# BAB 3

# ANALISIS DAN PERANCANGAN SISTEM

## 3.1 Analisis Sistem

Analisis Sistem adalah penguraian dari suatu masalah yang utuh ke dalam bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikannya. Tahap analisis sistem ini merupakan tahap yang sangat kritis dan sangat penting, karena kesalahan di dalam tahap analisis akan menyebabkan juga kesalahan di tahap selanjutnya. Hasil akhir dari analisis sistem adalah deskripsi dan pemetaan sistem yang akan dibangun yang digambarkan dengan diagram alur.

### 3.1.1 Analisis Masalah

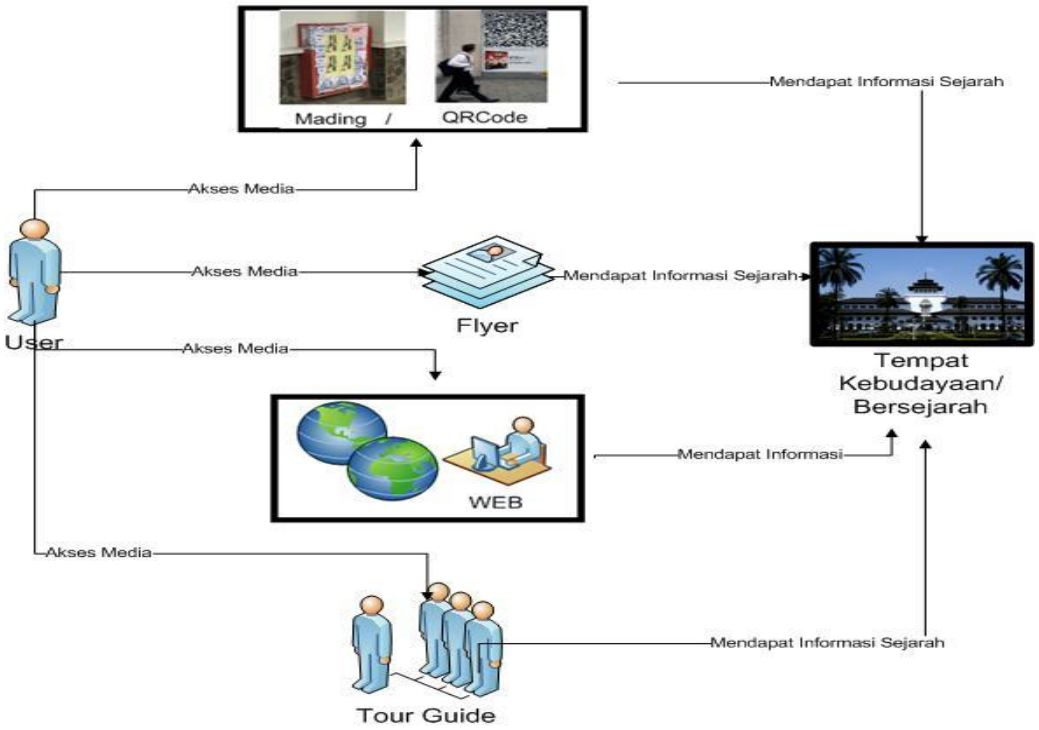
Analisis masalah adalah langkah awal dari analisis sistem. Langkah ini diperlukan untuk mengetahui pemasalahan apa saja yang terjadi didalam sistem yang telah berjalan.

Analisis masalah yang dimaksud disini adalah sulit diperolehnya media yang ada untuk memperkenalkan sejarah suatu tempat kebudayaan di Jawa Barat, khususnya kota Bandung. Banyaknya media yang tersedia**,** namun masih banyak kekurangan baik dari sisi media secara fisik maupun situasi dan kondisi yang kurang memungkinkan untuk mendapatkan informasi mengenai sejarah tempat kebudayaan tersebut.

Dengan 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. Merujuk terhadapa kekeurangan yang ada maka dibangun aplikasi mobile dengan menggunakan teknologi Augmented Realty (AR). Aplikasi ini bertujuan menampilkan informasi secara real time berupa text di layar telepon. Identifikasi pengenalan gambar tempat menggunakan *Library ARLab* dimana *Library* ini menggunakan metode *image processing* dan *Metode Euclidean Distance* untuk pengenalan gambar sehingga *gambar* yang diambil dari kamera telepon seluler dapat dikenali sebagai *markerless*. *Markerless* AR inilah yang berfungsi untuk menampilkan informasi tersebut secara real time di layar telepon seluler, khususnya android.

### 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. Analisa Prosedur pada proses media yang sedang berjalan sekarang adalah sebagai berikut:




**Gambar 3. 1 Alur sistem yang sedang berjalan**

1. User disini adalah pengunjung atau wisatawan sebuah tempat kebudayaan yang ingin mengetahui informasi lebih jelas tentang sejarah gedung/tempat tersebut.
2. Telah banyak media yang tersedia untuk mengetahui informasi sejarah tempat tersebut, diantaranya :
    a. Mading atau QR Code : Media ini dapat mengetahui informasi sejarah tempat kebudayaan dengan proses kerja mendekati sumber informasi untuk mengetahui informasi tempat tesebut. Mading atau QR Code biasanya di tempatkan di posisi-posisi tertentu di tempat kebudayaan tersebut, sehingga pengunjung yang ingin mengetahui informasi tempat tersebut harus mengetahui terlebih dahulu posisi mading atau QR Code yang bisa diakses.

b. *Flyer* atau *Brosur* : Media ini cukup mudah untuk mengetahui informasi sejarah tempat tersebut, namun terkadang tidak semua orang bisa mendapatkan *Brosur* atau kurang menariknya tampilan *Brosur* sehingga pengunjung terkadang mengabaikan info yang ada dalam *Brosur* tersebut.

c. *Web* : Media ini terkoneksi dengan internet, dan untuk mengakses web diperlukan perangkat hardware yang cukup nyaman atau layar akses cukup besar, serta diperlukan alamat web yang yang singkat agar mudah diingat oleh banyak orang agar mempermudah akses web tersebut, sehingga untuk akses web, pengunjung minimal mempunyai perangkat hardware yang memenuhi kriteria dan mengetahui alamat situs web yang menyediakan informasi tempat kebudayaan tersebut.

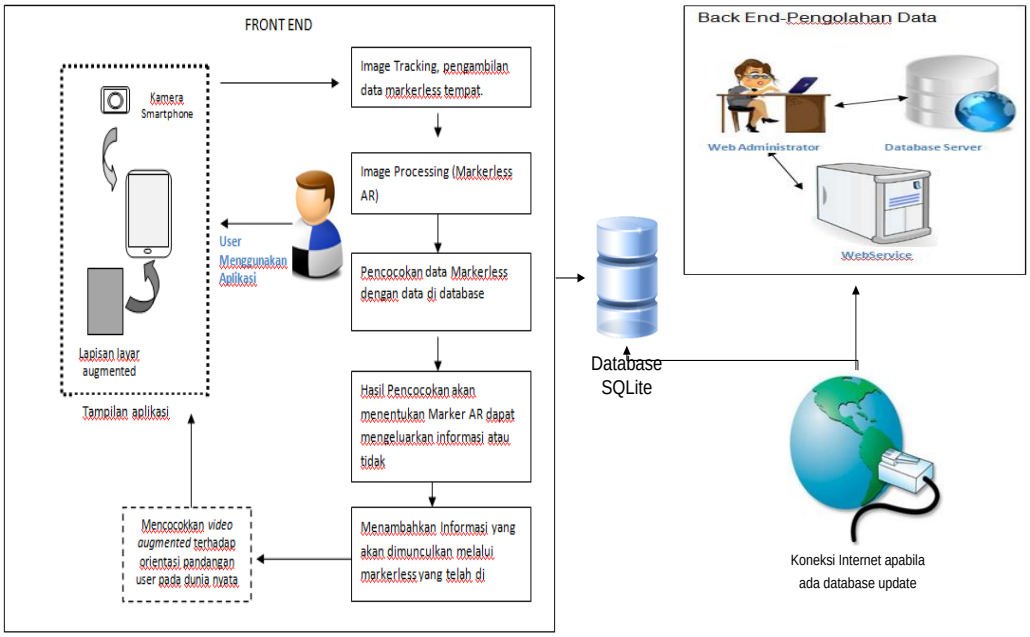
d. *Tour Guide* : Media ini juga cukup efektif berfungsi, namun tidak banyak orang yang mengunjungi tempat kebudayaan di damping oleh *tour guide* yang memberikan penjelasan mengenai tempat bersejarah tersebut.

Dari gambaran prosedur terlihat telah banyak media yang dapat memperkenalkan sejarah tempat kebudayaan namun masih terdapat kekurangan diantaranya tempat informasi yang kurang diketahui oleh pengunjung, tampilan browser atau flyer yang kurang menarik serta akses web yang terkadang bermasalah di jaringan koneksi dan alamat web yang kurang diketahui oleh user.

### 3.1.3 Analisis Sistem yang Akan Dibangun

Sistem yang akan dibangun adalah suatu aplikasi *mobile* menggunakan *platform* Android dengan teknologi Augmented Reality. Teknologi dengan aplikasi mobile 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 suatu tempat kebudayaan secara *real time.*

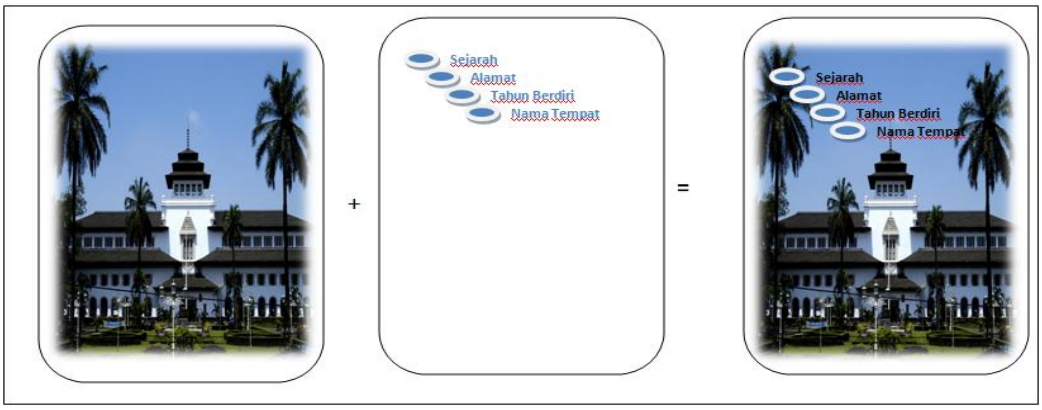
Proses Aplikasi Augmented Reality yang akan dibangun menggunakan sistem markerless yang disupport oleh Lib ARLab. Gambar 3.2 menunjukan deskripsi proses sistem yang akan dibangun.




**Gambar 3. 2 Alur Sistem yang akan Dibangun**

1. User adalah pengguna aplikasi. Aplikasi ini dapat berjalan dengan koneksi internet ataupun tanpa koneksi internet.
2. Pada saat user menjalankan aplikasi, aplikasi ini akan mejalankan kamera telepon seluler secara otomatis untuk mengambil gambar yang nantinya akan diproses.
3. Gambar yang diperoleh akan diproses menggunakan *Library ARLab*, sehingga akan mendapatkan hasil yang akan dibandingkan dengan nilai gambar yang berada di database.
4. Gambar yang di ambil dari kamera telepon seluler dibandingkan dengan nilai gambar di dengan data nilai di data database. Apabila nilai gambar yang diambil dari kamera telepon seluler tersebut nilainya paling kecil, gambar ini dinyatakan memiliki kemiripan pola gambar dengan gambar yang ada di database sehingga gambar ini akan dijadikan sebagai *markerless* dan menampilkan AR berupa informasi dalam bentuk text di layar telepon seluler.
5. Web Admin sebagai *backend*, yang bertujuan untuk mengolah data, seperti penambahan, pengeditan dan penghapusan data yang tidak digunakan. Apabila ada data baru, aplikasi di sisi *frontend* menyediakan tombol *download* yang bertujuan untuk mengambil data baru tersebut dari *web server*, sehingga data yang ada di SQLite tersebut bertambah.

Tahap ini merupakan hasil akhir penggabungan yang akan dilihat oleh pengguna di layar telepon seluler yang digunakan. Pada gambar 3.3 adalah gambar tampilan aplikasi yang akan dibangun.




**Gambar 3.3 Proses Gabungan text dan layar *augmented reality***

### 3.1.4 Analisis *Marker*

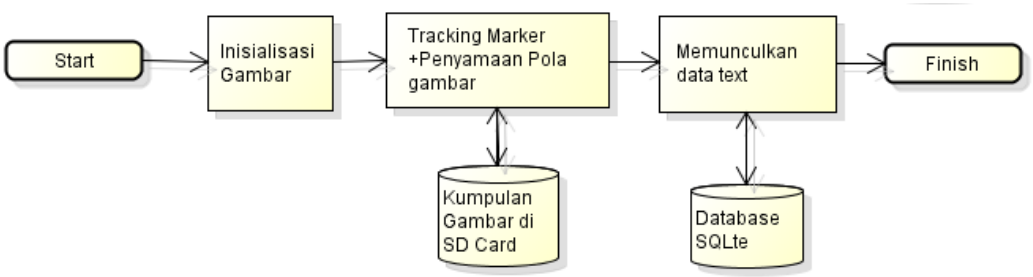
*Markerless* merupakan salah satu teknologi *tracking* yang disediakan oleh ARLab. *Tracking* jenis ini menggunakan gambar sebagai markernya. Proses *tracking* ini menggunakan tekstur gambar (berkas jenis gambar) yang disimpan dalam database sebagai sumber referensinya, dan membandingkan tekstur yang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

### 3.1.5 Analisis Penentuan *Markerless*

Analisis metode terhadap kasus merupakan analisis yang mendeskripsikan bagaimana proses *augmented reality* ARLab dari awal *inisialisasi*, *tracking marker*, sampai dengan proses memunculkan data text dengan metode *markerless* yang diterapkan oleh ARLab.

Dalam perancangan aplikasi dengan teknologi AR, menggabungkan objek *virtual* dengan objek nyata, dalam hal ini objek *virtual* berupa objek 2D atau 3D dan objek nyatanya berupa gambar

dengan pola tertentu (*markerless*). SDK dari ARLab ini menggunakan metode *Euclidean Distance* untuk proses pengenalan gambar.



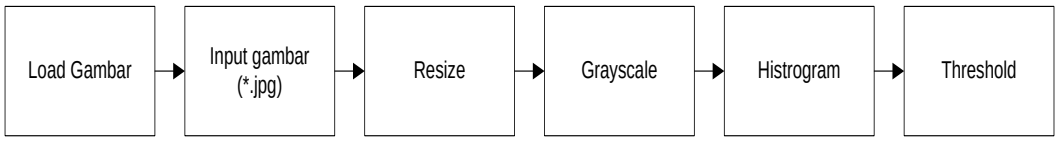

**Gambar 3.4 Alur Sistem Aplikasi *Frontend Augmented Reality***

Secara keseluruhan, proses sistem *augmented reality* dapat digambarkan dengan diagram alur pada gambar 3.4. Proses pengenalan deteksi *markerless* ini melalui beberapa tahapan. Secara garis besarnya, dalam perancangan ada tiga bagian utama yaitu :

1. *Inisialisasi* Gambar
2. *Tracking Markerless*
3. Memunculkan data text

### 3.1.5.1 *Inisialisasi* Gambar

Pada tahap ini ditentukan *proses* dimana ARLab berkerja untuk mengambil gambar sebagai data masukan yang akan di proses




**Gambar 3.5 Alur Sistem Proses Inisialisasi Gambar**

1. ***Load* Gambar**

Proses *inisialisasi* berawal dari pengambilan gambar secara otomatis menggunakan kamera telepon seluler. Sistem berjalan disupport oleh *Library* dari ARLab.

```
surfaceView=(SurfaceView)findViewById(R.id.surfaceView);
holder=surfaceView.getHolder();
holder.addCallback(this);
holder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
```

2. ***Input* Gambar**

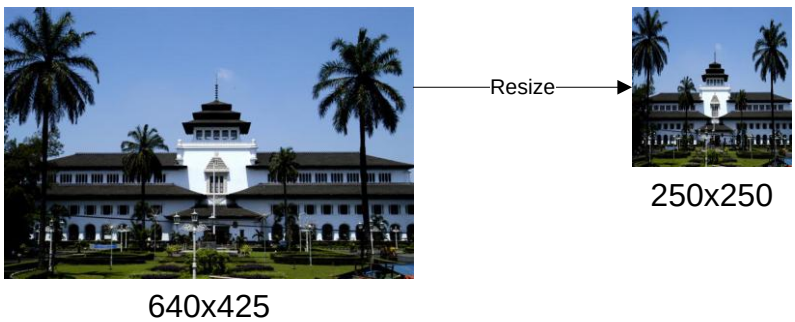
Pada saat aplikasi berjalan, kamera melakukan proses pengambilan gambar dalam format *.jpg.

```
/**Set the matching type. */
 aRmatcher.setMatchingType(ARmatcher.IMAGE_MATCHER);
```

Gambar masukan berbentuk format *.jpg diproses oleh library ARlab dalam class Image_Matcher.

3. ***Resize***

Gambar yang dijadikan sebagai markereles akan di *resize* terlebih dahulu. Gambar di *resize* menjadi 250x250 (Ketentuan dari ARLab).




**Gambar 3.6 Proses *Resize* Gambar.**

**Algoritma *Resize* :**

```
Read(image,imgx,imgy)
Max_x<- 250 (maksimal nilai lebar gambar)
Max_y-<250(maksimal nilai panjang gambar)
If imgx>=Max_x then
 Scale = Max_x div imgx
 new_imgx = imgx * scale
 new_imgy = imgy *scale
endif
If new_imgy > Max_y then
       Scale = Max_y/new_imgy
       new_imgx = new_imgx*scale
       new_imgy = new_imgy*scale
 endif
 If imgx<Max_x then
       new_imgx = imgx;
 If imgy<Max_y then
       new_imgy = imgy;
```

*4.* ***Grayscale***

*Grayscale* merupakan proses mengubah gambar menjadi monokrom dengan nilai intensitas 0 untuk hitam, 255 untuk putih, dan abu-abu untuk nilai yang berada pada rentang nilai 0 sampai 255. Dimana rumus secara garis besarnya adalah new pixel = (Red + Green +Blue )/3.



RGB Grayscale

**Gambar 3.7 Proses *Grayscale* Gambar**

**Algoritma *Grayscale***

```
Read (image,x,y)
For (i<-0 ; i<x; i++) do
   For (j<-0; j<y; j++) do
     Getpixel(i,j);
     Int grayscale = (int)((originalcolor.R)+
(originalcollor.G)+(originalcolor.Y)/3)
    Newimage(i,j,grayscale)
End for End for
```

5. ***Histogram***

Proses *Histogram* adalah suatu proses perataan *histogram*, dimana distribusi derajat keabuan pada suatu gambar dibuat rata.



Grayscale Histogram

**Gambar 3.8 Proses *Histogram* Gambar**

**Algoritma *Histogram***

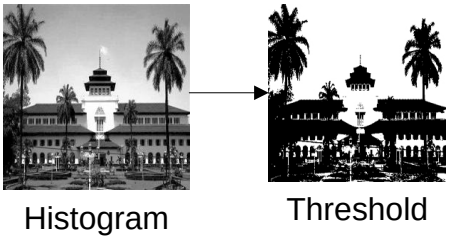
```
Deklarasi : (A,int N,int M) /*A=array citra N,
M=panjang,Lebar gambar*/
  /*inisialisasi Hist[0,255] dengan 0*/
  for(i=0;i<=255;i++)
    Hist[i]=0
  endfor
  /*mencacah frekuensi kemunculan*/
```

```
for(k=0;k<=255;k++)
for (i=0;i<=N-1;i++)
  for(j=0,j<=M-1;j++)
        if(A[i,k]==i-1)
              Hist[i]=Hist[i]+1
        endif
  endfor
endfor
endfor
```

6. **Ambang Batas (*Threshold*)**

Untuk *Library ARLab*, Threshold ditentukan besarnya agar gambar yang diambil dapat masuk kedalam pool untuk diproses lebih lanjut.

Proses *threshold* ini yang menjadi acuan untuk proses selanjutnya yaitu pencocokan pola gambar. Dalam *Library ARLab* ukuran *threshold* dapat diatur sesuai keakuratan gambar pada saat proses pencocokan pola dengan jarak tertentu. Berikut contoh perubahan gambar dari *Histogram* ke gambar *Threshold* dapat dilihat di gambar 3.9




**Gambar 3.9 Proses Threshold Gambar**

Gambar *Threshold* tersebut kemudian diproses untuk mendeteksi gambar dan pencocokan pola menggunakan metode *Euclidean Distance* yang telah disediakan pada SDK ARLab.

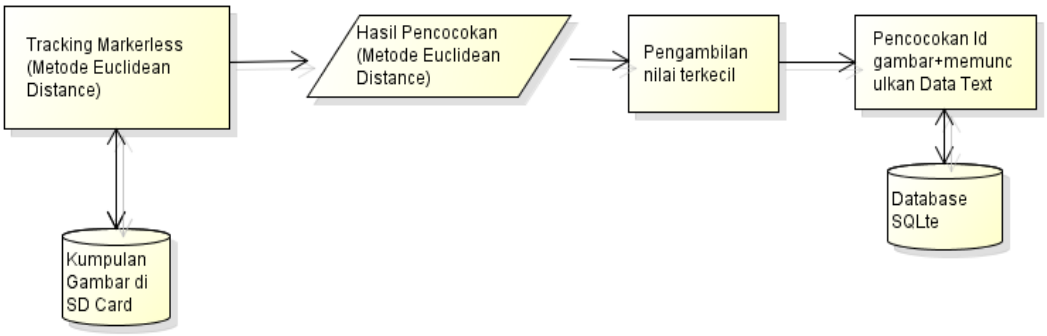
**Algoritma *Threshold***

```
Tmean = 0 /*deklarasi nilai awal t*/
for (k = 0; k < level; k++)
```

```
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
```

### 3.1.5.2 *Tracking Markerless*

Proses tracking yang terjadi adalah dengan pembandingan texture gambar pada database dengan tampilan pada *marker*. Seperti terlihat pada gambar 3.10.

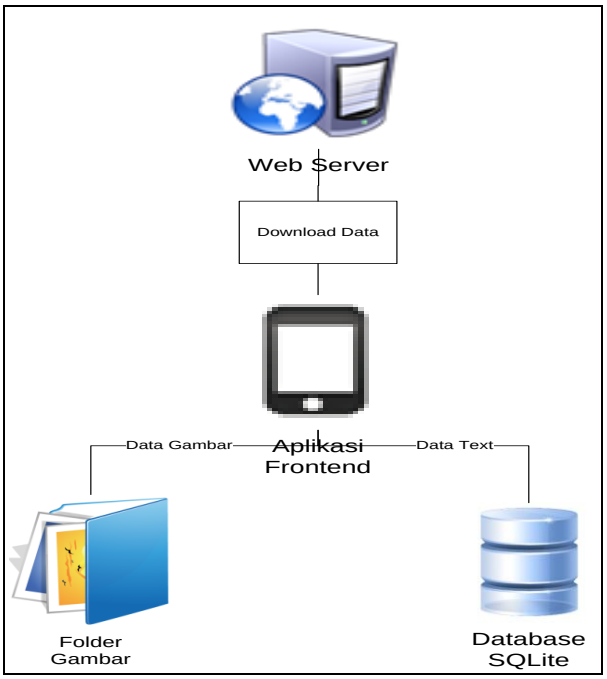



**Gambar 3.10 Proses kerja Tracking Marker ARLab**

*Tracking Markerless* menggunakan Library ARLab dengan SDK hasil dari implementasi Metode *Euclidean Distance* untuk menentukan apakah suatu gambar dikenali atau tidak. Gambar uji dihitung jarak euclidnya terhadap setiap vektor rata-rata gambar yang ada di database. Jika jarak minimum gambar tersebut didapatkan nilainya mendekati atau nilainya

paling kecil maka gambar tersebut dikenali dan akan menampilkan data text sesuai dengan database.

Data yang dijadikan acuan untuk pencocokan data diambil dari web server. Untuk data gambar akan disimpan dalam external data telepon seluler sedangkan data text dapat disimpan di SQLite. Tujuan penyimpanan data ini dibagi dua untuk mempermudah proses *tracking marker* dan pencocokan data.




**Gambar 3. 11 Proses penyimpanan data Aplikasi Frontend**

Untuk proses penamaan gambar didatabase disesuaikan dengan data text di SQLite. Satu Objek/tempat mempunyai 3 gambar acuan. Tabel 3.1 menunjukan proses penamaan gambar yang ada di folder *external memory* telepon seluler.

**Tabel 3.1 Format Penamaan Gambar**

| Gambar | Penamaan Gambar | IDGambar (DB Gambar di SQLite) | Id Marker (DB Marker di Sqlite) |
|---|---|---|---|
|  | 1.jpg | 1 | 1 |
|  | 2.jpg | 2 | 1 |
|  | 3.jpg | 3 | 1 |
|  | 4.jpg | 4 | 2 |
|  | 5.jpg | 5 | 2 |
|  | 6.jpg | 6 | 2 |

Pengurutan berdasarkan penamaan gambar Idgambar *.jpg sesuai dengan penginputan yang dilakukan diawal.

Misalkan kode data “1.jpg” dapat dijelaskan sebagai berikut:

1. “1” = merupakan nama field nm_gambar(table gambar di DB SQlite) yang disamakan dengan Idgambar(table gambar di DB SQlite), agar mudah dicocokan dengan Idmarker(table marker di DB SQlite) dengan INNERNJOIN Syntax pada saat pencocokan data pada saat perbandingan gambar.
2. “.jpg” = penamaan sesuai dengan syntax yang dilampirkan dengan tujuan melampirkan type data pada SDCard.

Sedangkan untuk pencocokan gambar yang diambil oleh kamera telepon seluler dengan gambar yang ada di External memory telepon seluler menggunakan metode Euclidean Distance (Ketentuan ARLab). Sebagai gambaran Perhitungan jarak dapat dihitung menggunakan rumus :

$$ED(X,Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n \quad y_n)^2} \qquad (2.8)$$

Sebagai contoh perhitungan yang dijabarkan :

1. Bagi Data Gambar menjadi dua bagian, *A=[a1,a2,…ap]* dan *B=[b1,b2…bp]* dimana *ai=[ai1,ai2,…ain]* dan *bj=[bi1,bi2,…,bim]*, dimana A adalah gambar dalam database sedangkan B adalah gambar uji yang diambil oleh kamera telepon seluler.

a. Gambar Dalam database

Tabel 3.2 Tabel Gambar Acuan dalam Database

| Nama Gambar | Gambar |
|---|---|
| Gambar I |  |
| Gambar II |  |
| Gambar III |  |

b. Uji (pengambilan gambar oleh kamrea telepon seluler)

**Tabel 3.3 Tabel Gambar Uji**

| Nama Gambar | Gambar |
|---|---|
| Gambar Ia |  |

c. Ambil nilai pixel gambar, sebagai contoh nilai pixel diambil dari titik kordinat (x,y) dan ordo gambar 3x3:

**Tabel 3.4 Tabel Kordinat pixel**

| (x[..],y[..]) | (x[..],y[..]) | (x[..],y[..]) |
|---|---|---|
| 50,90 | 51,90 | 52,90 |
| 50,91 | 51,91 | 52,91 |
| 50,92 | 51,92 | 52,92 |

d. Penjabaran nilai pixel gambar

1. Gambar A( Data Gambar di Database)

   Data gambar dan nilai pixel acuan di database dapat dilihat pada table 3.5

**Tabel 3.5 Gambar dan nilai pixel gambar acuan dalam database**

| Nama Gambar | Gambar | Nilai Pixel |
|---|---|---|
| Gambar I |  | 0 228 225<br>0 228 225<br>0 228 225 |
| Gambar II |  | 225 225 225<br>225 225 225<br>225 225 225 |
| Gambar III |  | 0 225 225<br>0 225 225<br>0 225 225 |

2. Gambar B (Gambar Yang diambil oleh Kamera)

Data gambar dan nilai pixel acuan gambar uji dapat dilihat pada table 3.6

**Tabel 3.6 Tabel Data dan nilai pixel gambar uji**

| Nama Gambar | Gambar | Nilai Pixel |
|---|---|---|
| Gambar Ia |  | 225 225 225<br>225 225 119<br>0 0 0 |

2. Buat vektor rata-rata A=(a1,a2,…ap) untuk masing-masing kelas dari data pelatihan persamaan 2.9

a. Data Gambar di Database

Data Gambar didatabase di proses untuk menghasilkan nilai vektor rata-rata. Pada tabel 3.7 dapat dilihat untuk proses perhitungan nilai gambar.

**Tabel 3.7. Proses Vektor Rata-Rata Gambar Database**

| Keterangan | Nilai Pixel | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Gambar I | 0 | 228 | 0 | 0 | 228 | 255 | 0 | 228 | 255 |
| Gambar II | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 |
| Gambar III | 0 | 255 | 255 | 0 | 255 | 255 | 210 | 255 | 255 |
| Dijumlahkan | 255 | 738 | 510 | 255 | 738 | 765 | 465 | 738 | 765 |
| Rata-Rata | 85 | 246 | 170 | 85 | 246 | 255 | 155 | 246 | 255 |
| Cari Rata-rata vektor gambar database | | | | | | | | | |
| Vektor Gambar I (Gambar II – rata) | 0 | 228 | 0 | 0 | 228 | 255 | 0 | 228 | 255 |
| | 85 | 246 | 170 | 85 | 246 | 255 | 155 | 246 | 255 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Vektor Gambar II (Gambar II – rata) | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 |
| | 85 | 246 | 170 | 85 | 246 | 255 | 155 | 246 | 255 |
| | 170 | 9 | 85 | 170 | 9 | 0 | 100 | 9 | 0 |
| Vektor Gambar | 0 | 255 | 255 | 0 | 255 | 255 | 210 | 255 | 255 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| III (Gambar II – rata) | 85<br>0 | 246<br>9 | 170<br>85 | 85<br>0 | 246<br>9 | 255<br>0 | 155<br>55 | 246<br>9 | 255<br>0 |

b. Data Gambar Uji (gambar yang diambil oleh Kamera)

Data Gambar didatabase di proses untuk menghasilkan nilai vektor rata-rata. Pada tabel 3.8 dapat dilihat untuk proses perhitungan nilai gambar.

**Tabel 3.8 Proses Vektor Rata-Rata Gambar Uji**

| Keterangan | Gambar | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Gambar Ia | 255 | 255 | 255 | 255 | 255 | 119 | 0 | 0 | 0 |
| Vektor Gambar Ia (Gambar Ia – rata) | 85<br>170 | 246<br>9 | 170<br>85 | 85<br>170 | 246<br>9 | 255<br>0 | 155<br>0 | 246<br>0 | 255<br>0 |

3. Untuk masing-masing data pengujian D=[d1,d2,…dp], hitung jaraknya terhadap vector rata-rata setiap kelas seperti persamaan 2.10

$$ED(X,Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n \quad y_n)^2}$$

**Tabel 3.9 Proses Perhitungan Jarak**

| Keterangan | X | Y | Perhitungan | Hasil |
|---|---|---|---|---|
| Vektor rata Gambar I-Gambar Uji | 0 | 170 | Abs(0-170) | 170 |
| | 0 | 9 | Abs(0-9) | 9 |
| | 0 | 85 | Abs(0-85) | 85 |
| | 0 | 170 | Abs(0-170) | 170 |
| | 0 | 9 | Abs(0-9) | 9 |
| | 0 | 0 | Abs(0-0) | 0 |
| | 0 | 0 | Abs(0-0) | 0 |
| | 0 | 0 | Abs(0-0) | 0 |
| | 0 | 0 | Abs(0-0) | 0 |
| | **ED** | | | 443 |
| Vektor rata Gambar II-Gambar Uji | 170 | 170 | Abs(170-170) | 0 |
| | 9 | 9 | Abs(9-9) | 0 |
| | 85 | 85 | Abs(85-85) | 0 |
| | 170 | 170 | Abs(170-170) | 0 |
| | 9 | 9 | Abs(9-9) | 0 |
| | 0 | 0 | Abs(0-0) | 0 |
| | 100 | 0 | Abs(100-0) | 100 |
| | 9 | 0 | Abs(9-0) | 9 |
| | 0 | 0 | Abs(0-0) | 0 |
| | **ED** | | | 109 |
| Vektor rata Gambar III-Gambar Uji | 0 | 170 | Abs(0-170) | 170 |
| | 9 | 9 | Abs(9-9) | 0 |
| | 85 | 85 | Abs(85-85) | 0 |
| | 0 | 170 | Abs(0-170) | 170 |
| | 9 | 9 | Abs(9-0) | 9 |
| | 0 | 0 | Abs(0-0) | 0 |
| | 55 | 0 | Abs(55-0) | 55 |
| | 9 | 0 | Abs(9-0) | 9 |
| | 0 | 0 | Abs(0-0) | 0 |
| | **ED** | | | 404 |

4. Dari contoh perhitungan diatas jarak minimum tersebut dapat dilihat dari hasil yang telah dijabarkan pada point 3. Untuk metode Euclidean distance, nilai yang paling minimum atau paling kecil setelah proses perbandingan dengan nilai database yang mempunyai nilai kecocokan paling mendekati.

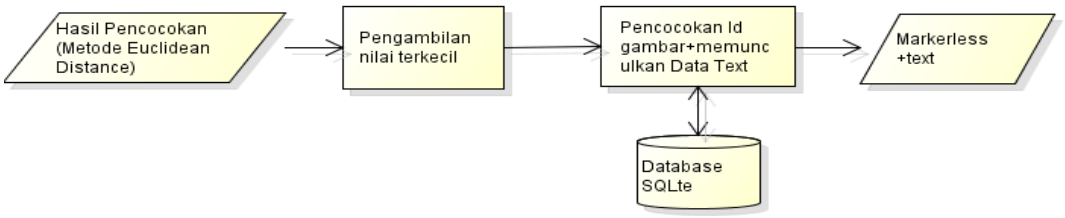
**Tabel 3.10 Hasil Pencocokan Gambar**

| Gambar Database | Nilai Euclidean Distance | Gambar Uji |
|---|---|---|
|  1.jpg | 443 | Tidak Cocok |
|  2.jpg | 109 |  Cocok |
|  3.jpg | 404 | Tidak Cocok |

Dengan kesimpulan bahwa metode yang digunakan SDK ARLab yaitu metode Euclidean Distance menjelas bahwa nilai yang paling minimum yang menunjukan bahwa pola gambar yang diambil oleh kamera telepon seluler cocok dengan pola gambar yang berada didatabase.

### *3.1.5.3* Memunculkan data Text

Hasil pencocokan pola matriks yang dilakukan di step sebelumnya oleh SDK ARLab merupakan data acuan untuk menampilkan objek yang sesuai dengan sebuah data di database SQlite.



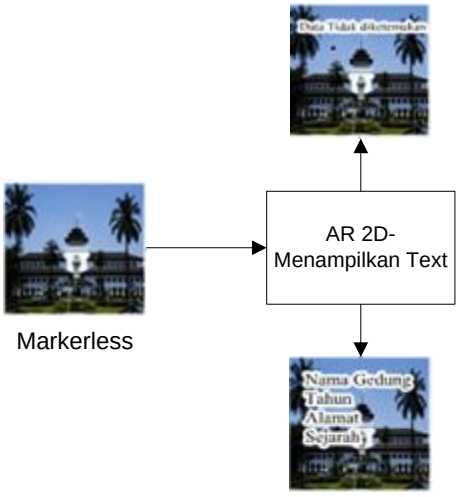

**Gambar 3.12 Proses Menampilkan Text**

ARLab disini menyediakan fungsi tersendiri untuk menampilkan objek 2D. Apabila inputan gambar sesuai dengan data didatabase, Objek 2D berupa text akan menampilkan data-data yang ada didatabase. Tabel 3.10 menampilkan proses untuk memuat text.

**Tabel 3.11 Proses Penampilan Text**

| | | |
|---|---|---|
|  Gambar Uji | Proses Pencocokan |  Gambar 2.jpg |
| Proses Pencocokan Id gambar dengan database | | |
| 1.jpg<br>2.jpg<br>3.jpg | 1 (Idm**ark**er) | Nama, tahun, alamat, sejarah |

Apabila tidak ditemukan nilai idgambar yang sama dengan database ini akan memunculkan Text 2D “Data Tidak Ditemukan”.

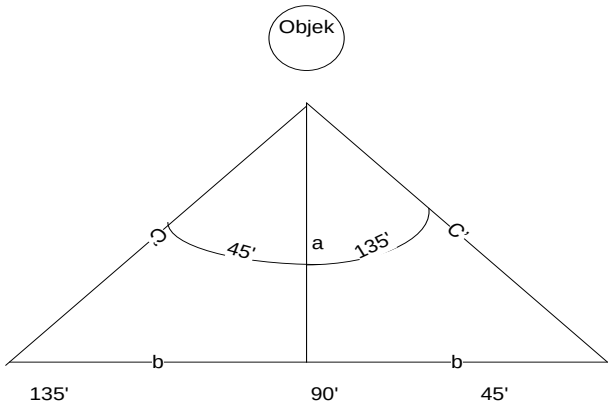



**Gambar 3.13 Hasil Proses Menampilkan Text**

### 3.1.6 Analisis Optimalisasi Sudut dan jarak

Analisis optimalisasi jarak sistem merupakan analisis yang berfungsi untuk menggambarkan secara rinci bagaimana sistem bekerja untuk deteksi

gambar dengan memperhitungkan jarak dan sudut. Untuk sudut dan jarak ditentukan secara eksplisit. Jarak 45° dan 135° menggunakan jarak pytagoras yang dapat dijelaskan di gambar 3.14 :




**Gambar 3.14 Proses perhitungan Jarak dan Sudut**

Sebagai contoh perhitungan :

Jarak : a : 3 meter

b : 5 meter

Panjang Jarak $C = \sqrt{a^2 + b^2}$

$C = \sqrt{3^2 + 5^2}$

C =6 Meter

**Tabel 3.12 Tabel Pengukuran Jarak dan Sudut**

| Gambar | Sudut | Jarak Minimal | Jarak Optimal | Jarak Maksimal |
|---|---|---|---|---|
|  | 45° | ±3 meter | ±6 meter | ±8 meter |

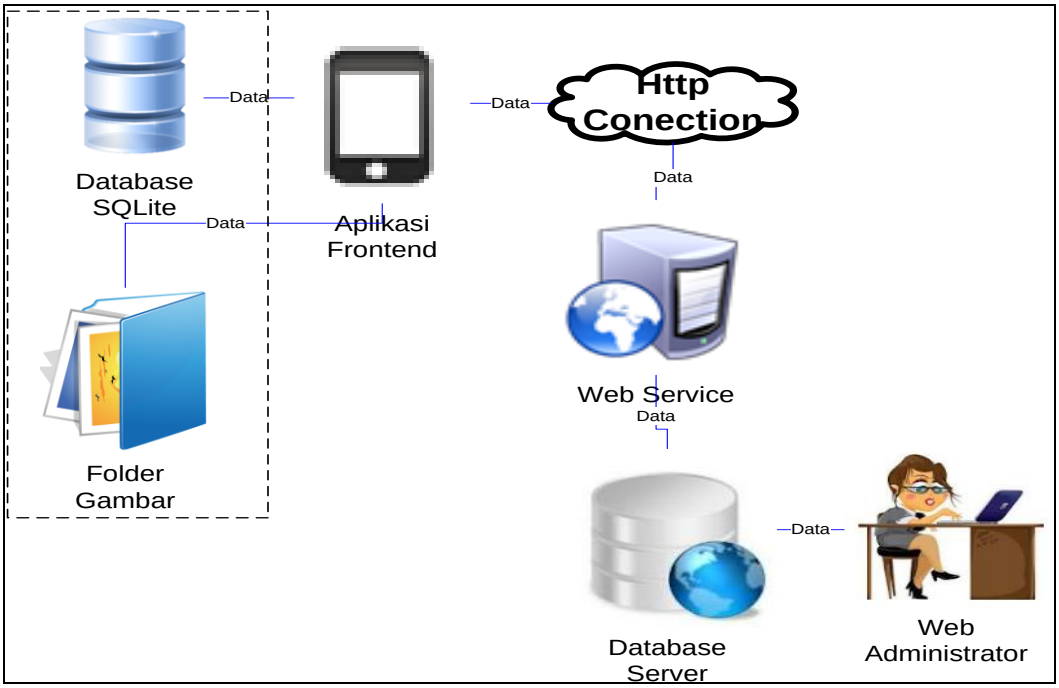
| | | | | |
|---|---|---|---|---|
|  | 90° | ±1 meter | ±3 meter | ±5 meter |
|  | 135° | ±3 meter | ±6 meter | ±8 meter |

Rincian dari tabel diatas sebagai contoh objek adalah gambar Museum geologi . Titik utama objek adalah tiang bendera. Jarak antara tiang bendera dengan pengambilan gambar menggunakan titik optimal yaitu untuk sudut 45° diambil dari jarak ±6 meter dari tiang, sudut 90° diambil dari jarak ±3 meter dari tiang dan jarak 135° diambil dari jarak ±6 meter dari tiang.

Dengan penetuan pengambilan gambar dari sudut 45°,90°, dan 135° dapat mengoptimalkan objet dapat terdeteksi khususnya dari arah depan.

### 3.1.7 Analisis Alur Data Sistem

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




**Gambar 3.15 Alur data**

Analisis ini meliputi :

1. Aplikasi *mobile* (*FrontEnd*)

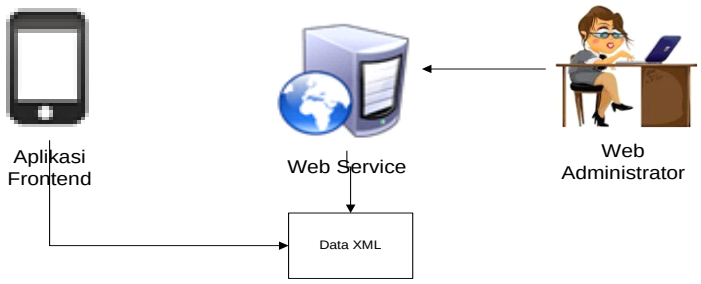
   Aplikasi *mobile* dalam hal ini adalah analisis fungsionalitas aplikasi untuk mengakses data dari database server melalui web service kemudian data yang diambil dari web service disimpan di database Sqlite dan folder gambr device tersebut. Aplikasi ini dibangun pada *platform* android dan komunikasi data antar aplikasi dengan database server menggunakan *parsing* data dari web service yang bertipe XML.

   Database SQLite berfungsi untuk menyimpan data xml yang diambil dari webservice melalui `Http Connection`. Data yang diambil berupa xml kemudian di parsing data dan diinputkan ke dalam database SQLite.

Sedangkan data gambar yang diambil dari Webservice melalu *Http Connection* disini disimpan di suatu folder di SDCard telepon seluler. Metode penyimapanan gambar di folder dipilih karena untuk penyimpanan gambar di SQLite cukup membutuhkan memory yang cukup besar dan belum menemukan metode yang cocok untuk penyimpanan dan akses gambar tipe blob dalam suatu DBMS SQlite.

2. Web *service*

Web service pada penelitian ini berfungsi sebagai *middle application* atau perantara aplikasi dengan database server. Cara kerja web service ini yaitu dengan mengambil data dari database server dan kemudian mengkonversi data tersebut kedalam format pertukaran data XML. Untuk data gambar di export kedalam sebuah folder, agar mempermudah aplikasi frontend download data gambar.




**Gambar 3. 16 Alur Data Web Service**

Untuk Link data :

1. Data String XML:

   a. XML Marker

   - Link Download

```
updatexml("http://wiki.commonroom.info/marker.x
ml");//hasil disimpan di SQlite
```

- Format XML Marker :

```
<?xml version=\"1.0\"?>\n
<ARDATA>
      <Data>
            <Idmarker></Idmarker>
            <Nama_Tempat></Nama_Tempat>
            <Tahun></Tahun>
            <Alamat></Alamat>
            <Sejarah></Sejarah>
      </Data>
</ARDATA>
```

b. XML Data Gambar

- Link Download

```
updatexml("http://wiki.commonroom.info/gambar.xml");//hasil disimpan di SQlite
```

- Format XML Data Gambar

```
<?xml version=\"1.0\"?>\n
<ARDATA>
      <Data>
            <Idmarker></Idmarker>
            <idgambar></idgambar>
      </Data>
</ARDATA>
```

c. XML Data MD5

- Link Download:

```
updatexml("http://wiki.commonroom.info/md5.xml");//hasil disimpan di SQlite
```

- Format XML Data Md5

```
<?xml version=\"1.0\"?>\n
<MD5HASH>
<FILE>
      <NAME>marker.xml</NAME>
      <MD5></MD5>
</FILE>
<FILE>
```

```
        <NAME>[idgambar]</NAME>
        <MD5>[gambar]</MD5>
    </FILE>
    <MD5HASH>
```

2. Data Gambar dari web server didownload di link :

```
updateimage("http://wiki.commonroom.info/gambar");
//hasil disimpan disuatu folder (sdcard)
```

3. Web Administrator(*Backend*)

Web Administrator disini berfungsi sebagai backend. Web Administrator bertugas untuk mengolah data diantaranya yaitu tambah data, cari data, export data, hapus data, edit data. Web Administrator disini melakukan konfigurasi *cross* antara *Web service* dengan *Aplikasi FrontEnd.*

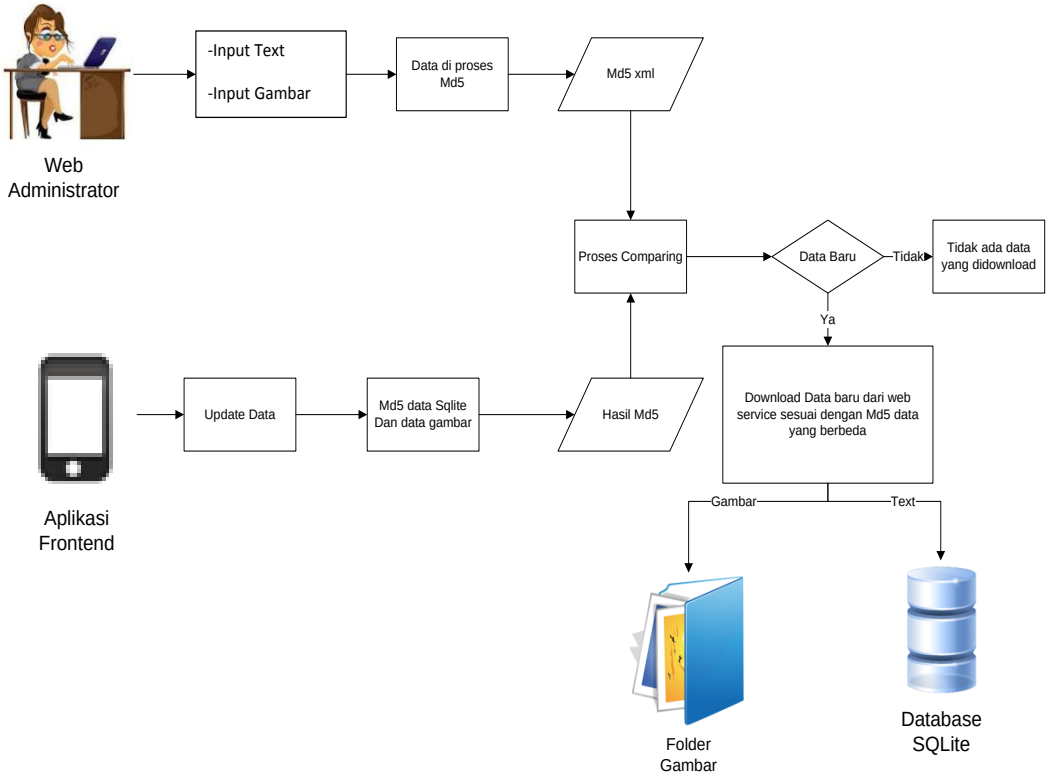
4. *Protocol* HTTP

*Protocol* HTTP merupakan *protocol* jaringan lapisan aplikasi yang digunakan untuk sistem informasi terdistribusi, kolaboratif, dan menggunakan hypermedia. Pada penelitian ini HTTP digunakan aplikasi *mobile* dan web administrator sebagai protocol yang dapat men-distribusikan data yang bersumber dari database server.

```
url=new URL(URL_domain + nmfile);
ucon= url.openConnection();
HttpURLConnection huc=(HttpURLConnection)
url.openConnection();
huc.setRequestMethod("GET");
huc.connect();
int code=huc.getResponseCode();
huc.disconnect();
```

### 3.1.8 Analisis Update Data

Update data dalam aplikasi ini dimulai dari web service yang menyimpan data baru kemudian akan diunduh oleh user di aplikasi frontend. Alur Update Data bisi diliat di gambar 3.17 :




**Gambar 3.17 Alur Sistem Proses Update Data**

Data di web server dan aplikasi frontend di export menjadi satu file yang sudah diproses menggunakan metode Md5. Setelah ditemukan nilai Md5, kemudian dicocokan, apabila data tidak sama atau data tidak ditemukan, aplikasi frontend akan mengambil data yang berbeda atau data yang tidak ada dari database server dalam betuk file gambar serta xml data gambar.

## 3.2 Analisis Dan 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 mencari informasi yang dibutuhkan.

### 3.2.1 Analisis dan Kebutuhan Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam pembuatan aplikasi ini adalah :

**Tabel 3.13 Kebutuhan Perangkat Keras**

| Aplikasi FrontEnd / User | Aplikasi Backend/ Admin |
|---|---|
| 1. Telepon Seluler (OS android 2.3-GingerBread) | 1. PC/Laptop dengan spesifikasi :<br>a. RAM 2GB<br>b. *Harddisk* 250GB |
| 2. Kamera Telepon Seluler Android (OS 2.2) 2 Megapixel | 2. Memiliki koneksi internet, seperti LAN WiFi, Modem. |
| 3. Memiliki koneksi internet HSDPA, 2G dan 3G | |

### 3.2.2 Analisis dan Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pembangunan aplikasi ini adalah :

**Tabel 3.14 Kebutuhan Perangkat Lunak**

| Aplikasi FrontEnd / User | Aplikasi Backend/ Admin |
|---|---|
| 1. Eclipse. | 1. Sistem Operasi : Windows xp/vista/seven. |
| 2. Librari Augmented Reality : ARLab | 2. Browser : Firefox, internet explorer, opera,google crohme. |
| 3. SQLite sebagai DDMS | 3. My SQL sebagai DBMS. |
| 4. Java JDK 1.6 | 4. Dream Weaver/Notepad ++, |
| | 5. 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 fasilitas umum (*end user*) yaitu para pengguna *smartphone* dengan *platform* android versi 2.3 atau lebih tinggi. Dalam menggunakan sistem ini, pengguna terhubung dengan koneksi internet GPRS/EDGE/UMTS/3G dan kamera.

2 Admin sebagai pengelola konten mempunyai kapabilitas dalam mengelola database seperti menambah, mengubah, dan menghapus data melalui web administrator.

Berikut ini karakteristik pengguna sistem yang ditunjukkan oleh tabel 3.15:

**Tabel 3.15 Karakteristik Pengguna**

| Penggun a | Hak Akses | Tingkat pendidikan | Pengalaman |
|---|---|---|---|
| End User | Mengakses data informasi melalui aplikasi front-end | Minimal SMA, atau sederajat | Minimal dapat menjalankan aplikasi pada perangkat android |
| Admin | Mengakses data serta mengelola data di aplikasi back-end | Min.D3 | Min. Dapat menjalankan fungsi-fungsi yang ada di web admin dan mengupdate/mengolah data-data pada web admin |

## 3.3 Analisis Kebutuhan Fungsional

Kebutuhan fungsional berhubungan dengan fitur software yang akan dibuat atau dikembangkan. Berikut adalah tahapan analisis kebutuhan fungsional Sistem Aplikasi mobile kebudayaan Jawa Barat, khususnya kota Bandung.

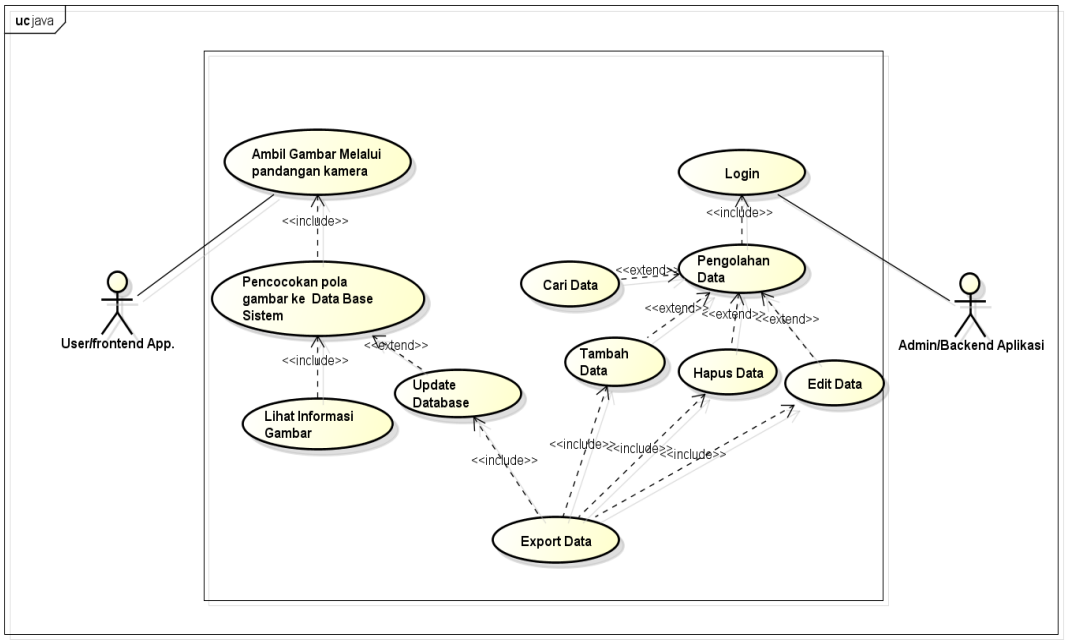
Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik.

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

### 3.3.1. Usecase Diagram

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

penggunaan sistem yang akan dibangun. Dari hasil analisis aplikasi yang ada maka *use case diagram* untuk aplikasi ini dapat di lihat pada gambar 3.18:




**Gambar 3.18 Usecase Diagram Aplikasi**

### 3.3.1.1 Definisi Usecase

*Use case diagram* merupakan konstruksi untuk mendeskripsikan hubungan-hubungan yang terjadi antar aktor dengan aktivitas yang terdapat pada sistem.

Sasaran pemodelan *use case* diantaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan yang disepakati antara pemakai dan pengembang.

**Tabel 3.16 Definisi Usecase**

| ***Use Case* Aplikasi *Frontend*** | | |
|---|---|---|
| **No** | ***Use Case*** | **Deskripsi** |
| UCF-01 | Ambil Gambar Melalui Pandangan Kamera. | Fungsionalitas untuk Mengambil Gambar menggunakan kamera Telepon Seluler. |
| UCF-02 | Pencocokan pola Gambar Ke Database Sistem | Fungsionalitas untuk membandingkan gambar yang diambil kamera Telepon Seluler dengan gambar yang ada di database SQLite. |
| UCF-03 | Lihat Informasi Gambar | Fungsionalitas untuk menampilkan informasi dengan teknologi Augmented Reality. |
| UCF-04 | Update Database | Fungsionalitas untuk update database SQLite dan data gambar dari Database Server. |
| | | |
| ***UseCase* Aplikasi *Backend*** | ***Use Case*** | **Deskripsi** |
| UCB-01 | Login | Fungsionalitas login untuk admin. |
| UCB-02 | Pengolahan Data | Fungsionalitas untuk admin dapat melakukan pencarian serta mengelola data tempat kebudayaan. |
| UCB-03 | Tambah Data | Fungsionalitas untuk menambahkan data. |
| UCB-04 | Cari Data | Fungsionalitas untuk mencari data sesuai Nama Gambar dan data Alamat. |
| UCB-05 | Ubah Data | Fungsionalitas untuk mengubah atau *update* data. |
| UCB-06 | Hapus Data | Fungsionalitas untuk menghapus data. |
| UCB-07 | Export Data | Fungsionalitas untuk export data di database menjadi xml dan export gambar di database ke dalam suatu folder. |

### 3.3.1.2 Skenario *Usecase*

Skenario *Use Case* 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 Skenario *Use Case Frontend*

Skenario proses-proses yang terdapat dalam *use case diagram* aplikasi *frontend* dapat dilihat pada table 3.17

**Tabel 3.17 Skenario *Use Case* Ambil Gambar Melalui Pandangan Kamera**

| **Identifikasi** | |
|---|---|
| Nama Use Case | Ambil Gambar Melalui Pandangan Kamera |
| Aktor | Pengguna *User* |
| Tujuan | Menjalankan aplikasi dengan menyorotkan kamera ke objek tertentu, guna untuk mengambil gambar secara sistem dan dilakukan proses pencocokan pola. |
| Keadaan awal | Sistem menampilkan icon aplikasi di layar telepon seluler. |
| **Skenario Utama** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1.Memilih icon aplikasi kebudaayn Jawa Barat. | |
| | 2. Menjalankan Kamera telepon seluler untuk deteksi gambar dan akan diproses sehingga menampilkan informasi dengan format teks. |
| | 3. Apabila data sesuai dengan gambar yang ada didatabae akan keluar informasi sejarah sesuai dengan data didatabase. |
| **Skenario Gagal** | |
| **Aksi Aktor** | **Reaksi Sistem** |

| | |
|---|---|
| | 1. “Data Tidak Ditemukan”. |
| **Keadaan Akhir** | Aplikasi menampilkan menu utama dan kamera berjalan kembali untuk proses deteksi. |

**Tabel 3.18 Skenario *Use Case* Pencocokan pola Gambar Ke Database Sistem**

| **Identifikasi** | |
|---|---|
| Nama Use Case | Sinkronisasi Gambar ke Database Sistem |
| Aktor | Pengguna *User* |
| Tujuan | Mencocokan gambar yang diambil melalui kamera telepon seluler dengan data yang berada didatabase. |
| Keadaan awal | Sistem menampilkan icon aplikasi di layar telepon seluler. |
| **Skenario Utama** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1.Menyorotkan Kamera Telepon Seluler ke Objek | |
| | 2. Menjalankan Kamera telepon seluler untuk mengambil gambar yang berfungsi sebagai markerless dan akan diproses pencocokan pola gambar sehingga menampilkan informasi sejarah menggunakan teknologi Augmented Reality. |
| | 3. Melakukan pencocokan markerless dengan data yang berada didatabase sistem (SQLite). |
| **Skenario Gagal** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| | 1. “Data tidak ditemukan”. |
| **Keadaan Akhir** | Aplikasi tetap menjalankan kamera untuk deteksi gambar. |

**Tabel 3.19 Skenario *Use Case* Lihat Informasi Sejarah**

| **Identifikasi** | |
|---|---|
| Nama *Use Case* | Lihat Informasi Sejarah |
| Aktor | Pengguna *User* |
| Tujuan | Menampilkan informasi sejarah tempat kebudayaan menggunakan teknologi Augmented Reality. |
| **Scenario Utama** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Menjalankan Menu ambil gambar dan menyorotkan kamera ke gedung kebudayaan. | |
| | 2. Data hasil pengenalan pola cocok dengan data didatabase sehingga menampilkan informasi sejarah tempat tersebut secara *real time* dengan menggunakan teknologi Augmented Reality. |
| **Keadaan Akhir** | Aplikasi menampilkan informasi sejarah tempat tersebut secara *real time* |
| **Skenario Gagal** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| | 1. Menampilkan pesan “ Data tidak ditemukan” |
| **Keadaan Akhir** | Aplikasi tetap menjalankan kamera untuk deteksi gambar. |

**Tabel 3.20 Skenario *Use Case* Update Database**

| **Identifikasi** | |
|---|---|
| Nama *Use Case* | Update Database |
| Aktor | Pengguna *User* |
| Tujuan | Mengupdate data dari database Server dengan cara mendowload data. |

| **Scenario Utama** | |
|---|---|
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Menekan tombol menu dan memilih tombol update | |
| | 2. Proses download berjalan dengan menampilkan progress bar yang menunjukan proses download data dari web server. |
| **Keadaan Akhir** | Progress bar otomatis menghilang apabila proses download data sudah berhasil dilakukan. |
| **Skenario Gagal** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| | 1. Menampilkan pesan "Tidak Ada Koneksi Internet" |
| **Keadaan Akhir** | Aplikasi berhenti dan kembali ke menu utama. |

#### *3.3.1.2.2* Skenario *Use Case Backend/Admin*

Skenario proses-proses yang terdapat dalam *use case diagram* aplikasi *backend* dapat dilihat pada tabel-tabel berikut :

**Tabel 3.21 Skenario *Use Case Login***

| **Identifikasi** | |
|---|---|
| Nama Use Case | *Login* |
| Aktor | Pengguna *Backend* |
| Tujuan | Masuk kedalam sistem |
| Keadaan Awal | Sistem menampilkan halaman login admin |
| **Skenario Utama** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Mengisi *username* dan *password* pada halaman web *login* admin | |

| | |
|---|---|
| | 2. Sistem melakukan autentikasi *username* dan *password* yang diiunputkan. |
| | 3. Bila sesuai maka menampilan halaman web menu utama. |
| **Skenario Alternatif – Validasi Gagal** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| | 1. Menampilkan pesan "*Login Gagal, pastikan username dan password anda*" |
| 2. Melakukan input *username* dan *password* pada halaman web login admin | |
| **Kondisi Akhir** | Halaman menu utama ditampilkan |

**Tabel 3.22 Skenario *Use Case* Pengolahan Data**

| **Identifikasi** | |
|---|---|
| Nama Use Case | Pengolahan Data |
| Aktor | Pengguna *Backend/Admin* |
| Tujuan | Pengolahan Data Tempat |
| Keadaan Awal | Menampilkan halaman Pencarian Data Tempat dan list data. |
| **Skenario Utama** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Memilih Menu Utama Pengolahan Data Tempat | |
| | 2. Menampilkan halaman Pengolahan Data Tempat, tediri dari input, cari data, hapus data dan edit data |
| **Kondisi Akhir** | Menampilkan data konten. |

**Tabel 3.23 *Use Case* Tambah Data**

| **Identifikasi** | |
|---|---|
| Nama Use Case | Tambah Data |
| Aktor | Pengguna *Backend/Admin* |
| Tujuan | Menambah data |
| Keadaan Awal | Sistem menampilkan halaman input data. |
| **Skenario Utama** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1.Memilih Tombol *Tambah* | |
| | 2. Menampilkan halaman pengisian tambah data. |
| 3. Mengisi *field-field* pada pengisian tambah *data*. | |
| | 4. Melakukan proses tambah data pada database. |
| | 5. Menyimpan hasil tambah data. |
| | 6. Export data secara otomatis oleh sistem. Data String di *Convert* ke XML, data gambar di *convert* ke satu folder |
| **Skenario Alternatif – Validasi Gagal** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| | 1. Menampilkan pesan bahwa "*Data tidak berhasil diinputkan*". |
| 2. Melengkapi *field-field* pada pengisian tambah kategori | |
| **Kondisi Akhir** | 3.Data kategori bertambah sesuai dengan input data. |

**Tabel 3.24 Skenario *Use Case Cari Data***

| **Identifikasi** | |
|---|---|
| Name Use Case | Cari Data |
| Aktor | Pengguna *Backend* |
| Tujuan | Mencari data. |
| Keadaan Awal | Sistem menampilkan halaman konten |
| **Skenario Utama** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1.klik *cari data* halaman Konten | |
| | 2. Menampilkan halaman pencarian data content. |
| 3. Mengisi *field* pada halaman cari data | |
| | 4. Melakukan proses pencarian data pada basis data dan menyimpannya. |
| **Skenario Alternatif – Validasi Gagal** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| | 1. Menampilkan menu utama kosong. |
| 2. Menekan tombol notifikasi (“cari”) | |
| **Kondisi Akhir** | 3. Kembali ke form list data. |

**Tabel 3.25 Skenario *Use Case* Ubah Data**

| **Identifikasi** | |
|---|---|
| Nama Use Case | Ubah Data |
| Aktor | Admin/Backend |
| Tujuan | Mengubah data |
| Keadaan Awal | Menampilkan list data . |

| **Skenario Utama** | |
|---|---|
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Memilih tombol edit | |
| | 2. Menampilkan halaman ubah konten |
| 3. Mengubah data pada field-field yang diinginkan | |
| | 4. Melakukan proses ubah data pada basis data |
| | 5. Menyimpan hasil perubahan data. |
| | 6. Export data secara otomatis oleh sistem. Data String di *Convert* ke XML, data gambar di *convert* ke satu folder |
| **Skenario Alternatif – Validasi Gagal** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| | 1. Menampilan pesan "*data tidak berhasil di edit"*. |
| 2.Menampilkan field-field kembali sesuai dengan ID yang akan di edit | |
| **Kondisi Akhir** | Perubahan data konten sesuai dengan ubah konten. |

**Tabel 3.26 Skenario *Use Case* Hapus Data**

| **Identifikasi** | |
|---|---|
| Nama Use Case | Hapus Data |
| Aktor | Admin |
| Tujuan | Menghapus data Alamat yang tidak diperlukan |
| Keadaan Awal | Sistem menampilkan halaman pengolahan alamat. |
| **Skenario Utama** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Memilih tombol | |

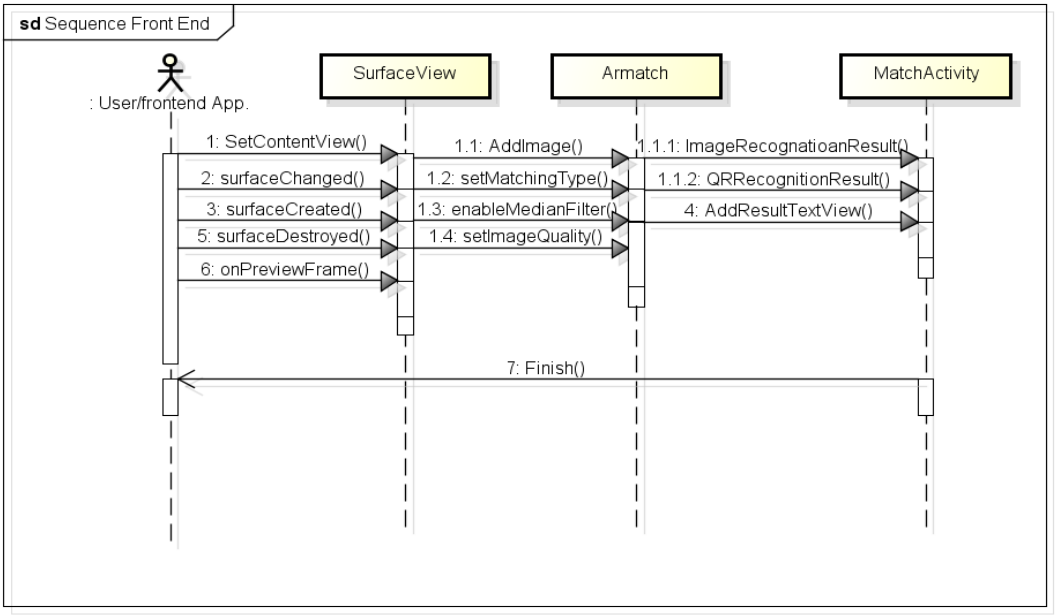
| | |
|---|---|
| hapus | |
| | 2. Menampilkan pesan konfirmasi hapus data. |
| | 3. Menghapus data konten |
| | 4. Menyimpan hasil perubahan data |
| | 5. Export data secara otomatis oleh sistem. Data String di *Convert* ke XML, data gambar di *convert* ke satu folder |
| **Kondisi Akhir** | Perubahan data konten sesuai data yang dihapus. |

### 3.3.2 Sequence Diagram

*Sequence diagram* 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.

#### 3.3.2.1 *Sequence Diagram* Ambil Gambar Melalui Pandangan Kamera

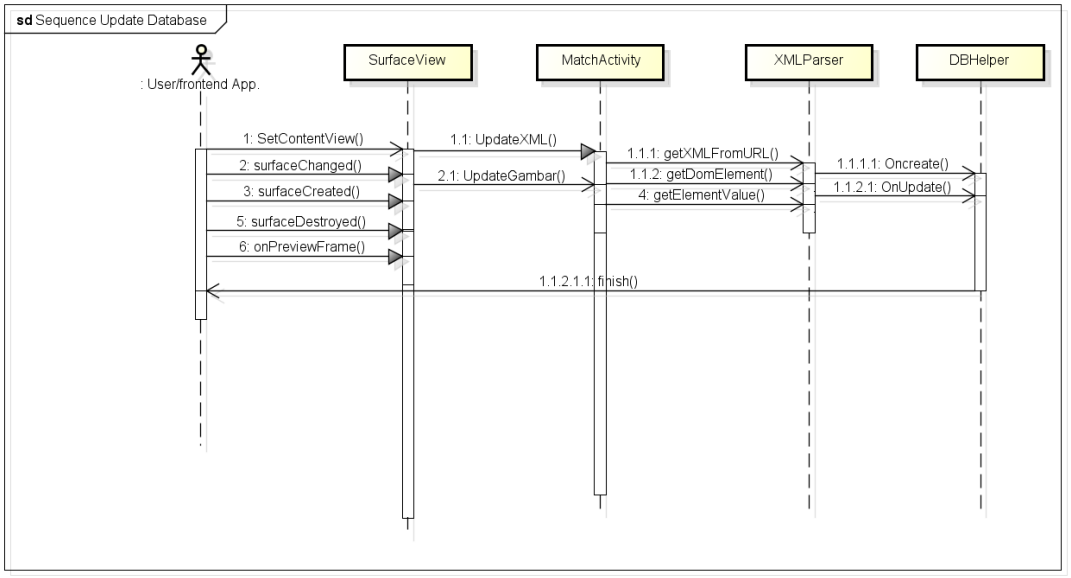
*Sequence diagram* Ambil Gambar Melalui Pandangan Kamera menunjukkan proses menu yang dipilih untuk menyorot sebuah tempat yang bertujuan untuk mengambil Gambar Melalui Pandangan Kamera kemudian diproses guna menghasilkan output informasi sejarah berupa text. *Sequence diagram* Ambil Gambar Melalui Pandangan Kamera dapat dilihat pada Gambar 3.19




**Gambar 3.19 *Sequence diagram* Ambil Gambar Melalui Pandangan Kamera**

### *3.3.2.2 Sequence Diagram Update Database*

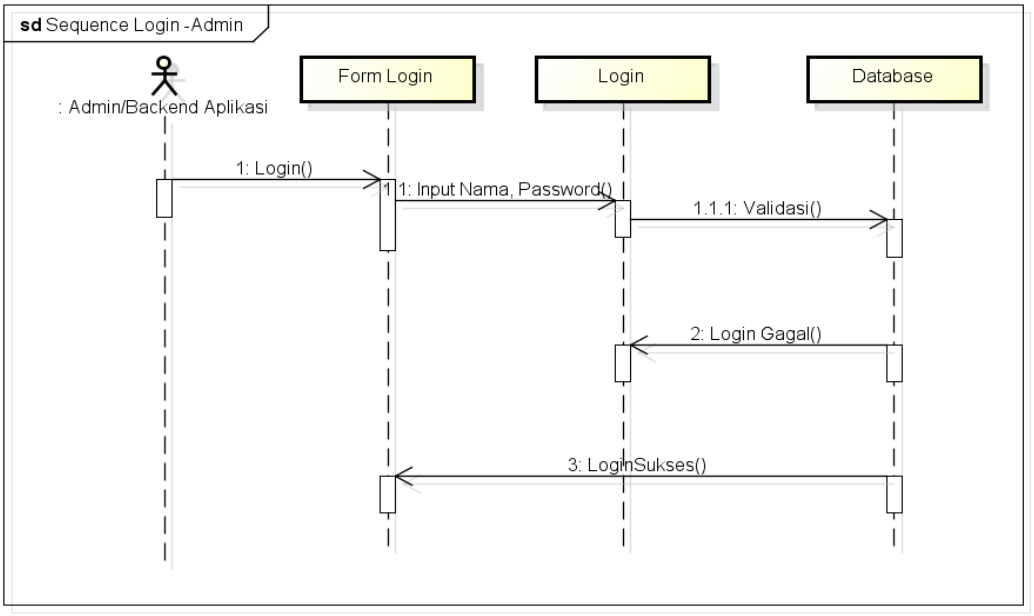
*Sequence diagram Update Database* menunjukkan rangkaian pesan yang dikirim antara user dalam memproses sistem Update Database. *Sequence diagram* Update Database dapat dilihat pada Gambar 3.20




**Gambar 3. 20 *Sequence diagram* Update Database**

### *3.3.2.3 Sequence Diagram Login -Admin*

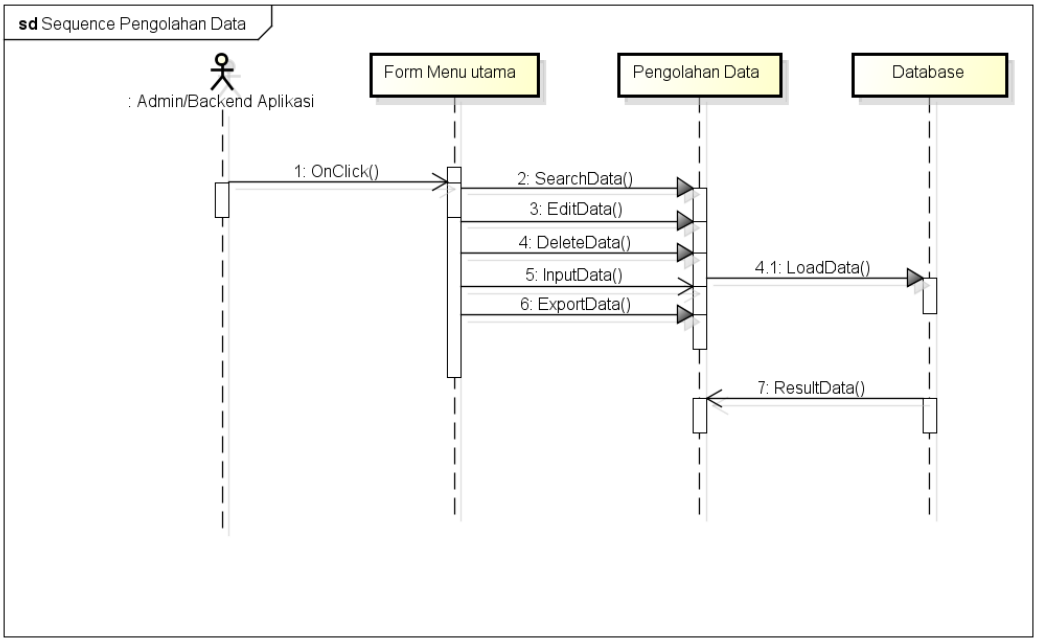
*Sequence diagram Login -Admin* menunjukkan rangkaian pesan yang dikirim antara admin dalam memproses sistem Login. *Sequence diagram* Login -Admin dapat dilihat pada Gambar 3.21




**Gambar 3. 21 *Sequence diagram* Login-Admin**

### *3.3.2.4 Sequence Diagram Pengolahan Data-Admin*

*Sequence diagram Pengolahan-Admin* menunjukkan rangkaian pesan yang dikirim antara admin dalam memproses sistem Pengolahan data yang terdiri dari tambah data, edit data, hapus data dan pencarian data. *Sequence diagram* Pengolahan Data-Admin dapat dilihat pada Gambar 3.22



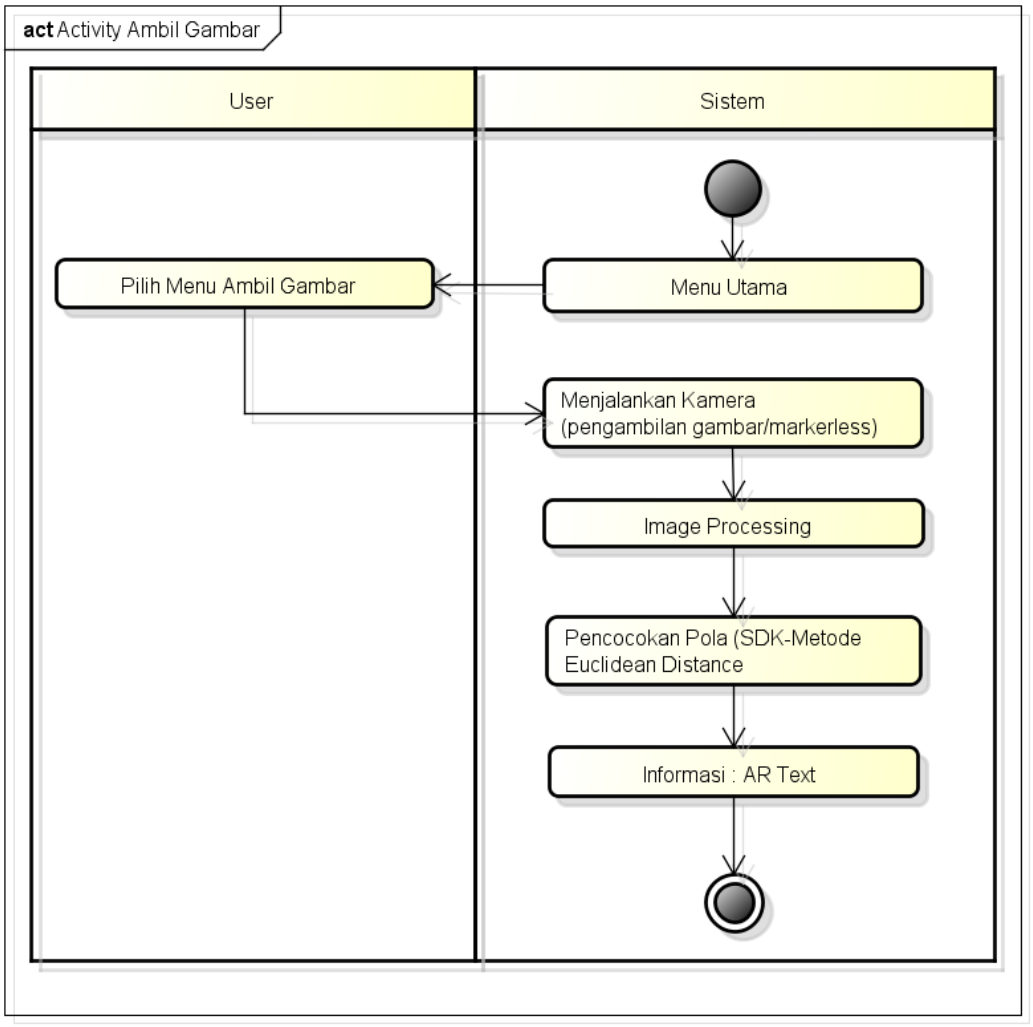

**Gambar 3.22 *Sequence diagram* Pengolahan Data-Admin**

### 3.3.3 *Activity Diagram*

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

#### 3.3.3.1 *Activity Diagram Ambil Gambar Melalui Pandangan Kamera*

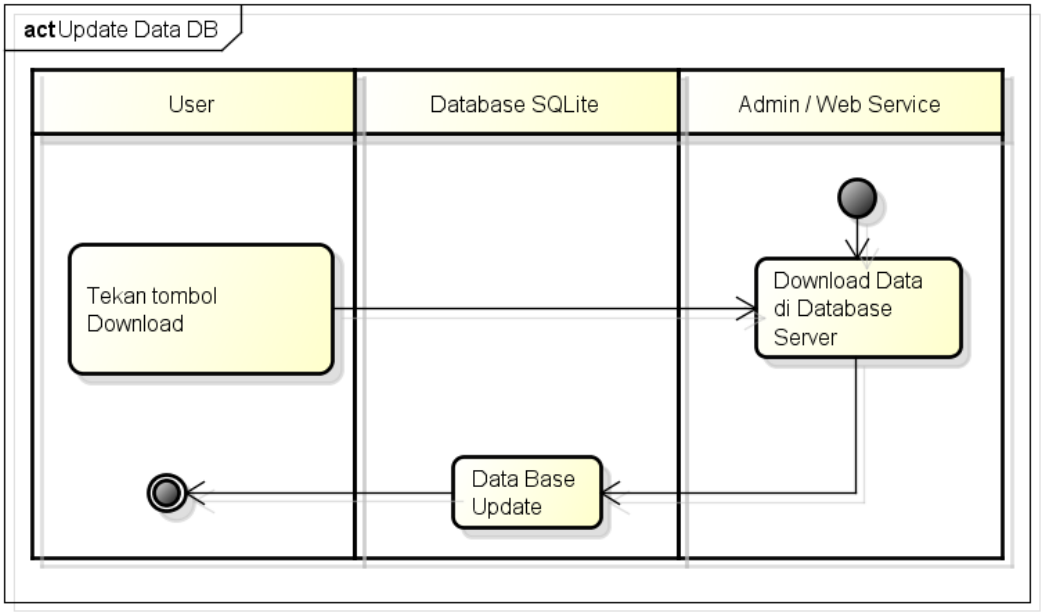
*Activity diagram Ambil Gambar Melalui Pandangan Kamera* menjelaskan aliran kerja aktor pengguna pada saat monyorot suatu tempat guna mendapatkan image yang akan diproses dalam aplikasi *mobile*. *Activity diagram Ambil Gambar Melalui Pandangan Kamera* dapat dilihat pada gambar 3.23




**Gambar 3.23** ***Activity diagram Ambil Gambar Melalui Pandangan Kamera***

### 3.3.3.2 *Activity Diagram Sinkronisasi Data Ke Database SQLite*

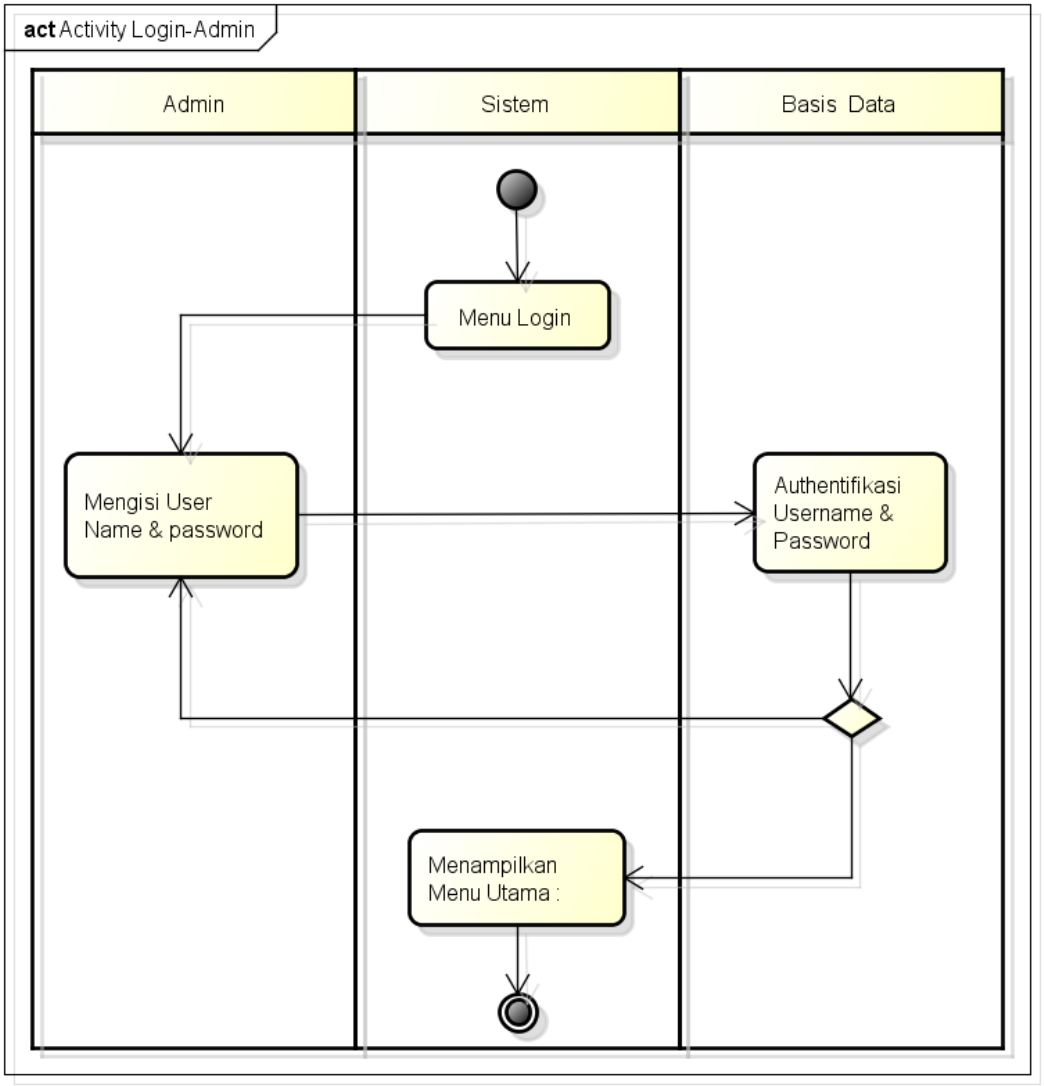
*Activity diagram Sinkronisasi Data Ke Database SQLite* menjelaskan aliran kerja admin pada saat memilih menu update data. *Activity diagram Sinkronisasi Data Ke Database SQLite* dapat dilihat pada gambar 3.24




**Gambar 3.24** ***Activity diagram Update Data***

### 3.3.3.3 *Activity Diagram Login-Admin*

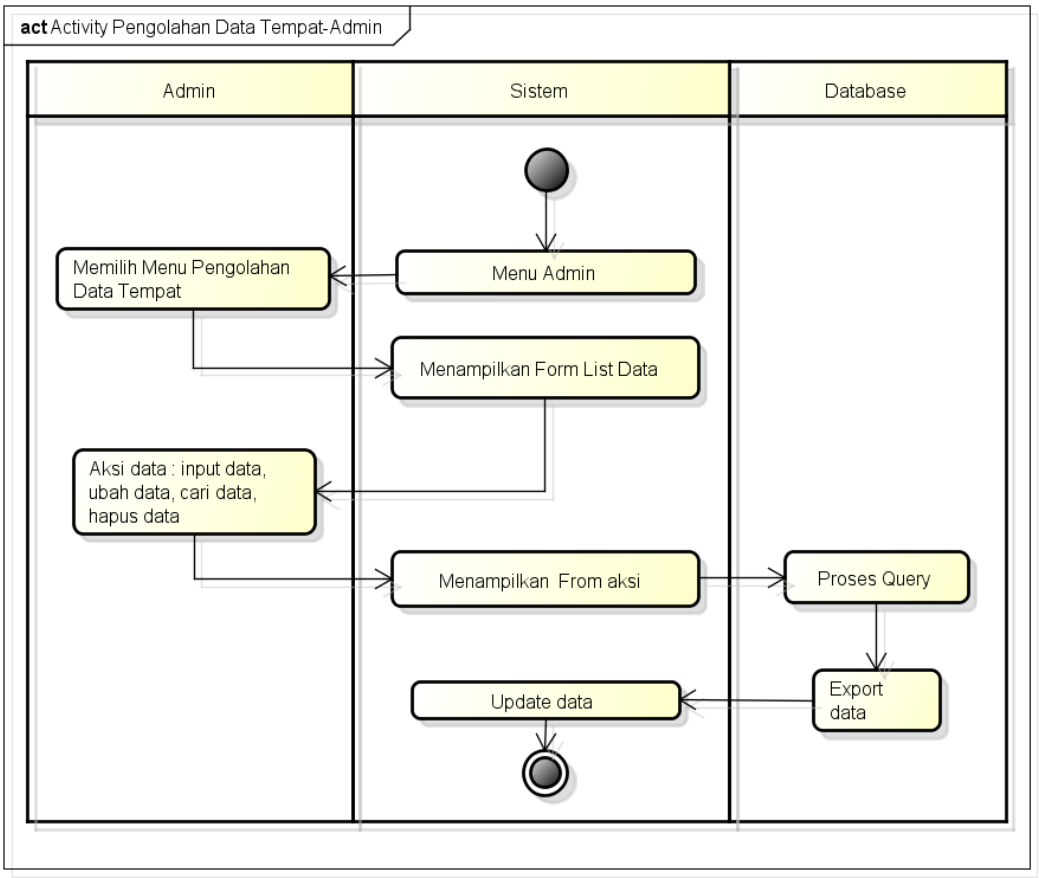
*Activity diagram Login-Admin* menjelaskan aliran kerja admin pada saat memilih menu Login. *Activity diagram Login*-Admin dapat dilihat pada gambar 3.25




**Gambar 3.25** ***Activity diagram Login Admin***

### 3.3.3.4 *Activity Diagram Pengolahan Data -Admin*

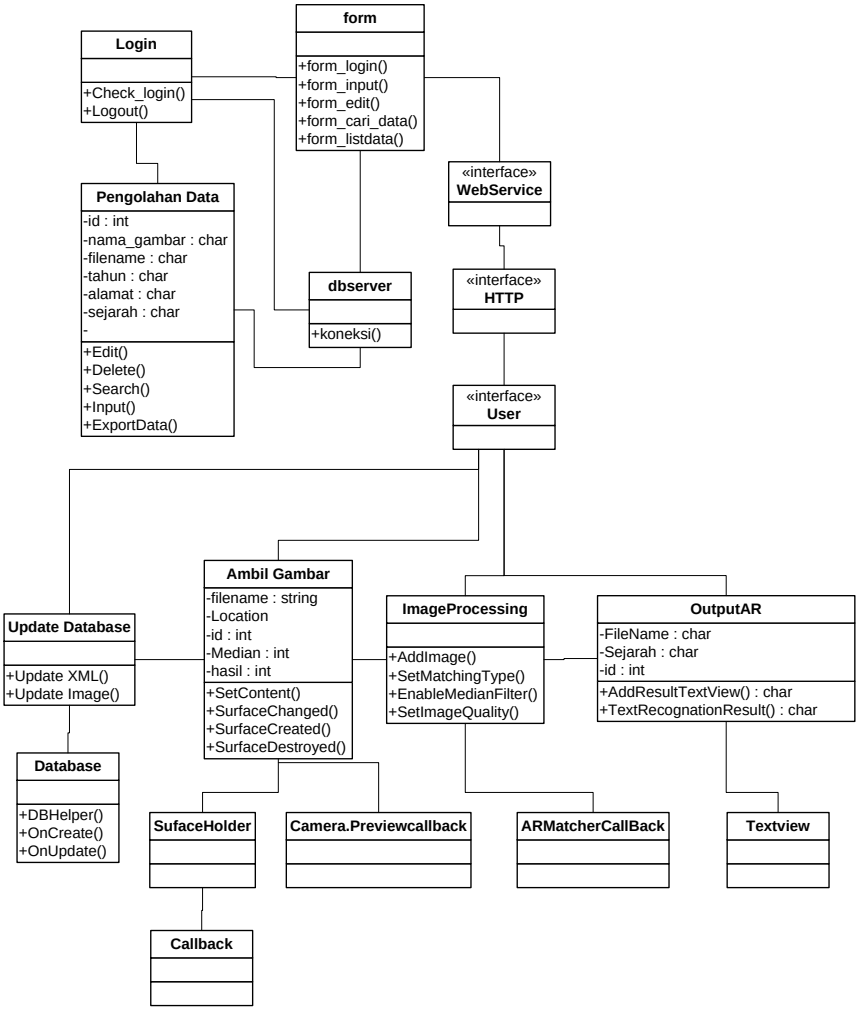
*Activity diagram Pengolahan Data -Admin* menjelaskan aliran kerja admin pada saat memilih menu Pencarian Data dan update data information. *Activity diagram Pengolahan Data Tempat*-Admin dapat dilihat pada gambar 3.26




**Gambar 3.26** ***Activity diagram Pengolahan Data-Admin***

### 3.3.4 Class Diagram

*Class Diagram* menggambarkan struktur dan hubungan antar objek- objek yang ada pada sistem. Struktur itu meliputi atribut-atribut dan metode- metode yang ada pada masing-masing *class*. Hubungannya dari masing- masing *class* yang ada tersebut digambarkan seperti pada gambar 3.27




**Gambar 3. 27 *Class Diagram***

**Tabel 3.27 Deskripsi *Class Diagram***

| ***Backend Application*** | | |
|---|---|---|
| **Class** | **Jenis Class** | **Deskripsi** |
| Login | Control | Kelas login merupakan kelas yang berisikan metode – metode yang berfungsi untuk menentukan hak akses untuk mengelola data pada server. |
| PengolahanData | Entity | Kelas pengolahan data merupakan kelas berisikan metode-metode untuk mengelola data pada server |

| | | |
|---|---|---|
| Form | Interface | Kelas yang berisikan metode- metode yang berfungsi menampilkan halaman yang untuk mengelola data pada *database server*. |
| DBServer | Control | Kelas DBServer merupakan kelas yang berisikan metode untuk mengkoneksikan *database server* dengan kelas – kelas yang terdapat pada aplikasi server. |
| ***Frontend Application*** | | |
| **Class** | **Jenis Class** | **Deskripsi** |
| Http Conection | Control | Class yang berisikan fungsi http connection. |
| DBHelpper | *Control* | Class yang berisikan method-method untuk *koneksi* ke database serta proses pencocokan gambar dengan data base local (SQLite). |
| *MatchActivity* | *Control* | Class yang berisikan *method-method* untuk manampilkan informasi dari setiap marker, Update data, HTTP Connection |
| XML Parser | *Control* | Class yang berisikan *method-method* untuk fungsionalitas mengambil data dari web service dengan format data xml. |
| SufaceHolder | *Control* | Class yang berisikan *method-method* untuk fungsionalitas menjalankan fungsi memanggil fram untuk tampilan Augmented Reality dalam deteksi gambar. |
| Camera.Previewcall back | *Control* | Class yang berisikan *method-method* untuk fungsionalitas menjalankan fungsi menjalankan kamera untuk deteksi gambar. |
| ARMatcherCallBack | *Control* | Class yang berisikan *method-method* untuk fungsionalitas menjalankan fungsi memanggil frame untuk tampilan Augmented Reality dalam deteksi gambar. |
| Textview | *Control* | Class yang berisikan *method-method* untuk fungsionalitas menjalankan fungsi |

| | | |
|---|---|---|
| | | teks untuk output Augmented Reality |
| Callback | *Control* | Class yang berisikan *method-method* untuk dari ARLab. |

## 3.4 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 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 basis data 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.

### 3.4.1.1 Perancangan Database Sistem SQLite (FrontEnd)

**1. Tabel Marker**

**Tabel 3. 28 Perancangan Struktur Tabel Marker**

| Nama Field | Tipe Data | Keterangan |
|---|---|---|
| Idmarker | Int(10) | Primary key |
| Nama_tempat | Varchar (50) | |
| Tahun | Varchar (50) | |
| Alamat | Varchar (50) | |
| Sejarah | Text | |

**2. Tabel Gambar**

**Tabel 3. 29 Perancangan Struktur Tabel Gambar**

| Nama Field | Tipe Data | Keterangan |
|---|---|---|
| Idmarker | Int(10) | Foreign key |
| Idinfo | Int(10) | Primary Key |

### 3.4.1.2 Perancangan Database Sistem Web Server (Back End)

**1. Tabel Admin**

**Table 3.30 Perancangan Struktur Tabel Admin**

| Nama Field | Tipe Data | Keterangan |
|---|---|---|
| Id_admin | Int | Primary key |
| Username | Varchar(50) | |
| Password | Varchar(50) | |

**2. Tabel Marker**

**Tabel 3.31 Perancangan Struktur Tabel Marker**

| Nama Field | Tipe Data | Keterangan |
|---|---|---|
| Idmarker | Int(10) | Primary key |
| Nama_tempat | Varchar (50) | |
| Tahun | Varchar (50) | |
| Alamat | Varchar (50) | |
| Sejarah | Text | |
| Id_admin | Int | Foreign Key |

### 3. Tabel Gambar

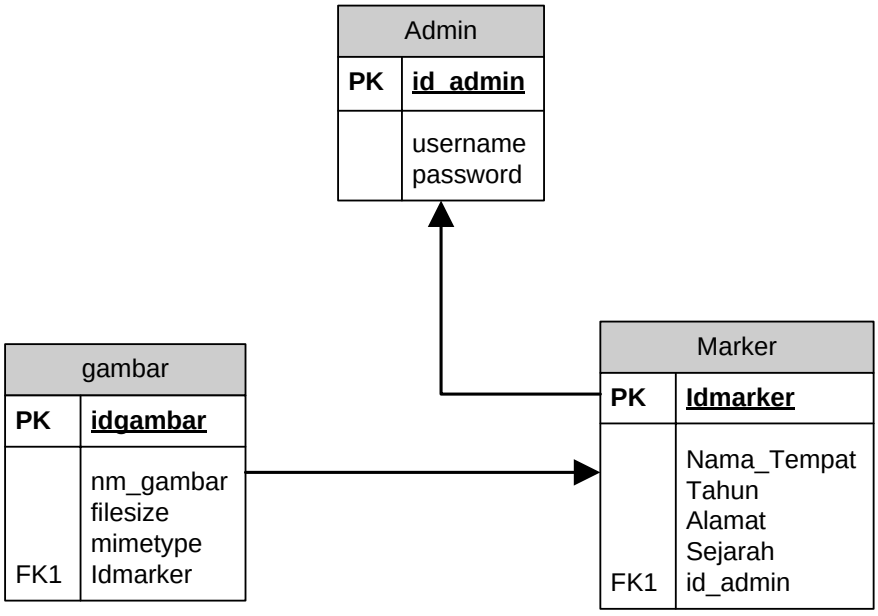
**Tabel 3.32 Perancangan Struktur Tabel Gambar**

| Nama Field | Tipe Data | Keterangan |
|---|---|---|
| Idmarker | Int(10) | Foreign Key |
| Idgambar | Int(10) | Primary Key |
| Gambar | Longblob | |
| Mimetype | Varchar (50) | |

### 3.4.2 Skema Relasi

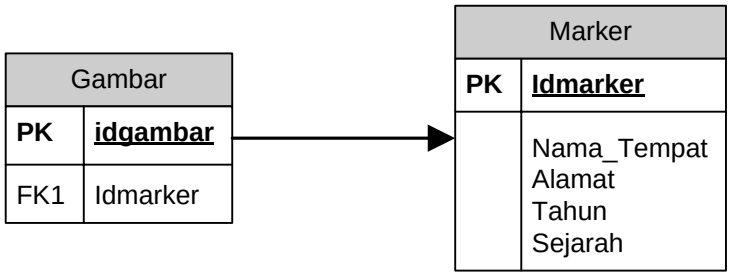
Skema relasi atau relasi antar tabel merupakan gabungan antar file yang mempunyai kunci utama yang sama, sehingga file-file tersebut menjadi satu kesatuan yang dihubungkan oleh field kunci (*primary key*). Pada proses ini elemen-elemen data dikelompokkan menjadi satu file database beserta entitas dan hubungan.

1. Skema Relasi Backend




Gambar 3. 28 Skema Relasi Aplikasi Backend

2. Skema Relasi Frontend



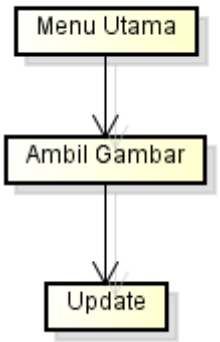

Gambar 3.29 *Skema Relasi Aplikasi Frontend*

### 3.4.3 Struktur Menu

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

#### 3.4.3.1 Struktur Menu FrontEnd/User

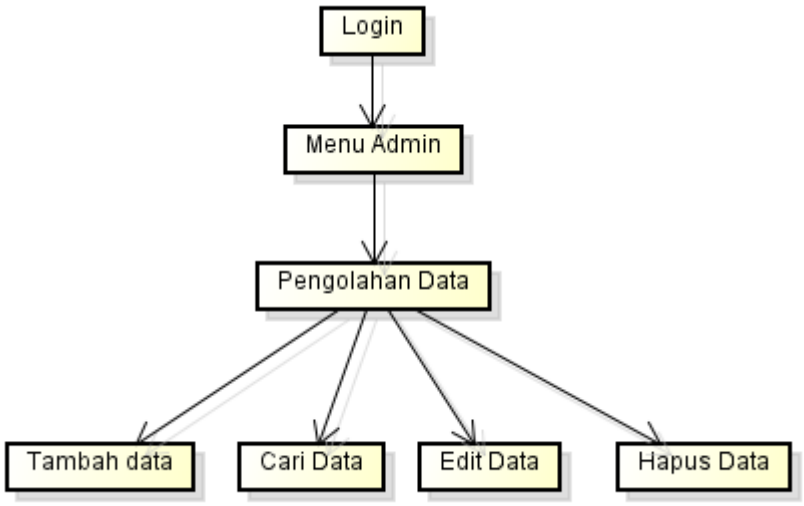
Perancangan struktur menu merupakan gambaran jalur pemakaian aplikasi user pengguna aplikasi mobile ini. Perancangan struktur menu dari aplikasi *ini* dapat dilihat pada gambar 3.30




**Gambar 3. 30 Struktur Menu Aplikasi FrontEnd**

### 3.4.2.2 Struktur Menu Aplikasi Backend/Admin

Perancangan struktur menu merupakan gambaran jalur pemakaian aplikasi admin pengelolaan aplikasi mobile ini. Perancangan struktur menu dari aplikasi *ini* dapat dilihat pada gambar 3.31




**Gambar 3.31 Struktur Menu Web Admin**

### 3.4.4 Perancangan Antarmuka

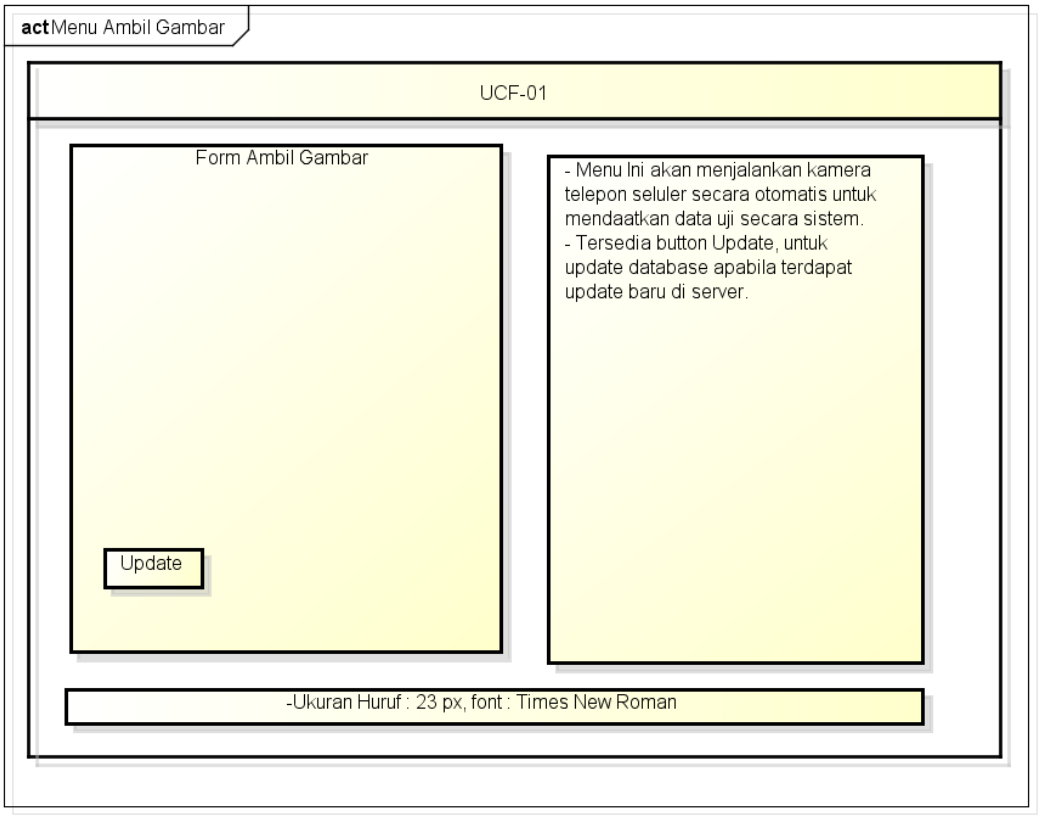
Perancangan antarmuka (*interface*) menggambarkan bagaimana perangkat lunak berkomunikasi dengan dirinya sendiri, dengan sistem yang berinteroperasi dengannnya, dan dengan manusia yang menggunakannya.

### 3.4.4.1 Perancangan Tampilan

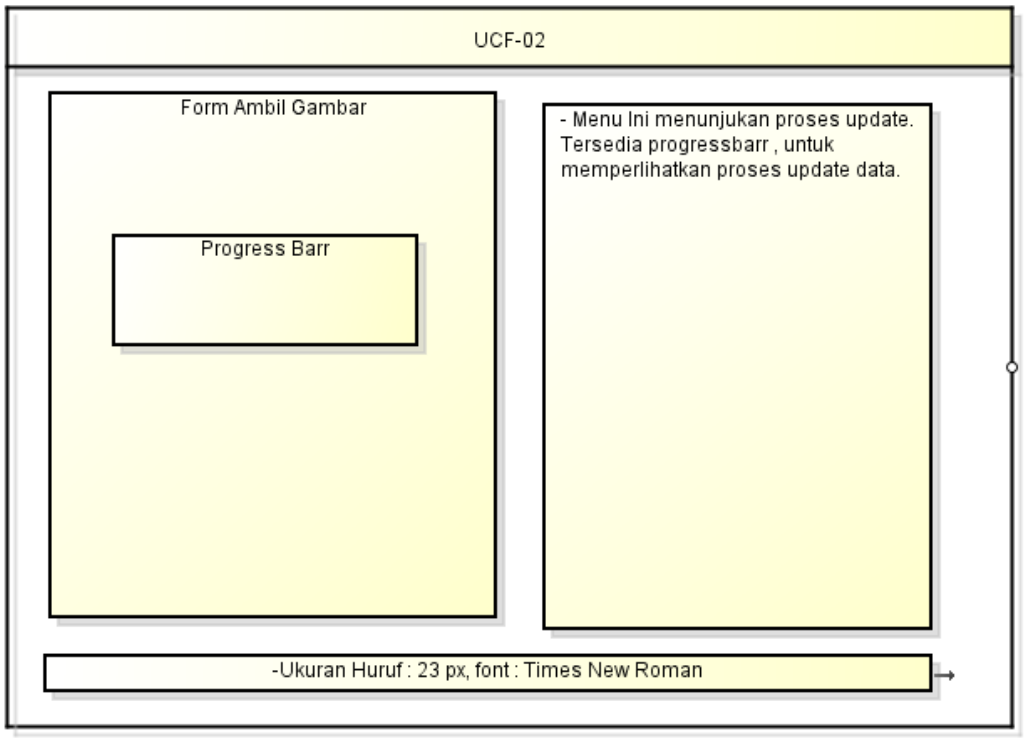
Perancangan *form* mendeskripsikan rencana tampilan dari setiap *form* yang akan digunakan pada tampilan aplikasi *ini.* Adapun gambaran dari perancangan antarmuka aplikasi *mobile* ini adalah sebagai berikut:

### 3.4.4.1.1 Perancangan Form Ambil Gambar Melalui Pandangan Kamera

Gambaran umum dari perancangan antarmuka form Pilih Ambil Gambar Melalui Pandangan Kamera Berita pada aplikasi *mobile* ini dapat dilihat pada Gambar 3.32



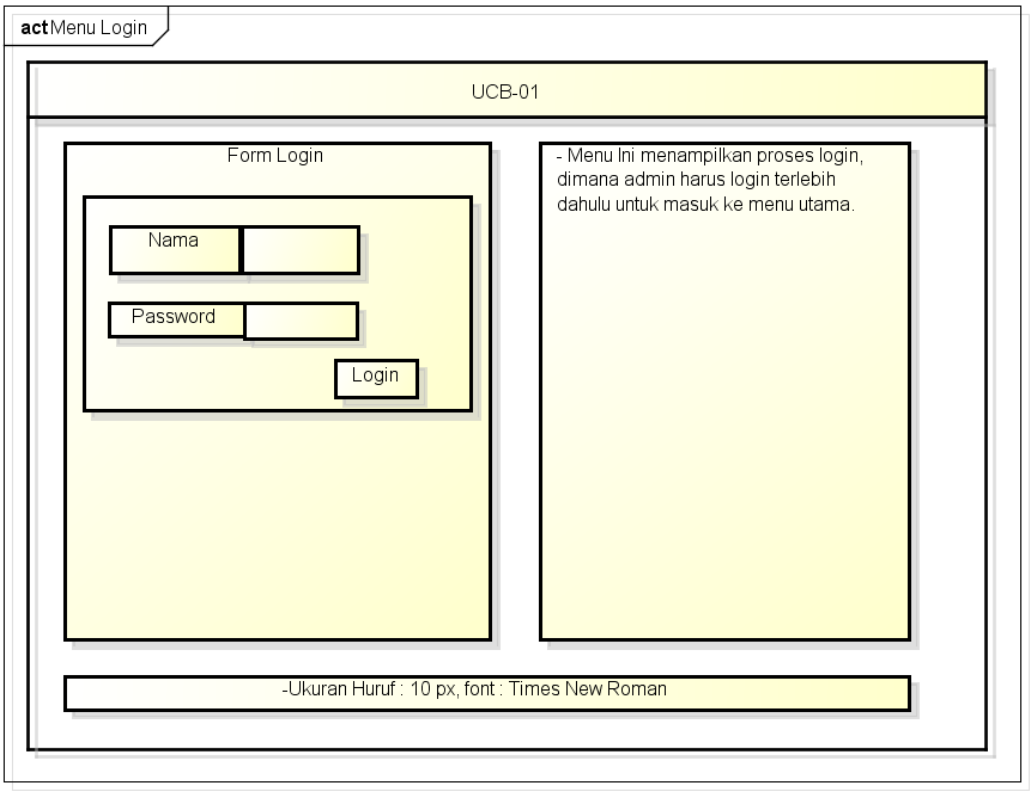

**Gambar 3.32 Tampilan Form Ambil Gambar Melalui Pandangan Kamera**




**Gambar 3.33 Tampilan Update-Frontend**

### 3.4.4.1.2 Perancangan Form Login -*Admin*

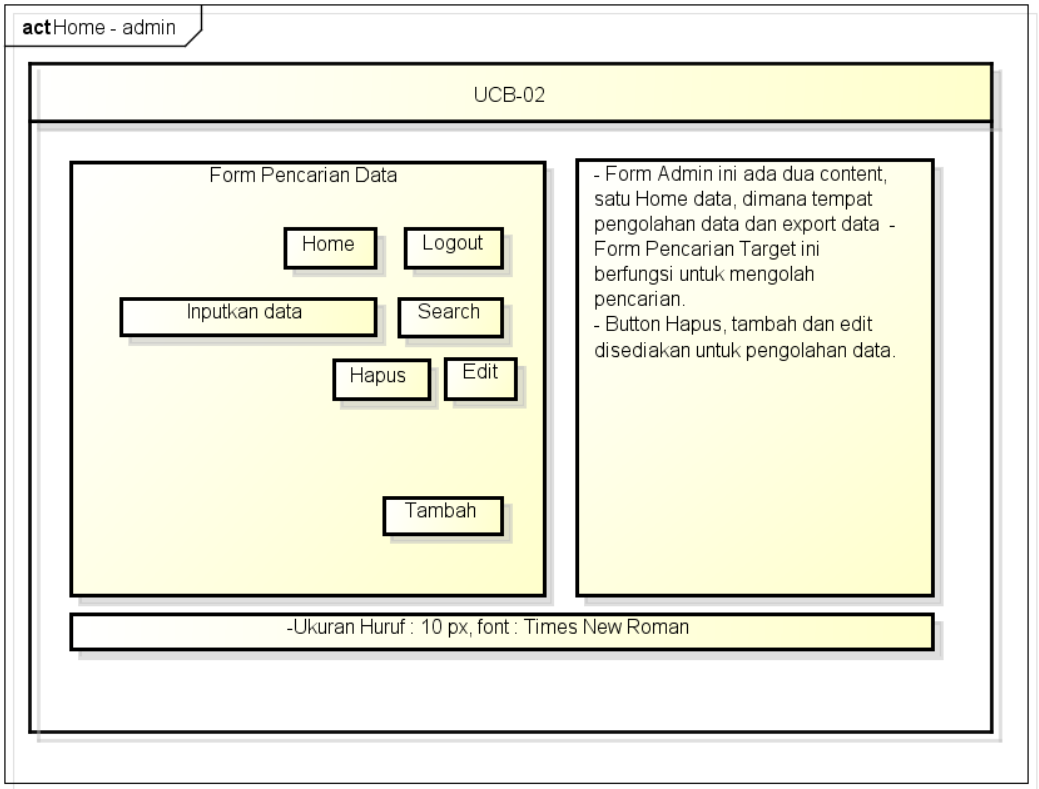
Gambaran umum dari perancangan antarmuka form *Login -Admin* pada aplikasi mobile ini dapat dilihat pada Gambar 3.34




**Gambar 3.34 Tampilan Form Login-Admin**

### 3.4.4.1.3 Perancangan Form *Pengolahan Data-Admin*

Gambaran umum dari perancangan antarmuka form Pengolahan Data-Admin pada aplikasi mobile dapat dilihat pada Gambar 3.35.




**Gambar 3.35 Tampilan Form Pengolahan Data Admin**

### 3.4.5 Jaringan Semantik

Jaringan semantic sistem yang akan dibangun dibedakan sesuai dengan jenis penggunanya yaitu pengguna, admin. Jaringan semantiknya dapat dilihat pada gambar dibawah ini :

#### 3.4.5.1 Jaringan Semantik Admin

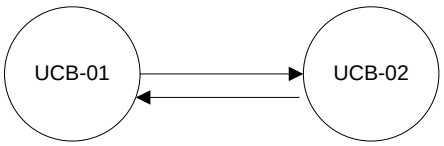
Jaringan semantik pengguna dapat dilihat pada Gambar 3.36




Gambar 3. 36 Jaringan Semantik Aplikasi Frontend

### 3.4.5.2 Jaringan Semantik Admin

Jaringan semantik admin dapat dilihat pada Gambar 3.37




Gambar 3. 37 Jaringan Semantik Aplikasi Backend