BAB III
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem merupakan penguraian dari suatu sistem informasi 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.

3.1.1 Analisis Masalah

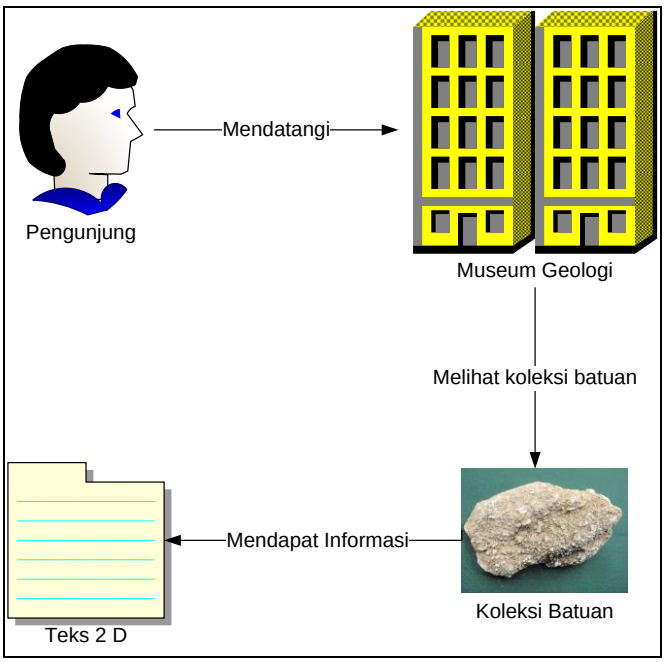
Semakin berkembangnya teknologi informasi sebagai sarana pemenuhan kebutuhan informasi bagi masyarakat dan penyampaian informasi dengan berupa teks informasi dari koleksi batuan yang di pajang, menjadi masalah yang dihadapi Museum Geologi Bandung dalam beberapa tahun terakhir ini. Hal itu juga menjadi alasan bagi para pengunjung kurang mendapat informasi secara lengkap mengenai bebatuan yang ada. Analisis perubahan media habit dalam mengkonsumsi media di era teknologi informasi terjadi penurunan eksistensi penyampaian informasi sederhana terhadap pengunjung dengan perkembangan teknologi informasi pada saat ini.

Keunggulan dari teknologi system rekognisi dalam segi pengenalan dan menampilkan suatu informasi dapat diterapkan pada koleksi batuan untuk dapat memberikan solusi dari permasalahan yang ada. Dengan demikian, tujuan dari pembangunan aplikasi yaitu memberikan suatu aplikasi berbasis teknologi rekognisi dalam mendeteksi batuan pada batuan di Museum Geologi yang telah disediakan untuk meningkatkan penyampaian informasi secara menarik dan lengkap kepada masyarakat dengan memberikan fasilitas tambahan teknologi *pattern recognition*

berupa penerapan metode deteksi pola. Dengan menerapkan teknologi ini, penyampaian informasi yang didapat akan lebih menarik dan interaktif yaitu berupa video atau animasi 3D. Sistem yang dibuat merupakan program untuk mendeteksi sebuah *markerless* sekaligus menampilkan objek tiga dimensi dan video yang bisa diputar. Objek yang di buat berupa video dan animasi serta menerapkan sebagian dari objek dengan metode *markerless*, agar pengunjung lebih berinteraksi dengan aplikasi dan memaksimalkan penyampaian informasi tersebut.

3.1.2 Analisis Sistem Yang Berjalan

Analisis sistem atau analisis proses adalah tahapan yang memberi gambaran tentang sistem yang sedang berjalan sekarang. Analisis ini bertujuan untuk memberi gambaran yang lebih detail bagaimana cara kerja dari sistem yang sedang berjalan. Prosedur pada proses media yang sedang berjalan sekarang dapat digambarkan ke dalam sebuah alur sistem dibawah ini.



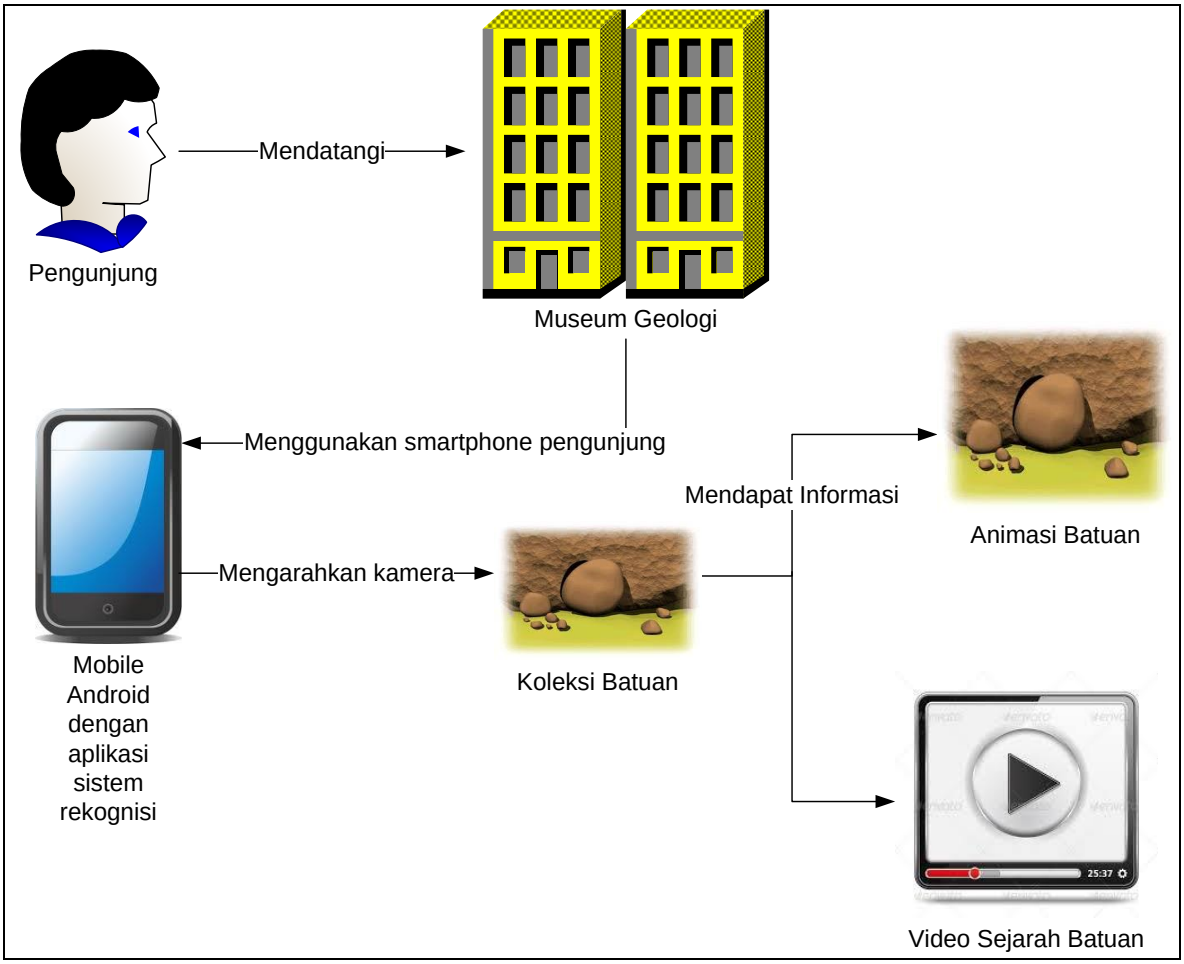
Gambar 3. 1 Alur sistem yang sedang berjalan

1. Pengunjung mendatangi Museum Geologi Bandung
2. Pengunjung melihat koleksi batuan yang dipajang di dalam Museum Geologi
3. Pengunjung mendapatkan informasi batuan berupa teks 2D

Dari gambaran prosedur yang didapat untuk mendapatkan informasi mengenai bebatuan menggunakan teknologi sistem rekognisi bisa dijadikan sebagai media alternatif sehingga penyampaian informasi menjadi lebih menarik dan interaktif dalam bentuk sebuah alat peraga berupa *mobile* berbasis *Android*.

3.1.3 Analisis Arsitektur Sistem

Arsitektur fisik sistem yang dibangun terdiri dari komponen aplikasi Deteksi Bebatuan, dan *database* aplikasi. Arsitektur sistem dalam pembangunan aplikasi ini terdapat aplikasi yang terhubung ke *database system*, menampilkan antarmuka pengunjung melalui mobile berbasis Android. Aplikasi ini berfungsi sebagai antarmuka antara pengunjung dengan *system database*, Pada bagian ini berisi informasi data menampilkan antarmuka pengguna dengan sistem. Fungsi utama database adalah menyimpan data dari aplikasi deteksi bebatuan dan objek 3D dan *video playback*.



Gambar 3. 2 Arsitektur Sistem

Pada bagian ini, pengunjung mendatangi Museum Geologi lalu Mobile Android dengan aplikasi sistem rekognisi untuk mendeteksi batuan. Setelah itu pengunjung mengarahkan kamera ke koleksi batuan yang telah disediakan maka muncul output berupa animasi batuan dan video sejarah batuan. Fungsi utama Aplikasi Sistem Rekognisi adalah aplikasi pendukung yang berfungsi sebagai media pembelajaran dalam bentuk aplikasi untuk penyampaian informasi mengenai bebatuan secara menarik, interaktif dan berbasis *mobile* Android.

3.1.4 Analisis Kebutuhan UNITY (*Software Engine*)

Unity merupakan software pembangun game. Masa kini unity dapat pula digunakan dalam pembangun aplikasi berbasis *recognition* seperti hal rupanya *augmented reality*.

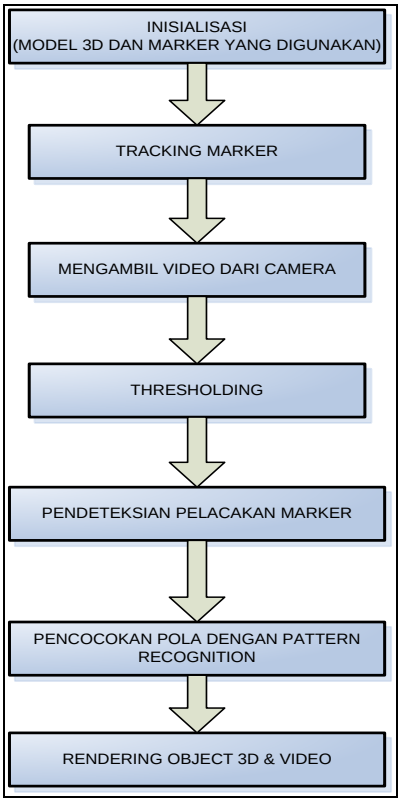
Unity adalah sebuah sebuah *tool* yang terintegrasi untuk membuat *game*, arsitektur bangunan dan simulasi. Kelebihan unity yaitu *multiple platform*, maksudnya banyak platform yang disupport oleh unity, misal Windows, Mac, iPhone, iPad, Android, Nintendo Wii, dan juga browser. Untuk browser, memerlukan sebuah *plugin*, yaitu Unity Web Player, sama halnya dengan Flash Player pada Browser.

Bahasa pemrograman yang digunakan bermacam-macam, mulai dari Javascript, C#, dan Boo. Banyak hal yang bisa di lakukan di unity , ada fitur *audio reverb zone* , *particle effect* , *Sky Box* Untuk menambahkan langit , dan masih banyak lagi tentunya juga bisa langsung mengedit tekstur dari editor seperti photoshop dll.

3.1.5 Analisis Metode Terhadap Kasus

Analisis metode terhadap kasus merupakan analisis yang mendeskripsikan bagaimana proses UNITY dari awal inisialisasi, *tracking marker*, sampai dengan proses *rendering* objek 3D dengan metode *markerless* yang diterapkan oleh UNITY.

Dalam perancangan aplikasi dengan teknologi sistem rekognisi, mendeteksi *pattern* yang disediakan dalam bentuk target batuan, dalam hal ini objek virtual berupa objek 3D animasi batuan dan objek nyatanya berupa koleksi batuan dengan pola tertentu (*markerless).*



Gambar 3.3 Diagram alir sistem rekognisi

Secara keseluruhan, proses sistem rekognisi dapat digambarkan dengan diagram alir pada gambar 3.3. Aplikasi melakukan inisialisasi terlebih dahulu sebelum melakukan *tracking marker*, *marker* dideteksi dari masukan video camera target koleksi batuan berupa mobile Android, jika marker terdeteksi maka dilakukan proses *thresholding marker*, lalu pendeteksian pelacakan marker. Pencocokan pola dengan *pattern recognition*, bila cocok maka dilakukan *rendering* objek 3D dan video.

Secara garis besarnya, dalam perancangan ada tiga bagian utama yaitu sebagai berikut :

1. Inisialisasi
2. Tracking Marker
3. Rendering Objek 3D

3.1.5.1 Inisialisasi

Pada tahap ini ditentukan *marker* yang akan digunakan, sumber input video nya, dan objek 3D yang akan digunakan .Pada bagian inisialisasi ini, objek 3D diinisialisasi terlebih dahulu karena loading objek 3D memerlukan waktu yang cukup lama.

3.1.5.1.1 Inisialisasi Model 3D

Model 3D yang akan ditampilkan di-*load* terlebih dahulu. Agar aplikasi dapat menampilkan objek 3D tertentu tanpa merubah atau membangun ulang aplikasi, diperlukan sebuah file konfigurasi untuk menentukan objek 3D yang akan di-*load* sesuai dengan pola *marker* yang dideteksi.

1. Proses Pembentukan Objek

Dalam proses pemodelan objek terdiri dari 3 langkah.

1) Menyesuaikan objek 3D dengan animasi atau bentuk yang akan dibuat.
2) Memasukan tekstur sesuai dengan objek 3D.
3) Mengekspor objek yang sudah dirancang dan dibuat ke dalam format Autodesk FBX (*.FBX).

2. Proses Input Data Objek

1) Input data objek

Proses pertama yaitu menambahkan data objek .FBX ke dalam library UNITY. File .FBX ini berfungsi untuk memanggil bentuk objek yang sudah di ekspor.

2) Input variabel tekstur objek

Proses ini berfungsi untuk menambahkan tekstur pada objek, tekstur pada objek tidak bisa digunakan atau muncul sebelum ada penambahan variabel dan pengaturan kode yang ditambahkan agar tekstur tersebut sesuai dengan model yang telah dibuat.

3. Penambahan variabel tekstur.

   Penambahan variabel tekstur disesuaikan dengan jumlah dan nama material id pada file .FBX yang digunakan pada objek.

   1) Penambahan kode class variabel tekstur

      Penambahan kode class pada gambar atau tekstur yang ditambah disesuaikan dengan variabel tekstur.

   2) Proses inisialisasi objek

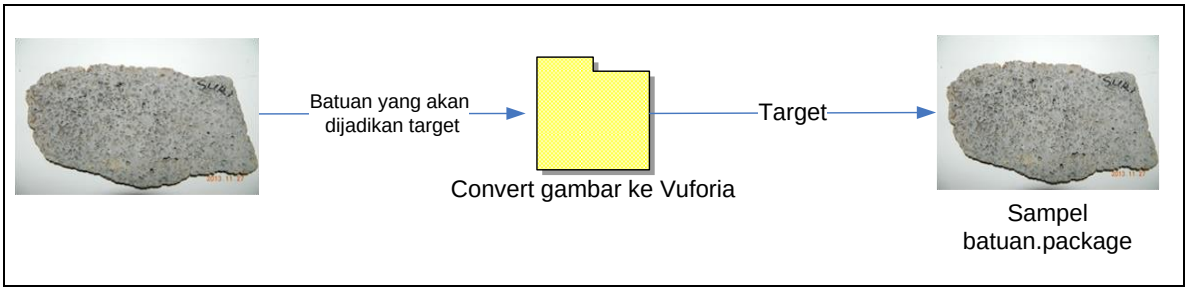
      Proses ini dilakukan pada saat objek di-*compile* menjadi aplikasi dan data objek beserta teksturnya objek akan di inisialisasi.

   3) Tampilan hasil objek

      Proses yang terakhir adalah menampilkan hasil objek yang telah melalui proses inisialisasi, dimana pada saat objek tampil disesuaikan dengan *marker* yang ada.

3.1.5.1.2 Penanda Yang Digunakan

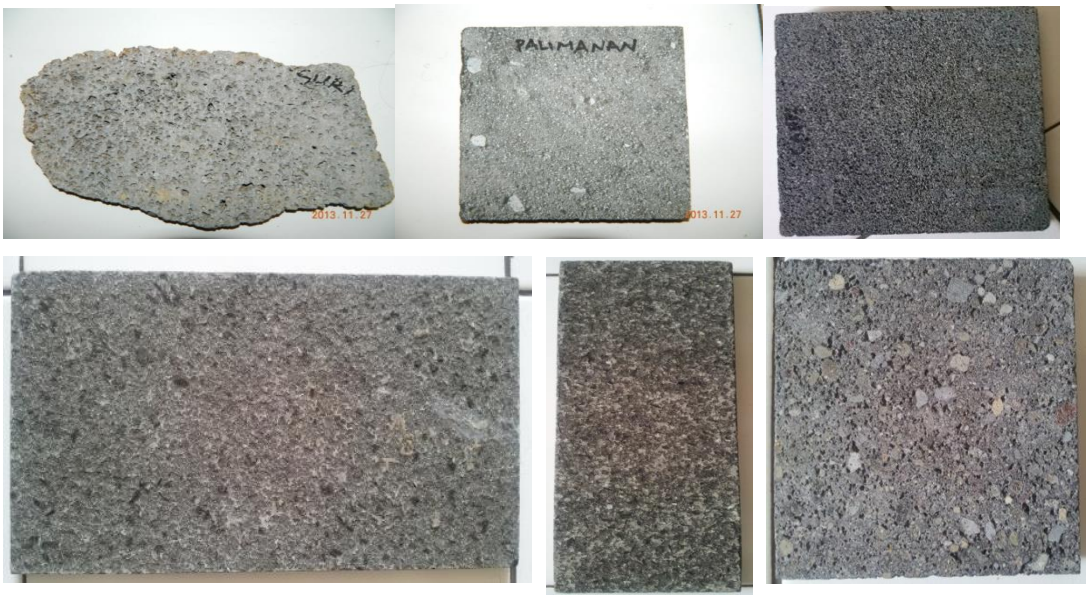
Target yang digunakan pada analisis ini didapatkan dari sampel koleksi batuan yang sudah ada, ini disebabkan karena aplikasi ini masih bersifat sementara belum di resmikan. Namun kedepannya setelah aplikasi diresmikan target yang digunakan akan sesuai dengan koleksi batuan dan informasi yang ditampilkan di Museum Geologi. Adapun proses pengambilan *target* adalah sebagai berikut :



Gambar 3.4 Proses Pembentukan *Marker*

Proses pembuatan target batuan menjadi target.package dilakukan oleh pihak Vuforia dengan cara mengirimkan gambar yang akan di-*convert*, melalui *marker engine* yang disediakan oleh pihak Vuforia. Setelah gambar di- *convert* menghasilkan file dengan format .package. File tersebut kemudian dijadikan masukan pada coding untuk mendeteksi gambar yang dijadikan *marker*. Spesikasi pola *target*:

1. Pola target minimum harus memiliki lebar 550 pixels
2. Format gambar yang dikirimkan .jpg



Gambar 3.5 Contoh Batuan

3.1.5.2 Tracking Marker

UNITY memiliki kemampuan untuk mendeteksi gambar dan menghitung posisi gambar tersebut menggunakan kamera standar. Informasi posisi yang didapatkan akan dipergunakan untuk menempatkan objek atau model tiga dimensi atau video ke dalam posisi gambar atau markerless. Ada empat langkah, dalam proses kerja *tracking marker* UNITY.

3.1.5.2.1 Mengambil video dari camera

Mendapatkan masukan video dari sebuah kamera alat peraga berupa mobile android adalah langkah awal yang harus dilakukan. 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.. Tujuan dari kalibrasi kamera adalah untuk menghitung tingkat distorsi dari sebuah lensa kamera yang digunakan agar citra yang dihasilkan mendekati citra ideal. Berikut persamaannya :

iconWidth = 0.5f * videoHeight / videoWidth; ...[P 3.1]



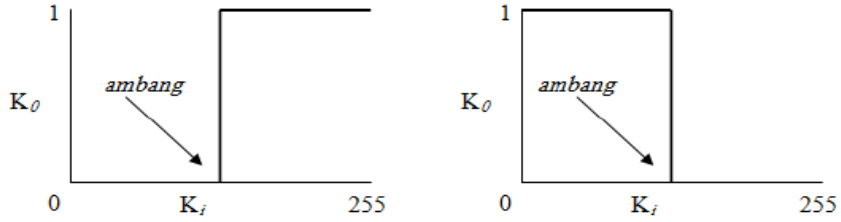
Gambar 3.6 Mengambil Citra Dari Batuan

3.1.5.2.2 Binarisasi citra masukan (*thresholding*)

Langkah pertama pada aplikasi visi komputer yang terletak pada deteksi tepi adalah untuk men-*threshold* sumber citra atau disebut juga binarisasi, fungsi dari proses ini adalah untuk membantu sistem agar dapat mengenali bentuk segi empat dan pola marker pada citra yang diterima. Proses tahapannya akan dijelaskan pada operasi pengambangan ini mengunakan pengambangan ganda. berikut persamaannya :

$$K_0 = \begin{cases} 0, \text{jika } K_i < \text{ambang} \\ 1, \text{jika } K_i >= \text{ambang} \end{cases} \quad \textit{atau} \quad K_0 = \begin{cases} 0, \text{jika } K_i >= \text{ambang} \\ 1, \text{jika } K_i < \text{ambang} \end{cases}$$

...[P 3.2]



Gambar 3.7 Ambang batas dari persamaan tunggal

Maka dihasilkan persamaan pada operasi *thresholding.* Berikut persamaannya :

for (var i = 0; i < targetData.Length; i++) ... [P 3.3]

3.1.5.2.3 Pendeteksian pelacakan marker

Langkah berikutnya dari Unity adalah menemukan area yang berdampingan dalam citra yang di-*treshold*. Area yang berdampingan diberi tanda sebagai persegi (*marker outline*). menggunakan metode ASSURF dengan algoritma utamanya algoritma SURF (*Speeded Up Robust Feature*).

Proses tahapannya akan dijelaskan sebagai berikut :

Langkah 1 : Menentukan *keypoint*

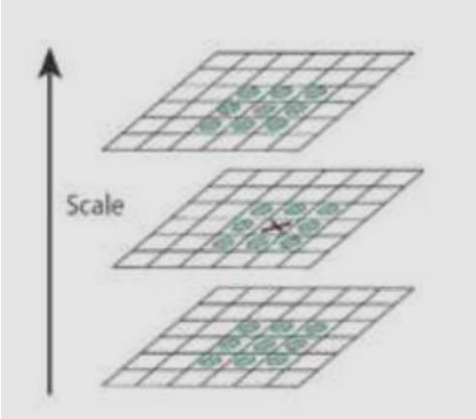
Proses pertama yaitu menentukan *keypoint* ketika treshold dinaikan, jumlah *keypoint* yang terdeteksi lebih kecil begitupun sebaliknya. Berikut persamaannya :

float offset = it.size.x * 0.5f; ...[P 3.4]

Langkah 2 : *Non-maxima Suppresion*

Proses ini dilakukan untuk mencari sekumpulan calon *keypoint* dengan membandingkan tiap-tiap pixel gambar pada scale *space* dengan 26 tetangga. 26 tetangga pixel itu terdiri atas 8 titik di scale asli dan 9 titik di tiap-tiap scale diatas dan

dibawahnya. Proses inilah yang menghasilkan keypoint dari suatu gambar. Gambar 3.8 menunjukkan *non-maxima suppresion.*



Gambar 3.8 ***Non-maxima suppression***

Maka dihasilkan persamaan untuk menentukan skala sebagai berikut :

float childScaleFactor = it.GetSize()[0] / size[0]; ...[P 3.5]

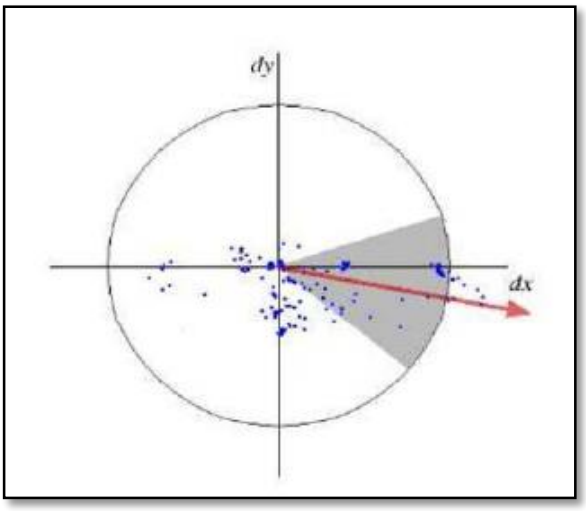
Langkah 3 : Proses mencari lokasi keypoint

Proses terakhir yaitu proses mencari lokasi keypoint menggunakan interpolasi data yang dekat dengan keypoint hasil proses sebelumnya. Ini dilakukan dengan mencocokan quadratic 3D yang diajukan oleh Brown H(x,y,∂)adalah determinan Hessian.

$$\begin{bmatrix} 0 & 1 & 2 \\ 2 & 1 & 3 \end{bmatrix} \quad \text{...[P 3.6]}$$

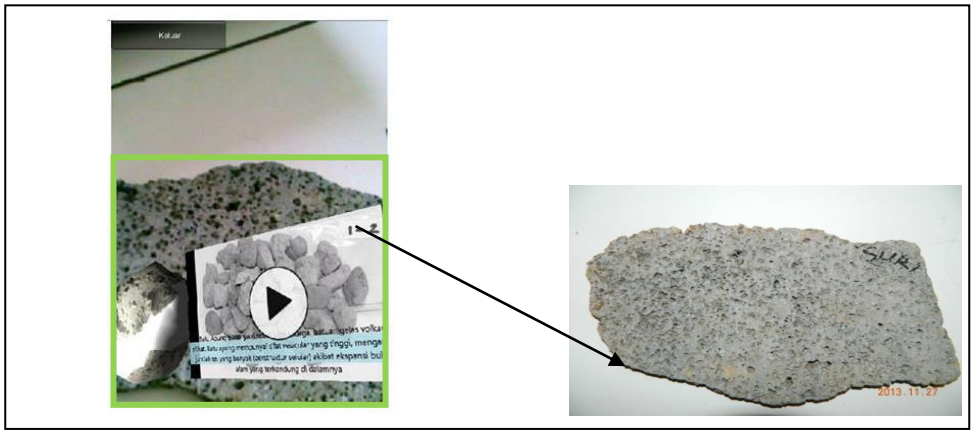
Langkah 4 : Proses ekstraksi deskriptor

Dilakukan dengan membuat daerah kotak persegi disekitar keypoint, dimana keypoint sebagai pusat dari daerah kotak persegi tersebut, dan orientasinya disekitar orientasi yang ditentukan dan ditunjukkan pada Gambar 3.9.



Gambar 3. 9 Jendela pergeseran orientasi

Kemudian, respon wavelet dx dan dy dijumlahkan untuk setiap sub-region. Hal ini akan memberikan informasi tentang polar dari perubahan intesitas, dan juga akan dihasilkan jumlah nilai absolut dari respon |dx| dan |dy|. Masing-masing sub-region mempunyai 4 dimensi deskriptor vektor v, yaitu dx ,dy, |dx| dan |dy|. Untuk 4 x 4 sub-region, maka panjang vektor deskriptornya berjumlah 64. Sehingga yang akan ditampilkan pada layar hanyalah area yang memiliki bentuk segi empat dan pola-pola gambar yang sudah di-labelling (Gambar 3.10).



Gambar 3.10 Pendeteksian pelacakan marker (*marker detection tracking*)

3.1.5.2.4 Pencocokan pola dengan Pattern Recognition (Pengenalan Pola)

Setelah semua area persegi dan pola-pola gambar ditandai, UNITY menganalisa citra yang berada di dalam persegi dan membandingkan polanya dengan sekumpulan pola yang telah ditentukan (pencocokan pola). Unity mengekstrak pola didalam persegi menggunakan transformasi homography. Unity memberikan memberikan nilai confidance kepada pola yang cocok, jika kecocokannya di atas nilai yang telah ditentukan maka polanya dinyatakan cocok.

3.1.5.3 Rendering Objek 3D

Transformasi matriks yang dikalkulasikan dilangkah sebelumnya yang digunakan Unity dan menampilkan objek yang sesuai dengan sebuah library 3D. Unity menyertakan kelas pendukung yang mengkonversikan transformasi matriks Unity ke setiap kelas matriks internal library 3D tersebut.

3.1.6 Analisis

Dalam analisis sistem akan dibahas mengenai apa saja yang dibutuhkan dalam sistem aplikasi yang akan dibuat seperti analisis kebutuhan non fungsional, analisis perangkat lunak, analisis perangkat lunak untuk pengembang dan pengunjung, analisis perangkat keras pengembang dan pengunjung, analisis pengunjung, analisis fungsional, analisis kebutuhan sistem dan konten, pemodelan sistem, jaringan semantik dan perancangan method.

3.1.7 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional menggambarkan kebutuhan sistem yang menitik-beratkan pada properti perilaku yang dimiliki oleh sistem, diantaranya kebutuhan perangkat keras, perangkat lunak, serta pengunjung sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan.

3.1.7.1 Analisis Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan kumpulan perintah-perintah yang diberikan kepada perangkat keras agar saling berinteraksi untuk melakukan suatu tugas. Dalam analisis perangkat lunak ini dibagi menjadi 2 (dua) yaitu analisis perangkat lunak untuk pengembang dan analisis perangkat lunak untuk pengunjung.

3.1.7.1.1 Analisis Perangkat Lunak untuk Pengembang

Perangkat lunak yang dibutuhkan untuk membangun aplikasi Recognition untuk pengembang adalah sebagai berikut :

1) Sistem operasi Windows 7 Ultimate
2) Unity 4.2.1
3) Vuforia sdk
4) Autodesk 3DMax
5) Windows Movie Maker

3.1.7.1.2 Analisis Perangkat Lunak untuk Pengunjung

Perangkat lunak yang dibutuhkan untuk membangun aplikasi Recognition untuk pengunjung adalah sebagai berikut :

1) Sistem Operasi Android
2) *Sistem rekognisi.apk*

3.1.7.2 Analisis Perangkat Keras

Alat peraga berupa mobile dan camera terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat lunak memberikan instruksi - instruksi kepada perangkat keras untuk melakukan tugas tertentu. Dalam analisis perangkat lunak ini dibagi menjadi 2 (dua) yaitu analisis perangkat lunak untuk pengembang dan analisis perangkat lunak untuk pengunjung.

3.1.7.2.1 Analisis Perangkat Keras untuk Pengembang

Dalam pembangunan aplikasi Recognition, perangkat keras yang digunakan untuk pengembang adalah sebagai berikut :

1) Processor Intel Core i5 2430M 2,4Ghz
2) RAM 4GB
3) Harddisk 750GB
4) Mouse dan Keyboard
5) Webcam 1.3 Mega Pixel
6) Printer Canon IP2770
7) Handphone berbasis Android

3.1.7.2.2 Analisis Perangkat Keras untuk Pengunjung

Dalam pembangunan aplikasi Recognition, perangkat keras yang digunakan untuk pengunjung adalah sebagai berikut :

1) Smartphone pengunjung
2) Koleksi Batuan di Museum Geologi

3.1.7.3 Analisis Pengunjung

Aplikasi deteksi bebatuan ini diakses oleh satu pengunjung saja (*Single User*). Dimana pengunjung tersebut dapat menjalankan aplikasi ini dengan menggunakan *mobile* berbasis android lalu cameranya diarahkan ke media *target* batuan yang sudah ditentukan. Secara umum alur sistem aplikasi yang dibuat adalah sebagai berikut :

1. Pengunjung mengarahkan smartphone kearah koleksi batuan untuk ditampilkan informasinya.
2. Ketika user mengarahkan kamera ke koleksi batuan yang akan ditampilkan informasinya, secara otomatis kamera akan melacak *target* yang sudah diregistrasi tersebut dan kemudian akan memunculkan informasi batuan.

3.1.8 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan. Analisis kebutuhan fungsional ini meliputi analisis kebutuhan sistem, analisis kebutuhan konten, dan pemodelan sistem.

3.1.8.1 Analisis Kebutuhan Sistem

Dalam pembangunan aplikasi perancangan sistem, sebelumnya dilakukan analisis kebutuhan sistem agar sistem yang dibangun sesuai dengan yang diharapkan atau dibutuhkan pengguna. Adapun kebutuhan sistem yang diperlukan antara lain :

1. Menampilkan informasi berita animasi atau video pada display.
2. Informasi bebatuan animasi atau video yang diinformasikan jelas.
3. Memberikan informasi visual mengenai batuan yang terkait dengan tampilan berupa video dan animasi 3D.
4. Informasi mengenai batuan animasi atau video yang sudah diterapkan teknologi sistem rekognisi segera ditampilkan.

3.1.8.2 Analisis Kebutuhan Konten

Analisis ini bertujuan untuk menganalisis konten yang diperlukan agar memudahkan perancangan informasi setelah sistem yang dibutuhkan telah diketahui. Adapun analisis kebutuhan konten dengan kriteria sebagai berikut :

1. Konten Video

   Menampilkan konten video berupa video informasi yang sesuai dengan batuan.

2. Konten Animasi

   Menampilkan konten animasi berupa objek 3D.

3.1.8.3 Pemodelan Sistem

Pemodelan sistem berfungsi untuk menentukan fungsi-fungsi yang dapat dilakukan oleh sistem pada aplikasi serta menentukan kelas yang dibutuhkan untuk realisasi fungsi-fungsi sistem yang telah dianalisis sebelumnya dan mendeskripsikannya kedalam bentuk diagram.

1. ***Use Case Diagram***

Digunakan untuk memodelkan atau menggambarkan batasan sistem dan fungsi-fungsi utamanya. Mendiskripsikan fungsi dari sebuah sistem dari perspektif pengguna, *use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebauh sistem dipakai. *Use case* terdiri dari tiga bagian yaitu identifikasi aktor, identifikasi *use case,* dan *scenario use case.*

1.1 Identifikasi Aktor

Aktor adalah abstraksi dari orang dan sistem yang lain yang mengaktifkan fungsi dari target sistem. Berikut adalah aktor yang berperan dalam menjalankan sistem yang dibangun.

Tabel 3. 1 *Actor Use Case*

Actor	Deskripsi
Pengunjung	Merupakan actor atau pengguna akhir dari aplikasi yang dibangun, (*Sistem rekognisi* deteksi Batuan)
Koleksi Batuan	Merupakan aktor sebagai objek target yaitu koleksi batuan di Museum Geologi
Kamera	Merupakan actor sebagai pengambil gambar dari aplikasi yang dibangun, (*Sistem rekognisi* deteksi Batuan)

Use case diagram menunjukkan bagaimana interaksi antara aktor dan sistem, dimana terdapat aktor yang berperan dan sistem yang terdiri dari aplikasi. Aktor pengunjung yaitu yang menjalankan Aplikasi Sistem Rekognisi deteksi Batuan.

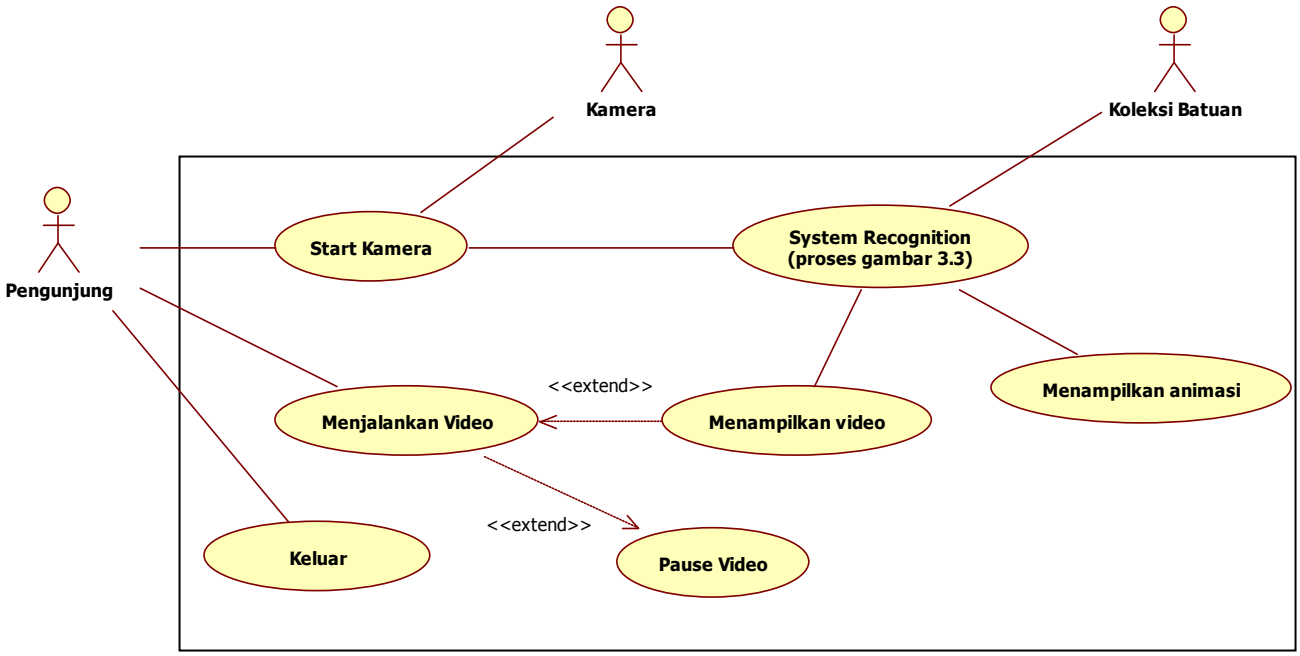
1.2 *Use Case Diagram*

Berikut spesifikasi dari *use case diagram.*

Nama : Aplikasi Sistem Rekognisi untuk mendeteksi batuan

Aktor : Pengunjung, koleksi batuan dan kamera

Pre Kondisi : Koleksi Batuan belum terdeteksi



Gambar 3.11 *Use Case Diagram*

1.3 Identifikasi *Use Case Diagram*

Berikut tabel identifikasi *use case* yang terdapat dalam sistem :

Tabel 3. 2 Identifikasi *Use Case Diagram*

***Use Case* Aplikasi Sistem Rekognisi Deteksi Batuan**		
No	*Use Case*	Deskripsi

UC-01	Start Kamera	Fungsionalitas menjalankan kamera pada *smartphone* android
UC-02	Sistem rekognisi	Fungsionalitas untuk pencocokan pola marker yang ditangkap dengan marker yang sudah terdaftar di vuforia sebagai marker
UC-03	Menampilkan Animasi	Fungsionalitas untuk melihat animasi 3D batuan tertentu.
UC-04	Menampilkan Video	Fungsionalitas untuk melihat video.
UC-05	Menjalankan Video	Fungsionalitas untuk menjalankan video yang akan diputar.
UC-06	Pause Video	Fungsionalitas untuk memberhentikan sementara video yang sedang diputar
UC-07	Keluar	Fungsionalitas untuk keluar dari aplikasi sistem rekognisi deteksi batuan

1.4 Skenario *Use Case*

Skenario proses-proses yang terdapat dalam use case diagram Aplikasi Sistem Rekognisi deteksi Batuan dapat dilihat pada tabel berikut :

Tabel 3. 3 Skenario *Use Case* Start Kamera

Identifikasi	
Nama *Use Case*	Start Kamera
Aktor	Pengunjung, kamera
Tujuan	Menjalankan kamera di *smartphone* Android
Keadaan Awal	Sistem menampilkan halaman dialog start kamera
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Pengunjung meng-klik tombol	

start	
2. Kamera mengambil gambar	
	3. Sistem menjalankan sistem rekognisi
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
1. Pengunjung membuka aplikasi ulang	
2. Pengunjung meng-klik tombol start	
3. Kamera mengambil gambar	
Kondisi Akhir	4. Sistem menjalankan system recogniton

Tabel 3. 4 Skenario *Use Case* Sistem rekognisi

Identifikasi	
Nama *Use Case*	Sistem rekognisi
Aktor	Koleksi Batuan,pengunjung, kamera
Tujuan	Mendeteksi pola (*pattern recognition*) dari koleksi batuan
Keadaan Awal	koleksi batuan tidak terdeteksi
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Pengunjung mengarahkan kamera smartphone ke Koleksi Batuan	
	2. Sistem akan menginisialisasi model 3D dan marker yang digunakan
	3. Sistem melakukan tracking marker
	4. Mengambil video dari kamera

	5. Sistem melakukan proses *thresholding*
	6.Sistem melakukan pendeteksian pelacakan marker
	7. Sistem melakukan pencocokan pola dengan *pattern recognition*
	8. Rendering objek 3D
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
1. Pengunjung mengarahkan *smartphone* ke koleksi batuan	
	2. Sistem akan menginisialisasi model 3D dan koleksi batuan yang digunakan
	3. Sistem melakukan tracking marker
	4. Mengambil video dari kamera
	5. Sistem melakukan proses *thresholding*
	6.Sistem melakukan pendeteksian pelacakan marker
	7. Sistem melakukan pencocokan pola dengan *pattern recognition*
Kondisi Akhir	8. Rendering objek 3D

Tabel 3. 5 Skenario Menampilkan Video

Identifikasi	
Nama *Use Case*	Menampilkan Video
Aktor	Koleksi Batuan
Tujuan	Menampilkan videoplay
Keadaan Awal	Sistem belum menampilkan Video
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Pengunjung mengarahkan *smartphone*	

ke koleksi batuan	
	2. Menampilkan videoplay yang belum di-play
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
1, Pengunjung mengarahkan *smartphone* ke koleksi batuan	
Kondisi Akhir	2. Menampilkan video yang belum di-play

Tabel 3. 6 Skenario Menjalankan Video

Identifikasi	
Nama *Use Case*	Menjalankan Video
Aktor	Pengunjung
Tujuan	Memulai videoplay
Keadaan Awal	Sistem menampilkan tombol play
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan video yang belum di-play
2. menekan tombol play pada *smartphone*	
	3. Setelah menekan tombol play maka sistem akan memulai video mengenai bebatuan sesuai marker.
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan video yang belum di-play
2. Menekan tombol play kembali pada	

display *smartphone*	
Kondisi Akhir	2. sistem akan memulai video mengenai bebatuan sesuai target.

Tabel 3. 7 Skenario *Pause* Video

Identifikasi	
Nama *Use Case*	*Pause* Video
Aktor	Pengunjung
Tujuan	Memberhentikan sementara video
Keadaan Awal	Sistem sedang menjalankan video
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
	1. Videoplay tentang batuan yang sedang di putar
2. Menekan video yang sedang berjalan	
	3. Sistem akan mem-*pause* video yang sedang diputar
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
	1. Videoplay tentang batuan yang sedang di putar
2. Menekan kembali video yang sedang diputar	
Kondisi Akhir	3. Sistem akan mem-*pause* video yang sedang diputar

Tabel 3. 8 Skenario Menampilkan Animasi

Identifikasi	
Nama *Use Case*	Menampilkan Animasi

Aktor	koleksi batuan
Tujuan	Menampilkan animasi 3D
Keadaan Awal	Sistem belum menampilkan Animasi
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Pengunjung mengarahkan *smartphone* ke koleksi batuan	
	2. Menampilkan animasi 3D batuan sesuai target.
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
1. Pengunjung mengarahkan *smartphone* ke koleksi batuan	
Kondisi Akhir	2. Menampilkan animasi 3D batuan sesuai target.

Tabel 3. 9 Skenario Keluar

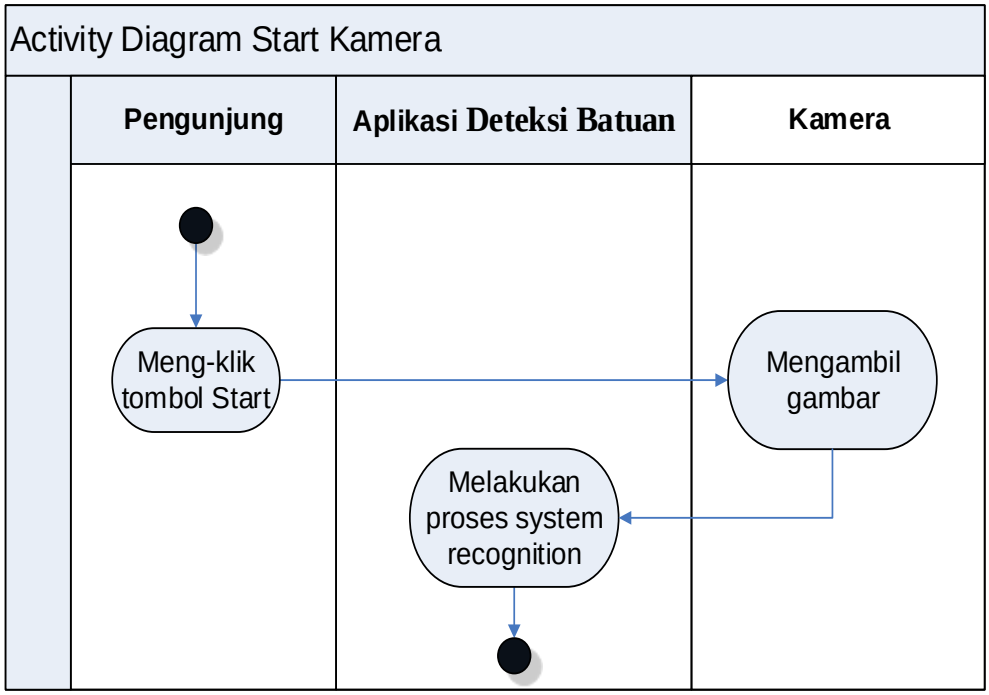
Identifikasi	
Nama *Use Case*	Keluar
Aktor	Pengunjung
Tujuan	Keluar dari aplikasi sistem rekognisi deteksi batuan
Keadaan Awal	Sistem sedang menjalankan aplikasi
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Pengunjung menekan tombol “Keluar”	
	2. Sistem akan keluar dari aplikasi

Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
1. Pengunjung menekan kembali tombol keluar	
Kondisi Akhir	2. sistem akan keluar dari aplikasi

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.

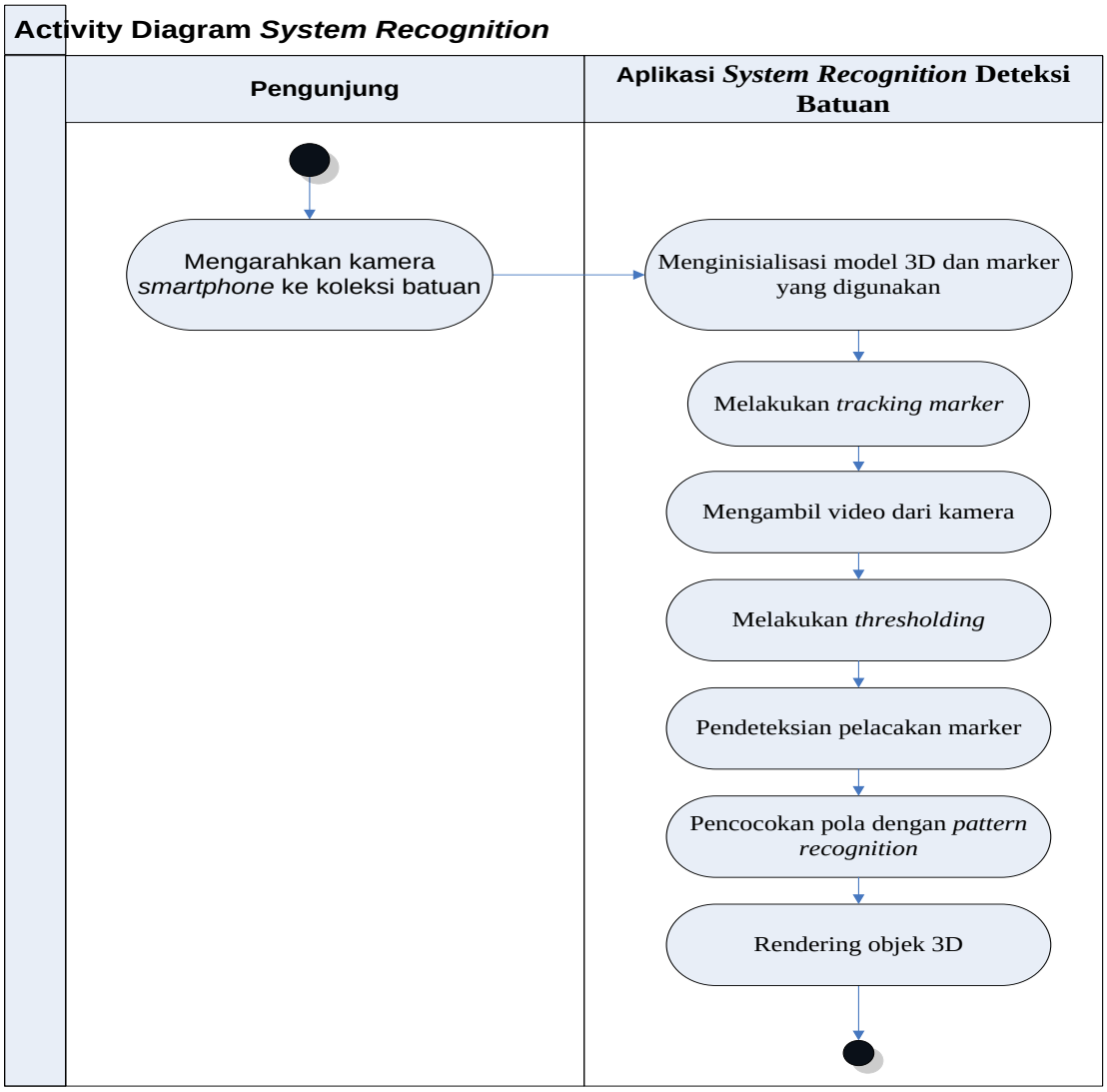
2.1 *Activity Diagram* Start Kamera



Gambar 3. 12 *Activity Diagram* Start Kamera

Gambar 3.12**,** menggambarkan aktivitas yang terjadi pada aplikasi saat pengunjung pertama kali menggunakan aplikasi. Dimana pada saat penggunaan menjalankan aplikasi pengunjung meng-klik tombol start pada tampilan dialog dari system, selanjutnya kamera akan mengambil gambar lalu system akan menjalankan proses sistem rekognisi.

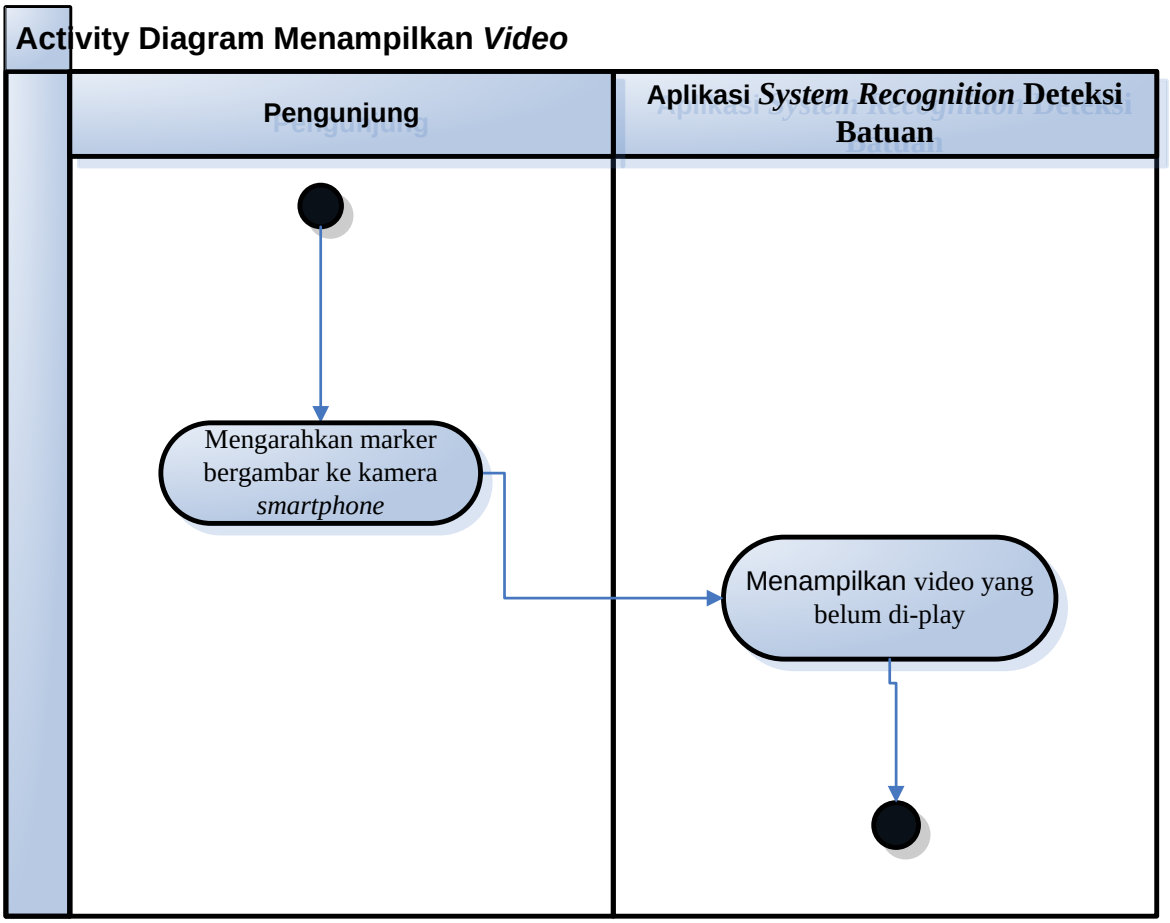
2.2 Activity Diagram Sistem rekognisi



Gambar 3. 13 ***Activity Diagram Sistem rekognisi***

Gambar 3.13**,** menggambarkan aktivitas ketika pengunjung mengarahkan kamera smartphone ke koleksi batuan. Selanjutnya sistem melakukan inisialisasi, *tracking marker*,mengambil video dari kamera, *thresholding*, pendeteksian pelacakan marker,pencocokan pola (pattern recognition) dan proses terakhir yaitu rendering objek 3D.

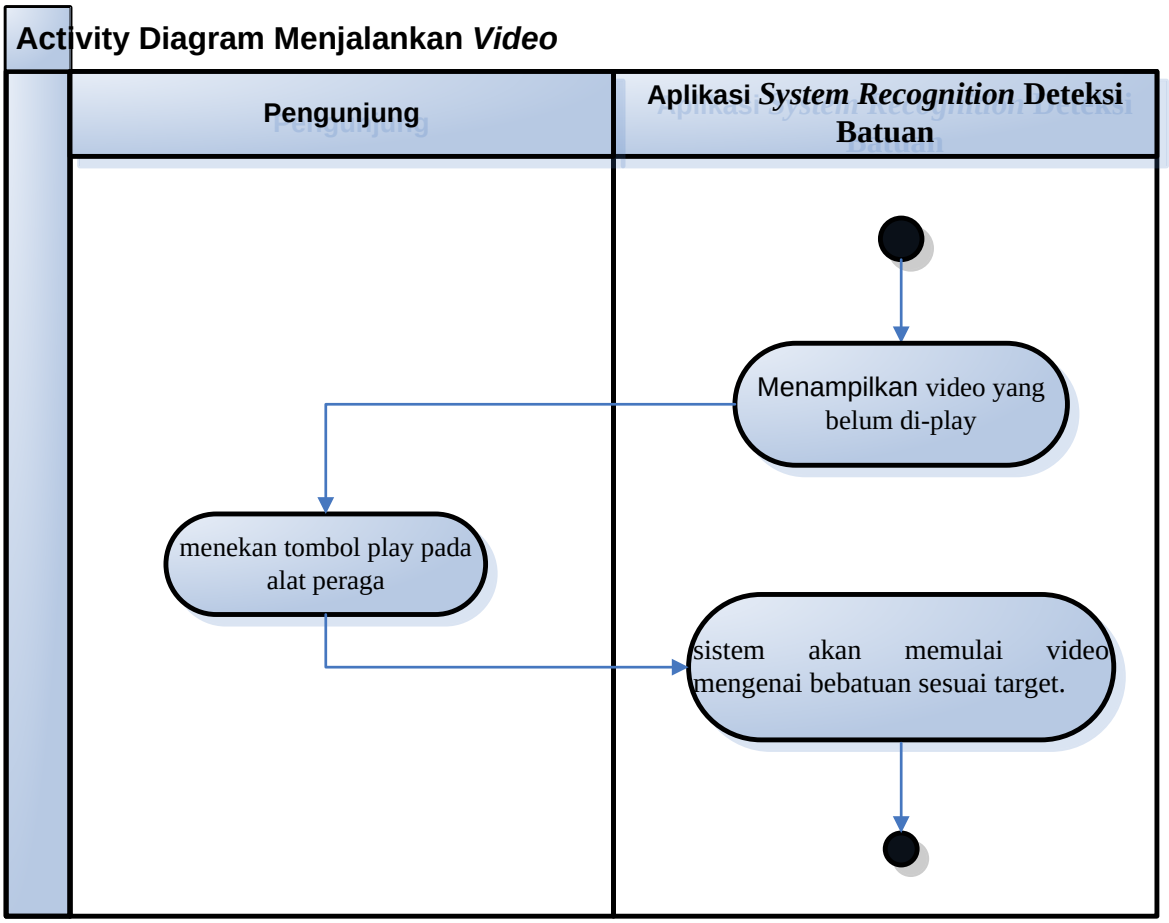
2.3 Activity Diagram Menampilkan Video



Gambar 3. 14 *Activity Diagram* Menampilkan Video

Gambar 3.14**,** menggambarkan aktivitas ketika pengunjung mengarahkan kamera smartphone ke koleksi batuan. Sistem menampilkan video yang belum di-play.

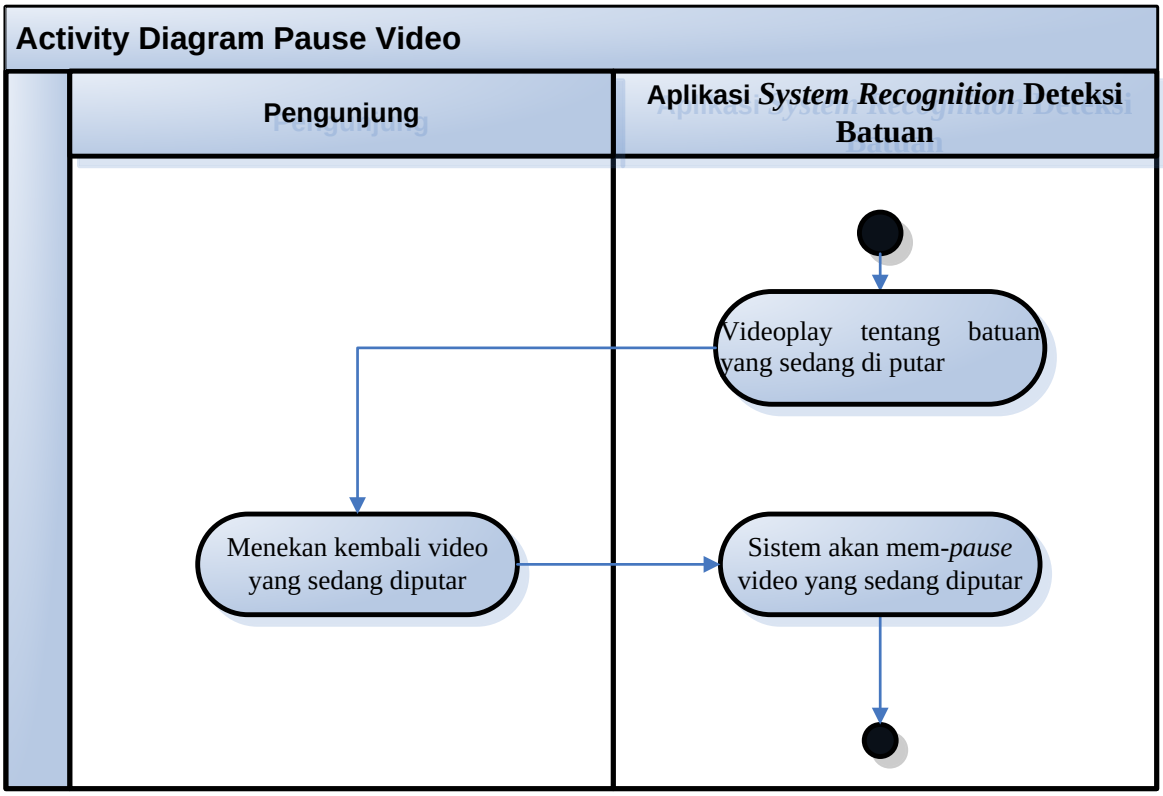
2.4 Activity Diagram Menjalankan Video



Gambar 3. 15 *Activity Diagram* Menjalankan Video

Gambar 3.15**,** menggambarkan aktivitas ketika pengunjung menekan *icon play* video. Sistem menampilkan video yang belum di-play, lalu pengunjung menekan tombol play pada *smartphone*. Selanjutnya sistem akan memulai video mengenai bebatuan sesuai target.

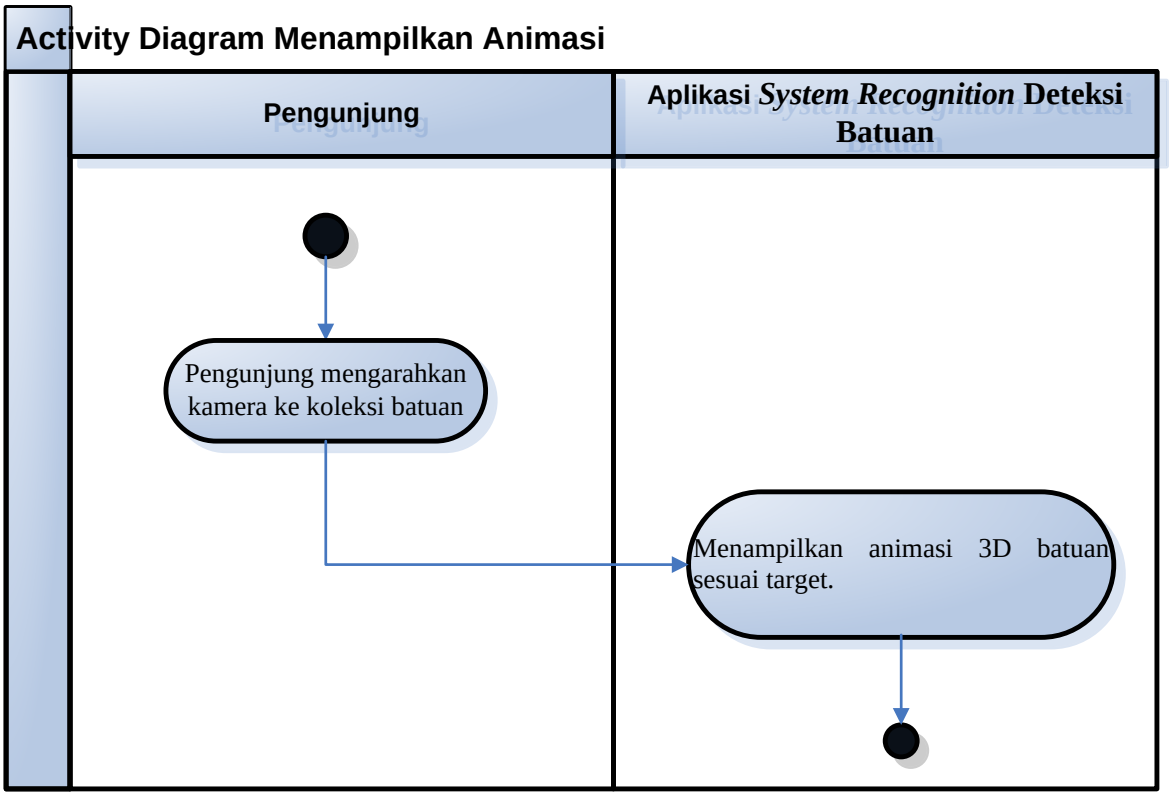
2.5 *Activity Diagram* Pause Video



Gambar 3. 16 *Activity Diagram Pause* Video

Gambar 3.16**,** menggambarkan aktivitas ketika pengunjung ingin mem-*pause* video. Sistem menampilkan video tentang batuan yang sedang diputar, lalu pengunjung menekan video yang sedang diputar pada smartphone, selanjutnya sistem akan mem-*pause* video mengenai bebatuan sesuai jenis batuan.

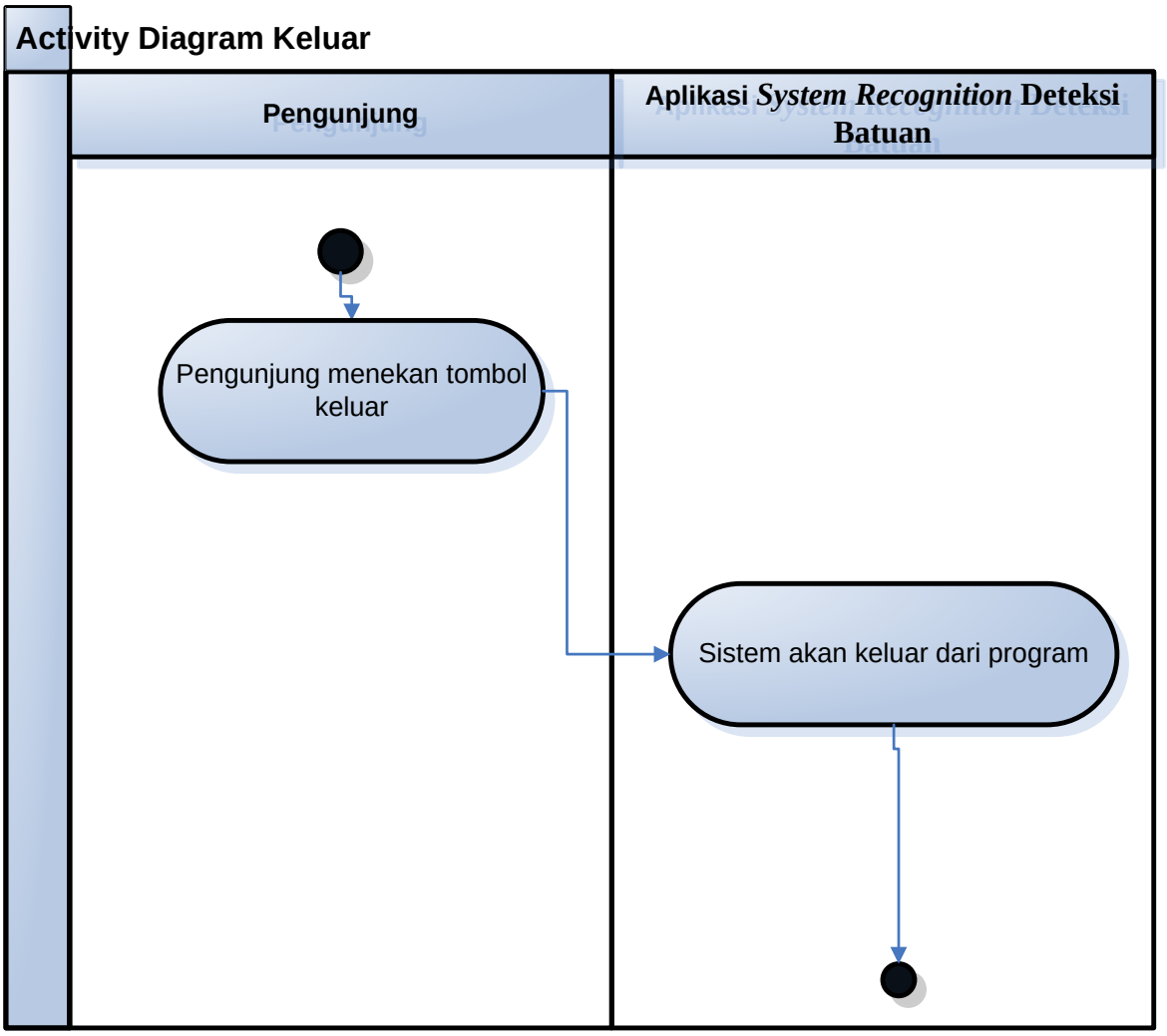
2.6 *Activity Diagram* Menampilkan Animasi



Gambar 3. 17 *Activity Diagram* Animasi

Gambar 3.17**,** menggambarkan aktivitas ketika pengunjung akan melihat tampilan animasi. Pengunjung mengarahkan kamera ke koleksi batuan, selanjutnya sistem akan menampilkan animasi 3D mengenai bebatuan sesuai target.

2.7 Activity Diagram Keluar



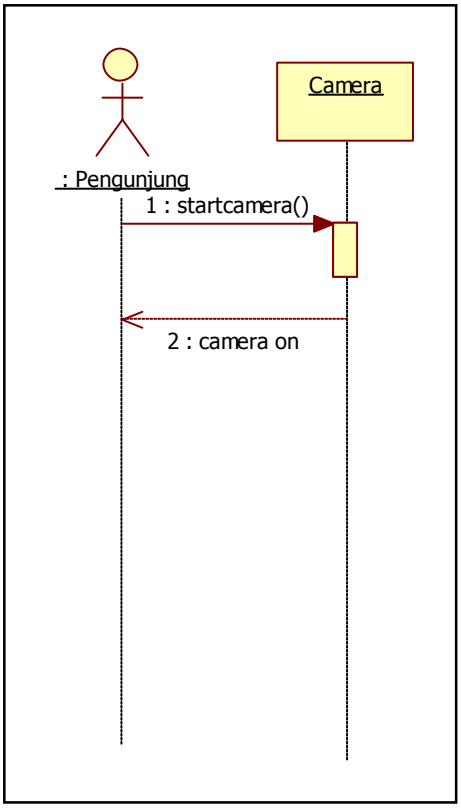
Gambar 3. 18 *Activity Diagram* Keluar

Gambar 3.18**,** menggambarkan aktivitas ketika Pengunjung menekan tombol keluar. Lalu sistem akan keluar dari aplikasi sistem rekognisi deteksi bebatuan.

3. *Sequence Diagram*

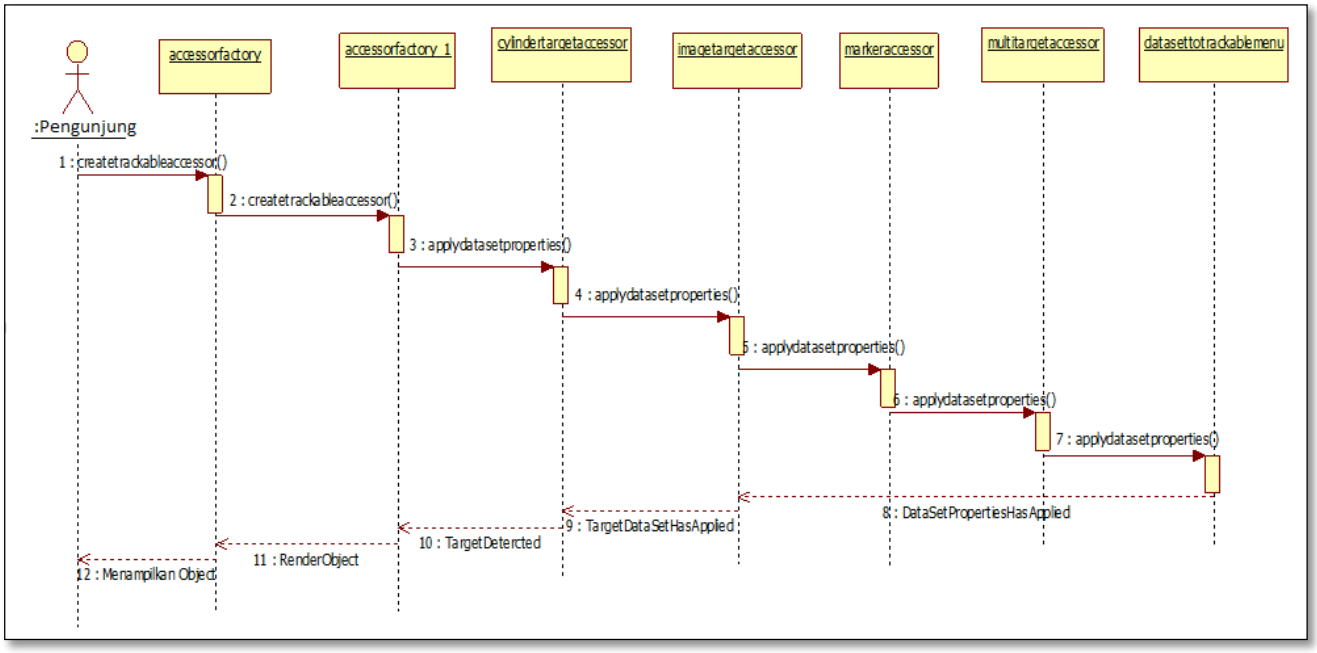
Sebuah diagram yang menggambarkan interaksi antar objek di dalam sebuah system. Interaksi tersebut berupa message yang digambarkan terhadap waktu. *Sequence diagram* terdiri dari dimensi horizontal (objek-objek) dan dimensi vertical (waktu). Dimensi vertikal adalah poros waktu, dimana waktu berjalan ke arah bawah, sedangkan dimensi horizontal merepresentasikan objek-objek individual. Tiap objek (termasuk *actor*) tersebut mempunyai waktu aktif yang direpresentasikan dengan kolom vertikal yang disebut dengan *lifeline*. Adapun *sequence diagram* yang digunakan dalam Aplikasi Sistem Rekognisi deteksi batuan ini adalah sebagai berikut.

3.1 Sequence Diagram Start Kamera



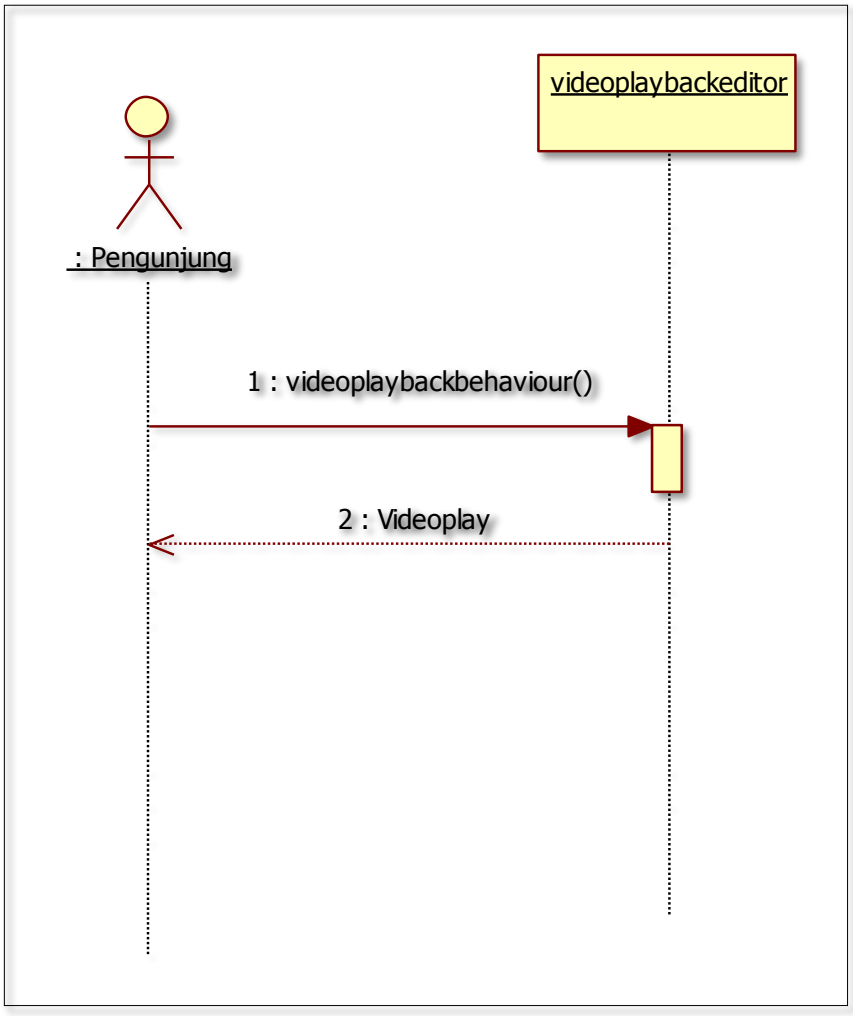
Gambar 3. 19 *Sequence Diagram Start* Kamera

3.2 Sequence Diagram Sistem rekognisi



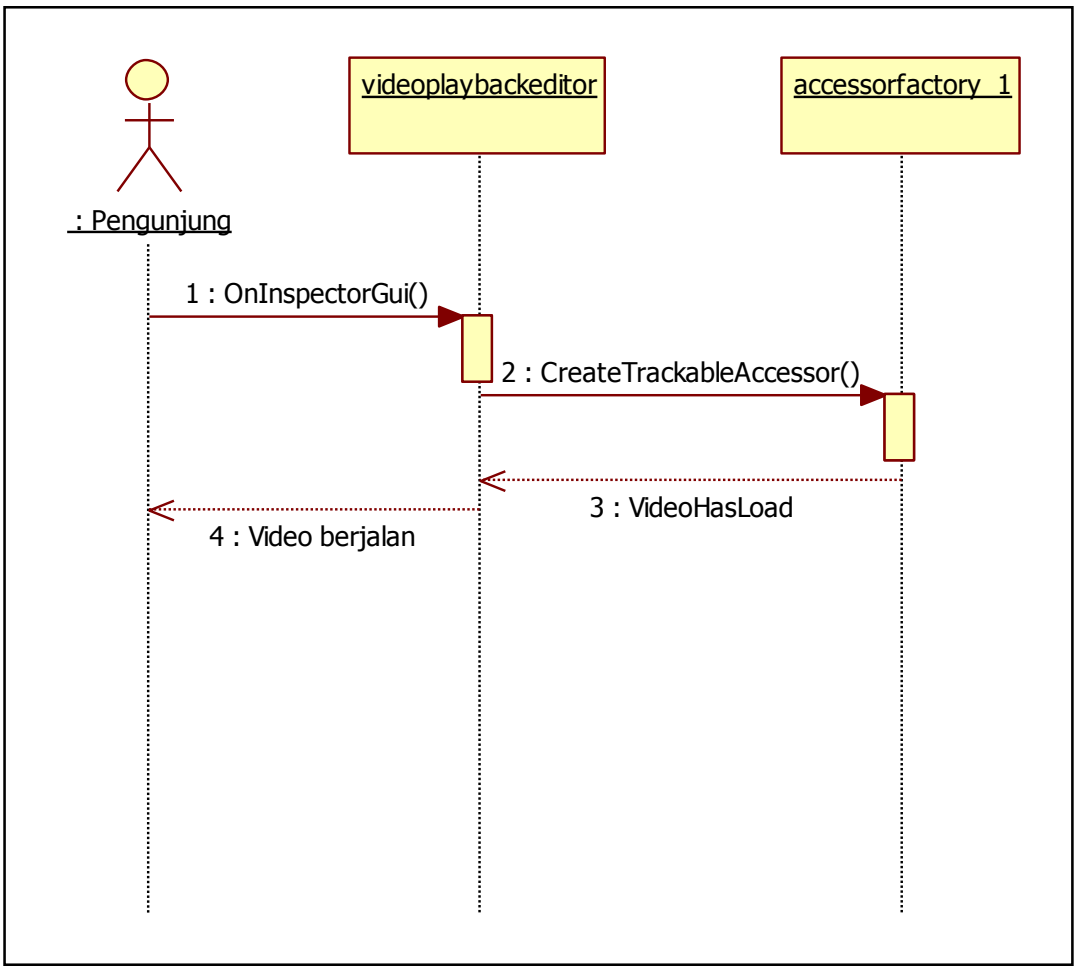
Gambar 3. 20 ***Sequence Diagram Sistem rekognisi***

3.3 Sequence Diagram Menampilkan Video



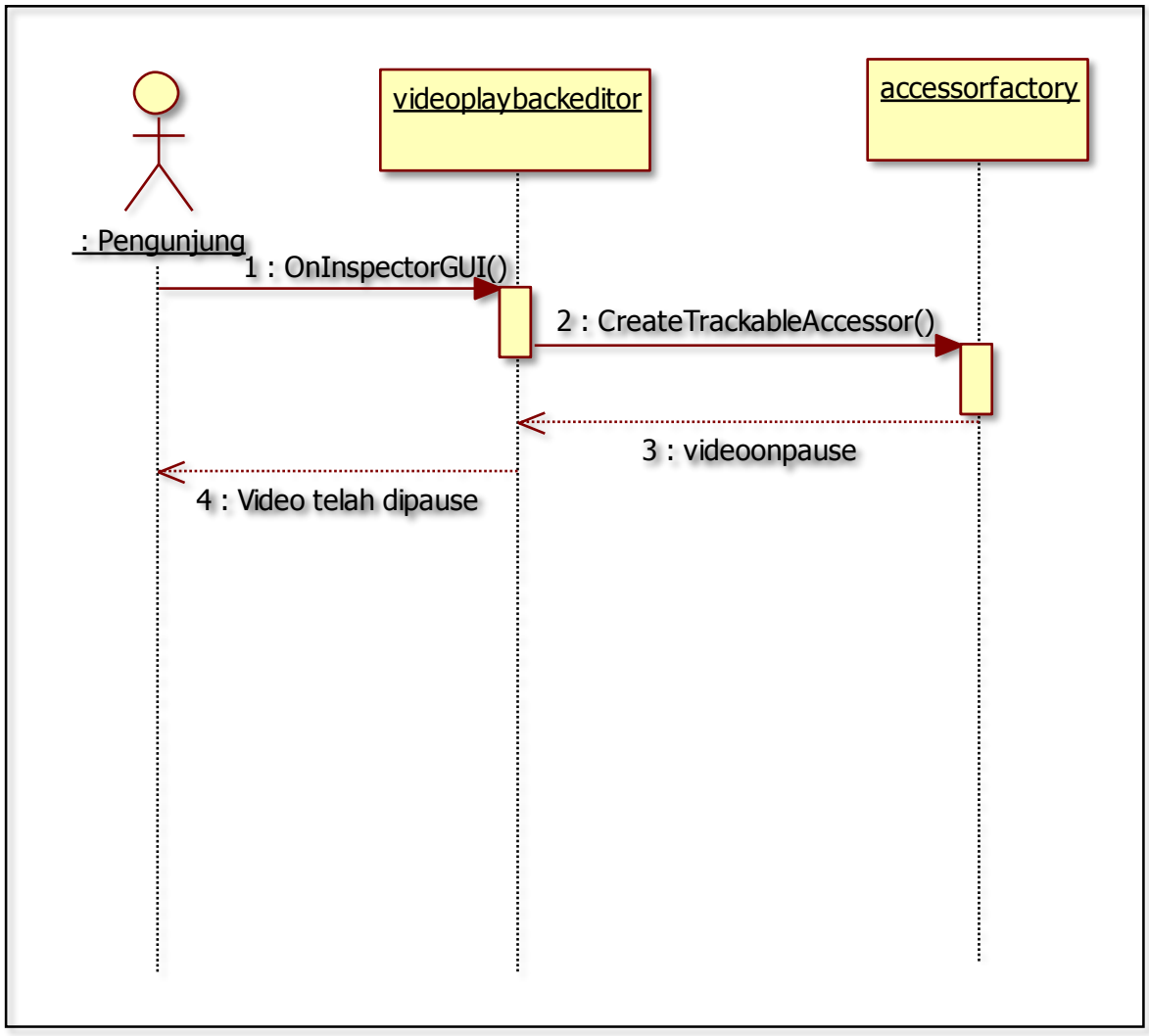
Gambar 3. 21 *Sequence Diagram* Menampilkan Video

3.4 Sequence Diagram Menjalankan Video



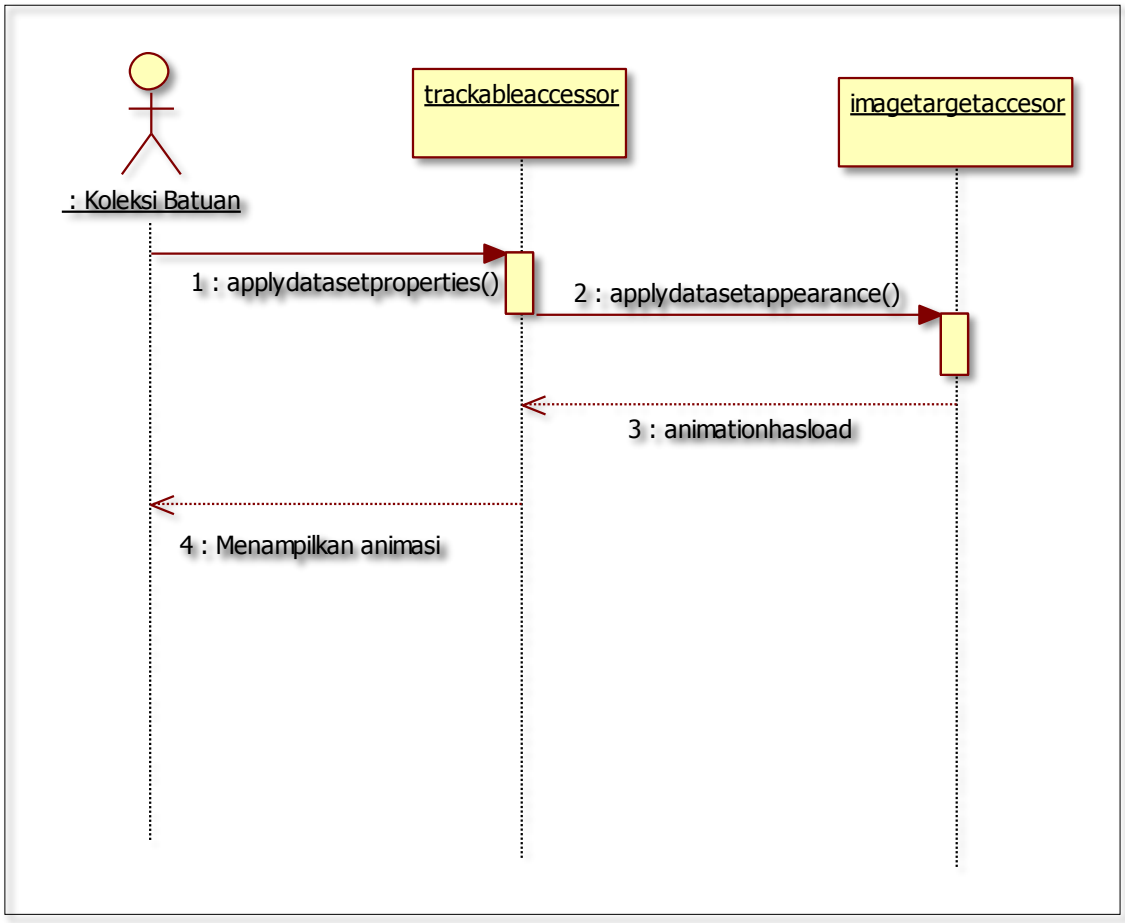
Gambar 3. 22 *Sequence Diagram* Menjalankan Video

3.5 Sequence Diagram Pause Video



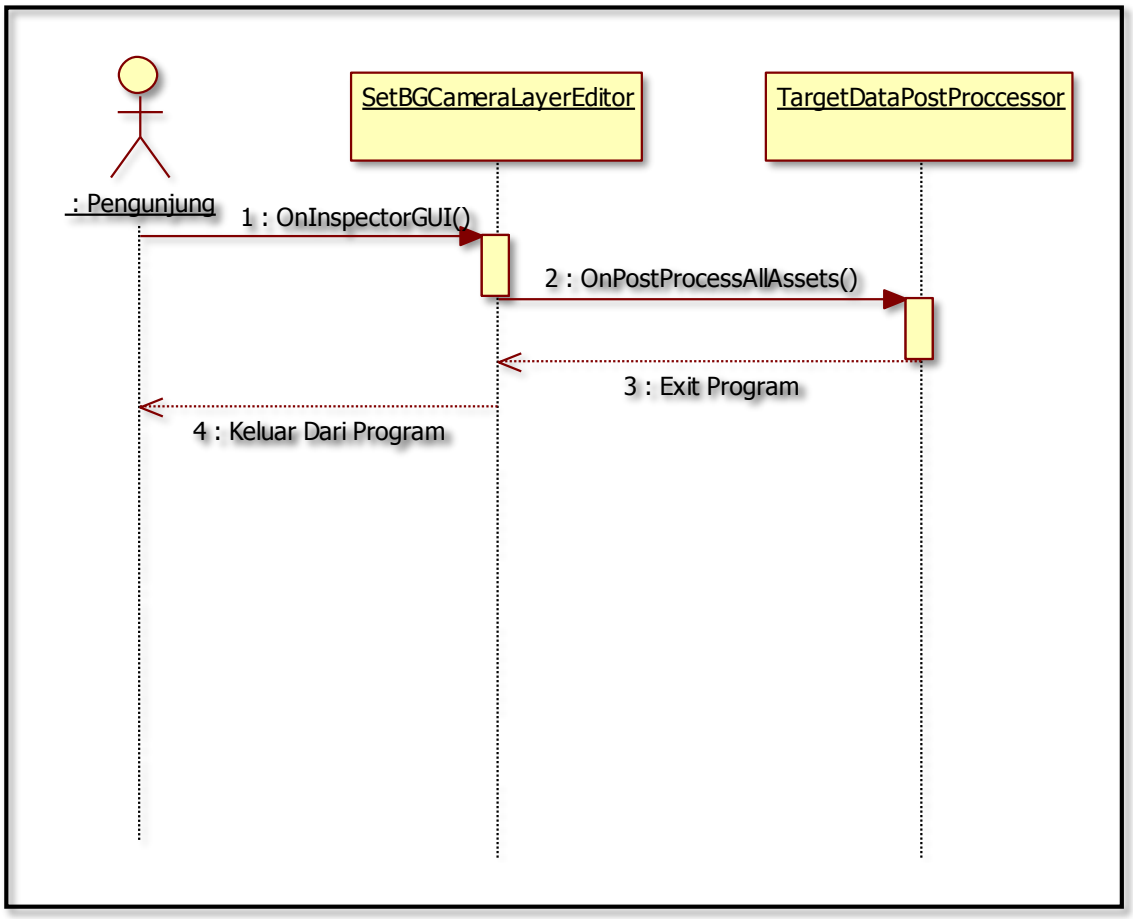
Gambar 3. 23 *Sequence Diagram Pause* Video

3.6 Sequence Diagram Menampilkan Animasi



Gambar 3. 24 *Sequence Diagram* Menampilkan Animasi

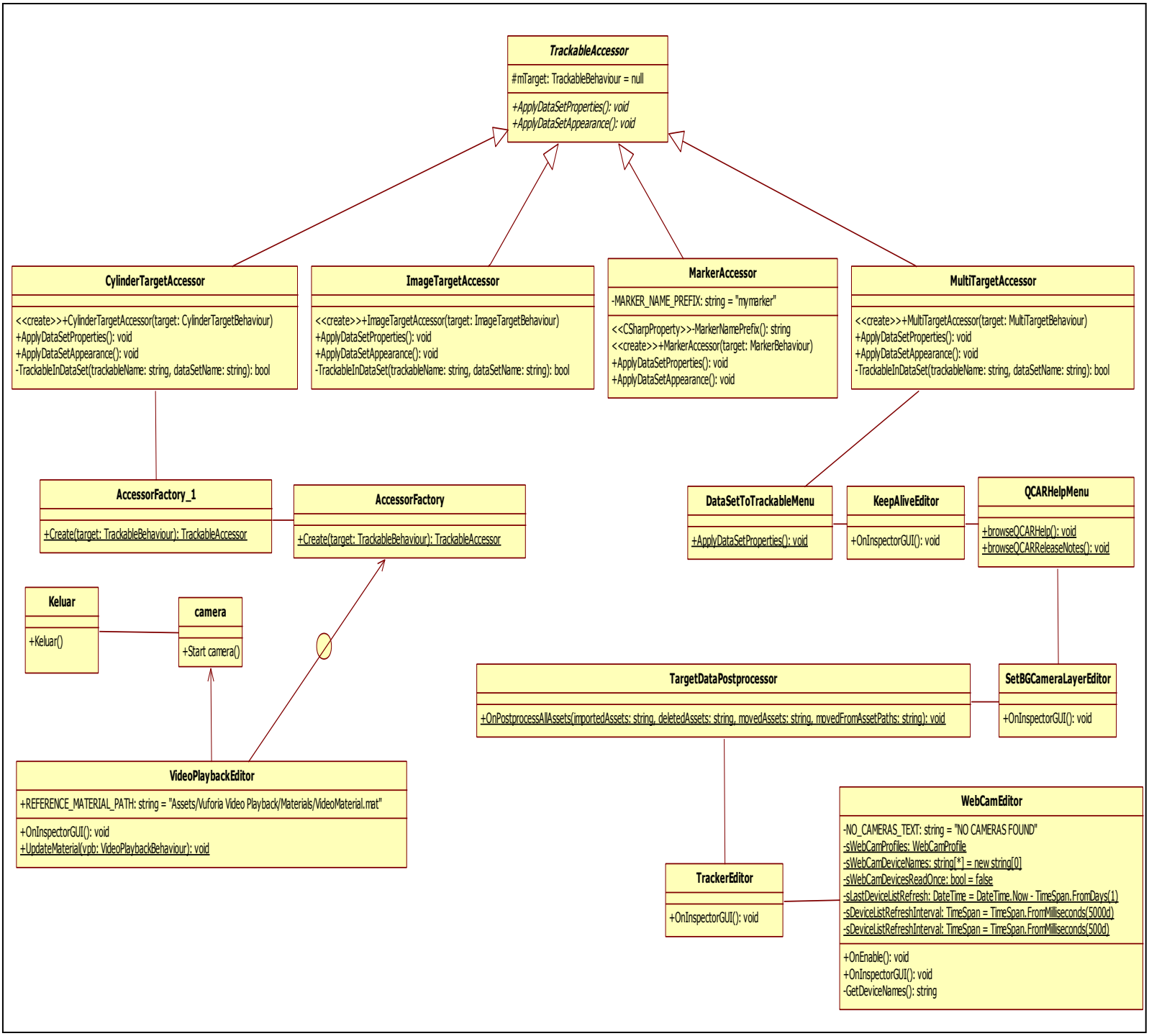
3.7 Sequence Diagram Keluar



Gambar 3. 25 *Sequence Diagram* Keluar

4. *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 sistem aplikasi sistem rekognisi Deteksi Batuan.



Gambar 3. 28 Class Diagram Aplikasi Sistem Rekognisi Deteksi Batuan

3.2 Perancangan Sistem

Perancangan akan dimulai setelah tahap analisis terhadap sistem selesai dilakukan. Perancangan dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi.

3.2.1 Perancangan Grafis dan Antarmuka

Tahap perancangan desain bertujuan untuk mencari bentuk yang optimal dari aplikasi yang akan dibangun dengan pertimbangan faktor - faktor permasalahan dan kebutuhan yang ada pada sistem seperti yang telah ditetapkan pada tahap analisis. Dalam tahap ini upaya yang dilakukan yaitu dengan cara mengkombinasikan penggunaan teknologi perangkat keras dan perangkat lunak yang tepat sehingga diperoleh desain yang optimal dan mudah di implementasikan.

3.2.1.1 Perancangan Antarmuka Aplikasi Sistem Rekognisi Deteksi Batuan

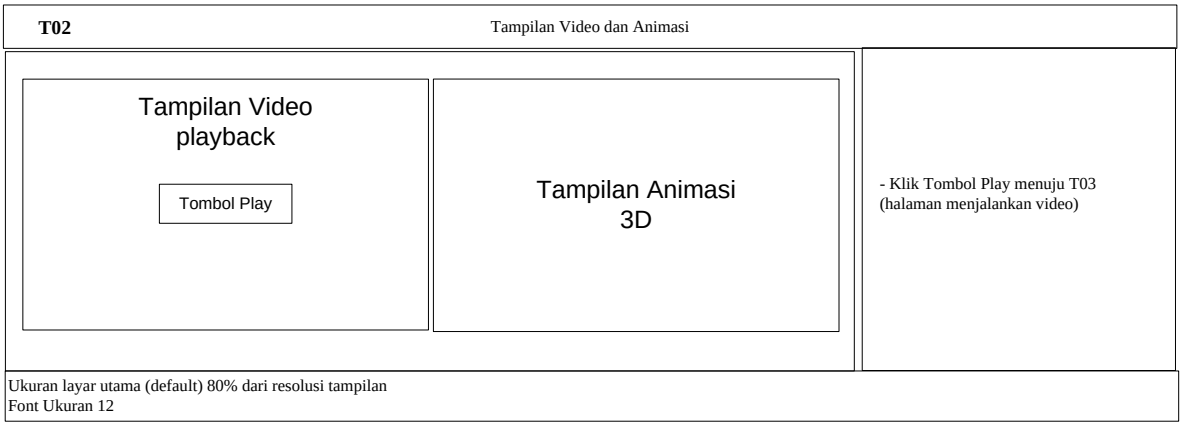
Tahap perancangan antarmuka dilakukan untuk merancang antarmuka agar dapat digunakan oleh pengguna baik admin maupun user yang menggunakan *Sistem rekognisi* Deteksi Batuan ini.

1) Form Start Camera (T01)



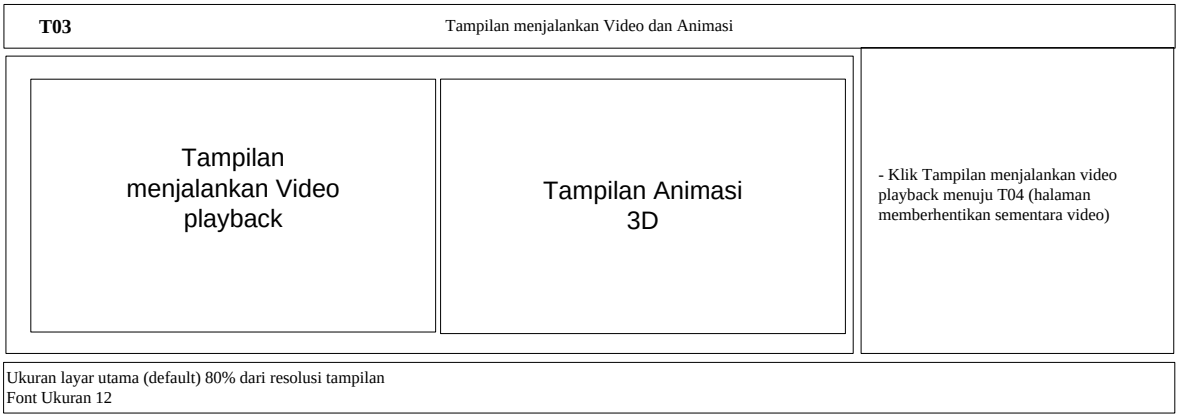
Gambar 3. 29 Form *start camera*

2) Form video dan animasi (T02)



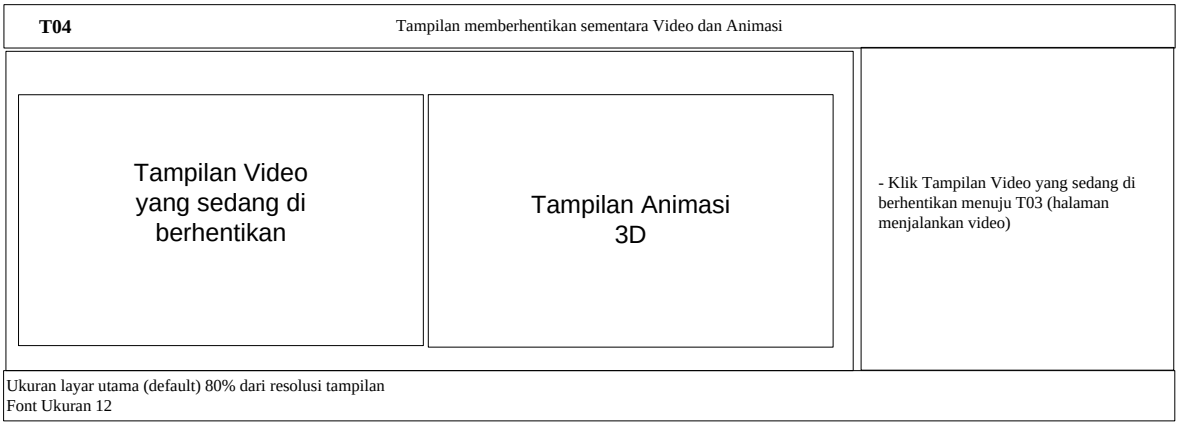
Gambar 3. 30 Form video dan animasi

3) Form menjalankan video (T03)



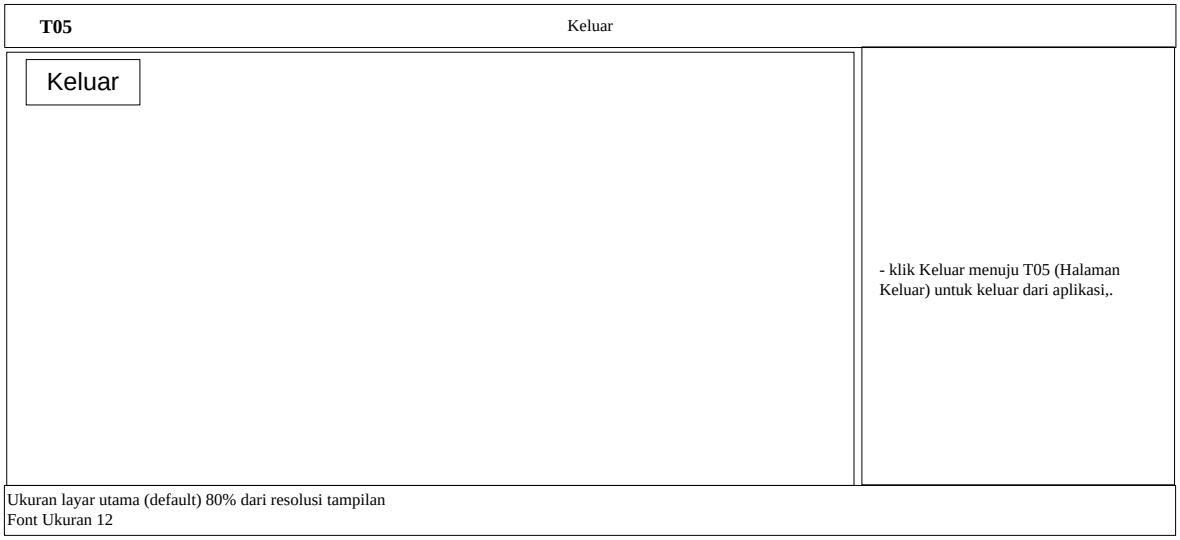
Gambar 3. 31 Form menjalankan video

4) Form memberhentikan sementara video (T04)



Gambar 3. 32 Form memberhentikan sementara video

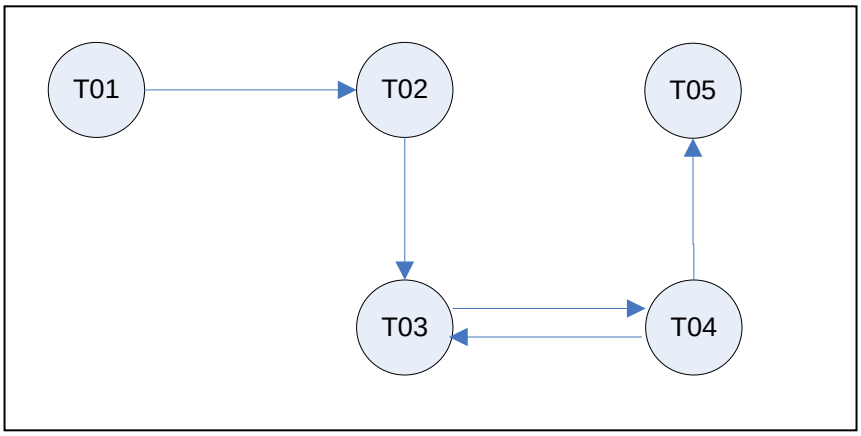
5) Form Start Camera (T05)



Gambar 3. 33 Form Keluar

3.2.1.2 Jaringan Semantik

Jaringan semantic sistem yang akan dibangun dibedakan sesuai dengan jenis penggunanya. Jaringan semantiknya dapat dilihat pada Gambar 3.34 dibawah ini :



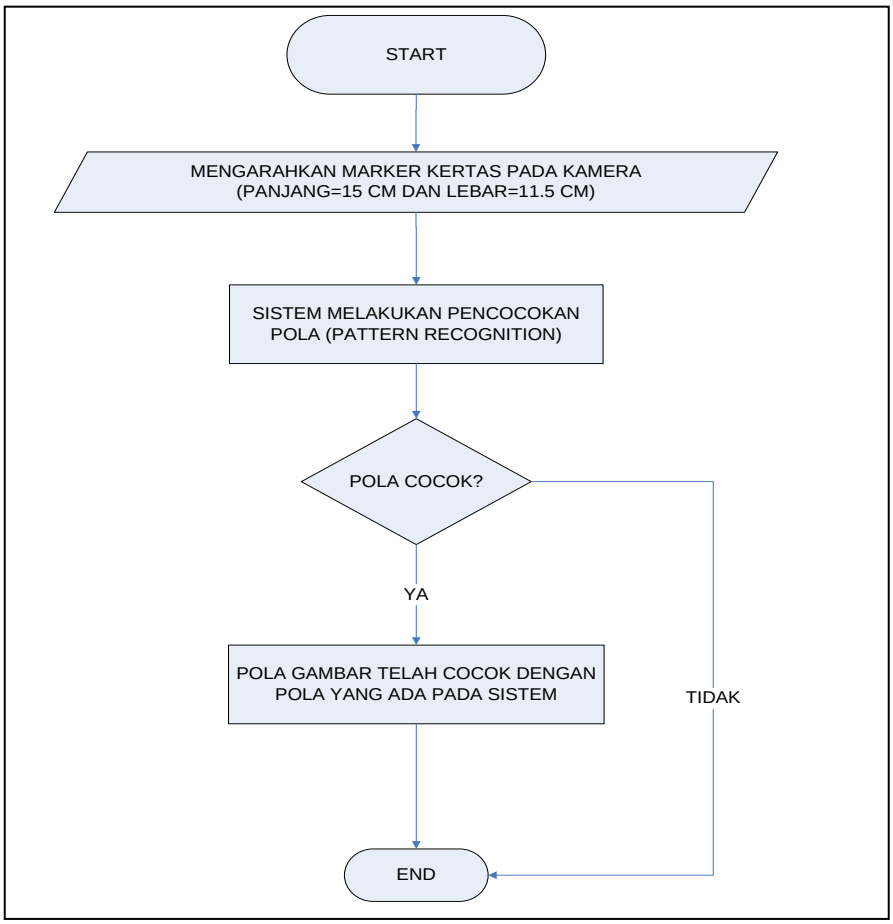
Gambar 3.34 Jaringan Semantik

3.2. 2 Perancangan Method

Perancangan komponen method merupakan perancangan yang dibuat setelah perancangan arsitektur menu, antar muka, dan jaringan semantik. Perancangan ini berfungsi untuk mendeskripsikan method-method yang berada di dalam aplikasi. Adapun method-method yang terdapat dalam Aplikasi Sistem Rekognisi yang akan dibangun adalah sebagai berikut.

A. Method apply dataset properties

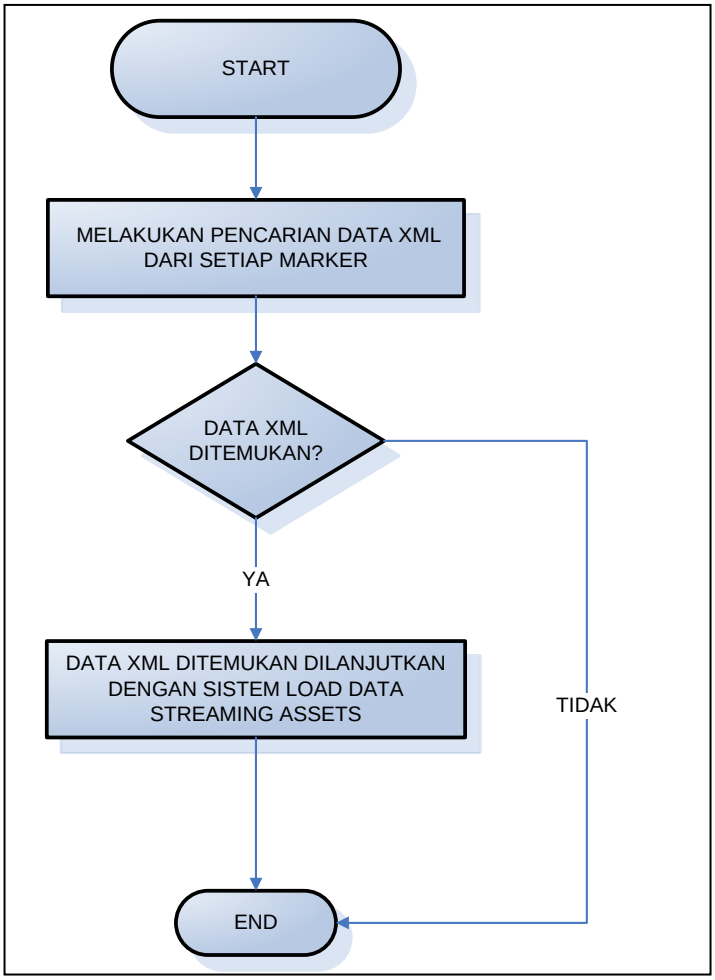
Method ini digunakan untuk mencocokan marker yang telah dibuat di Vuforia sebagai marker yang akan digunakan di Unity.



Gambar 3.35 Flowchart apply dataset properties

B. Method browseQCARHelp

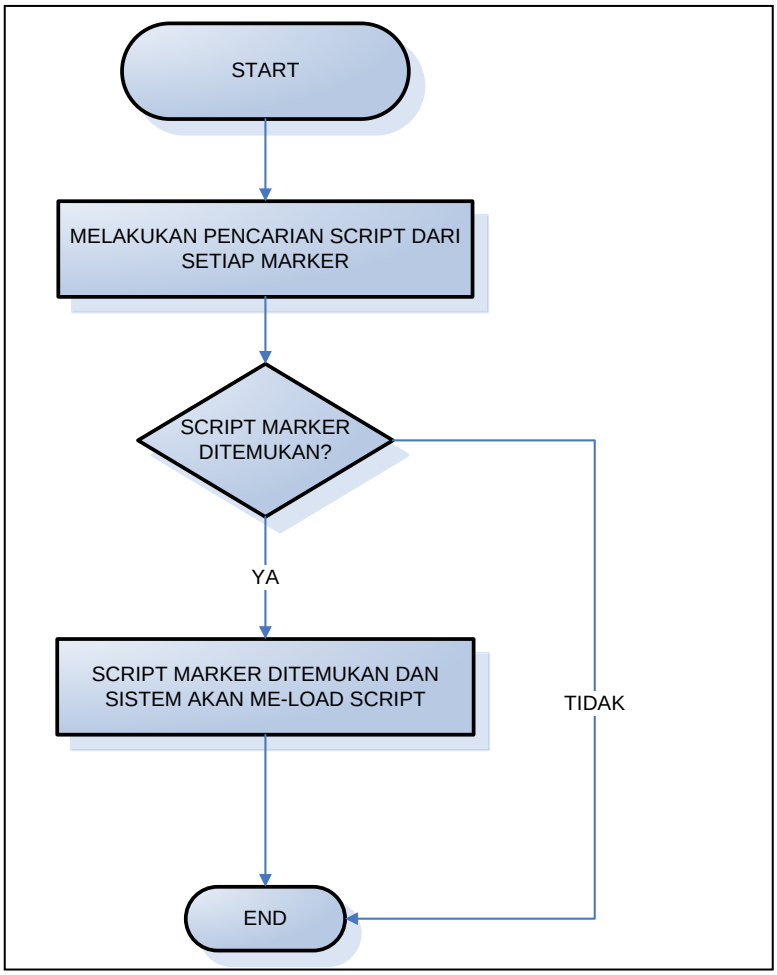
Method ini digunakan untuk melakukan pencarian script data marker yang telah di imporkan.



Gambar 3.36 Flowchart browseQCARHelp

C.Method browseQCARReleaseNotes

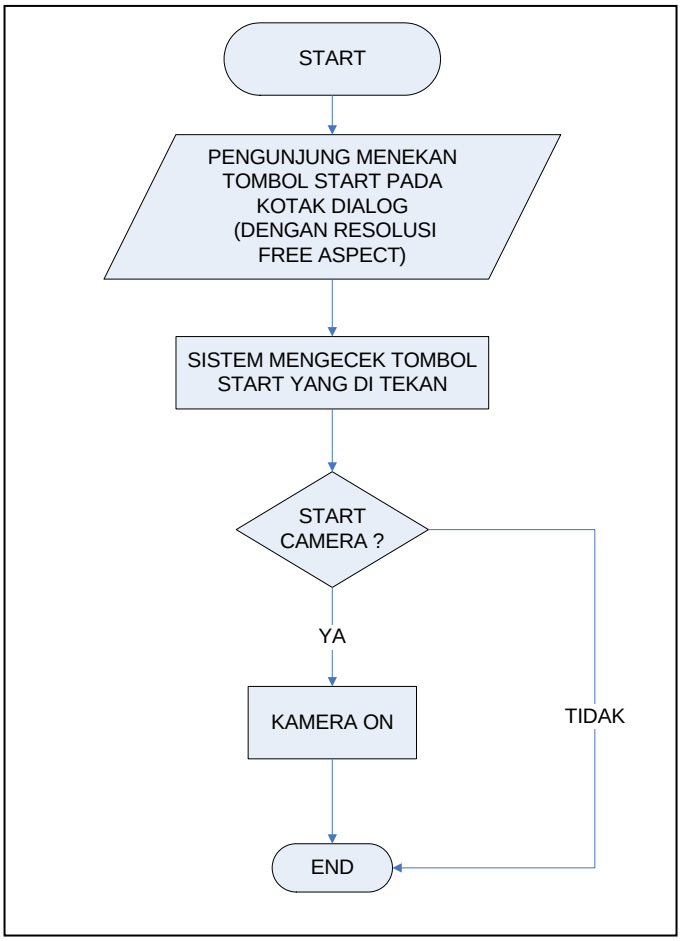
Method ini digunakan untuk *browse script* dari setiap marker yang digunakan dan akan me-*load script* tersebut.



Gambar 3.37 Flowchart browseQCARReleaseNotes

D. Method startcamera

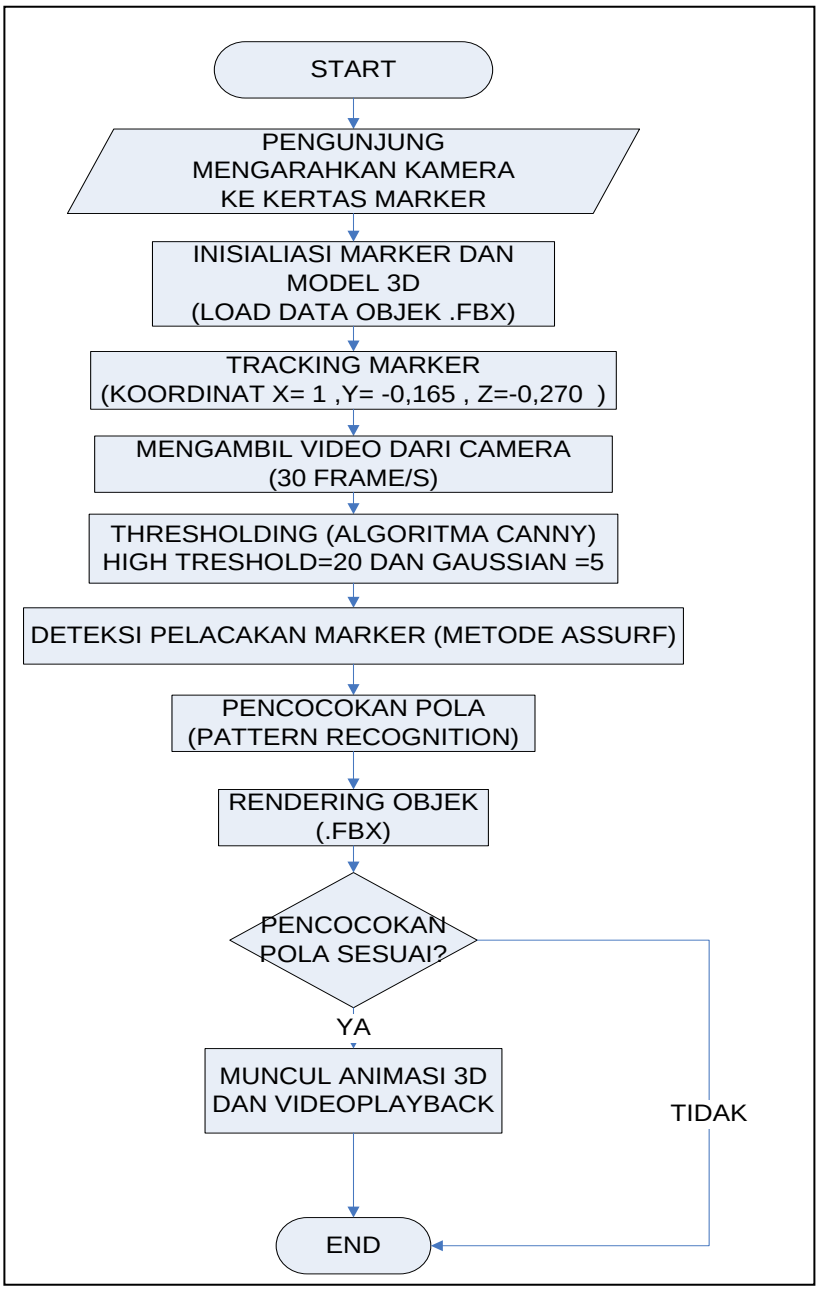
Method ini digunakan untuk mengaktifkan kamera pada alat peraga.



Gambar 3.38 Flowchart startcamera

E. Method OnPostProcessAllAssets

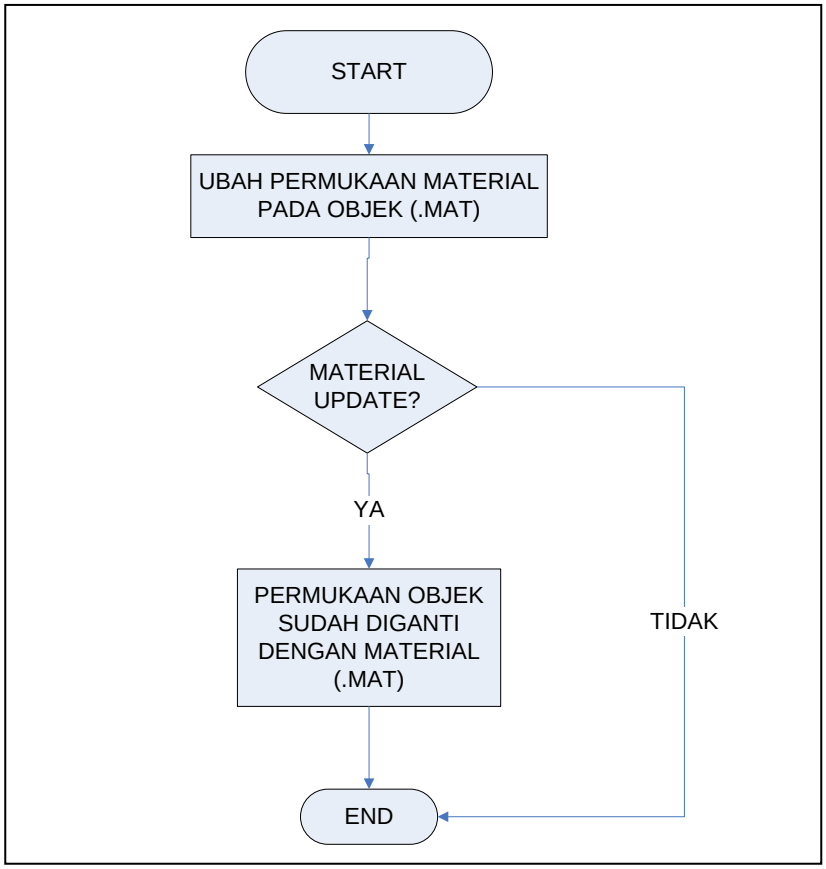
Method ini digunakan untuk mendeteksi batuan dengan sistem rekognisi (*sistem rekognisi*) dan menampilkan *videoplayback* dan animasi 3D.



Gambar 3.39 Flowchart OnPostProcessAllAssets

F. Method update material

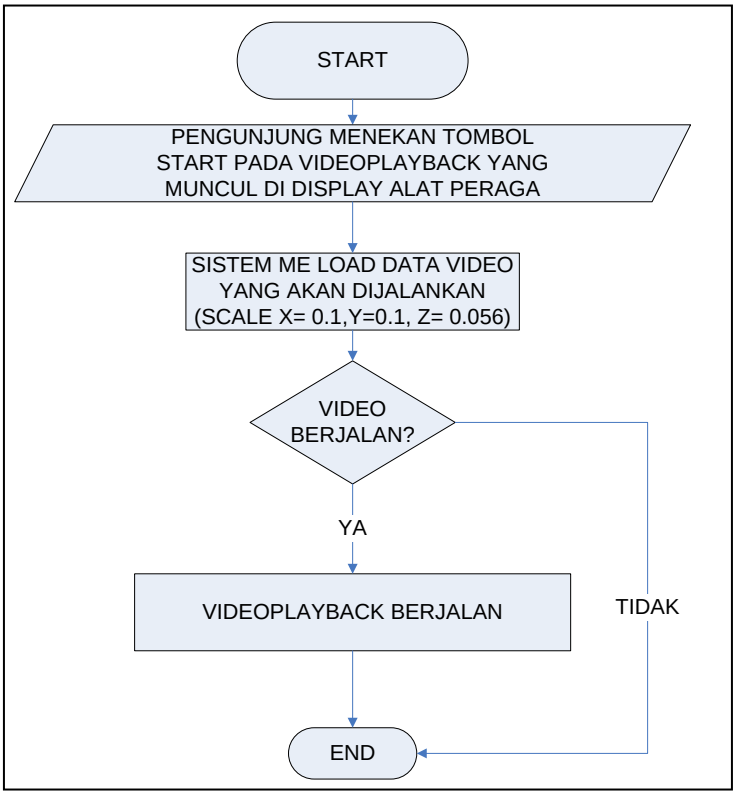
Method ini digunakan untuk merubah atau mengganti material pada objek 3D (permukaan dan tekstur pada batuan).



Gambar 3.40 Flowchart update material

G. Method OnEnable

Method ini digunakan untuk menjalankan *videoplayback*.



Gambar 3.41 Flowchart OnEnable