetakan nilai koordinatnya.

Setelah melalui tahapan-tahapan tersebut, maka pada hasil akhir akan diperoleh suatu citra dengan *feature point* yang invarian terhadap berbagai macam perubahan, seperti yang ditunjukan pada gambar dibawah ini :

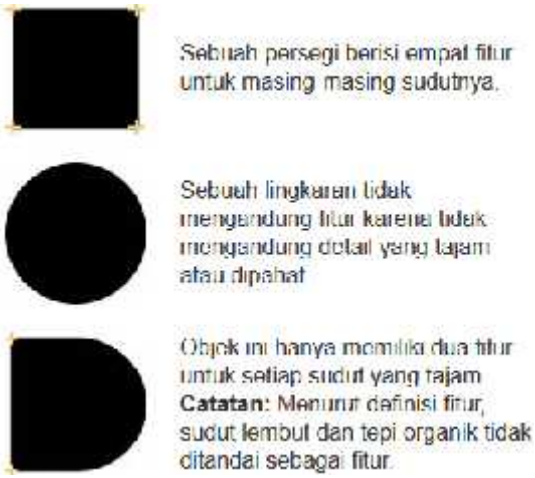


Gambar 3.17 (a) Gambar Awal



Gambar 3.18 (b) Hasil NFT

Sebuah *rating augmentable* pada suatu target gambar akan terlihat semakin kuat kemampuan deteksi dan pelacakan yang dikandungnya dengan menunjukan rating pola tersebut berkisar antara 0 hingga 5 bintang. Sebuah rating dari nol menunjukkan bahwa target tidak dilacak sama sekali oleh sistem *Augmented Reality*, sedangkan rating bintang 5 menunjukkan bahwa sebuah gambar dengan mudah dilacak oleh sistem Augmented Reality.



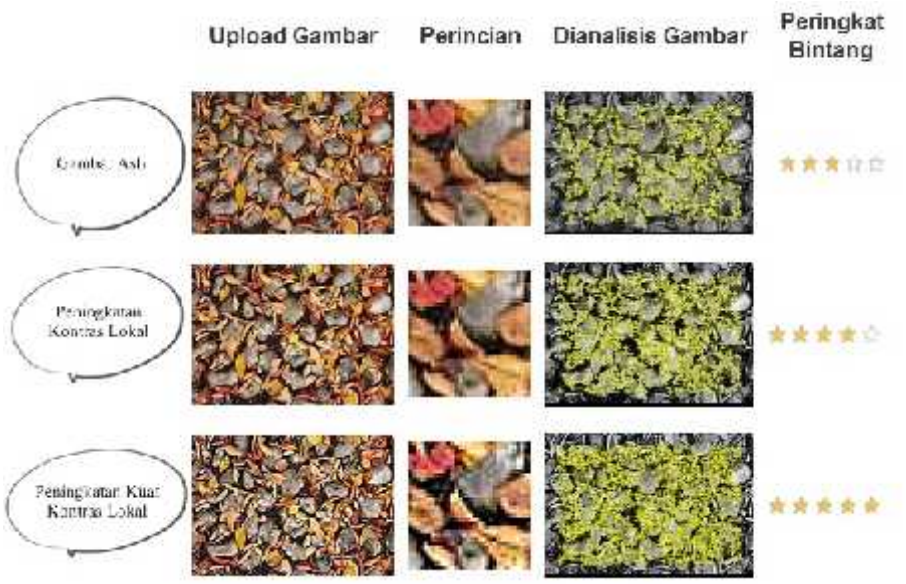
Gambar 3.19 Contoh bentuk objek gambar yang mengandung fitur [21]

Baik atau buruknya kontras dapat mempengaruhi terdeteksi fitur dengan meningkatkan kontras gambar secara umum atau memilih gambar dengan detail rincian bulat, kabur dan gambar yang dikompresi berlebihan maka tidak akan memberikan kekayaan yang cukup rinci untuk dideteksi dan dilacak dengan benar.



Gambar 3.20 Contoh *Rating Augmentable* di Vuforia[21]

Sebuah Fitur adalah tajam, berduri dan seringkali berupa sudut seperti yang hadir dalam benda berstruktur. Di bawah ini merupakan contoh struktur gambar yang memiliki fitur.



Gambar 3.21 Contoh Perubahan Kontras Terhadap Perubahan Nilai Deteksi[21]

Gambar diatas merupakan contoh yang lebih praktis mengenai bagaimana meningkatkan kontras lokal target. Latar belakang adalah permukaan bertekstur. Lapisan hanya ada dalam editor grafis kami ketika meng-upload pada Target manajer yang selalu menggunakan gambar pipih, misalnya, format PNG. Meskipun beberapa gambar mengandung cukup fitur dan kontras yang baik, pola berulang menghambat kinerja deteksi. Untuk hasil terbaik, pilih gambar tanpa motif berulang-ulang (bahkan jika diputar dan bersisik) atau simetri rotasi yang kuat. Sebuah kotak-kotak adalah contoh dari pola berulang yang tidak dapat terdeteksi, karena pola 2x2 kotak hitam dan putih tampak persis sama sehingga tidak dapat dibedakan dengan detektor.



Gambar 3.22 Contoh Gambar yang memliki pola berulang[21]

Adapun komponen-komponen dalam proses pelacakan adalah sebagai berikut :

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

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

   Dalam pelacakan memiliki informasi status yang terkait dengannya, setiap informasi satus dilacak akan diperbaharui sebagai frame kamera diproses. Status yang berbeda dilacak dapat menjadi salah satu dibawah ini :
   a. UNKNOW : tempat atau lokasi pelacakan tidak diketahui biasanya dikembalikan sebelum tracker initialization
   b. UNDEFINE : tempat atau lokasi pelacakan tidak didefinisikan
   c. NOT_FOUND : tempat atau lokasi tidak ditemukan pada database yang dituju
   d. DETECTED : tempat atau lokasi pelacakan diseteksi dalam frame
   e. TRACKED : pelacakan telah terlacak dalam frame

3.1.5.1 Deskripsi Fungsi pada Vuforia

Adapun beberapa anggota fungsi pada vuforia [24], yaitu sebagai berikut:

1. *Class CameraDevice*

CameraDevice

Fungsi class ini untuk menyediakan akses ke kamera dan properti.

2. *Class QCARBehaviour*

QCARBehaviour

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

3. *Class DataSetLoadBehaviour*

DataSetLoadBehaviour

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

4. *TrackableBehaviour*

`TrackableBehaviour.Trackable [get]`

Pelacakan pada saat runtime.

5. *RegisterTrackableEventHandler*

TrackableBehaviour.RegisterTrackableEventHandler

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

6. *UnregisterTrackableEventHandler*

TrackableBehaviour.UnregisterTrackableEventHandler

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

7. *Class QCARAbstractBehaviour*

`Status` **TrackableBehaviour.CurrentStatus [get]**

Class yang berisikan method-method yang menangani pelacakan.

8. *Class QCARBehaviour*

`Status` **TrackableBehaviour.CurrentStatus [get]**

Status pelacakan dari `TrackableBehaviour.`

9. *TrackableName*

String TrackableBehaviour.TrackableName [get]

Pelacakan nama yang ada pada `TrackableBehaviour.`

10. *OnTrackerUpdate*

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

Dipicu oleh `TrackerBehaviour` setelah itu diperbarui.

11. *ITrackableEventHandler*

```
ITrackableEventHandler
```

Fungsi ini berisikan antarmuka untuk menangani perubahan yang dilacak

12. *OnTrackableStateChanged*

```
DevaultTrackableEventHandler.OnTrackableStateChanged
```

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

13. *ImageTargetAbstractBehaviour*

```
ImageTargetAbstractBehaviour
```

Class ini berfungsi sebagai definisi sasaran image target ketika dilacak pada saat *runtime*

14. *ImageTarget*

```
ImageTarget
```

Fungsi untuk berisikan prilaku pelacakan untuk mewakili *natural feature* target

3.1.6 Spesifikasi Kebutuhan Perangkat Lunak

Pada tahap ini dijelaskan 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 kebutuhan funsional berisi mengenai fungsi atau layanan apa yang harus disediakan atau 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 mendeteksi marker
SKPL-F-002	Sistem harus dapat menampilkan informasi teks, suara dan video
SKPL-F-003	Sistem harus dapat menampilkan menu utama
SKPL-F-004	Sistem harus dapat menampilkan menu cara penggunaan
SKPL-F-005	Sistem harus dapat menampilkan menu penyajian AR
SKPL-F-006	Sistem harus dapat menampilkan menu tentang aplikasi

2. Spesifikasi Kebutuhan Non-Fungsional

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

Tabel 3.2 Spesifikasi Kebutuhan Non-Fungsional

Nomor	Spesifikasi Kebutuhan 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 memerlukan sudut yang cukup untuk membaca marker dari arah kamera
SKPL-NF-004	Sistem memerlukan jarak yang cukup namun harus mencangkup kamera ketika akan mendeteksi marker

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 untuk menjalankan aplikasi ini meliputi kebutuhan perangkat keras, kebutuhan perangkat lunak dan pengguna sistem yang akan memakai aplikasi. Analisis kebutuhan non-fungsional bertujuan agar aplikasi yang dibangun dapat digunakan sesuai dengan kebutuhan pengguna aplikasi dalam mencari informasi yang dibutuhkan.

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, aplikasi yang akan dibuat tidak akan dapat berjalan. Berikut spesifikasi standar perangkar keras yang dapat dipergunakan untuk membangun aplikasi media pengenalan monumen bersejarah. Spesifikasi perangkat keras yang digunakan dalam pembangunan aplikasi menggunakan teknologi *augmented reality* berbasis android adalah :

Tabel 3.3 Kebutuhan Perangkat Keras Pengembang

No	Perangkat Keras	Spesifikasi
1	*Processor*	Minimum Intel Core Duo 2.27 GHz
2	Monitor	14 inch
3	*Graphic Card* (VGA)	2 GB
4	Memori	4 GB
5	*Webcam*	3 MP
6	*Harddisk drive*	Free Space 500 MB
7	*Keyboard* dan *Mouse*	Standar
8	*Speaker*	Standar

Sedangkan untuk perangkat mobile android dapat dilihat pada Tabel 3.4.

Tabel 3.4 Spesifikasi Kebutuhan Perangkat Mobile Android

No	Perangkat Keras	Spesfikasi
1	Layar	4,6'(inch)
2	RAM	1 GB
3	Prosessor	ARmV7
4	Kamera	3,2 MP

3.1.7.2 Analisis Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan guna mendukung kinerja sebuah sistem dengan melakukan perintah-perintah yang diberikan kepada perangkat keras agar dapat saling berinteraksi antara keduanya.

Berikut spesifikasi perangkat lunak yang digunakan dalam pembangunan aplikasi ini adalah :

Tabel 3.5 Kebutuhan Perangkat Lunak Pengembang

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Windows 7
2	Unity 3.4.3	Tools pengembang sebagai editor dalam pembuatan aplikasi
2	*Vuforia* SDK (QCAR)	*Library AR* yang digunakan sebagai *tools* pendukung pembuatan aplikasi
3	Tool Desain	Adobe Photoshop CS5
4	Android SDK	Tools Pengembangan Android

3.1.7.3 Analisis Pengguna

Aplikasi ini akan digunakan hanya untuk pengunjung yang datang dan ingin mengetahui secara langsung informasi pada monumen bersejarah tersebut. Berikut kebutuhan pengguna yang terlibat dalam penggunaan aplikasi media pengenalan monumen bersejarah kota Bandung. Adapun spesifikasi kebutuhan pengguna yang akan dijelaskan pada tabel 3.6.

Tabel 3.6 Kebutuhan Pengguna

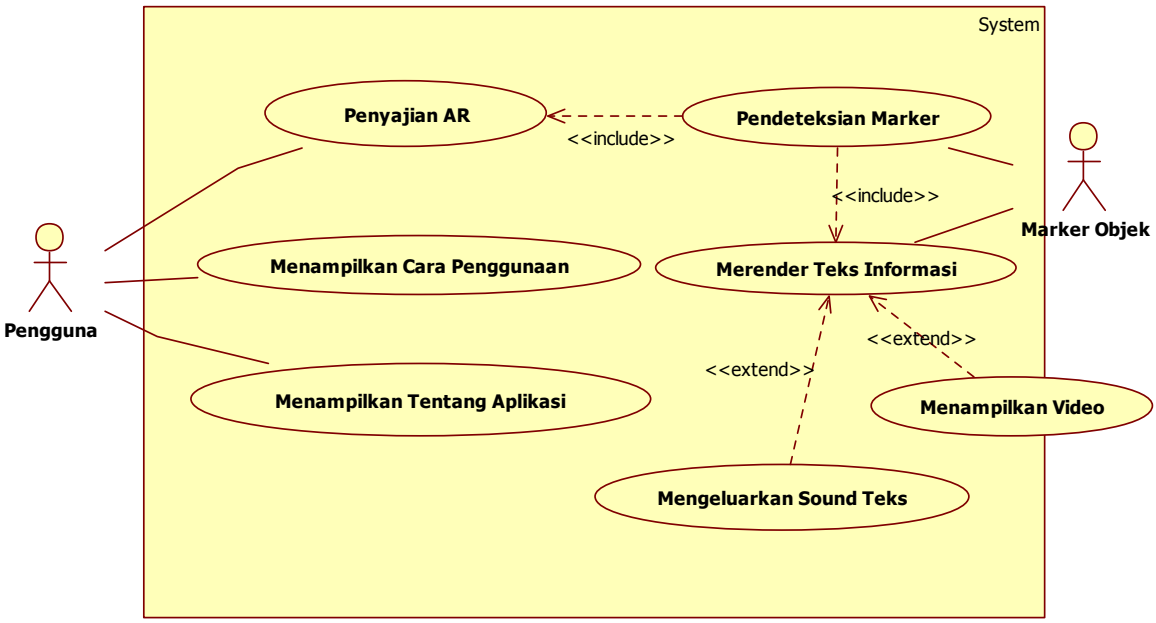
No	Kebutuhan Penguna
1	Hak akses pengguna adalah menggunakan aplikasi
2	Mengerti dan paham cara menggunakan telepon seluler Android
3	Pengguna tidak memiliki kelainan pada penglihatan mata (buta warna)
4	Minimal menggunakan telepon seluler (minimal OS android 2.2-Froyo)

3.1.8 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan kebutuhan sistem yang akan dibangun pada aplikasi media pengenalan monumen bersejarah menggunakan teknologi *augmented reality*. 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*.

3.1.8.1 Use Case Diagram

Use Case merupakan gambaran skenario dari interaksi antara *user* dengan sistem. Sebuah diagram *Use Case* yang menggambarkan hubungan antara aktor dan kegiatan yang dapat dilakukannya terhadap aplikasi. Didalam sistem terdapat pengguna yaitu pemakai aplikasi. Peran aktor yang ada dapat terlihat pada diagram *Use Case* pada gambar dibawah ini:



Gambar 3.23 *Use Case Diagram*

3.1.8.1.1 Definisi *Actor*

Definsi *Actor* berfungsi untuk menjelaskan actor yang terdapat pada *Use Case Diagram.* Definsi *Actor* akan dijelaskan pada tabel 3.7 dibawah ini:

Tabel 3.7 Definisi Actor

No.	*Actor*	Deskripsi
1	Pengguna (*User*)	Orang yang menggunakan aplikasi pada sistem, yaitu pengguna yang mengakses semua fungsi yang disediakan sistem.
2	Marker Objek	Pola atau bentuk yang akan dikenali kamera telepon akan dijadikan sebagai penanda atau inputan dimana informasi dari objek tersebut akan ditampilkan.

3.1.8.2 Definisi Use Case

Definisi *Use Case* berfungsi untuk menjelaskan fungsi *Use Case* yang terdapat pada *Use Case Diagram*. Definisi *Use Case* diterangkan pada Tabel 3.8 dibawah ini.

Tabel 3.8 Definisi *Use Case*

NO	*Use Case*	Deskripsi
UC-1	Menampilkan Cara Penggunaan	Fungsionalitas untuk menampilkan info cara menggunakan
UC-2	Penyajian AR	Fungsionalitas untuk memulai aplikasi dalam menyajikan AR
UC-3	Menampilkan Tentang Aplikasi	Fungsionalitas untuk menampilkan informasi tentang aplikasi ARMonumen
UC-4	Pendeteksian Marker	Fungsionalitas mendeteksi ketersediaan kamera dan mendeteksi marker
UC-5	Merender Teks Informasi	Fungsionalitas untuk menampilkan informasi berupa teks
UC-6	Mengeluarkan SoundTeks	Fungsionalitas untuk mengeluarkan suara terkait informasi monumen
UC-7	Menampilkan Video	Fungsionalitas untuk menampilkan video

3.1.8.3 Skenario Use Case

Skenario *Use Case* mendeskripsikan urutan langkah-langkah dalam proses kinerja, baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor. Adapun skenario usecase dari AR Monumen ini adalah sebagai berikut:

1. Skenario Use Case Penyajian AR

   Use case memilih menu ini menggambarkan proses dimana Penyajian AR pada aplikasi dapat dilihat pada tabel 3.9.

Tabel 3.9 Skenario Use Case Penyajian AR

Nama Use Case	**Penyajian AR**	
Persyaratan Terkait	-	
Tujuan Dalam Konteks	Sistem dapat menampilkan informasi	
Kondisi Sebelum	Kamera belum terdeteksi	
Keberhasilan Kondisi Akhir	Sistem dapat mengenali marker dan dapat menampilkan informasi	
Kegagalan Kondisi Akhir	Sistem tidak dapat menginisialisasi marker	
Aktor Utama	Pengguna (*user*)	
Aktor Kedua	Marker Objek	
Include	Pendeteksian marker	
Extend	-	
Pemicu	*User* memilih menu penyajian AR Monumen ini	
Aliran Utama	**Langkah**	**Aksi**
	1	*User* memilih menu penyajian AR
	2	*User* menunjukkan marker
	3 include::mendeteksi marker	Sistem mendeteksi marker
	4	Sistem merender objek teks informasi
	5	Sistem menampilkan informasi berupa teks
Perluasan	**Langkah**	**Aksi Percabangan**
	4.1	Sistem tidak dapat mengenali marker yang terdeteksi

	5.1	Sistem tidak dapat merender objek sesuai dengan marker

2. Skenario Use Case Menampilkan Cara Penggunaan

Use case memilih menu ini menggambarkan proses dimana menampilkan cara penggunaan pada aplikasi dapat dilihat pada tabel 3.10

Tabel 3.10 Skenario Menampilkan Cara Penggunaan

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

3. Skenario Use Case Menampilkan Tentang Aplikasi

Use case ini menggambarkan proses dimana menampilkan tentang aplikasi monumen dapat dilihat pada tabel 3.11.

Tabel 3.11 Skenario Use Case Menampilkan Tentang Aplikasi

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

4. Skenario Use Case Pendeteksian Marker

Skenario use case pendeteksian marker dapat dilihat pada Tabel 3.12.

Tabel 3.12 Skenario Use Case 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	Pengguna (*User)*
Aktor Kedua	Marker Objek
Include	-

Pemicu	Kamera dalam kondisi aktif dan 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* dan binarisasi citra masukan
	5	Sistem dapat melakukan proses pendeteksian fitur
	6	Sistem mampu mendeteksi marker sesuai dengan marker yang ada pada *library*
Perluasan	**Langkah**	**Aksi Percabangan**
	1.1	Sistem tidak dapat mendeteksi kamera
	6.1	Sistem tidak dapat mengidentifikasi marker

5. Skenario Use Case Merender Teks Informasi

   Skenario use case merender objek teks dapat dilihat pada Tabel 3.13.

Tabel 3.13 Skenario Use Case Merender Teks Informasi

Nama Use Case	**Merender Teks Informasi**	
Persyaratan Terkait	-	
Tujuan Dalam Konteks	Sistem menampilkan informasi berupa objek teks	
Kondisi Sebelum	Marker yang terdeteksi belum dapat dikenali	
Keberhasilan Kondisi Akhir	Sistem menampilkan objek teks sesuai marker yang dideteksi dan muncul dilayar telepon	
Kegagalan Kondisi Akhir	Sistem tidak dapat menampilkan informasi berupa objek teks informasi	
Aktor Utama	Pengguna (*User)*	
Aktor Kedua	Marker Objek	
Include	Merender teks informasi, menampilkan informasi berupa teks	
Pemicu	Marker dapat dikenali dengan baik dan sesuai dengan pola marker yang ada	
Aliran Utama	**Langkah**	**Aksi**
	1	*User* mengarahkan kamera pada marker

	2	Kamera membaca marker
	3	Sistem mengenali marker
	4	Sistem merender objek
	5	Objek yang ditampilkan berupa informasi teks
Perluasan	**Langkah**	**Aksi Percabangan**
	3.1	Sistem tidak dapat menemukan pola marker yang sesuai
	3.2	Sistem tidak merender objek teks
	3.3	Sistem tidak menampilkan informasi

6. Skenario Use Case Mengeluarkan *SoundText*

Skenario use case mengeluarkan *soundtext* dapat dilihat pada Tabel 3.14.

Tabel 3.14 Skenario Use Case Mengeluarkan SoundText

Nama Use Case	**Mengeluarkan SoundText**	
Persyaratan Terkait	-	
Tujuan Dalam Konteks	Sistem mengeluarkan suara terkait sejarah monumen	
Kondisi Sebelum	Objek teks informasi sudah ditampilkan	
Keberhasilan Kondisi Akhir	Sistem dapat mengeluarkan suara ketika menekan tombol audio saat muncul informasi berupa teks pada layar telepon	
Kegagalan Kondisi Akhir	Sistem tidak dapat mengeluarkan suara	
Aktor Utama	Pengguna *(User)*	
Aktor Kedua	Marker Objek	
Pemicu	Objek teks informasi sudah dapat ditampilkan dan pengguna menekan tombol *audio play*	
Extend	Merender *soundtext*, menampilkan informasi berupa suara	
Aliran Utama	**Langkah**	**Aksi**
	1	*User* menekan tombol audio play

	2	Sistem mengeluarkan suara terkait informasi monumen tersebut
Perluasan	**Langkah**	**Aksi Percabangan**
	3.1	Sistem tidak dapat menemukan pola marker yang sesuai

7. Skenario Use Case Menampilkan Video

   Skenario use case menampilkan video dapat dilihat pada Tabel 3.15.

Tabel 3.15 Skenario Use Case Menampilkan Video

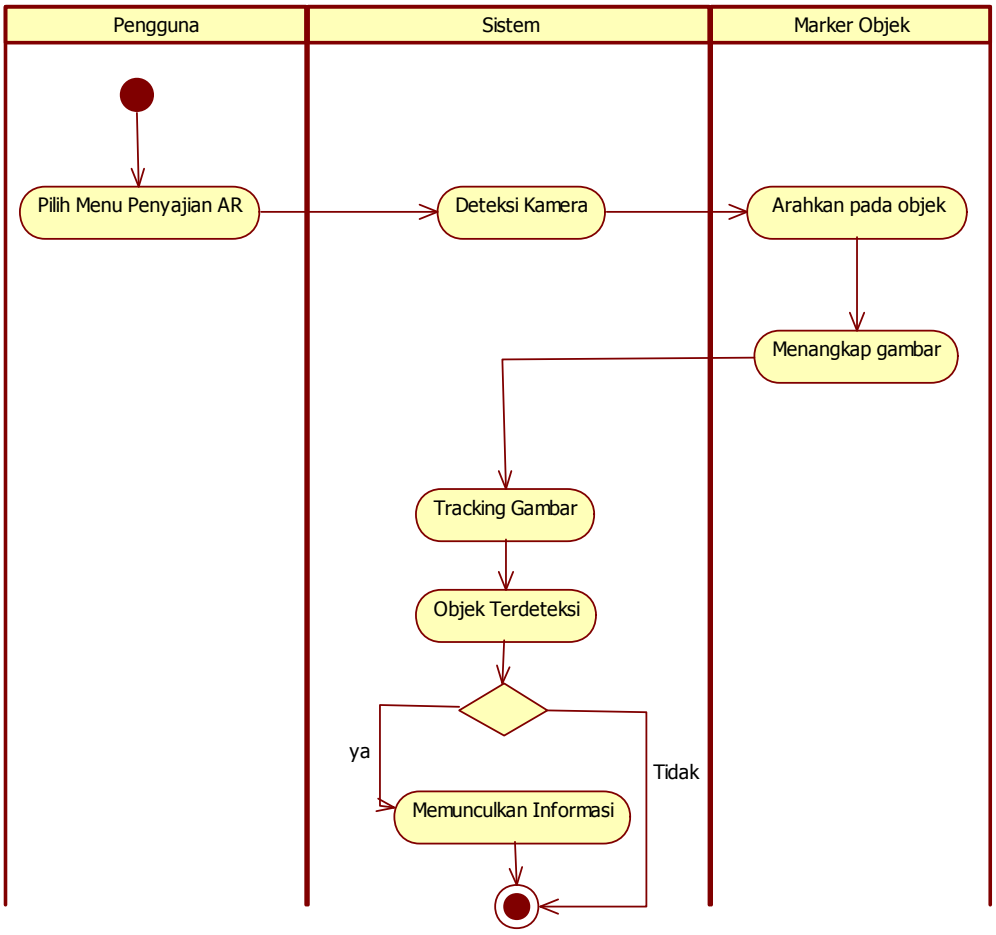
Nama Use Case	**Menampilkan Video**	
Persyaratan Terkait	-	
Tujuan Dalam Konteks	Sistem menampilkan video sejarah monumen	
Kondisi Sebelum	Objek teks informasi sudah ditampilkan	
Keberhasilan Kondisi Akhir	Sistem dapat mengeluarkan suara ketika menekan tombol *videoplay* saat muncul informasi berupa teks pada layar telepon	
Kegagalan Kondisi Akhir	Sistem tidak dapat menampilkan video	
Aktor Utama	Pengguna *(User)*	
Aktor Kedua	Marker Objek	
Pemicu	Objek teks informasi sudah dapat ditampilkan dan pengguna menekan tombol *videoplay*	
Extend	Merender *videoplay*, menampilkan informasi berupa video	
Aliran Utama	**Langkah**	**Aksi**
	1	*User* menekan tombol *video play*
	2	Sistem mengeluarkan suara terkait informasi monumen tersebut
Perluasan	**Langkah**	**Aksi Percabangan**
	3.1	Sistem tidak dapat menemukan pola marker yang sesuai

3.1.8.4 Activity Diagram

Diagram activity menggambarkan berbagai alir aktivitas dalam sistem yang dirancang, bagaimana masing-masing aliran data dimulai dan berakhir. Penggambaran *activity diagram* memiliki kemiripan dengan *flowchart* atau *Data Flow Diagram* pada perancangan tersruktur. Berikut ini merupakan *activity diagram* pada sistem yang akan dibangun.

3.1.8.4.1 *Activity Diagram* Penyajian AR

Activity Diagram terkait skenario penyajian AR dengan alur proses yang akan digambarkan sebagai berikut:

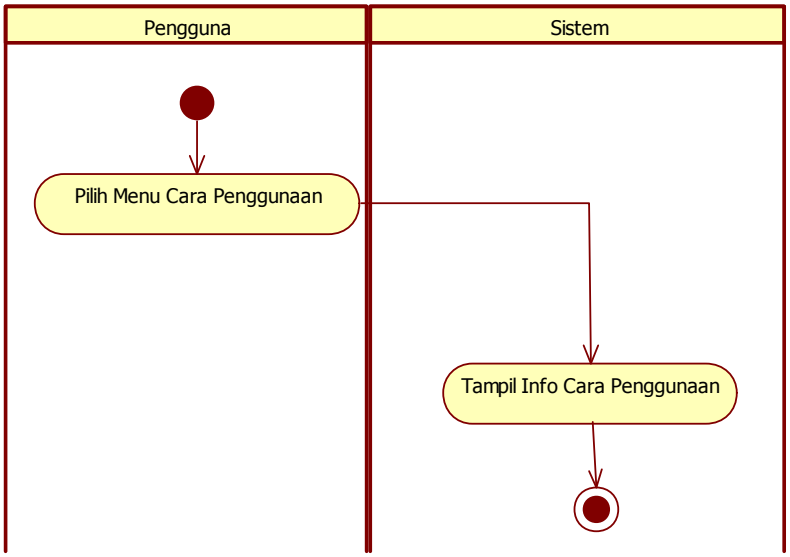


Gambar 3.24 *Activity Diagram* Penyajian AR

Pada gambar diatas menggambarkan aktivitas ketika pengguna memilih penyajian AR. Pengguna akan menekan tombol penyajian AR, lalu akan mendeteksi kamera yang sudah dalam keadaan aktif. Selanjutnya arahkan kamera ke objek yang akan dijadikan marker lalu sistem akan menangkap gambar, setelah itu akan dilakukan *tracking marker* apakah pola objek terdeteksi atau tidak, apabila terdeteksi akan memunculkan informasi pada layar telepon.

3.1.8.4.2 *Activity Diagram* Menampilkan Cara Penggunaan

Activity Diagram terkait skenario menampilkan cara penggunaan memiliki alur proses seperti digambarkan sebagai berikut :

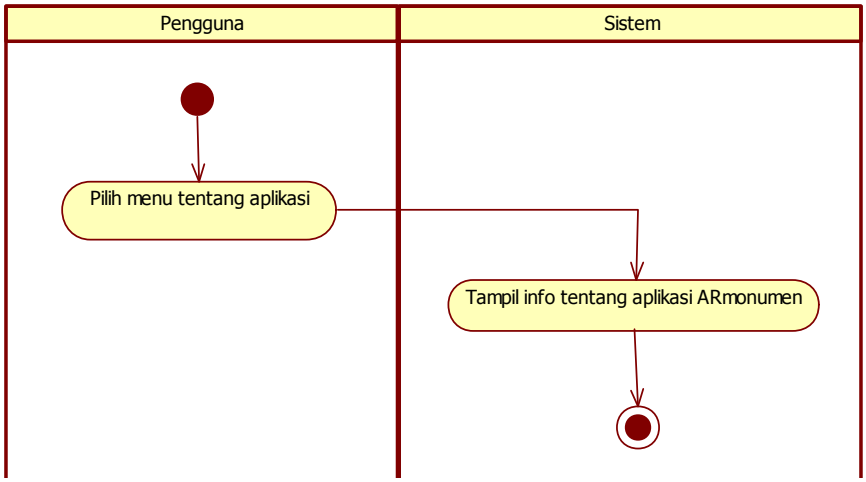


Gambar 3.25 *Activity Diagram* Menampilkan Cara Penggunaan

Pada gambar diatas menggambarkan aktivitas ketika pengguna memilih menu cara penggunaan. Ketika pengguna menekan tombol menu ini maka sistem akan menampilkan menu cara penggunaan yang berisikan informasi singkat mengenai teknolologi *augmented reality* dan cara menggunakan aplikasi.

3.1.8.4.3 *Activity Diagram* Menampilkan Tentang Aplikasi

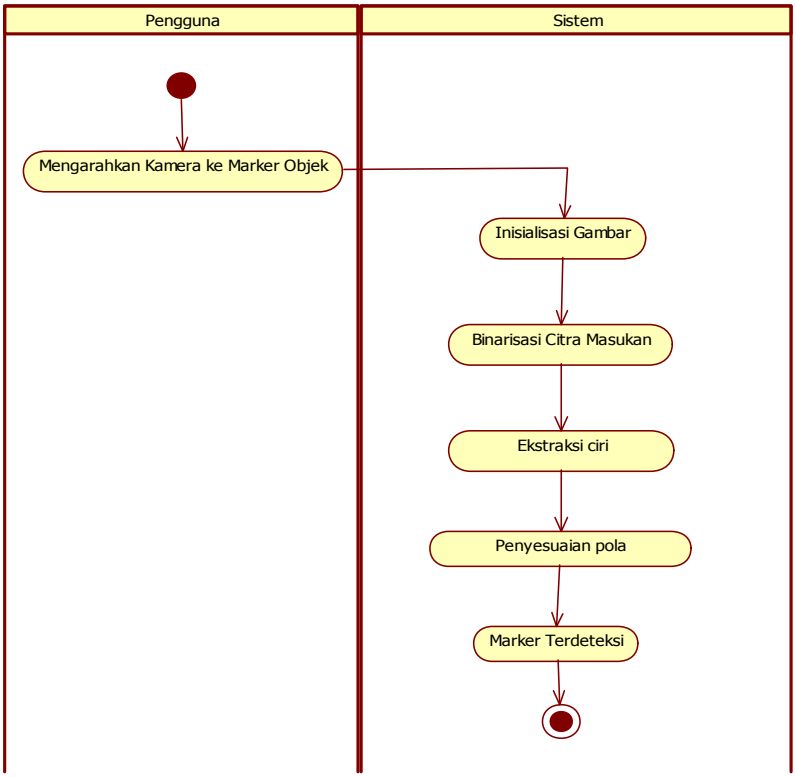
Activity Diagram terkait skenario Menampilkan Tentang Aplikasi memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3.26 *Activity Diagram* Menampilkan Tentang Aplikasi

3.1.8.4.4 *Activity Diagram* Pendeteksian Marker

Activity Diagram terkait skenario Pendeteksian Marker memiliki alur proses seperti yang digambarkan dibawah ini:

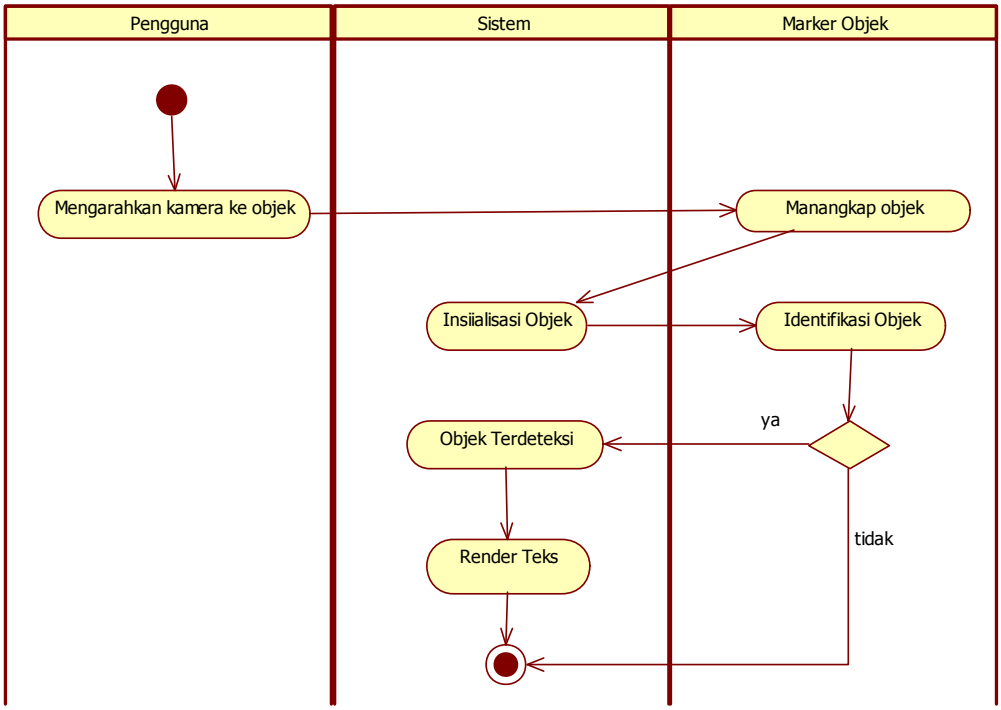


Gambar 3.27 *Activity Diagram* Pendeteksian Marker

Pada gambar diatas menggambarkan aktivitas ketika sistem sedang melakukan proses *tracking marker*. Ketika pengguna mengarahkan kamera ke objek marker dan melakukan penangkapan gambar maka sistem akan melakukan inisialisasi gambar diikuti dengan binarisasi citra masukan, ekstraksi ciri, penyesuaian pola gambar. Tahapan berikut merupakan proses *tracking* marker dimana pola akan dikenali lalu akan menampilkan informasi pada layar telepon.

3.1.8.4.5 *Activity Diagram* Merender Text Informasi

Activity Diagram terkait skenario merender teks informasi memiliki alur proses yang digambarkan dibawah ini:

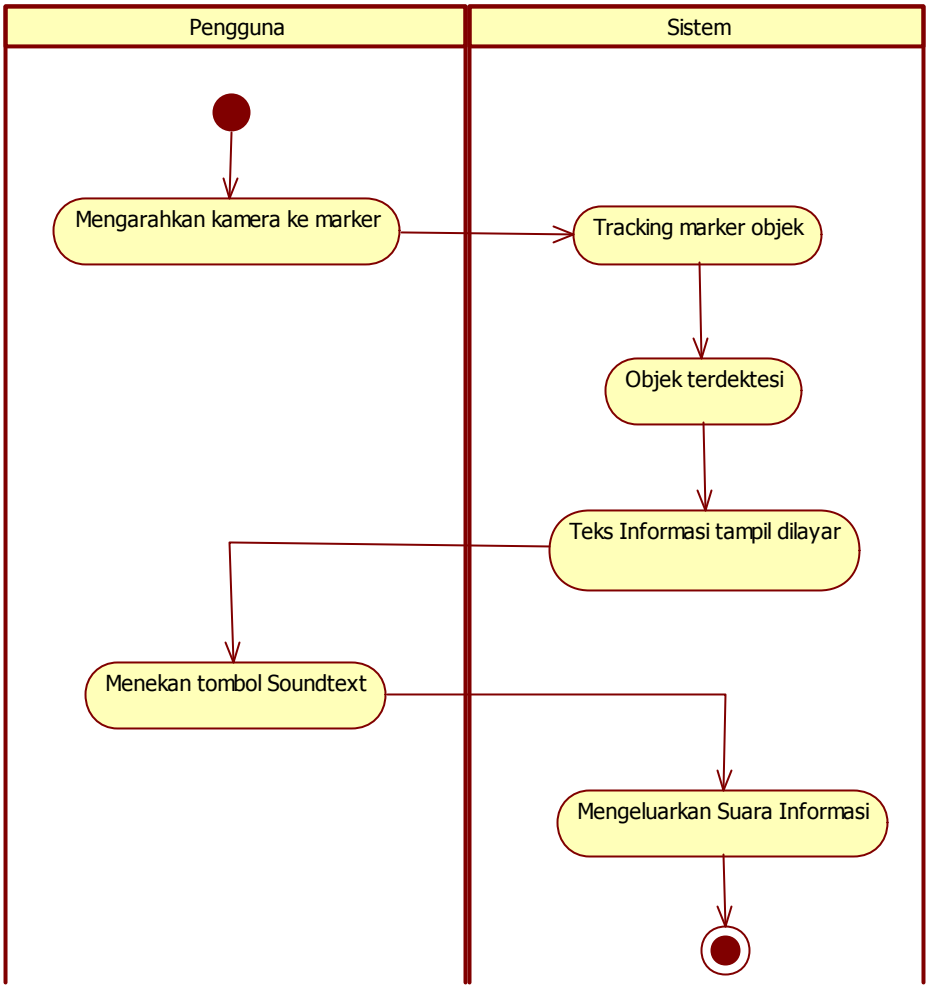


Gambar 3.28 *Activity Diagram* Merender Text informasi

Pada gambar diatas merupakan gambaran aktivitas ketika melakukan render text. Ketika kamera diarahkan dan menangkap gambar maka akan dilakukan proses *tracking* dengan inisialisasi selanjutnya dilakukan identifikasi objek terhadap pola gambar. Setelah itu dilakukan pendeteksian jika terdeteksi maka akan muncul atau menampilkan informasi berupa text.

3.1.8.4.6 *Activity Diagram* Mengeluarkan SoundText

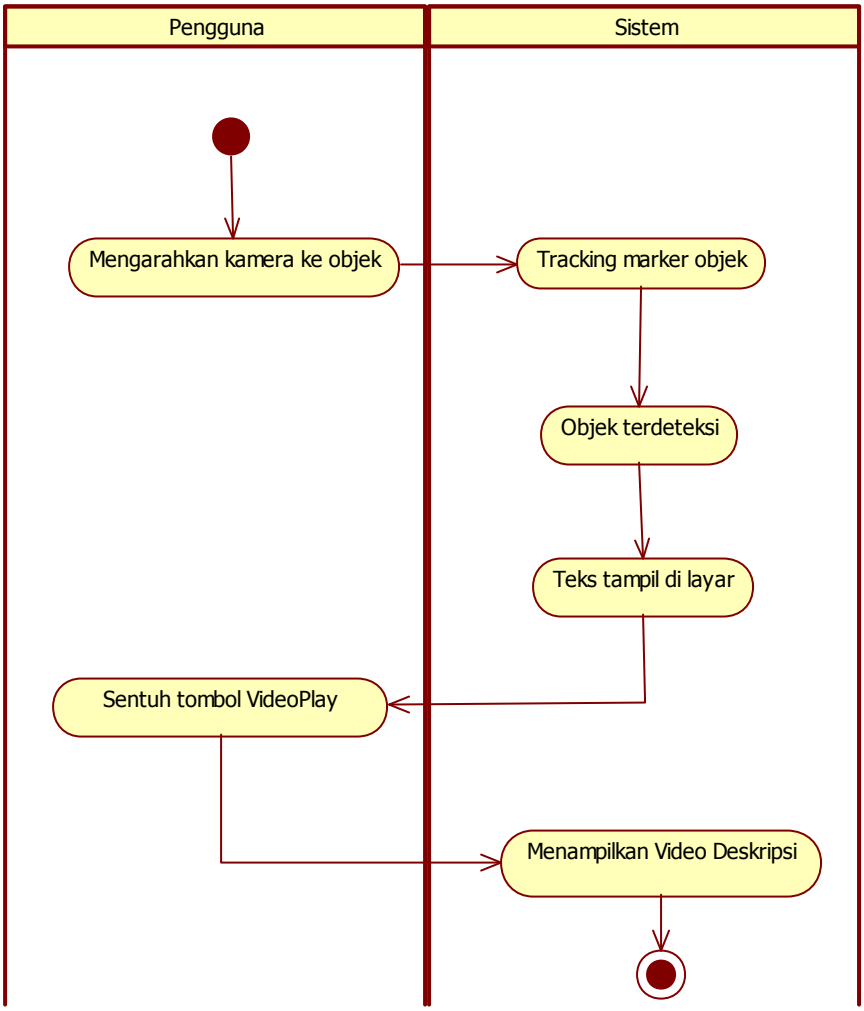
Activity Diagram terkait skenario mengeluarakan *soundtext* memiliki alur proses yang digambarkan dibawah ini:



Gambar 3.29 *Activity Diagram* Mengeluarkan *SoundText*

3.1.8.4.7 *Activity Diagram* Menampilkan Video

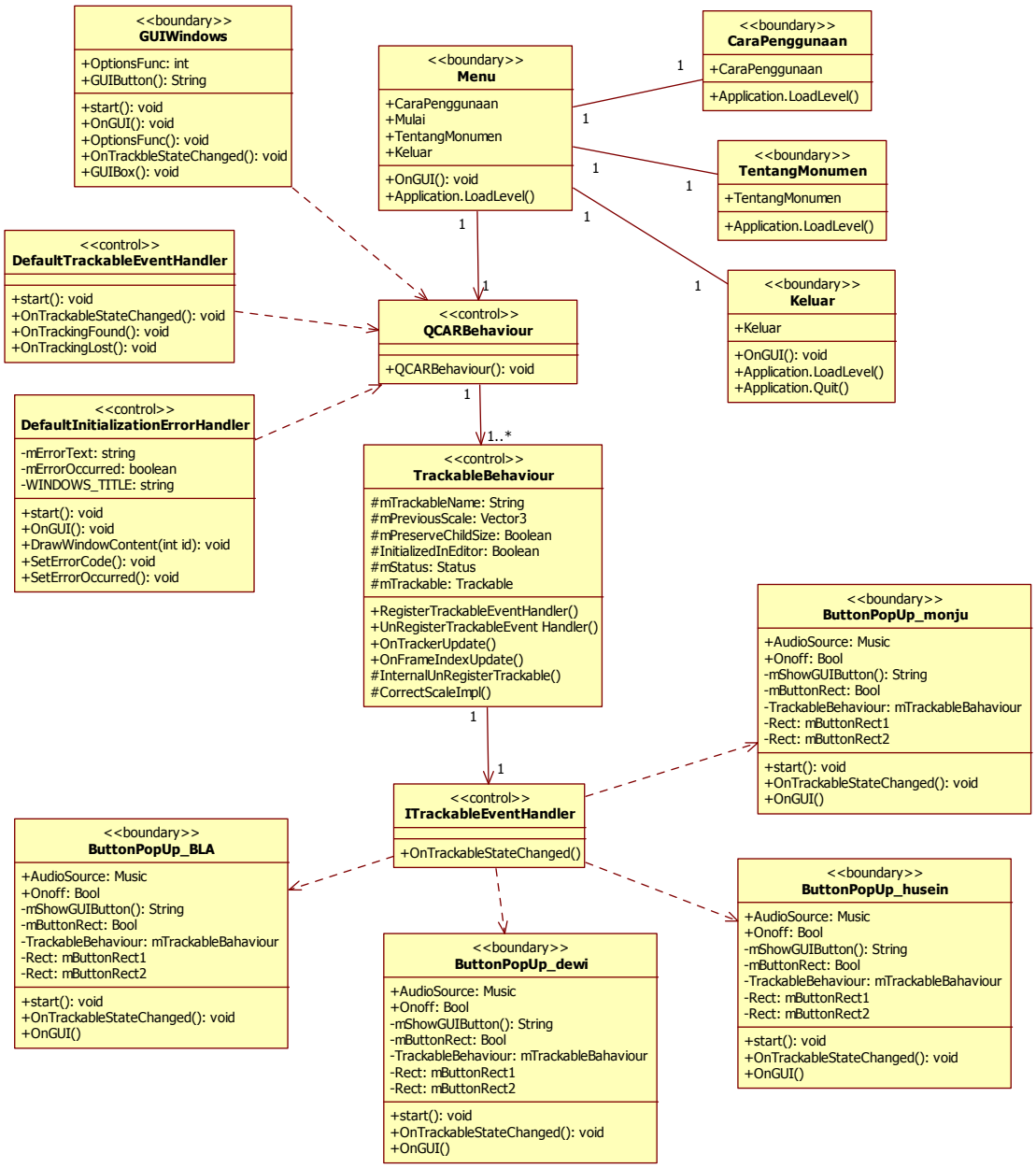
Activity Diagram terkait skenario menampilkan Video memiliki alur proses yang digambarkan dibawah ini:



Gambar 3.30 *Activity Diagram* Menampilkan Video

3.1.8.5 Class Diagram

Class diagram merupakan 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 pengenalan bangunan monumen bersejarah :



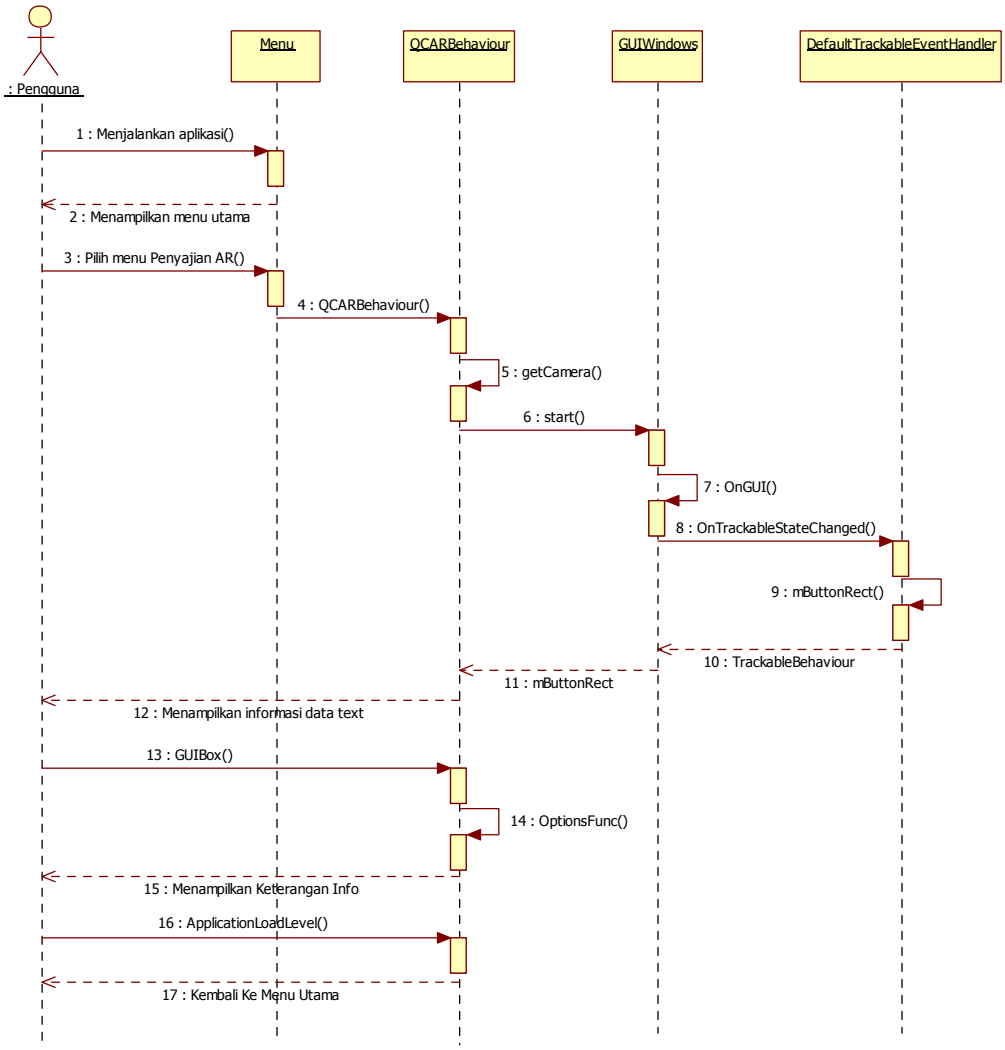
Gambar 3.31 *Class Diagram*

3.1.8.6 *Sequence Diagram*

Sequence diagram menjelaskan gambaran interaksi antar objek dalam urutan waktu. Interaksi ini berupa pengiriman serangkaian data antar objek-objek yang saling berinteraksi. *Sequence diagram* pada aplikasi ini terdiri dari cara penggunaan, penyajian AR, tentang aplikasi, pendeteksian marker, merender text informasi, mengeluarkan *soundtext* dan menampilkan video deskripsi.

1. ***Sequence Diagram* Penyajian AR**

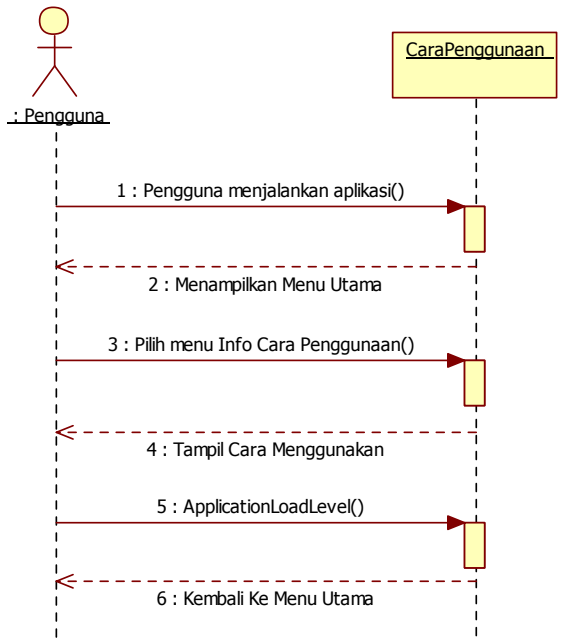
Berikut alur *sequence diagram* pada menu mulai dalam aplikasi ini yang akan digambarkan sebagai berikut:



Gambar 3.32 *Sequence Diagram* Penyajian AR

2. *Sequence Diagram* Menampilkan Cara Penggunaan

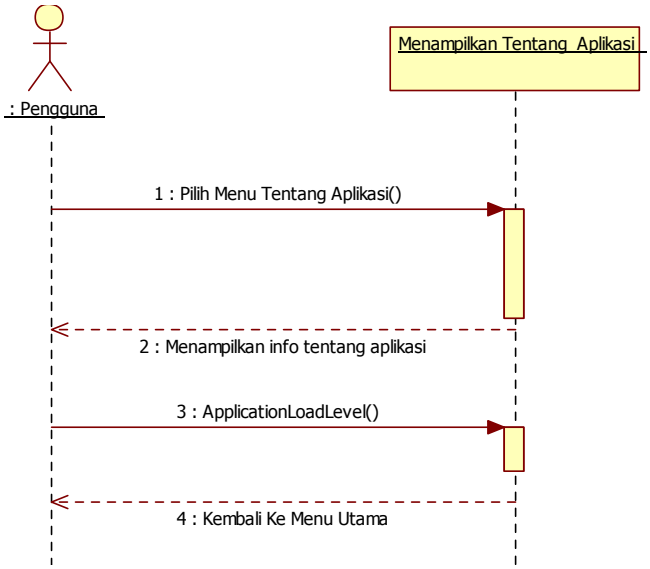
Berikut gambaran *sequence diagram* pada menu cara penggunaan dalam aplikasi ini yang akan digambarkan sebagai berikut :



Gambar 3.33 *Sequence Diagram* Cara Penggunaan

3. *Sequence Diagram* Menampilkan Tentang Aplikasi

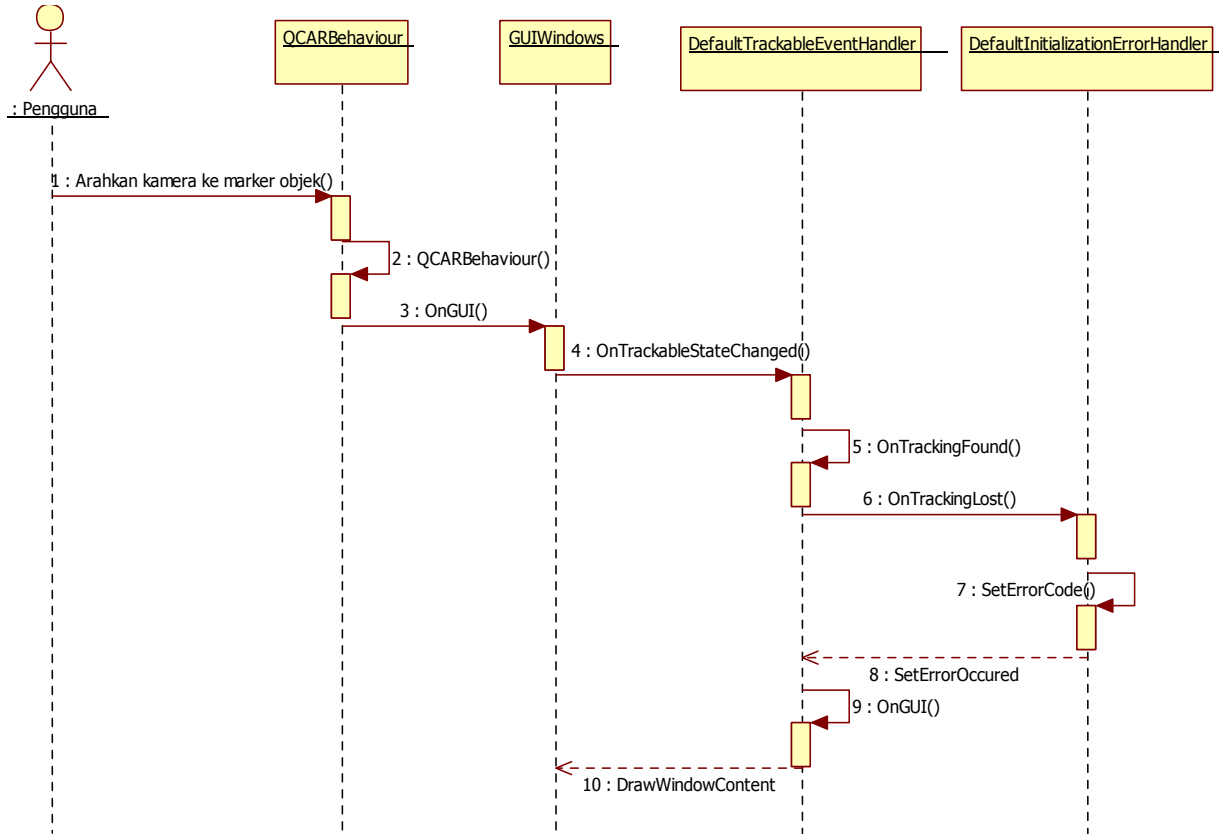
Berikut alur *sequence diagram* pada menu Tentang Aplikasi dalam aplikasi ini yang akan digambarkan sebagai berikut:



Gambar 3.34 *Sequence Diagram* Menampilkan Tentang Aplikasi

4. *Sequence Diagram* Pendeteksian Marker

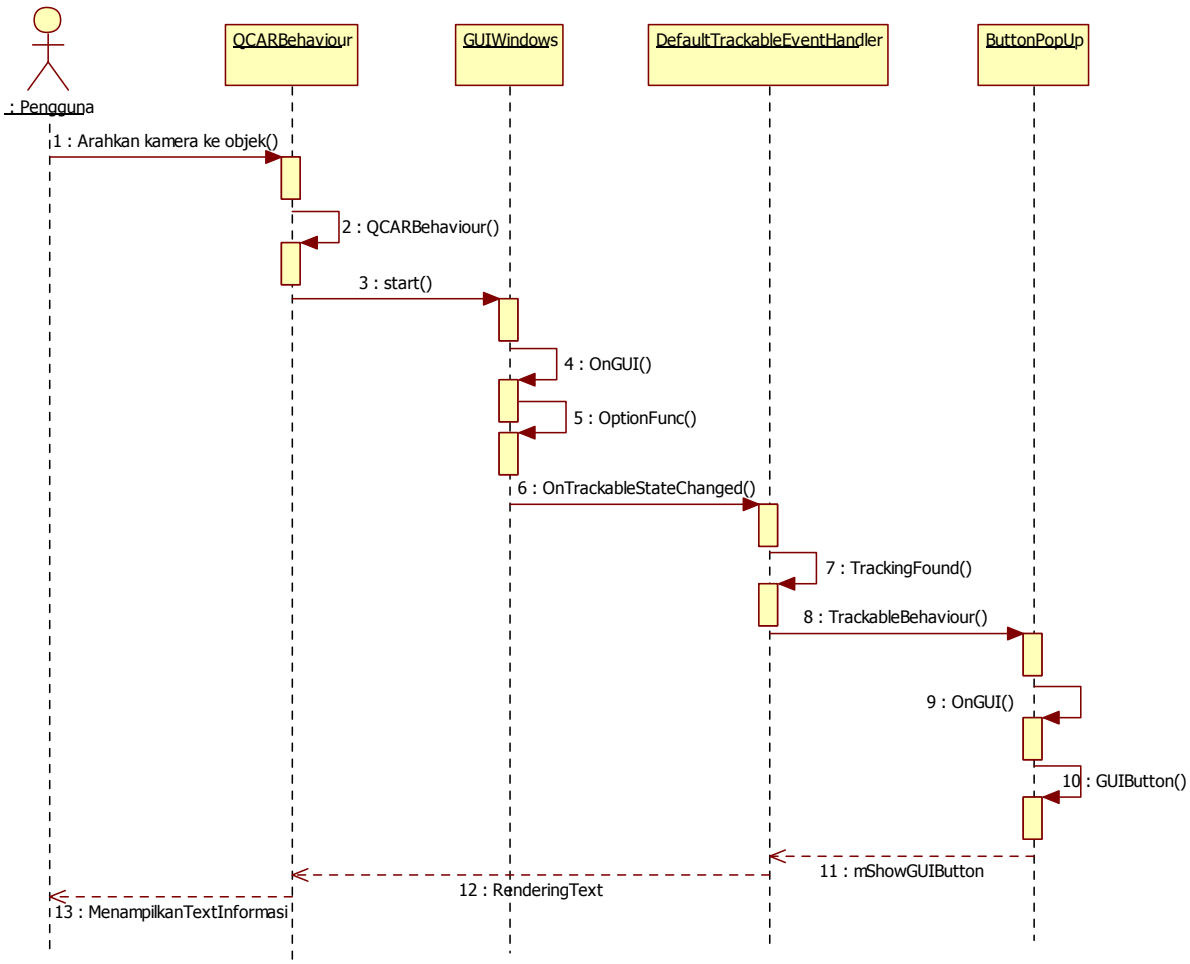
Berikut alur *sequence diagram* pada pendeteksian marker dalam aplikasi ini yang akan digambarkan sebagai berikut:



Gambar 3.35 *Sequence Diagram* Pendeteksian Deteksi Marker

5. *Sequence Diagram* Merender Text Informasi

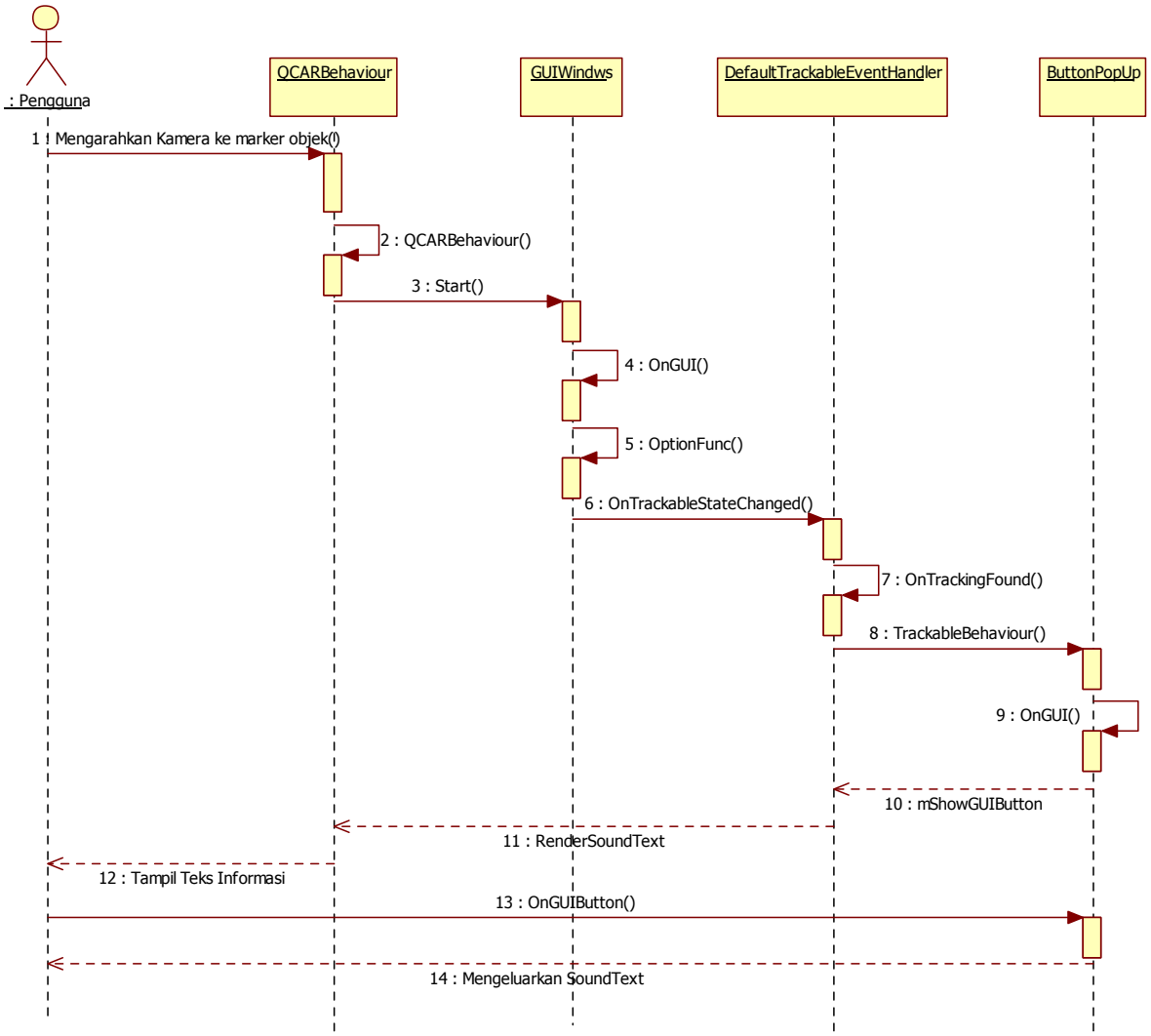
Berikut alur *sequence diagram* merender teks informasi dalam aplikasi ini yang akan digambarkan sebagai berikut:



Gambar 3.36 *Sequence Diagram* Merender Text Informasi

6. *Sequence Diagram* Mengeluarkan Sound Text

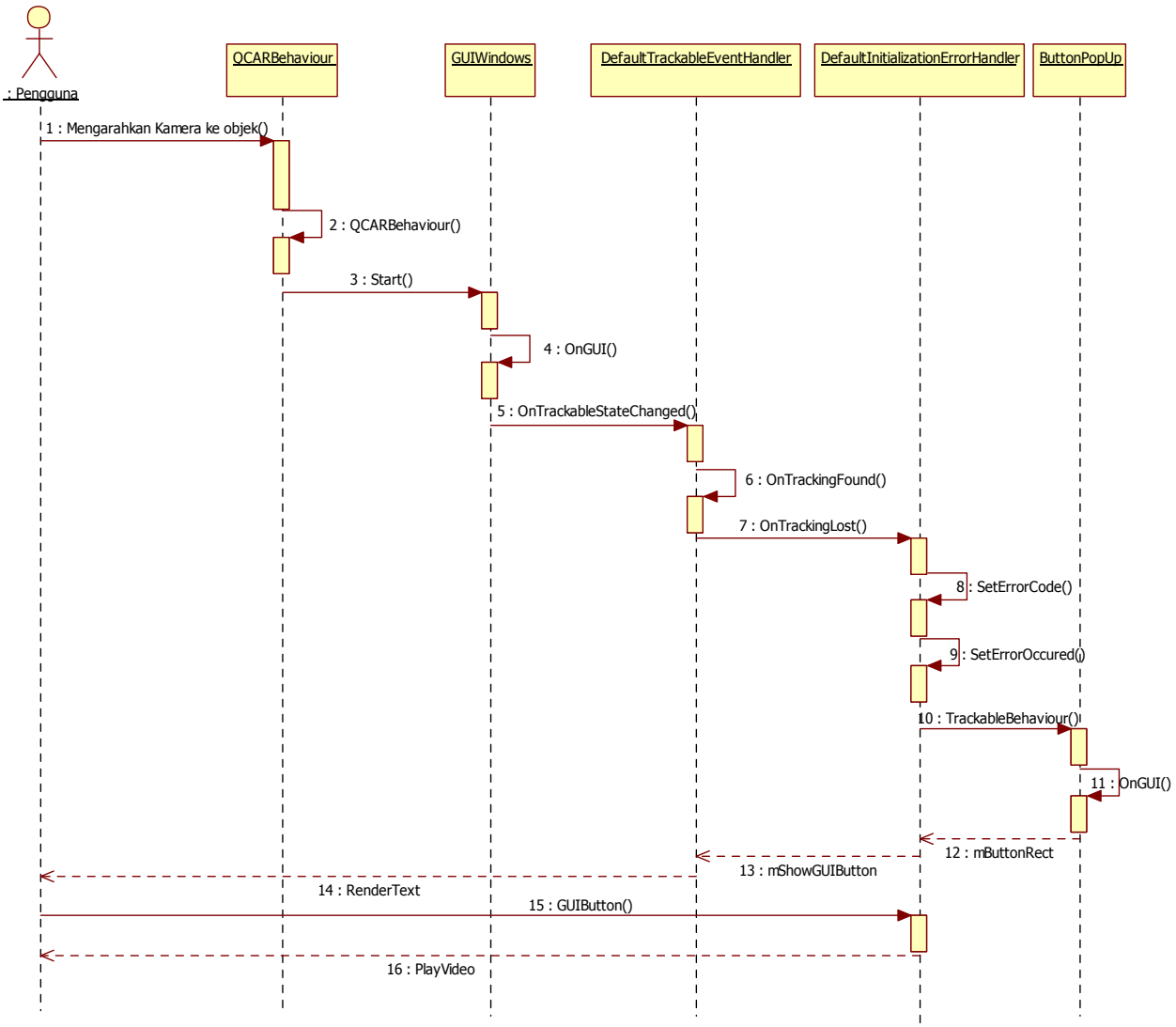
Berikut alur *sequence diagram* mengeluarkan *soundtext* dalam aplikasi ini yang akan digambarkan sebagai berikut:



Gambar 3.37 *Sequence Diagram* Mengeluarkan *SoundTe*xt

7. *Sequence Diagram* Menampilkan Video

Berikut alur *sequence diagram* menampilkan video pada aplikasi ini yang akan digambarkan sebagai berikut:



Gambar 3.38 *Sequence Diagram* Menampilkan Video Deskripsi

Berikut adalah deskripsi mengenai class diagram AR monumen yang akan dijelaskan pada tabel 3.16.

Tabel 3.16 Deskripsi Class Diagram AR Monumen

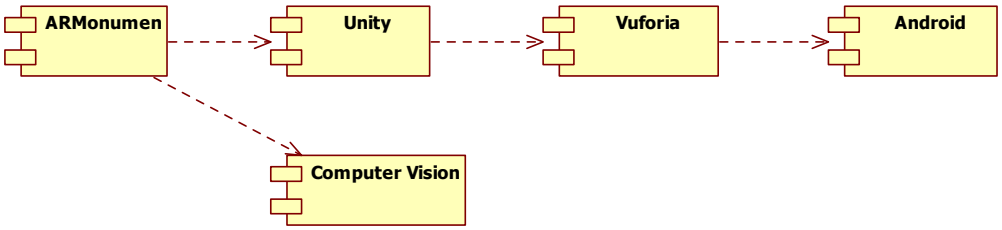
Class	Jenis Class	Deskripsi
CaraPenggunaan.cs	*Boundary*	Class yang berisikan tampilan data cara menggunakan aplikasi
Menu.cs	*Boundary*	Class yang berisikan method-method untuk fungsionalitas tampilan utama aplikasi yaitu menampilkan kategori menu utama
TentangAplikasi.cs	*Boundary*	Class yang berisikan tampilan data tentang penjelasan aplikasi
Keluar.cs	*Boundary*	Class yang berisikan keluar dari aplikasi
`Buttonpopup_monju.cs`	*Boundary*	Class ini berisikan method-method yang menampilkan sebuah deskripsi objek informasi berupa teks, audio, dan button.
`Buttonpopup_dewi.cs`	*Boundary*	Class ini berisikan method-method yang menampilkan sebuah deskripsi objek informasi berupa teks, audio, dan button.
`Buttonpopup_BLA.cs`	*Boundary*	Class ini berisikan method-method yang menampilkan

		sebuah deskripsi objek informasi berupa teks, audio, dan button.
Buttonpopup_husein.cs	*Boundary*	Class ini berisikan method-method yang menampilkan sebuah informasi objek berupa teks, audio dan button
GUIWindows.cs	*Boundary*	Class ini berisikan method yang menampilkan button keterangan info ketika kamera telah aktif
DevaultTrackableEventHandler	*Control*	Class ini berisikan method yang dapat menangani pelacakan secara default
DevaultInitializationError Handler	*Control*	Class ini berisikan method untuk menginisialisasi kesalahan terhadap defaulttrackablehandler
ITrackableEventHandler	*Control*	Class yang berisi Antarmuka untuk menangani perubahan yang dilacak.
CameraDevice	*Control*	Class yang berisikan method-method untuk menghidupkan kamera
QCARAbstractBehaviour	*Control*	Class yang berisikan method-method yang menangani pelacakan
QCARBehaviour	*Control*	Class yang menangani pelacakan

DataSet	Control	Class yang berisi method-method yang dapat dimuat dan memegan koleksi Trackables
DataSetLoadAbstractBehaviour	Control	Class yang berisi perilaku untuk secara otomatis memuat dan mengaktifkan satu atau lebih DataSet pada Startup
DataSetLoadBehaviour	Control	Class yang berisi perilaku untuk secara otomatis memuat dan mengaktifkan satu atau lebih DataSet pada Startup
TrackbleBehaviour	Control	Class yang berisikan method-method sebagai definisi augmentasi untuk dilacak dalam editor serta dilacak hasil pada saat runtime
DataSetTrackableBehaviour	Control	Class ini adalah dasar untuk semua Trackable yang merupakan bagian dari DataSet
ImageTargetAbstractBehaviour	Control	Class ini berisi sebagai definisi untuk image target dalam editor serta hasil sasaran image target dilacak saat runtime
ImageTarget	Control	Class ini berisikan sebuah prilaku pelacakan untuk mewakili natural feature target

3.1.8.7 Component Diagram

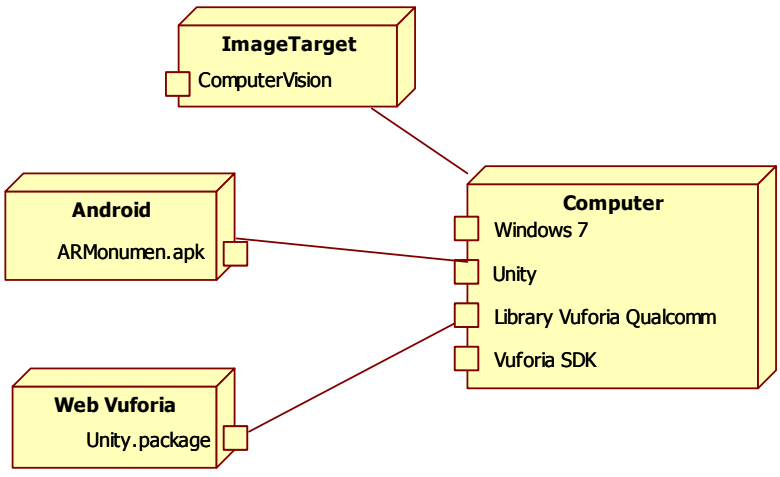
Component diagram merupakan bagian fisik dari sebuah system, karena menetap dikomputer, bukan dibenak para analis. Komponen bisa berupa tabel, file data, file .exe, dokumen dan lain-lain. Berikut *component diagram* aplikasi media pengenalan monumen bersejarah kota Bandung yang akan dibangun dapat dilihat pada gambar dibawah ini.



Gambar 3.39 *Component Diagram*

3.1.8.8 Deployment Diagram

Deployment Diagram menunjukan tata letak sebuah sistem secara fisik, menampilkan bagian-bagian *software* yang berjalan pada bagian-bagian hardware yang digunakan untuk mengimplementasikan sebuah sistem dan keterhubungan antara komponen-komponen *hardware* tersebut. *Deployment diagram* aplikasi media pengenalan monumen bersejarah kota Bandung yang akan dibangun dapat dilihat pada gambar dibawah ini.



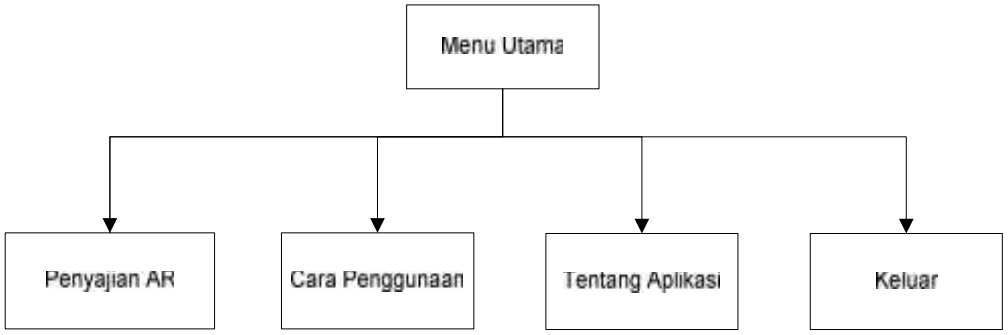
Gambar 3.40 *Deployment Diagram*

3.2 Perancangan Sistem

Perancangan akan dimulai setelah tahap analisis terhadap sistem selesai dilakukan. Perancangan dapat didefinisikan sebagai proses aplikasi berbagai teknik dan prinsip bagi tujuan pendefinisian suatu perangkat, suatu proses atau sistem dalam detail yang memadai untuk memungkinkan realisasi fisiknya. Perancangan ini digambarkan dengan melakukan beberapa langkah diantaranya meliputi perancangan struktur menu, perancangan antarmuka, perancangan pesan, jaringan semantik dan perancangan method.

3.2.1 Perancangan Struktur Menu

Struktur *menu* adalah bentuk umum dari suatu rancangan program untuk memudahkan pemakai dalam menjalankan program komputer. Sehingga saat menjalankan program, pengguna tidak mengalami kesulitan dalam memilih *menu-menu* yang diinginkan. Berikut ini perancangan struktur *menu* pada aplikasi pengenalan monumen bersejarah yang ditunjukan pada gambar berikut ini.

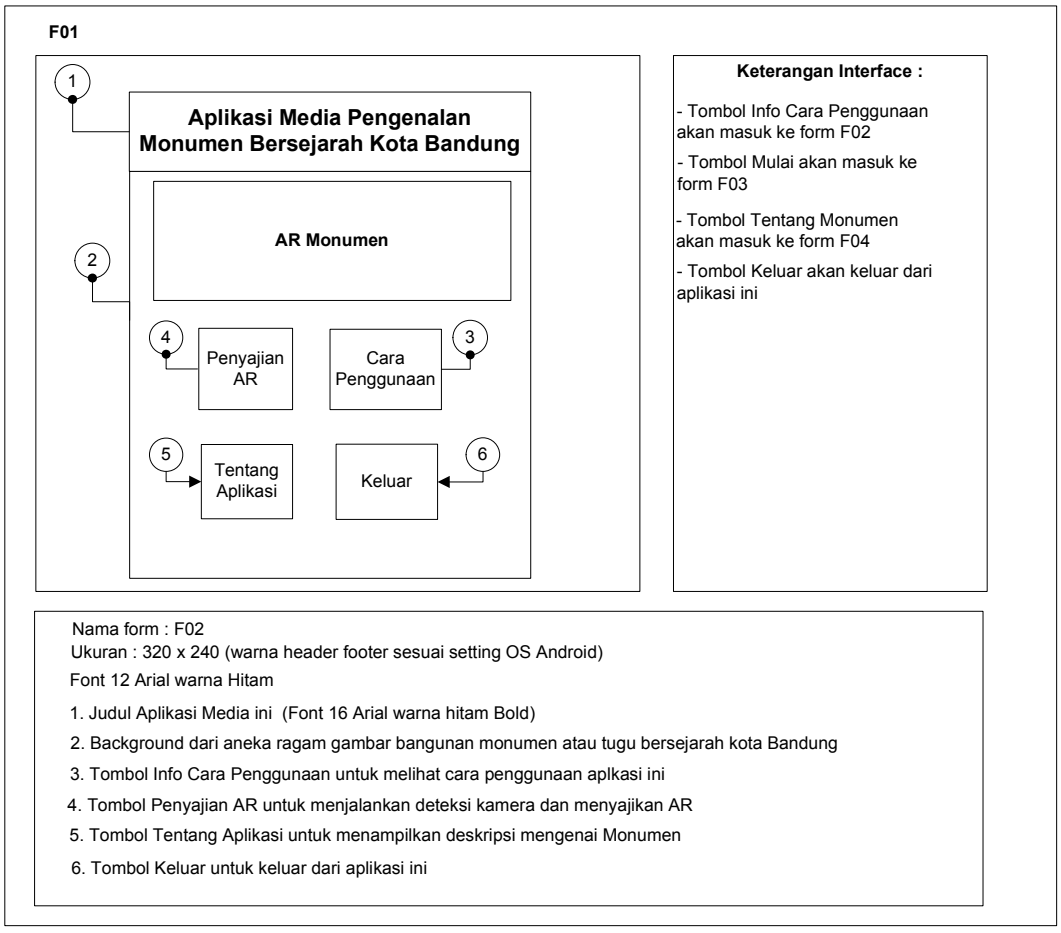


Gambar 3.41 Struktur Menu

3.2.2 Perancangan Antarmuka

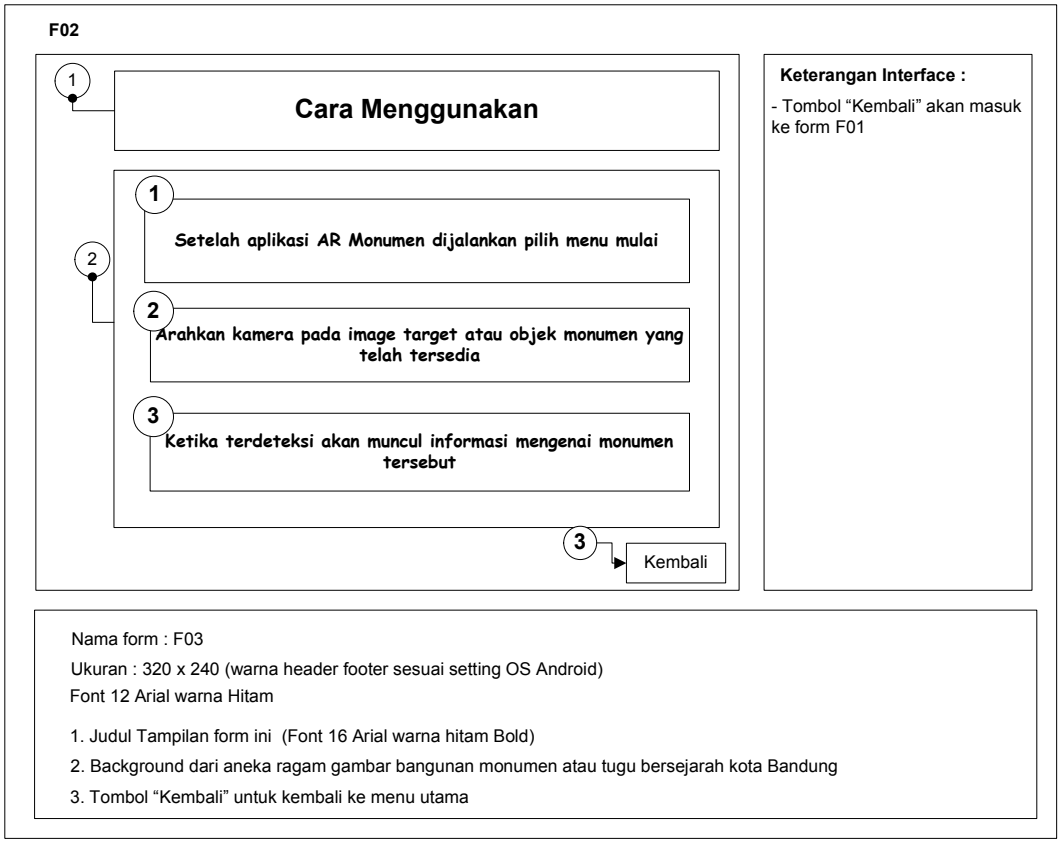
Perancangan antarmuka mendeskripsikan rancangan tampilan dari setiap *form* yang akan digunakan pada tampilan aplikasi ini. Perancangan antarmuka dibutuhkan untuk mawakili keadaan sebenarnya dari aplikasi yang akan dibangun. Berikut ini gambaran perancangan antarmuka yang ada dari aplikasi yang akan dibangun yaitu :

3.2.2.1 Perancangan Antarmuka Menu Utama



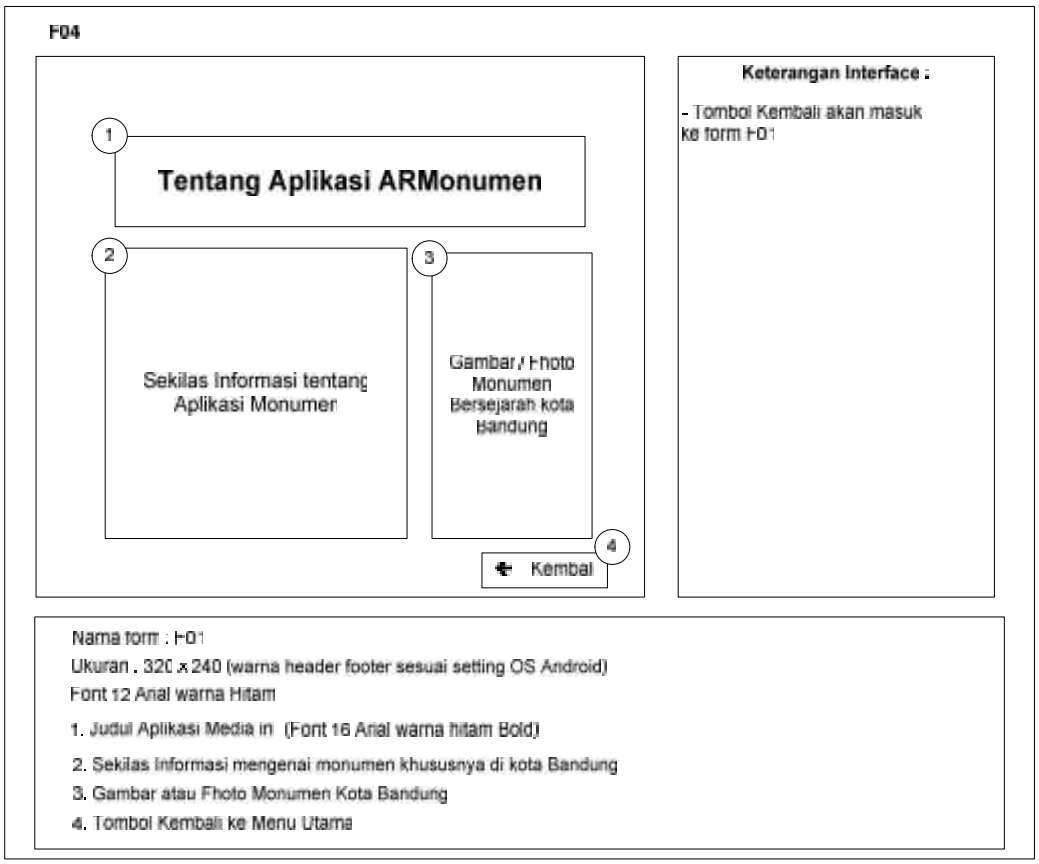
Gambar 3.42 Antarmuka Menu Utama

3.2.2.2 Perancangan Antarmuka Cara Penggunaan



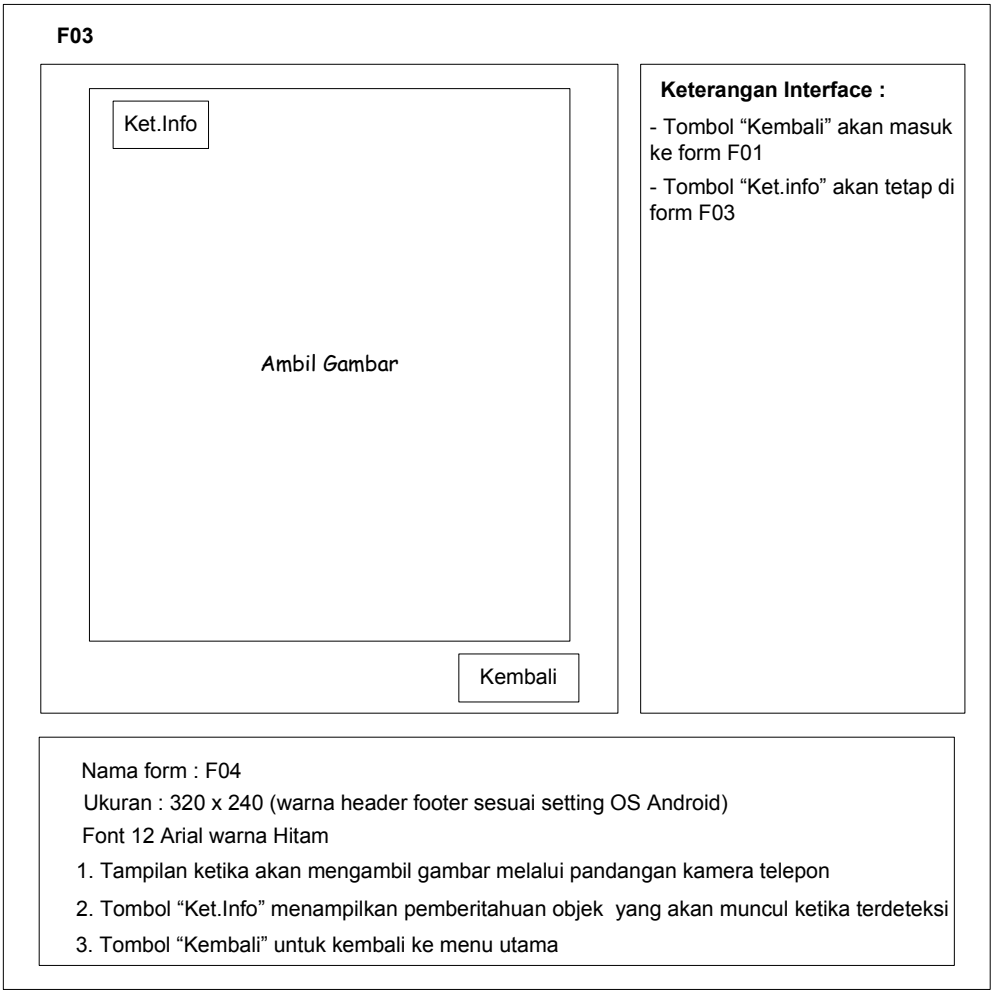
Gambar 3.43 Antarmuka Cara Penggunaan

3.2.2.3 Perancangan Antarmuka Tentang Aplikasi



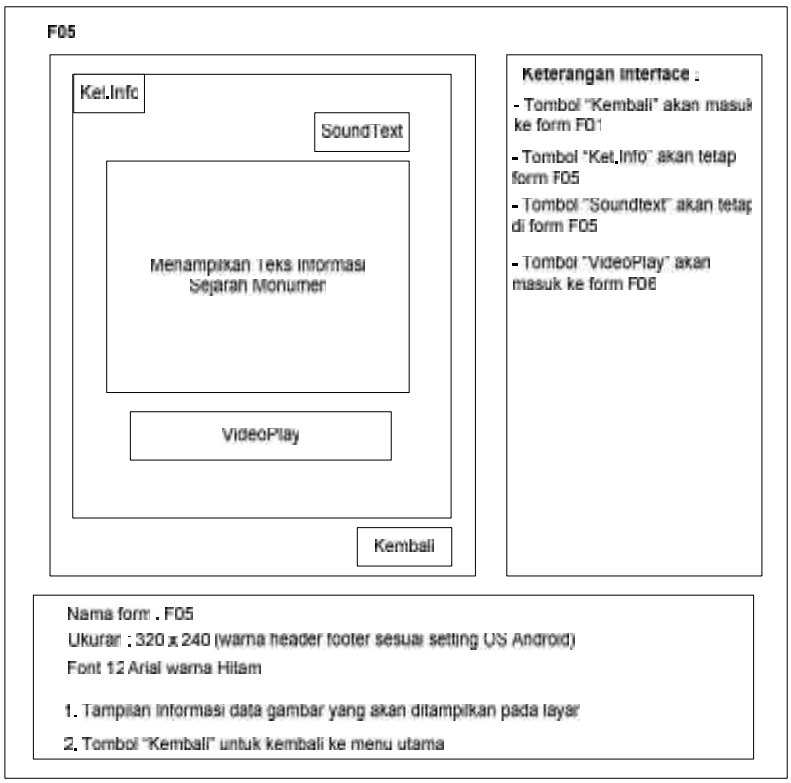
Gambar 3.44 Antarmuka Tentang Aplikasi

3.2.2.4 Perancangan Antarmuka Penyajian AR Ketika Kamera Aktif



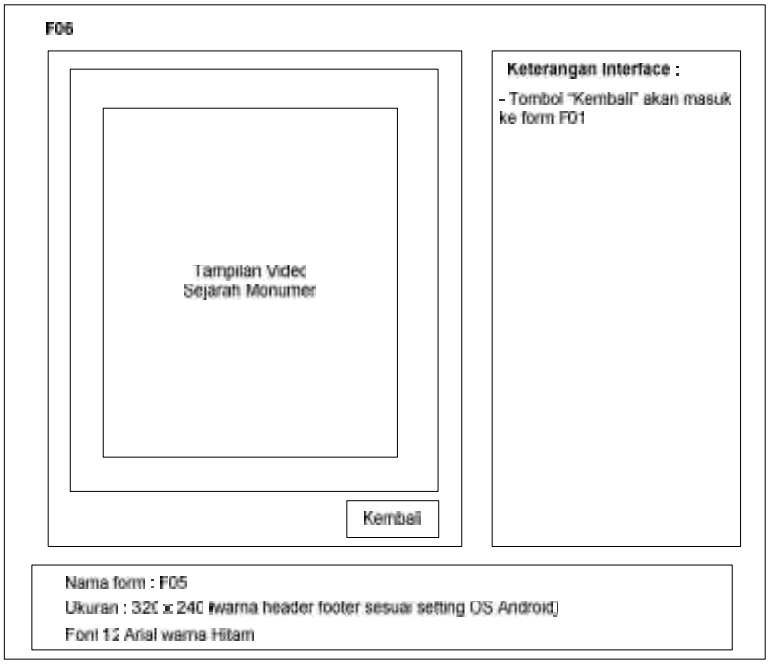
Gambar 3.45 Perancangan Antarmuka Penyajian AR Ketika Kamera Aktif

3.2.2.5 Perancangan Antarmuka Penyajian ARText Informasi



Gambar 3.46 Perancangan Antarmuka Penyajian ARText Informasi

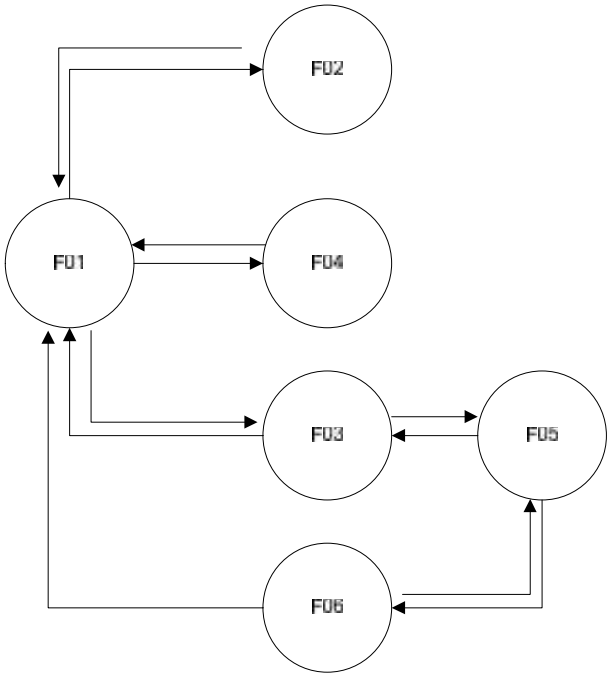
3.2.2.6 Perancangan Antarmuka Penyajian ARVideo



Gambar 3.47 Perancangan Antarmuka Penyajian ARVideo

3.2.3 Jaringan Semantik

Jaringan semantik ini menunjukan form-form yang bisa diakses oleh pengguna seperti ditunjukan pada gambar 3.48 dibawah ini.



Gambar 3.48 Jaringan Semantik

Keterangan :

F01 : Tampilan Menu Utama

F02 : Tampilan Menu Cara Penggunaan

F03 : Tampilan Menu Penyajian AR Ketika Kamera Aktif

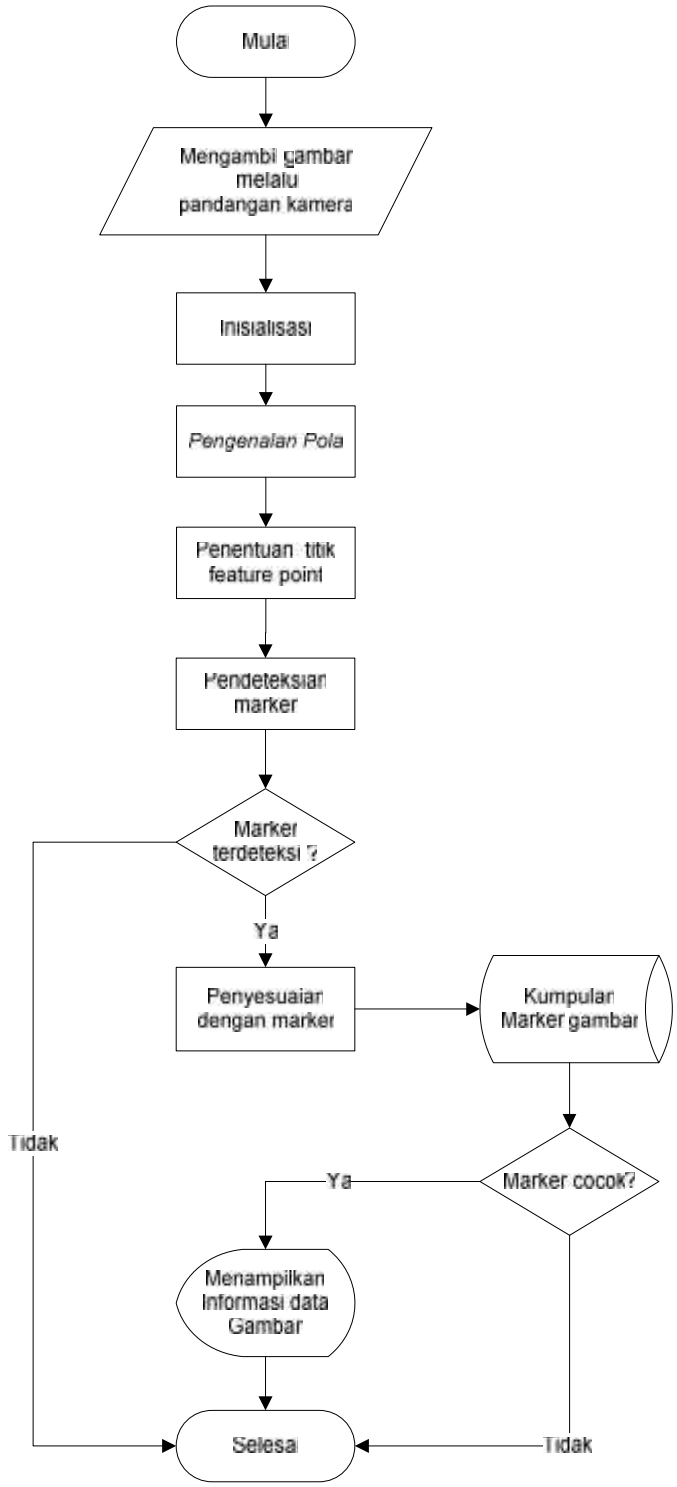
F04 : Tampilan Menu Tentang Aplikasi

F05 : Tampilan Penyajian ARTeks Informasi

F06 : Tampilan Penyajian ARVideo

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 didalam aplikasi media pengenalan monumen/tugu bersejarah menggunakan teknologi *augmented reality*. Adapun method yang terdapat dalam aplikasi yang akan dibangun adalah sebagai berikut :



Gambar 3.49 Perancangan Method