BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem merupakan penguraian dari suatu sistem informasi untuk mengidentifikasi masalah-masalah dan hambatan-hambatan yang terjadi pada sistem yang sedang berjalan, sehingga dapat diusulkan kebutuhan-kebutuhan untuk memperbaikinya. Tahap analisis sistem adalah tahapan dasar sebelum perancangan untuk memperbaiki sistem yang sedang berjalan, sehingga dari hasil analisis tersebut dapat dirancang atau diperbaiki sebuah sistem yang lebih efektif dan efisien.

3.1.1 Analisis Masalah

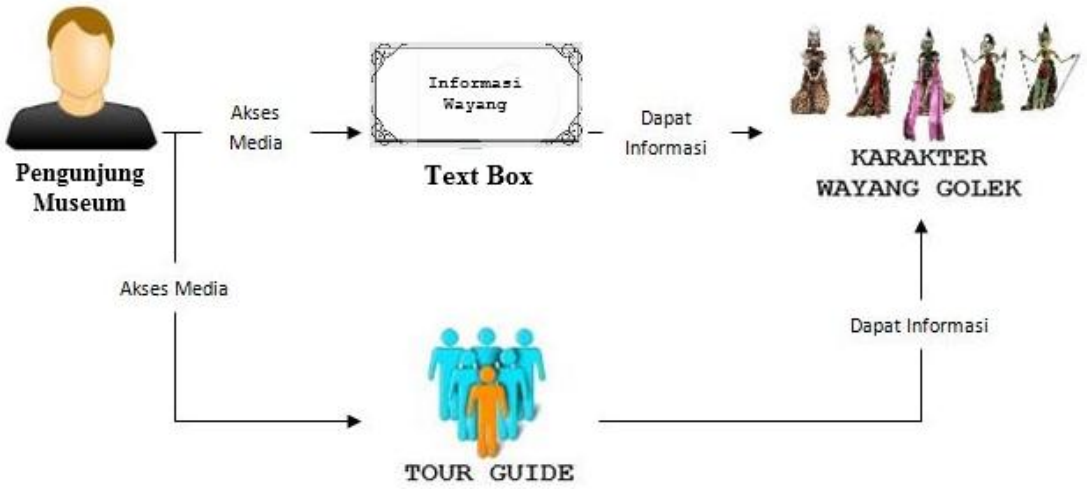
Analisis masalah adalah langkah awal dari proses analisis sistem. Langkah ini bertujuan untuk mengetahui masalah yang terjadi dalam sistem yang sedang berjalan. Analisis masalah yang dimaksud disini adalah sulit diperolehnya media untuk mendapatkan informasi tentang karakter-karakter yang ada dalam kesenian wayang golek. Beberapa media yang ada masih memiliki beberapa kekurangan baik dari segi media secara fisik maupun dari segi situasi dan kondisi yang kurang memungkinkan untuk mendapatkan informasi tentang karakter-karakter yang ada dalam kesenian wayang golek secara mudah dan cepat.

Salah satu teknologi informasi yang sedang berkembang saat ini adalah teknologi *mobile*. Kelebihan teknologi *mobile* adalah sisi ruang gerak (*mobilitas*) yang tidak terbatas ruang, *user friendly*, serta mudah diakses. Oleh karena itu dengan menggunakan teknologi *mobile*, diharapkan dapat memudahkan pengguna aplikasi dalam memperoleh informasi tentang karakter wayang golek serta memberikan solusi terhadap kekurangan yang ada pada media yang sedang digunakan saat ini. Merujuk pada kekurangan yang ada, maka dibangun aplikasi *mobile smartphone* berbasis *android* dengan menggunakan teknologi Augmented Realty (AR). Aplikasi ini bertujuan menampilkan informasi tentang karakter-karakter yang ada dalam kesenian wayang golek pada layar *mobile smartphone* berbasis *Android* secara mudah dan cepat (*real time*). Aplikasi ini menggunakan

metode *markerless based* dengan teknik *object tracking*. Markerless berfungsi sebagai kunci untuk menentukan informasi tentang karakter wayang golek yang akan ditampilkan. Dalam proses pengenalan *markerless* digunakan bantuan dari *Library vuforia SDK.*

3.1.2 Analisis Sistem yang Sedang 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. Analisis prosedur pada proses media yang sedang berjalan adalah sebagai berikut:



Gambar 3. 1 Alur Sistem Yang Sedang Berjalan

Keterangan :

1. User disini adalah seorang pengunjung museum wayang yang ingin mengetahui informasi tentang karakter-karakter dalam kesenian wayang golek.
2. Beberapa media yang digunakan untuk memperoleh informasi tentang karakter dalam kesenian wayang golek, diantaranya :
   a. *Teks Box* atau Kotak Informasi : Media ini biasanya terdapat pada lemari kaca tempat wayang dipamerkan di museum wayang. Cukup mudah untuk mengetahui informasi kesenian wayang golek dengan media ini, namun biasanya hanya terdapat satu *teks box* pada kotak kaca tempat wayang

ditaruh. Karena jumlahnya yang hanya satu pada tiap lemari kaca tempat wayang dipamerkan, informasi yang ditampilkan pun tidak banyak, hanya nama karakter wayang saja yang ada dalam teks box ini, sehingga pengunjung sulit untuk mendapatkan informasi tentang karakter wayang yang ada dalam lemari kaca secara lebih detail.

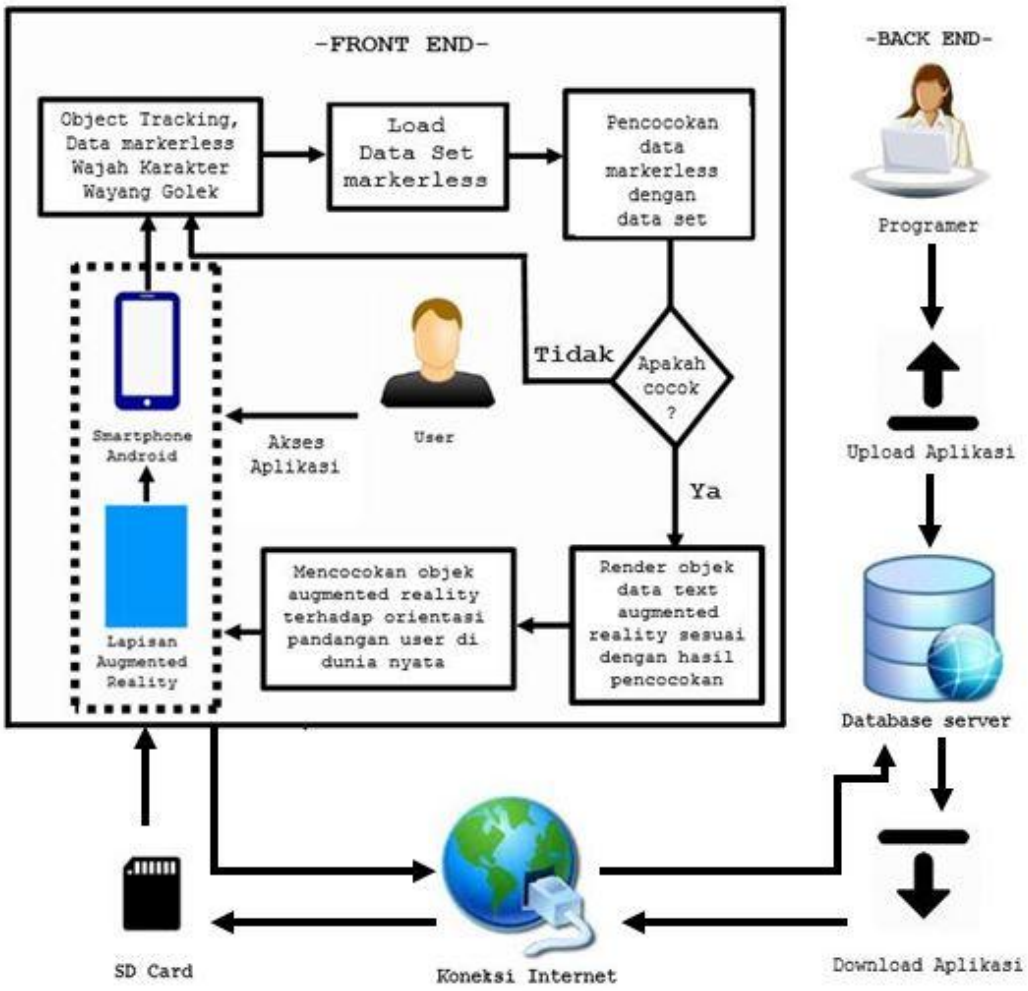
b. *Tour Guide* : Media ini juga cukup efektif untuk memperoleh informasi tentang karakter wayang, namun tidak semua pengunjung akan didamping oleh *tour guide* yang memberikan informasi tentang karakter- karakter dalam kesenian wayang golek tersebut.

Dari gambaran prosedur terlihat beberapa media yang saat ini digunakan sebagai media untuk memperoleh informasi tentang karakter-karakter yang ada dalam kesenian wayang golek masih memiliki beberapa kekurangan, diantaranya terbatasnya informasi yang ada dalam teks box, informasi tentang karakter wayang golek yang dirasa kurang *detail,* serta *tour guide* yang tidak semua pengunjung dapat didampingi olehnya, sehingga proses untuk mendapatkan informasi tentang karakter-karakter yang ada dalam wayang golek dirasa cukup sulit dan masih lambat.

3.1.3 Analisis Sistem yang Akan Dibangun

Sistem yang akan dibangun adalah sebuah aplikasi *mobile smartphone* yang menggunakan *platform* Android dengan teknologi Augmented Reality. Teknologi *mobile smartphone* dipilih karena memiliki beberapa kelebihan, diantaranya dapat diakses dengan mudah, kapan saja dan dimana saja, sehingga memudahkan user dalam mengakses dan menggunakan aplikasi. Teknologi *Augmented Reality* dipilih karena memiliki beberapa kelebihan, salah satu diantaranya adalah teknologi ini dapat menampilkan objek *Augmented Reality* berupa teks informasi tentang karakter wayang golek dengan cepat, sehingga dapat menjadi solusi dari permasalahan yang ada. Untuk dapat menampilkan objek *augmented reality* digunakan *metode markerless based* dengan teknik *object tracking*. Input yang digunakan sebagai *markerless* berupa wajah karakter wayang golek. Teknik *object tracking* digunakan untuk mengenali wajah wayang golek. Dalam proses *object tracking* wajah karakter wayang golek digunakan bantuan dari *Library Vuforia*

SDK. *Library Vuforia SDK* dipilih karena library ini memiliki kelebihan yaitu komputasinya yang sangat cepat dalam mendeteksi objek, karena hanya bergantung pada jumlah *feature* yang ada dalam sebuah objek target. Tujuan yang ingin dicapai dari perancangan aplikasi ini adalah membuat aplikasi yang dapat menampilkan informasi yang lebih detail tentang karakter-karakter yang ada dalam kesenian wayang golek secara cepat serta mudah digunakan oleh pengguna aplikasi. Gambar 3.2 menunjukan deskripsi proses sistem yang akan dibangun.



Gambar 3. 2Alur Sistem yang Akan ibangun

Keterangan :

a. User adalah pengunjung museum yang menggunakan *smartphone Android*. Aplikasi ini berjalan tanpa koneksi internet, tetapi memerlukan koneksi Internet hanya jika user hendak melakukan update aplikasi.

b. Pada saat user menjalankan aplikasi, aplikasi ini akan mejalankan fungsi kamera *smartphone* secara otomatis untuk mendeteksi objek wajah karakter wayang golek yang berfungsi sebagai *markerless*. Proses deteksi objek wajah karakter wayang golek ini menggunakan bantuan dari *library Vuforia SDK* .
c. Wajah karakter wayang golek yang terdeteksi akan dibandingkan dengan data markerless yang ada di database.
d. Apabila data markerless yang diambil dari kamera *smartphone* tersebut sesuai dengan data markerless yang berada di database, maka aplikasi akan menampilkan objek *augmented reality* berupa informasi tentang karakter wayang golek dalam bentuk teks di layar *smartphone* berbasis *platform Android*.
e. *Programer* sebagai *backend*, bertugas untuk membuat aplikasi dengan versi terbaru serta meng*upload* aplikasi ke *web server*. Layanan *Web service* yang digunakan adalah dropbox. Apabila ada aplikasi dengan versi yang baru, aplikasi di sisi *frontend* menyediakan tombol *CEK Update* yang berfungsi untuk mengecek apakah ada aplikasi dengan versi yang baru dari *web server*, apabila ada aplikasi dengan versi terbaru, user dapat mendownload aplikasi. Hasil *download* aplikasi akan tersimpan di dalam *memory sd card smartphone*.

3.1.4 Analisis Markerless

Markerless Augmented Reality merupakan salah satu metode Augmented Reality tanpa menggunakan frame marker sebagai obyek yang dideteksi. Dengan teknologi *Markerless Augmented Reality* dan teknik Pengenalan Pola (*Pattern Recognition*), maka penggunaan *marker* sebagai *object target* yang selama ini menghabiskan ruang dapat digantikan dengan wajah karakter wayang golek sebagai *object target* (obyek yang dilacak), sehingga tidak lagi mengurangi efisiensi ruang dengan adanya marker.

Terdapat perbedaan antara pelacakan berbasis *marker* (*marker based tracking*) dan pelacakan berbasis *markerless* (*markerless based tracking*). Pada pelacakan berbasis *marker* posisi kamera dan orientasi kamera dhitung dengan marker yang telah ditetapkan. Sementara pelacakan *markerless*, menghitung posisi antara kamera/pengguna dan dunia nyata tanpa referensi apapun, hanya

menggunakan titik-titik fitur alami (edge, corner. garis atau model 3D). Metode *Markerless* memerlukan objek 3D atau gambar referensi untuk inisialisasi agar keakuratan informasi yang didapat dari *object* yang dilacak lebih baik.

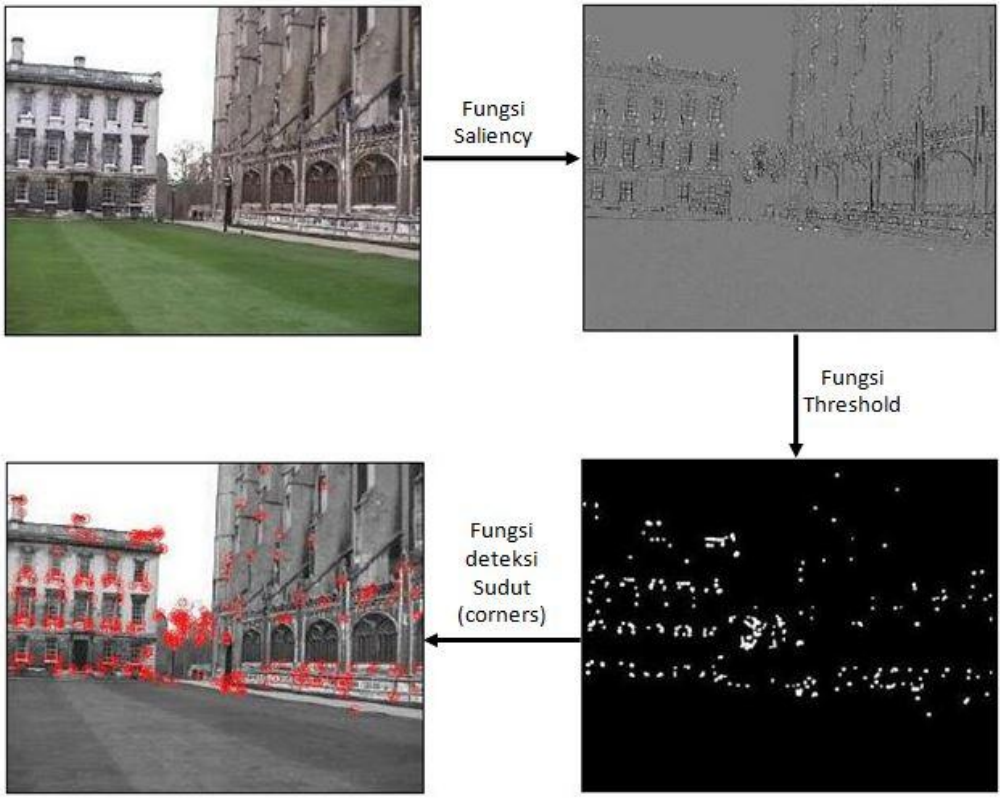
3.1.4.1 Analisis Algoritma untuk Tracking Object Target

Analisis algoritma ini menjelaskan algoritma yang digunakan *Library vuforia SDK* dalam mendektesi *object 3D* sebagai *markerless* yang berupa wajah karakter wayang golek. *Library Vuforia SDK* menggunakan algoritma *FAST Corner detection* untuk melacak dan mendeteksi keberadaan *object target*.

Vuforia menggunakan *point* sebagai *rating augmentabl*e untuk menilai kualitas pelacakan sebuah *object target*. *Object target* akan dirubah menjadi *markerless* oleh pihak *vuforia*. Sebelum *object target* dirubah menjadi markerless, sebelumnya *object target* melalui proses *scanning* terlebih dahulu. Proses *scanning object* menggunakan bantuan aplikasi *Vuforia Object Scanner*. *VuforiaObjectScanner* menggunakan *point* sebagai tolak ukur berapa banyak *feature* yang berhasil di-*capture* dari sebuah *Object target*. Jumlah *point* akan bertambah ketika kamera berhasil meng-*capture feature* dari sebuah *object target*. *Feature object target* di-*capture* dari berbagai macam sudut pandang dengan tujuan untuk menambah jumlah *point* sebuah *object target*. Semakin banyak jumlah *point* dari sebuah *object target,* semakin banyak *feature* yang terdeteksi. Semakin banyak *feature* yang terdeteksi, semakin baik kualitas pelacakan sebuah *object target.*

3.1.4.2 Algoritma FAST Corner Detection

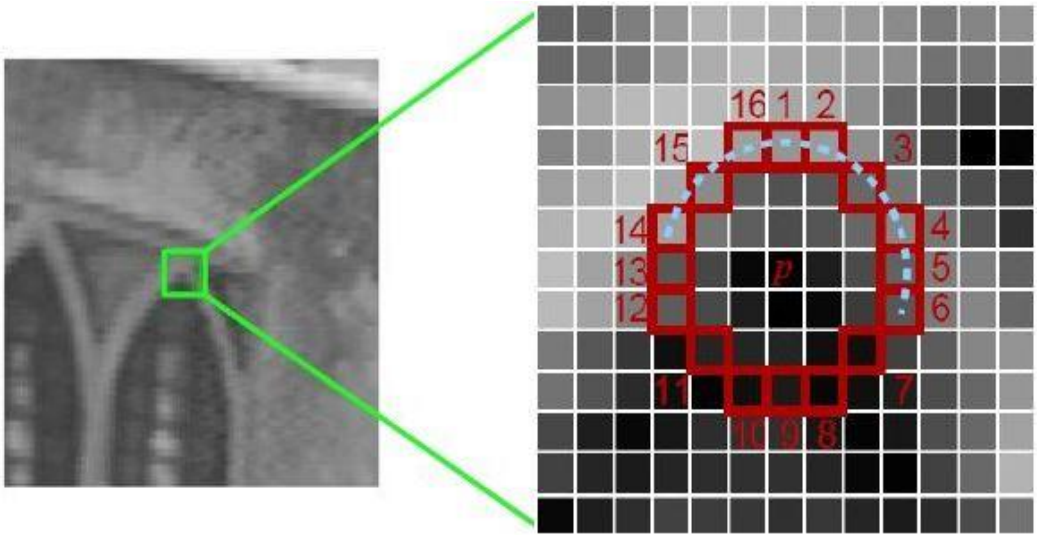
FAST *(Features from Accelerated Segment Test)* adalah suatu algoritma yang dikembangkan oleh Edward Rosten, Reid Porter, and Tom Drummond. FAST *corner detection* ini dibuat dengan tujuan mempercepat waktu komputasi secara *real-time* dengan konsekuensi menurunkan tingkat akurasi pendeteksian sudut.



Gambar 3. 3 FAST Flow Map

FAST *corner detection* bekerja pada suatu citra sebagai berikut:

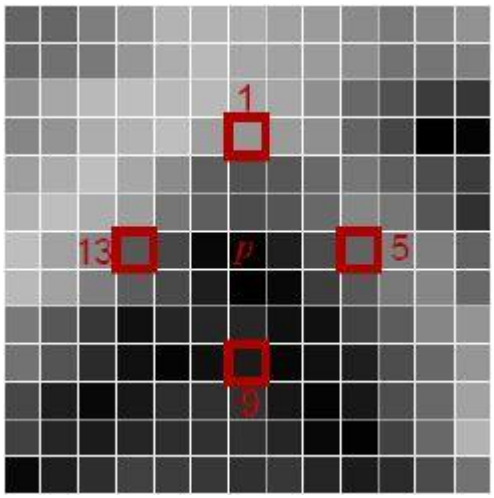
a. Tentukan sebuah titik *p* pada citra dengan posisi awal (x_p,y_p),



Gambar 3. 4 Sample Gambar Untuk Menentukan Titik P Sebagai Titik Awal

b. Tentukan lokasi keempat titik pengujian untuk dibandingkan dengan titik p, dengan ketentuan sebagai berikut :

- Titik pertama (n=1) terletak pada koordinat (x_p,y_{p-3}),
- Titik kedua (n=2) terletak pada koordinat (x_{p+3},y_p),
- Titik ketiga (n=3) terletak pada koordinat (x_p,y_{p+3}),
- Titik keempat (n=4) terletak pada koordinat (x_{p-3},y_p)



Gambar 3. 5 Menentukan Titik Pengujian

c. Intensitas pixels terbagi menjadi tiga subset, yaitu *Much Brigter*(lebih cerah), *Much darker*(lebih Gelap), *Similar*(sama). Bandingkan intensitas titik pusat *p* dengan keempat titik di sekitar. Jika terdapat paling sedikit 3 titik yang memenuhi syarat berikut, maka titik pusat *p* adalah sudut.

$$C_p = \begin{cases} 1, & In < Ip - t \quad or \quad In > Ip + t \\ 0, & otherwise \end{cases}$$

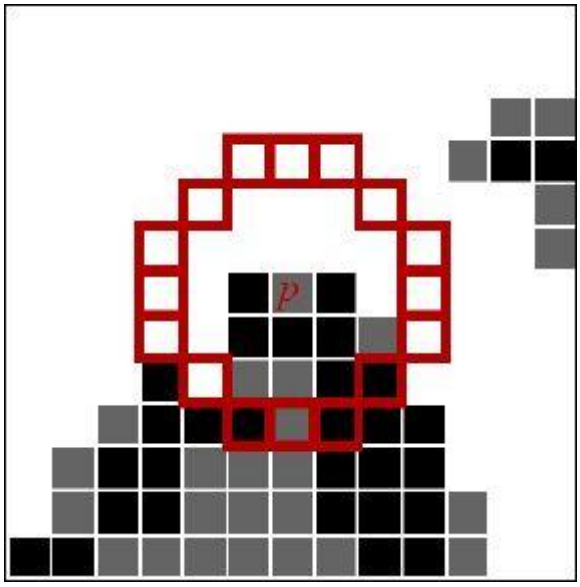
Keterangan :

C_p : Keputusan titik p sebagai sudut atau bukan sudut. Nilai 1 menunjukkan bahwa titik p merupakan suatu sudut, dan nilai 0 menunjukkan bahwa titik p bukan suatu sudut.

I_n : nilai intensitas piksel ke-n

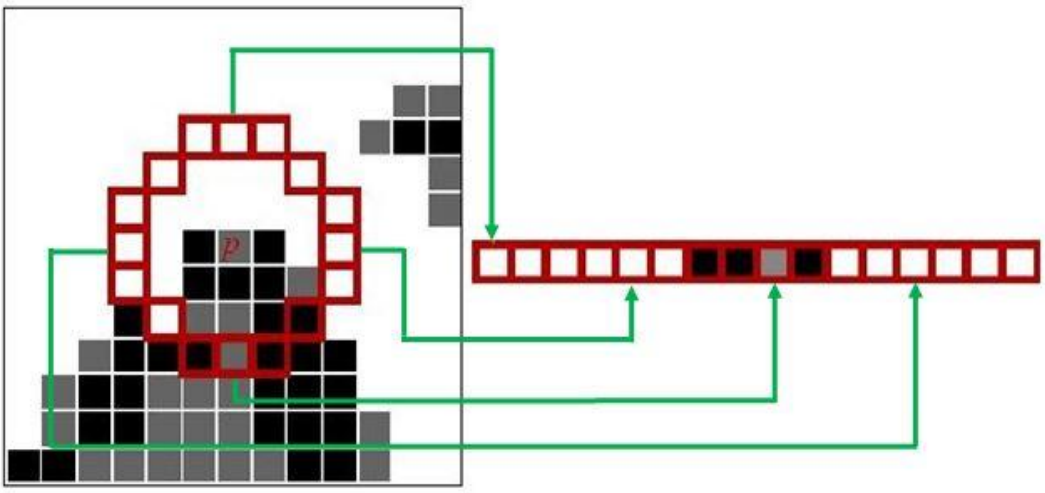
I_p : nilai intensitas titik p

t : batas ambang nilai intensitas yang ditoleransi (Threshold)



Gambar 3. 6 Hasil Proses Perbandingan Intensitas Pixels

d. Representasi hasil proses perbandingan intensitas pixels dalam bentuk vektor. Titik p merupakan sebuah sudut karena terdapat 4 titik yang memenuhi syarat.



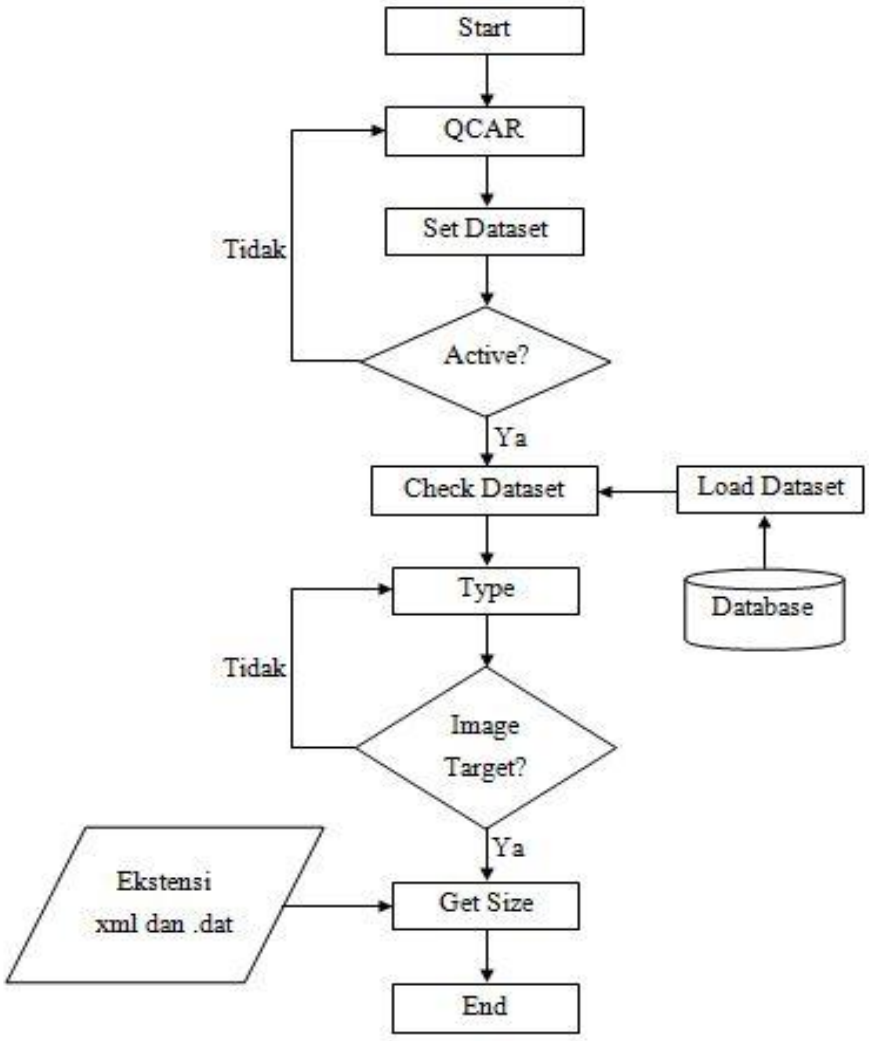
Gambar 3. 7 Repersentasi Intensitas Pixels Dalam Bentuk Vektor

e. Ulangi proses sampai seluruh titik pada citra sudah dibandingkan intensitasnya.

3.1.4.3 Analisis Penentuan Markerless

Analisis metode terhadap kasus merupakan analisis yang mendeskripsikan bagaimana proses aplikasi menampilkan objek *augmented reality* dari awal *inisialisasi, tracking marker,* sampai dengan proses memunculkan data teks.

Dalam perancangan aplikasi dengan teknologi *Augmented Reality* akan menggabungkan objek *virtual* dengan objek nyata, dalam hal ini objek *virtual* berupa objek 2D atau 3D dan objek nyatanya berupa wajah karakter wayang golek.



Gambar 3. 8 Alur Sistem Aplikasi Augmented Reality

Secara garis besar, proses sistem augmented reality dapat digambarkan dengan diagram alur pada gambar 3.8. Proses pengenalan deteksi markerless ini melalui beberapa tahapan. Secara garis besarnya, dalam perancangan ada tiga bagian utama yaitu :

1. *Inisialisasi*
2. *Tracking Object Target*
3. Memunculkan *Object Augmented Reality*

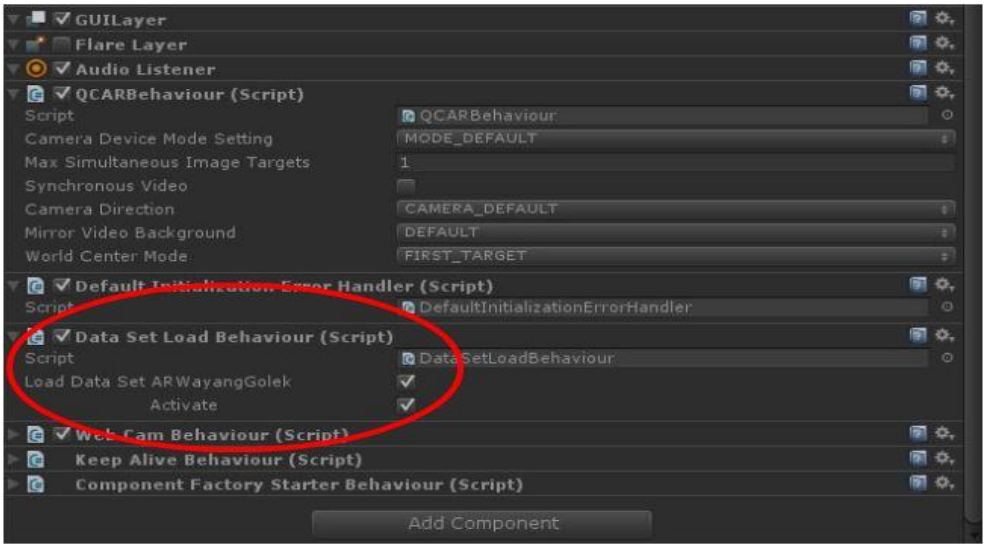
3.1.4.3.1. Inisialisasi

Pada tahap inisialisasi ini ditentukan *object target* dan *object Augmented reality* yang akan digunakan.

3.1.4.3.1.1. Inisialisasi Object Target

Object Target yang digunakan pada analisis ini berupa wajah karakter-karakter wayang golek yang berada di Museum Wayang. Proses pembuatan sebuah objek wayang golek menjadi *Object Target* digunakan bantuan *Vuforia.* Proses pembuatan sebuah Objek 3D menjadi *Object target* dapat dilihat pada bahasan bab sebelumnya.

Database *Object Target* yang telah dibuat sebelumnya di-*import* ke *unity3D* untuk dijadikan sebagai penanda(*markerless*) yang akan dilacak oleh aplikasi. Setelah itu aktifkan database yang telah di-*import* dengan cara ceklist pada *Load Data Set* dan pada *Aktivate*.



Gambar 3. 9 Menu Load Data Set

3.1.4.3.1.2. Inisialisasi object Augmented reality

Setelah *Object Target* telah diinisialisasi, maka tahap selanjutnya adalah menyisipkan *text augmented reality* untuk menampilkan informasi dari *objek target*. Penyisipan yang digunakan adalah dengan cara membuat *text mesh* yang disediakan oleh *game engine Unity 3D*, sehingga ketika objek target berhasil dilacak, maka informasi berupa *text* akan muncul. Proses menampilkan *text*

informasi ini hanya akan terjadi ketika *object target* berhasil dilacak oleh kamera *smartphone android*.

3.1.4.3.2. Tracking Object Target

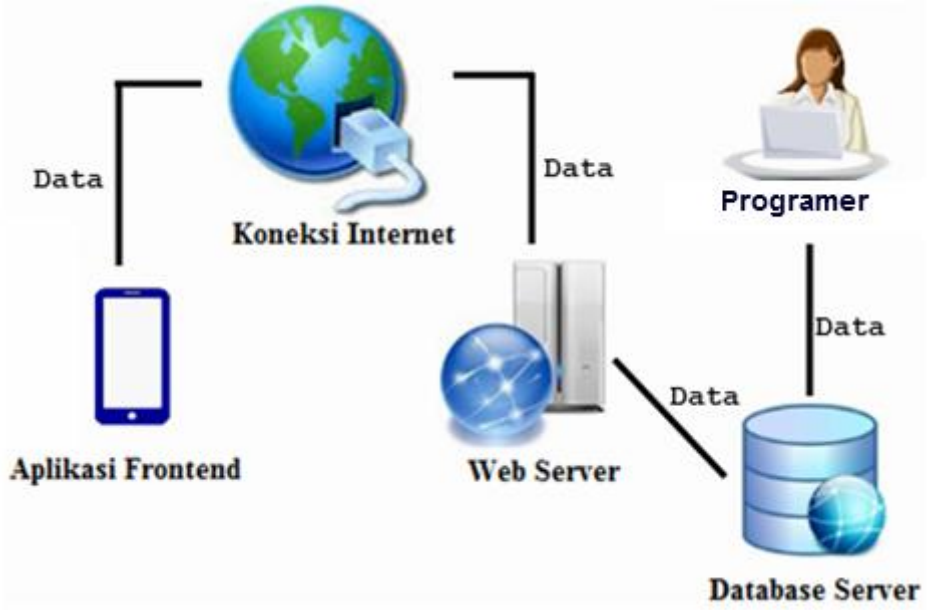
Tracking object target adalah proses dimana aplikasi akan melakukan proses deteksi atau pelacakan sebuah *object target* dengan tujuan untuk menampilkan *Object Augmented reality* berupa *text* informasi tentang sebuah objek sesuai dengan objek yang terdeteksi. Setiap *Object Target* yang telah diinisialisaikan akan digunakan dalam proses pelacakan (*tracking*) dan pencocokan pola objek 3D dengan pola *object target*. Jika pola dinyatakan cocok atau berhasil dikenali, maka akan dilakukan pencocokan *object target* dengan *object augmented reality* yang akan ditampilkan. Tracking objek target ini menggunakan metode *FAST Corner Detection* yang diimplementasikan pada *Library Vuforia SDK.*

3.1.4.3.3. Menampilkan Object Augmented Reality

Hasil pencocokan objek target yang telah dilakukan pada proses sebelumnya menjadi data acuan untuk menampilkan *object Augmented reality* yang sesuai dengan data *object target* di dalam database. *Library Vuforia SDK* menyediakan fungsi tersendiri untuk menampilkan objek *Augmented reality*. Aplikasi akan menampilkan *object augmented reality* berupa data *text* sesuai dengan data *object target* yang ada di database.

3.1.5 Analisis Alur Data pada Sistem Aplikasi

Analisis alur data sistem merupakan analisis yang berfungsi untuk menggambarkan secara rinci bagaimana sistem bekerja.



Gambar 3. 10 Alur Data pada Sistem Aplikasi

1. Aplikasi *mobile* (*FrontEnd*)

   Aplikasi *mobile* dalam hal ini adalah analisis fungsionalitas aplikasi untuk mengakses data dari database server melalui web service kemudian data yang diambil dari web server disimpan di *memory SD Card*.

2. Web *service*

   Web service pada penelitian ini berfungsi sebagai *middle application* atau aplikasi penghubung yang menghubungkan aplikasi frontend dengan database server. Cara kerja web server ini yaitu dengan mengambil data dari database server. Web Service yang digunakan untuk mennyimpan data aplikasi adalah *Dropbox.*

3. Programer

   Programer pada penelitian ini berfungsi sebagai pembuat aplikasi serta uuntuk meng*upload* aplikasi ke *database serve.r*

3.2 Analisis Kebutuhan Non Fungsional

Analisis dan kebutuhan non fungsional meliputi analisis dan kebutuhan perangkat keras serta analisis dan kebutuhan perangkat lunak. 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 memperoleh informasi yang dibutuhkan.

3.2.1 Analisis dan Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang dibutuhkan untuk pembuatan aplikasi ini dapat dilihat pada Tabel 3.1

Tabel 3. 1 Spesifikasi Kebutuhan perangkat Keras

Aplikasi Frontend/User	**AplikasiBackend/Admin**
Mobile Android sengan spek minimal OS android 2.3 - Ginger Bread	Laptop dengan spesifikasi : a. RAM 2GB b. Harddisk 250GB
Kamera *Smartphone* Android (OS 2.3) 2 Megapixel	Koneksi internet, Wifi, Modem.
Koneksi internet HSDPA, 2G dan 3G	

3.2.2 Analisis dan Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak yang dibutuhkan untuk pembuatan aplikasi ini dapat dilihat pada Tabel 3.2 :

Tabel 3. 2 Kebutuhan Perangkat Lunak

Aplikasi Frontend/User	Aplikasi Backend/Admin
Library Augmented Reality : Unity3D dengan *vuforia SDK*	Sistem Operasi : Windows xp/vista/seven.
	Browser : Firefox, internet explorer, opera,google crohme.
	My SQL sebagai DBMS.
	Dream Weaver/Notepad ++,
	Menggunakan PHP

3.2.3 Analisis Kebutuhan User

Spesifikasi pengguna sistem dimaksudkan untuk mengetahui siapa saja *actor* yang terlibat dalam menjalankan sistem. teknologi *mobile* dan web.

Pengguna sistem dibagi menjadi dua bagian, yaitu :

1. Pengguna aplikasi (*end user*), yaitu para pengunjung museum wayang yang menggunakan *smartphone* dengan *platform* android dengan versi 2.3 atau lebih tinggi.
2. Programer sebagai pengelola konten mempunyai kapabilitas dalam mengelola database seperti meng*upload* aplikasi versi terbaru dan menghapus aplikasi dengan versi sebelumnya. Karakteristik pengguna sistem dapat dilihat pada Tabel 3.3:

Tabel 3. 3 Karakteristik User

Pengguna	**Hak Akses**	**Pendidikan**	**Kemampuan**
End User	Menggunakan Data dengan cara Mengakses Aplikasi Front End	Minimal SD, usia 11 tahun keatas	Dapat Menggunakan *Smartphone* Android
Programmer	Mengelola Data pada Aplikasi Back End	Minimal S1	Dapat menjalankan fungsi-fungsi aplikasi serta dapat mengolah data wayang golek.

3.3 Analisis Kebutuhan Fungsional

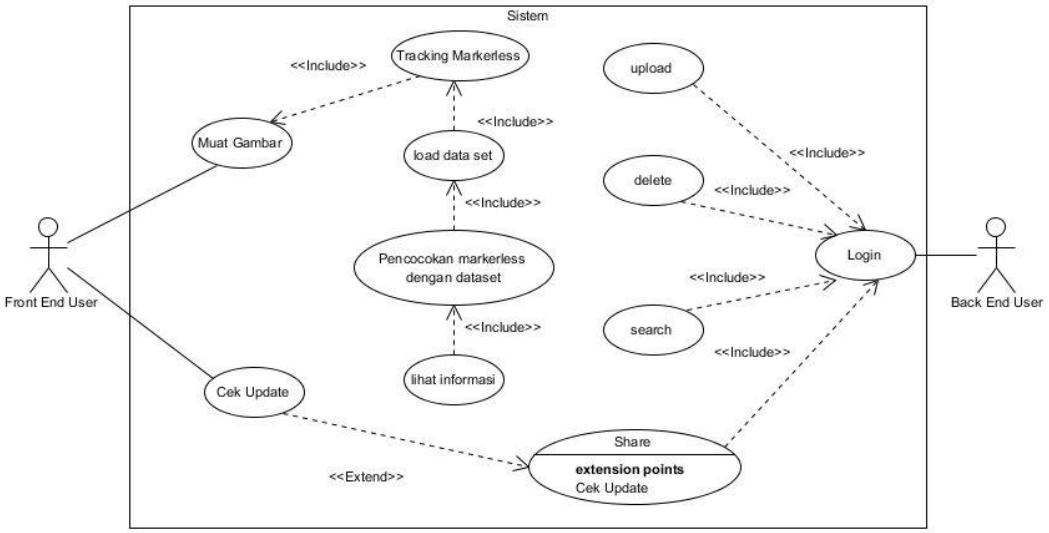
Kebutuhan fungsional berhubungan dengan fitur aplikasi yang akan dibuat atau dikembangkan. Berikut adalah tahapan analisis kebutuhan fungsional sistem aplikasi mobile *ARWayangGolek*.

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik.

Analisis yang dilakukan dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain identifikasi aktor, *use case diagram*, *skenario*, *sequence diagram*, *activity diagram*, *class diagram*.

3.3.1 Use Case Diagram

Use case Diagram merupakan konstruksi untuk mendeskripsikan hububungan yang terjadi antar aktor dengan aktivitas yang terdapat pada sistem. Sasaran pemodelan *use case* diantaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan sistem yang akan dibangun. Dari hasil analisis aplikasi yang ada maka *use case diagram* untuk aplikasi ini dapat di lihat pada gambar 3.16:



Gambar 3. 11 Usecase Diagram

3.3.1.1 Deskripsi *Usecase*

Use case diagram adalah sebuah diagram untuk mendeskripsikan hubungan-hubungan yang terjadi antar aktor dengan aktivitas yang terdapat pada sistem. Tujuan pembuatan *use case diagram* adalah untuk mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan yang disepakati antara pemakai dan pembangun aplikasi. Deskripsi Usecase dapat dilihat pada tabel 3.4

Tabel 3. 4 Deskripsi *Use case*

Use case* Aplikasi *Frontend		
No *Use case*	**Nama *Use case***	**Deskripsi**
UCF-01	Muat Gambar	Fungsionalitas untuk memuat gambar dengan menggunakan kamera *smartphone.*
UCF-02	Tracking Markerless	Fungsionalitas untuk *Tracking markerless.*
UCF-03	Load Data Set	Fungsionalitas untuk mengaktifkan *Data Set Markerless.*
UCF-04	Pencocokan Markerless dengan Data Set	Fungsionalitas untuk mencocokan data *markerless* yang dimuat kamera dengan *data set markerless*.
UCF-05	Lihat Informasi Gambar	Fungsionalitas untuk menampilkan informasi ke layar telepon dengan menggunakan teknologi *Augmented Reality.*
UCF-06	Cek Update	Fungsionalitas untuk mengunjungi URL dengan tujuan cek update aplikasi versi terbaru.
Use case* Aplikasi *Backend		
No Use case	**Nama Use case**	**Deskripsi**
UCB-01	Login	Fungsionalitas untuk masuk kedalam aplikasi web server.

UCB-02	Upload	Fungsionalitas untuk meng-*upload* data aplikasi dengan versi terbaru ke database web server.
UCB-03	Delete	Fungsionalitas untuk menghapus data aplikasi yang berada di database server.
UCB-04	Search	Fungsionalitas untuk mencari data yang tersimpan di database server.
UCB-05	Share	Fungsionalitas share link download data aplikasi yang berada di database server.

3.3.1.2 Use case Scenario

Use case Scenario/ Skenario *Use Case* adalah sebuah scenario untuk mendeskripsikan urutan/ langkah-langkah dalam proses bisnis, baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor.

3.3.1.2.1 Use case Scenario Frontend

Usecase Scenario aplikasi *frontend* dapat dilihat pada tabel 3.5 sampai tabel 3.13.

Tabel 3. 5 Use case Scenario Muat Gambar

Identifikasi	
Nama Use case	Muat Gambar
Aktor	Frontend User
Tujuan	Menjalankan aplikasi lalu menyorotkan kamera *smartphone* ke objek untuk memperoleh data uji markerless secara sistem yang kemudian akan dilakukan proses

	pencocokan dengan dataset markerless.
Kondisi Awal	Sitem menampilkan Menu Utama Aplikasi pada layar telepon seluller.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Mengakses Aplikasi	2. Menampilkan Menu Utama.
3. Mengeksekusi tombol Muat Gambar	4. Menjalankan fungsionalitas kamera *smartphone*.
5. Mengarahkan kamera ke Object Target berupa wajah karakter wayang golek.	6. Menjalankan fungsionalitas *tracking markerless* untuk mendeteksi keberadaan *markerless* sebagai data uji.
	7. Melakukan pencocokan data uji *markerless* hasil *tracking* dengan *data set markerless*
	8. Me-*render* objek *Augmented reality* sesuai dengan hasil pencocokan dan menampilkannya pada layar *smartphone*.
Skenario Gagal	
Aksi Aktor	**Reaksi Sistem**
	1. Data hasil *tracking markerless* tidak cocok dengan *data set markerless*.
	2. Sistem tetap menjalankan fungsi kamera smartphone untuk mendeteksi *markerless*.
Kondisi Akhir	Aplikasi tetap menjalankan fungsionalitas Kamera *smartphone* sampai dilakukan aksi berikutnya.

Tabel 3. 6 Usecase Skenario Tracking Markerless

Identifikasi	
Nama Use case	Tracking Markerless
Aktor	Frontend User
Tujuan	Mendeteksi keberadaan markerless yang diambil melalui mata kamera *smartphone*.
Kondisi Awal	Sitem menampilkan Menu Utama Aplikasi pada layar telepon seluller.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Mengakses Aplikasi	2. Menampilkan Menu Utama.
3. Mengeksekusi tombol Muat Gambar	4. Menjalankan fungsionalitas kamera *smartphone*.
5. Mengarahkan kamera ke Object Target berupa wajah karakter wayang golek.	6. Menjalankan fungsionalitas *tracking markerless* untuk mendeteksi keberadaan *markerless* sebagai data uji.
	7. Melakukan pencocokan data uji *markerless* hasil *tracking* dengan *data set markerless*
	8. Me-*render* objek *Augmented reality* sesuai dengan hasil pencocokan dan menampilkannya pada layar *smartphone*.
Skenario Gagal	
Aksi Aktor	**Reaksi Sistem**
	1. Data hasil *tracking markerless* tidak cocok dengan *data set markerless*.
	2. Sistem tetap menjalankan fungsi

	kamera smartphone untuk mendeteksi *markerless*.
Kondisi Akhir	Aplikasi tetap menjalankan fungsionalitas Kamera *smartphone* sampai dilakukan aksi berikutnya.

Tabel 3. 7 Usecase Skenario Pencocokan Markerless dengan Data Set

Identifikasi	
Nama Use case	Pencocokan Markerless dengan Data Set
Aktor	Frontend User
Tujuan	Mencocokan data uji *markerless* dengan *data set markerless.*
Kondisi Awal	Sitem menampilkan Menu Utama Aplikasi pada layar telepon seluller.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Mengakses Aplikasi	2. Menampilkan Menu Utama.
3. Mengeksekusi tombol Muat Gambar	4. Menjalankan fungsionalitas kamera *smartphone*.
5. Mengarahkan kamera ke Object Target berupa wajah karakter wayang golek.	6. Menjalankan fungsionalitas *tracking markerless* untuk mendeteksi keberadaan *markerless* sebagai data uji.
	7. Melakukan pencocokan data uji *markerless* hasil *tracking* dengan *data set markerless*
	8. Me-*render* objek *Augmented reality* sesuai dengan hasil pencocokan dan menampilkannya pada layar *smartphone*.
Skenario Gagal	

Aksi Aktor	**Reaksi Sistem**
	1. Data hasil *tracking markerless* tidak cocok dengan *data set markerless*.
	2. Sistem tetap menjalankan fungsi kamera smartphone untuk mendeteksi *markerless*.
Kondisi Akhir	Aplikasi tetap menjalankan fungsionalitas Kamera *smartphone* sampai dilakukan aksi berikutnya.

Tabel 3. 8 Use case Scenario Lihat Informasi

Identifikasi	
Nama Use case	Lihat Informasi
Aktor	Frontend User
Tujuan	Menampilkan informasi di layar smartphone sesuai dengan hasil pencocokan.
Kondisi Awal	Sitem menampilkan Menu Utama Aplikasi pada layar telepon seluller.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Mengakses Aplikasi	2. Menampilkan Menu Utama.
3. Mengeksekusi tombol Muat Gambar	4. Menjalankan fungsionalitas kamera *smartphone*.
5. Mengarahkan kamera ke Object Target berupa wajah karakter wayang golek.	6. Menjalankan fungsionalitas *tracking markerless* untuk mendeteksi keberadaan *markerless* sebagai data uji.
	7. Melakukan pencocokan data uji *markerless* hasil *tracking* dengan

	data set markerless
	8. Me-*render* objek *Augmented reality* sesuai dengan hasil pencocokan dan menampilkannya pada layar *smartphone*.
Skenario Gagal	
Aksi Aktor	**Reaksi Sistem**
	1. Data hasil *tracking markerless* tidak cocok dengan *data set markerless*.
	2. Sistem tetap menjalankan fungsi kamera smartphone untuk mendeteksi *markerless*.
Kondisi Akhir	Aplikasi tetap menjalankan fungsionalitas Kamera *smartphone* sampai dilakukan aksi berikutnya.

Tabel 3. 9 Use case Scenario Cek Update

Identifikasi	
Nama Use case	Cek Update
Aktor	Frontend User
Tujuan	Mengecek versi aplikasi terbaru.
Kondisi Awal	User menampilkan Menu Awal Aplikasi
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Mengakses Aplikasi	2. Menampilkan Menu Awal Aplikasi.
3. Mengeksekusi tombol Cek Update	4. Aplikasi mencoba terhubung ke webserver dengan koneksi internet menggunakan web

	browser
	5. Aplikasi terhubung ke web server dan menampilkan tombol *Download* untuk men-*download* aplikasi.
6. Mengecek versi terbaru aplikasi	
7. Mengeksekusi tombol *Download*	8. Melakukan proses pengunduhan aplikasi
	9. Aplikasi menampilkan progress bar sebagai tanda bahwa aplikasi sedang mengunduh data baru.
Kodisi Akhir	Progress bar secara otomatis menghilang dan data aplikasi dengan versi terbaru berhasil di *download*.
Skenario Gagal	
Aksi Aktor	**Reaksi Sistem**
	Menampilkan pesan "Tidak Ada Koneksi Internet" atau "Tidak ada Data Baru"
Kondisi Akhir	Aplikasi tidak dapat terhubung ke *web server*.

3.3.1.2.2 Use case Scenario Backend

Skenario proses-proses yang terdapat dalam *use case diagram* aplikasi *backend* dapat dilihat pada tabel 3.27 sampai tabel 3.33

Tabel 3. 10 Use case Scenario Login

Identifikasi	
Nama Use case	Login
Aktor	Backend User
Tujuan	Masuk kedalam sistem

Kondisi Awal	Sitem menampilkan halaman Login
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memasukan *user name* dan *password* pada halaman *Login*	
2. Mengeksekusi tombol *Login*	3. Melakukan validasi antara *user name* dan *password*
	4. Menampilkan halaman Menu Utama web server.
Skenario Gagal	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan pesan error "Login gagal, pastikan *user name* dan *password* anda benar "
Kondisi Akhir	Menampilkan Halaman Menu Utama web

Tabel 3. 11 Use case Scenario Upload

Identifikasi	
Nama Use case	Upload
Aktor	Backend User
Tujuan	Menambahkan data aplikasi dengan versi terbaru ke dalam *database server*.
Kondisi Awal	Sistem menampilkan *Icon Upload*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Mengeksekusi tombol Upload	2. Menampilkan halaman form *Upload*
3. Mengisi form upload	
4. Mengeksekusi tombol upload	5. Melakukan proses upload data ke database web server.

Kondisi Akhir	Data berhasil ditambahkan ke database server
Skenario Gagal	
Aksi Aktor	**Reaksi Sistem**
1. Mengisi form tambah data dengan tidak lengkap	2. Menampilkan pesan "Upload Failed"
Kondisi Akhir	Data tidak berhasil ditambahkan ke dalam database server

Tabel 3. 12 Use case Scenario Delete

Identifikasi	
Nama Use case	Delete
Aktor	Backend User
Tujuan	Menghapus data aplikasi dengan versi lama dari database server
Kondisi Awal	Sistem menampilkan *icon Delete*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Klik Kanan pada data aplikasi yang akan di-*delete*	2. Menampilkan opsi *share, download, delete, rename, move, copy, previous versions*
3. Mengklik opsi delete	4. Men-*delete* data aplikasi dari dalam *database server*
Kondisi Akhir	Data Aplikasi berhasil di-*delete* dan hilang dari *list* data.

Tabel 3. 13 Use case Scenario search

Identifikasi	
Nama Use case	Search
Aktor	Backend User
Tujuan	Menemukan data yang dicari

Kondisi Awal	Sistem menampilkan konten *search*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Meng-*input*-kan *key word* data yang akan dicari pada kotak isian *search*	2. Menampilkan konten *search*
3. Mengeksekusi tombol cari	Melakukan proses pencarian data di database server sesuai key word yang di-*input*-kan.
	4. Menemukan data yang dicar dan menampilkannya ke layar
Kondisi Akhir	Sistem menampilkan data hasil pencarian
Skenario Gagal	
Aksi Aktor	**Reaksi Sistem**
1. Meng-*input*-kan *key word* data yang akan dicari pada kotak isian *search* dengan salah	2. Menampilkan pesan "No Result Found, Please update your search and try again"
Kondisi Akhir	Sistem menampilkan pesan "No Result Found, Please update your search and try again"

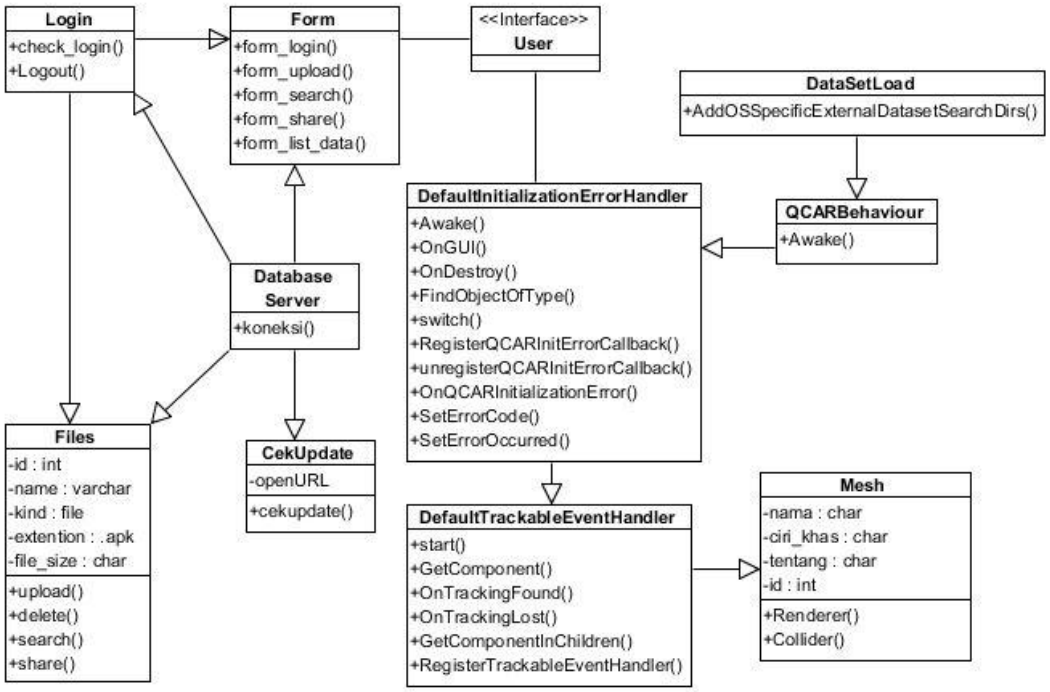
Tabel 3. 14 Use case Scenario Share

Identifikasi	
Nama Use case	Share
Aktor	Backend User
Tujuan	Share Data dalam database server agar dapat di *download.*
Kondisi Awal	Sistem menampilkan daftar data
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**

1. Memilih data yang akan di-*share*	2. Menampilkan halaman *list* Data
3. Mengeksekusi tombol *share*	4. Menampilkan link download data yang akan *di-share.*
Kondisi Akhir	Sistem menampilkan form share sampai dilakukan aksi berikutnya.

3.3.2 Class Diagram

Class Diagram menggambarkan struktur dan hubungan antara setiap objek yang ada pada sistem. Struktur itu meliputi atribut-atribut dan operasi-operasi yang ada pada masing-masing *class*. Keterkaitan antara masing- masing *class* yang ada tersebut digambarkan seperti pada gambar 3.25



Gambar 3. 12. Class Diagram

Tabel 3. 15 Deskripsi Class Diagram

Backend		
Class	**Jenis Class**	**Deskripsi**
Login	Control	Kelas login merupakan kelas yang

		berisikan metode – metode yang berfungsi untuk menentukan hak akses untuk mengelola data pada database server.
Files	Entity	Kelas Files merupakan kelas berisikan metode-metode untuk mengelola data pada database server
Form	Interface	Kelas yang berisikan metode- metode yang berfungsi menampilkan halaman untuk mengelola data pada *database server*.
Database Server	Control	Kelas Database Server merupakan kelas yang berisikan metode untuk mengkoneksikan *database server* dengan kelas – kelas yang terdapat pada aplikasi server.
Frontend		
Class	**Jenis Class**	**Deskripsi**
DefaultInitializationErrorHandler	Control	Kelas ini khusus mengimplementasikan antarmuka IQCARErrorHandler (interface QualComm Augmented Reality ErrorHandler).
VideoBackgroundBehaviour	Control	Kelas ini berfungsi untuk me-render background video asli
BackgroundPlaneBehaviour	Control	Kelas ini menciptakan mesh(penyaring) pada kamera sampai fungsi VideoBackground di jalankan.
HideExcessAreaBehaviour	Control	Kelas ini menciptakan empat object Plane di dekat object plane kliping kamera untuk menyembunyikan objek augmentation pada batas latar belakang video.
QCARBehaviour	Control	Kelas ini berfungsi dalam proses pelacakan dan memicu video yang asli..
DataSetLoad	Control	Kelas ini secara otomatis memuat dan

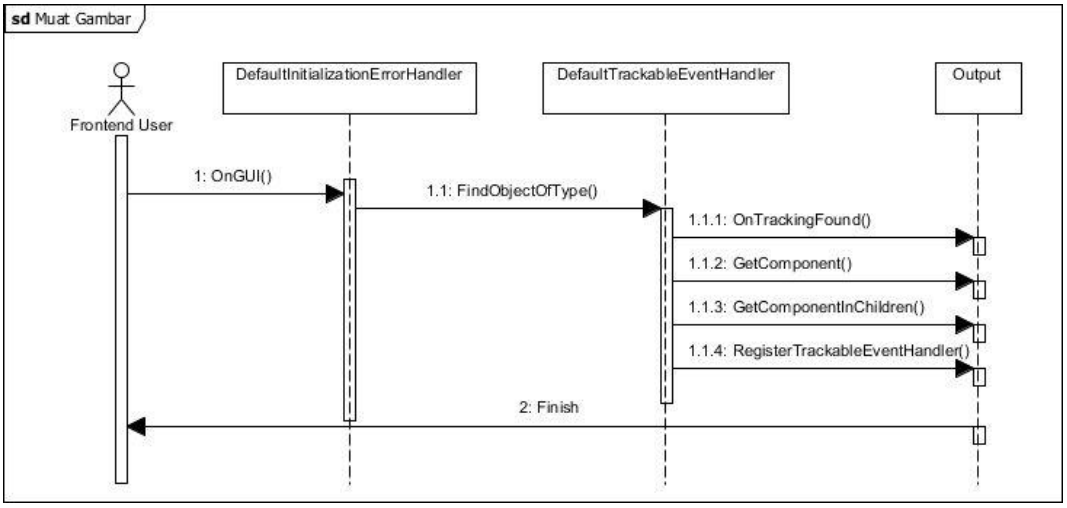
		mengaktifkan satu atau lebih DataSet pada startup
DefaultTrackableEventHandler	Control	Kelas ini khusus untuk mengimplementasikan antarmuka ITrackableEventHandler.
ObjectTarget Behaviour	Control	Kelas ini berfungsi sebagai definisi augmentasi untuk ObjectTarget di dalam editor.
Output	Control	Kelas ini berisikan object Augmented Reality yang akan ditampilkan.
MeshRenderer	Control	Kelas untuk me-render Object Mesh
TextMesh	Control	Object Mesh bertipe Text
QuadMesh	Control	Object Mesh bertipe Quad

3.3.3 Sequence Diagram

Sequence diagram adalah diagaram yang menggambarkan interaksi antar masing-masing objek pada setiap *use case* dalam urutan waktu. Interaksi ini berupa pengiriman serangkaian data antar objek-objek yang saling berinteraksi.

3.3.3.1 Sequence Diagram Muat Gambar

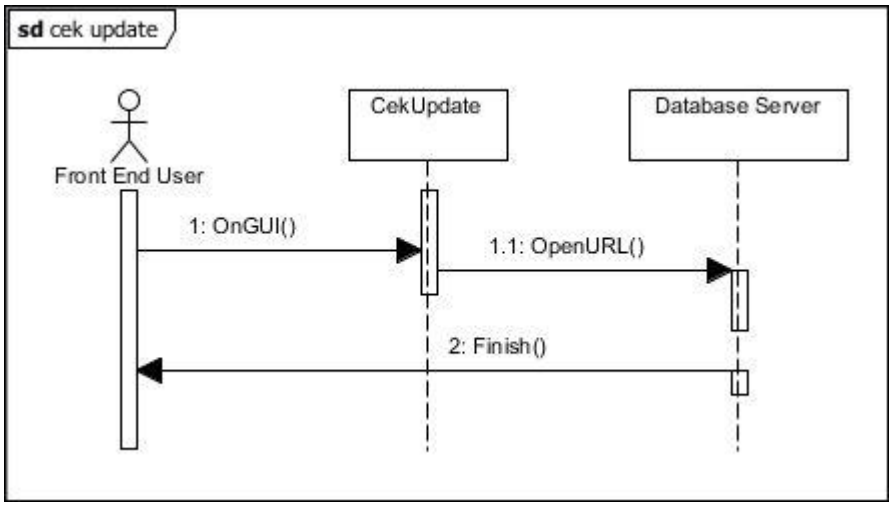
Sequence diagram Muat Gambar Melalui Mata Kamera menunjukkan proses menu yang dipilih untuk menyorot wajah karakter wayang golek yang bertujuan untuk memuat gambar yang kemudian diproses guna menghasilkan output informasi berupa teks tentang karakter wayang golek dengan menggunakan teknologi *Augmented Reality*. *Sequence diagram* Muat Gambar Melalui Mata Kamera dapat dilihat pada Gambar 3.17



Gambar 3. 13 Sequence Diagram Muat Gambar

3.3.3.2 Sequence Diagram Cek Update

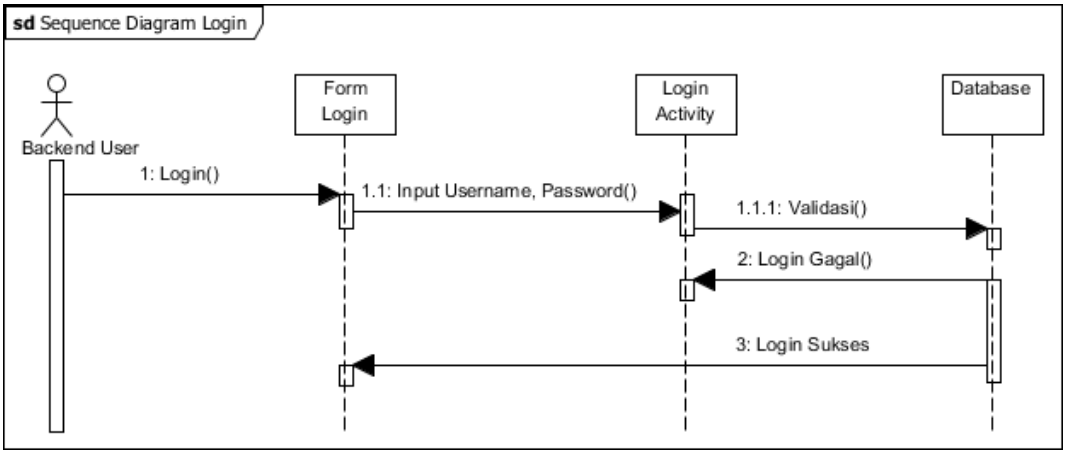
Sequence diagram Update Database menunjukkan rangkaian pesan yang dikirim antara user dalam melakukan proses Update Database. *Sequence diagram* Update Database dapat dilihat pada Gambar 3.18



Gambar 3. 14 Sequence diagram Cek Update

3.3.3.3 Sequence Diagram Login

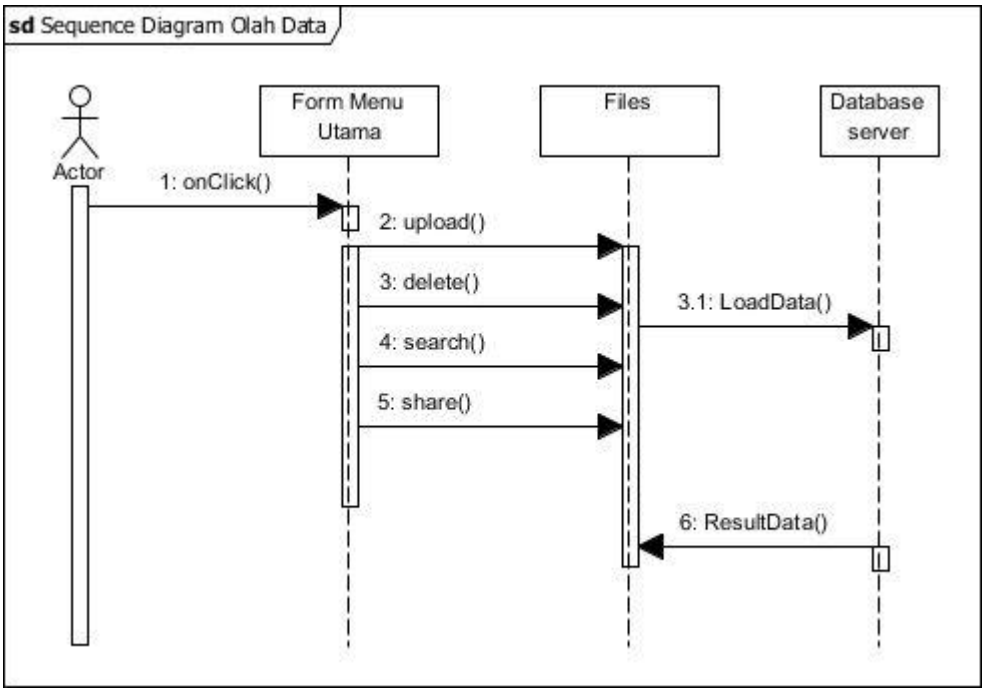
Sequence diagram Login menunjukkan rangkaian pesan yang dikirim antara *programer* dalam sistem Login. *Sequence diagram* Login dapat dilihat pada Gambar 3.19.



Gambar 3. 15 Sequence diagram Login

3.3.3.4 Sequence Diagram Files

Sequence diagram Files menunjukkan rangkaian pesan yang dikirim antara programer dengan sistem pengolahan data yang terdiri dari *upload, delete, search* data dan *share. Sequence diagram* Files dapat dilihat pada Gambar 3.20.



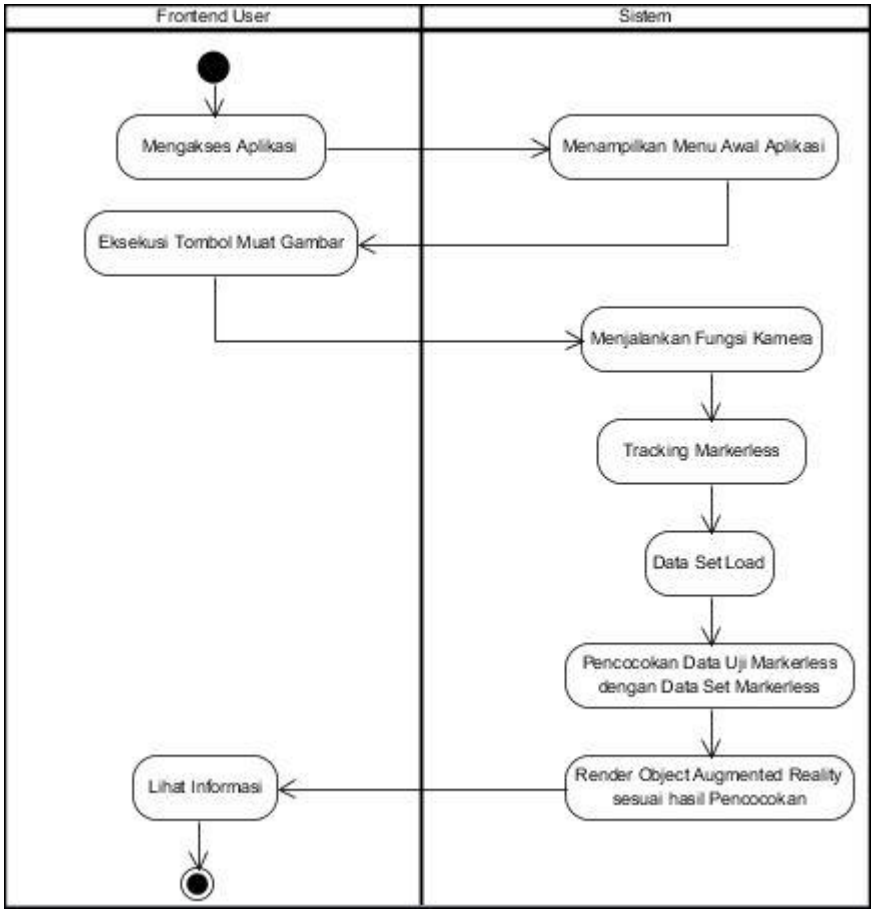
Gambar 3. 16 *Sequence diagram* Files

3.3.4 Activity Diagram

Activity diagram adalah diagram yang memodelkan aliran kerja atau *workflow* dari urutan aktivitas dalam suatu proses yang mengacu pada *use case diagram* yang ada. Berikut ini penjelasan dari tiap *activity diagram:*

3.3.3.1 Activity diagram Muat Gambar

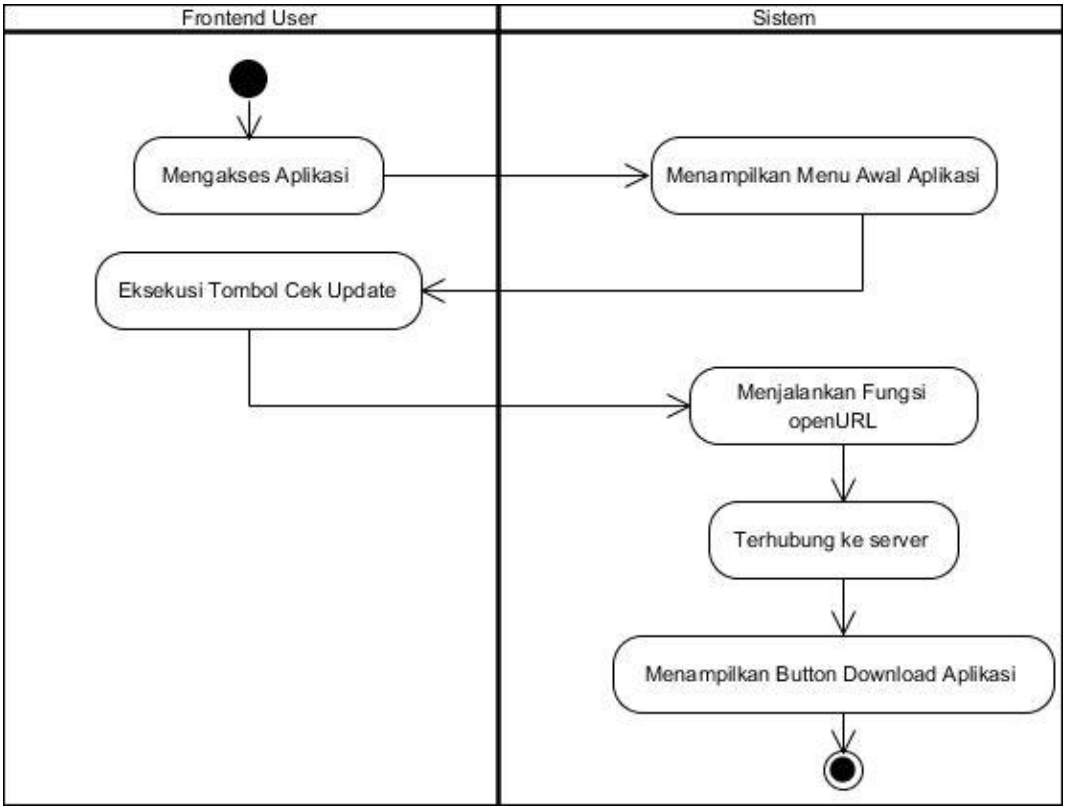
Activity diagram Muat Gambar menjelaskan aliran kerja aktor pengguna pada saat monyorot suatu tempat guna mendapatkan image yang akan diproses dalam aplikasi *mobile*. *Activity diagram* Muat Gambar dapat dilihat pada gambar 3.21.



Gambar 3. 17 Activity diagram Muat Gambar

3.3.3.2 Activity Diagram CekUpdate

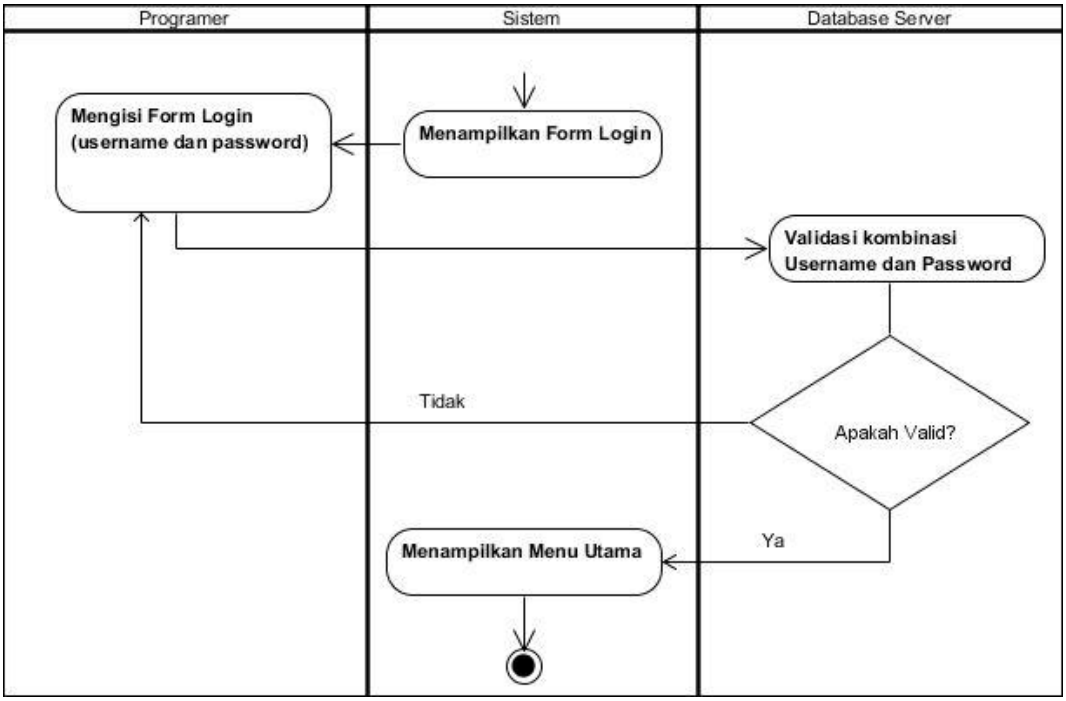
Activity diagram CekUpdate menjelaskan aktivitas yang dilakukan user ketika akan melakukan proses update Versi Aplikasi. *Activity diagram CekUpdate* dapat dilihat pada gambar 3.22.



Gambar 3. 18 Activity Diagram Update Database SQLite

3.3.3.3 Activity Diagram Login

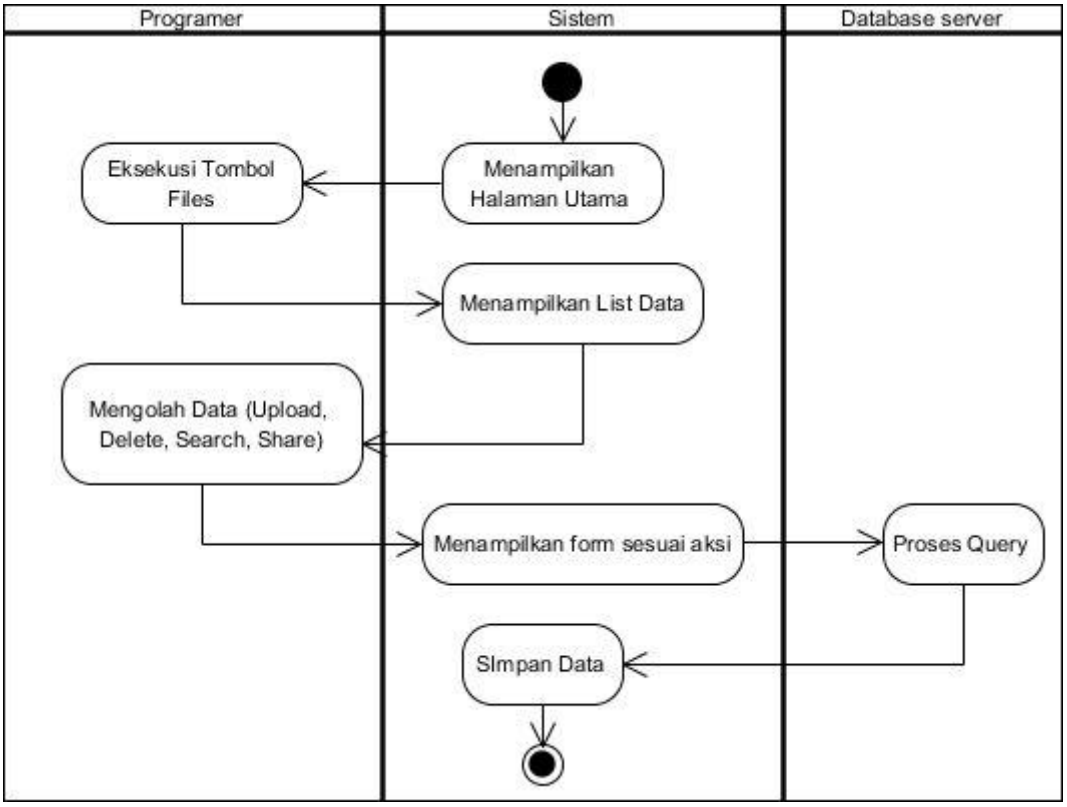
Activity diagram Login-Admin menjelaskan aliran kerja admin pada saat memilih menu Login. *Activity diagram Login*-Admin dapat dilihat pada gambar 3.23



Gambar 3. 19 Activity Diagram Login

3.3.3.4 Activity Diagram Olah Data

Activity diagram Olah Data menjelaskan aliran kerja web administrator pada saat mengolah data dalam database server. *Activity diagram* Olah Data dapat dilihat pada gambar 3.24



Gambar 3. 20 Activity Diagram Olah Data

3.4 Perancangan Sistem

Perancangan Sistem adalah kegiatan yang dilakukan setelah tahap analisis terhadap sistem selesai dilakukan. Perancangan sistem 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 digambarkan sebagai proses multi-langkah dimana representasi struktur data, struktur program, karakteristik *interface*, dan detail prosedur, disintesis dari persyaratan informasi.

3.4.1 Perancangan Data

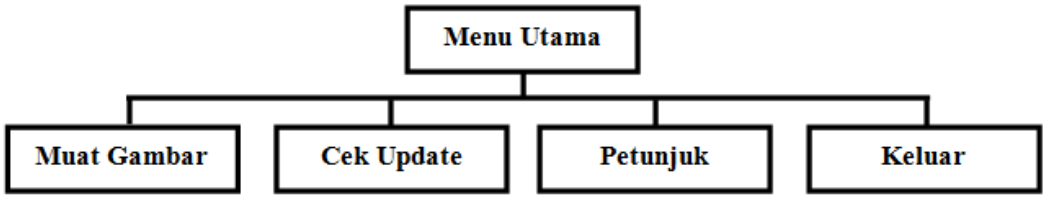
Perancangan data merupakan tahapan untuk memetakan model konseptual ke model database yang akan digunakan. Perancangan data dari aplikasi *mobile* merupakan perancangan struktur tabel dari tabel kategori. Struktur tabel kategori menggambarkan detail tabel yang berisi field, tipe data, panjang data, dan keterangan lainnya. Berikut ini gambaran struktur tabel pengaturan database sistem (SQLite) dan database web Service.

3.4.2 Struktur Menu

Perancangan struktur menu merupakan gambaran jalur pemakaian aplikasi. Perancangan struktur menu dari aplikasi *ini* dapat dilihat pada gambar 3.28 dan 3.29.

3.4.3.1 Struktur Menu Frontend user

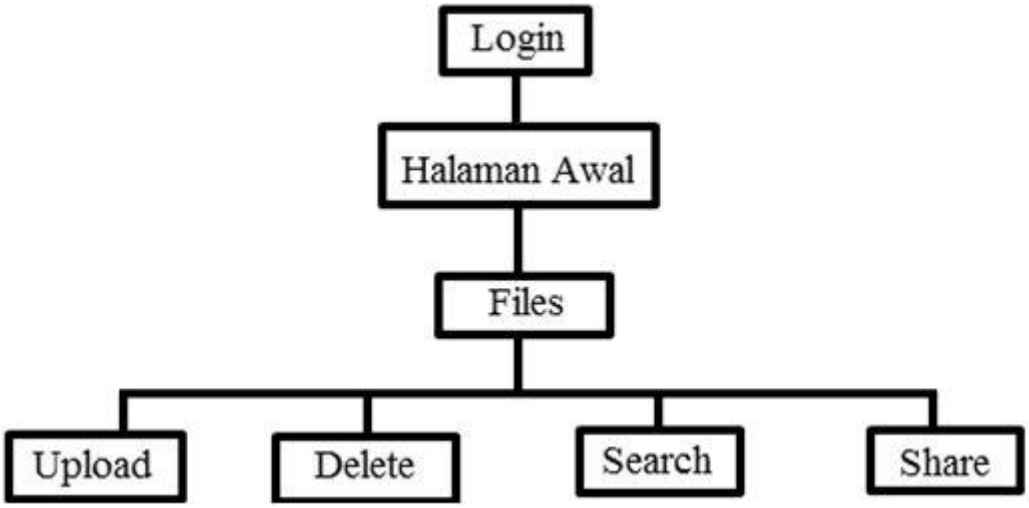
Perancangan struktur menu dari aplikasi *ini* dapat dilihat pada gambar 3.28



Gambar 3. 21 Struktur Menu Aplikasi Untuk Frontend User

3.4.3.2 Struktur Menu Aplikasi Backend user

Perancangan struktur menu dari aplikasi *ini* dapat dilihat pada gambar 3.29



Gambar 3. 22 Struktur Menu Aplikasi Untuk Backend User

3.4.3 Perancangan Antarmuka

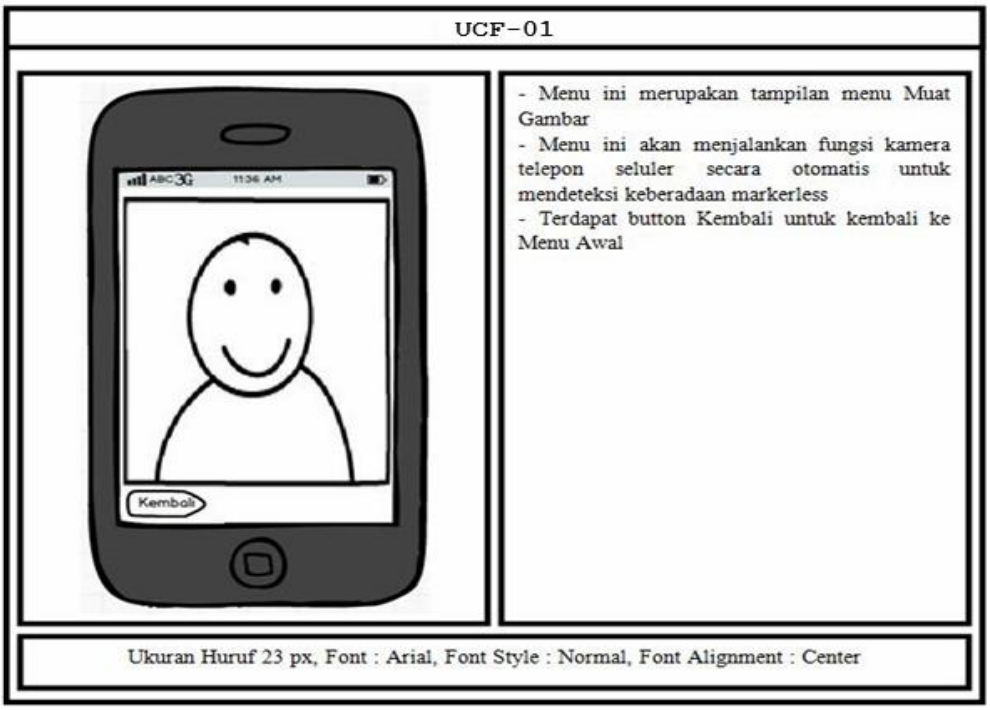
Perancangan antarmuka (*interface*) menggambarkan bagaimana aplikasi berkomunikasi dengan dirinya sendiri, dengan sistem yang berinteroperasi dengannnya, dan dengan penggunanya.

3.4.4.1 Perancangan Tampilan

Perancangan *form* mendeskripsikan rencana tampilan dari setiap *form* yang akan digunakan pada tampilan aplikasi *ini.* Gambaran perancangan antarmuka aplikasi *mobile* ini dapat dilihat pada gambar 3.30 sampai 3.33.

3.4.4.1.1 Perancangan Form Muat Gambar

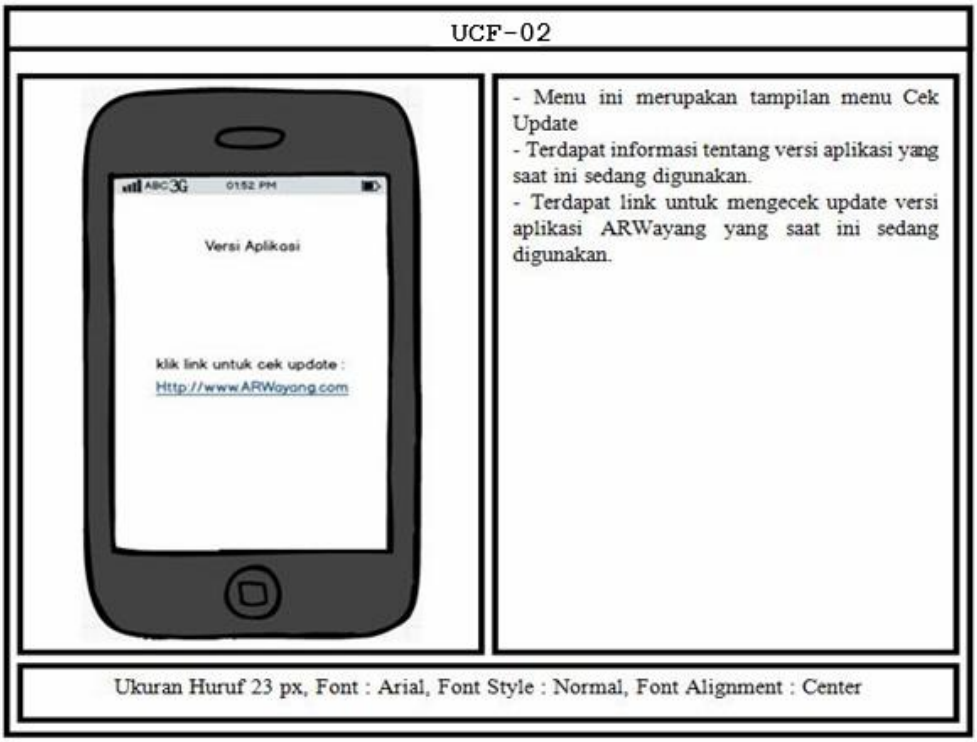
Gambaran umum dari perancangan antarmuka form Muat Gambar pada aplikasi *ARWayangGolek* ini dapat dilihat pada gambar 3.30.



Gambar 3. 23 Form Muat Gambar Melalui Mata Kamera

3.4.4.1.2 Perancangan Form Cek Update

Gambaran umum dari perancangan antarmuka form Cek Update pada aplikasi *ARWayangGolek* ini dapat dilihat pada gambar 3.31.



Gambar 3. 24 Form Cek Update

3.4.4 Jaringan Semantik

Jaringan semantik sistem yang akan dibangun dibedakan sesuai dengan jenis penggunanya, yaitu Frontend user dan Backend user. Jaringan semantik sistem dapat dilihat pada gambar 3.34 sampai 3.35.

3.4.5.1 Jaringan Semantik Frontend User

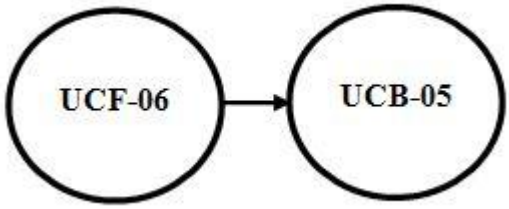
Jaringan semantik Frontend User dapat dilihat pada gambar 3.34



Gambar 3. 25 Jaringan Semantik Aplikasi Frontend User

3.4.5.2 Jaringan Semantik Backend User

Jaringan semantik Backend User dapat dilihat pada 3.35



Gambar 3. 26 Jaringan Semantik Aplikasi Frontend User