BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem bertujuan untuk mengidentifikasikan permasalahan-permasalahan yang terdapat pada sistem serta menentukan kebutuhan-kebutuhan dari sistem yang dibangun. Analisis tersebut meliputi analisis masalah, analisis arsitektur sistem, analisis kebutuhan non fungsional dan analisis data.

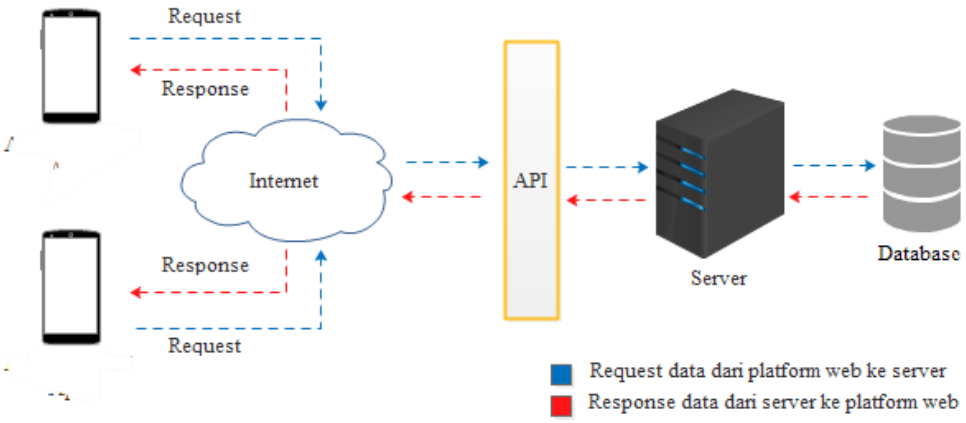
3.1.1 Analisis Masalah

Bagi pengguna jalan Kabupaten Purwakarta, adanya fasilitas yang mewadahi informasi mengenai tempat tujuan wisata dan jalan yang aman untuk dilalui sangat dibutuhkan, namun dewasa ini belum ada fasilitas yang dapat mewadahi permasalahan tersebut, masalah-masalah tersebut diantaranya :

1. Tidak adanya informasi pemetaan tempat-tempat usaha dan wisata.
2. Minimnya informasi mengenai keadaan jalan yang dapat menjadikan indikasi/ faktor terjadinya kecelakaan.

3.1.2 Analisis Arsitektur Sistem

Platform mobile adalah sistem yang dipilih untuk pembangunan perangkat lunak ini. Arsitektur perangkat lunak pada platform mobile mengambarkan bagaimana perangkat lunak saling berinteraksi seperti diilustrasikan pada Gambar 3.1 Arsitektur Perangkat Lunak pada Platform Mobile. Gambar tersebut menggambarkan keseluruhan arsitektur sistem pada platform mobile.



Gambar 3.1 Arsitektur Perangkat Lunak

Berikut adalah deskripsi dari Gambar 3.1 Arsitektur Perangkat Lunak pada Platform Mobile:

a. Perangkat mobile pengguna melakukan request data ke server melalui API
b. Server menerima request data dari server dan menentukan jenis request yang diminta
c. Server menerima permintaan data dan server akan mengambil data yang ada di database
d. Setelah server menerima data yang diminta data tersebut akan dikembalikan dalam bentuk JSON untuk diproses perangkat mobile pengguna.

Arsitektur sistem yang digambarkan pada Gambar 3.1 Arsitektur Perangkat Lunak pada Platform Mobile menggambarkan arsitektur sistem secara keseluruhan.

3.1.3 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak terbagi kedalam dua kebutuhan yaitu kebutuhan non fungsional dan fungsional. Kebutuhan fungsional dapat dilihat pada Tabel 3.1 Spesifikasi Kebutuhan Fungsional.

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-001	Sistem mobile menyediakan fasilitas bagi pengguna untuk register kedalam sistem
SKPL-F-002	Sistem mobile menyediakan fasilitas bagi pengguna untuk login kedalam sistem
SKPL-F-003	Sistem mobile menyediakan fasilitas bagi pengguna untuk menampilkan timeline informasi kondisi jalan
SKPL-F-004	Sistem mobile menyediakan fasilitas bagi pengguna untuk membuat route ke tempat tujuan dari lokasi awal,
SKPL-F-005	Sistem mobile menyediakan fasilitas bagi pengguna untuk menampilkan lokasi pelaku usaha tambal ban, bensin eceran dan hambatan yang terdekat dari user.
SKPL-F-006	Sistem mobile menyediakan fasilitas bagi pengguna untuk melakukan upload kondisi jalan
SKPL-F-007	Sistem mobile menyediakan fasilitas bagi pengguna untuk melakukan lihat profil pengguna
SKPL-F-008	Sistem mobile menyediakan fasilitas bagi pengguna untuk melakukan edit profil pengguna
SKPL-F-009	Sistem mobile menyediakan fasilitas bagi pengguna untuk melakukan pencarian pelaku usaha tambal ban dan bensin eceran
SKPL-F-010	Sistem mobile menyediakan fasilitas bagi pengguna untuk melakukan rating/penilaian terhadap pelaku usaha tambal ban dan bensin eceran
SKPL-F-011	Sistem mobile menyediakan fasilistas recovery lupa password

Tabel 3.1 Spesifikasi Kebutuhan Fungsional

Kebutuhan non fungsional pada perangkat lunak yang dibangun dapat dilihat pada Tabel 3.2 Spesifikasi Kebutuhan Non Fungsional.

Tabel 3.2 Spesifikasi Kebutuhan Non Fungsional

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-NF-001	Sistem yang dibangun adalah platform mobile
SKPL-NF-002	Sistem yang dibangun minimal menggunakan sistem operasi android versi 4.0 Ice Cream Sandwich
SKPL-NF-003	Sistem yang dibangun menggunakan perangkat smartphone
SKPL-NF-004	Sistem dibangun dengan spesifikasi hardware yang memenuhi standar minimum kebutuhan
SKPL-NF-005	Desain *user interface* pada perangkat lunak yang akan dibangun mengacu kepada *design guidelines* android dari google

3.1.4 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional pada sistem ini meliputi analisis perangkat lunak, analisis perangkat keras dan perangkat pikir.

1. Product Requirement

Berikut adalah kebutuhan non fungsional berdasarkan klasifikasi product requirement.

a. SKPL-NF-001 Sistem yang dibangun merupakan platform mobile.
b. SKPL-NF-002 Sistem yang dibangun minimal menggunakan sistem operasi android versi 4.0 Ice Cream Sandwich dan dibangun dengan kebutuhan perangkat lunak minimum:
    a) Sistem Operasi (platform bebas)
    b) Web server XAMPP versi 1.1.7 ke atas sebagai web server
    c) Kode editor berupa Notepad++
    d) Integrated development environment (IDE) berupa Android Studio.
    e) Android Development Tools versi 23.0.0 sebagai plugins Android Studio.
    f) Android Software Development Kit sebagai framework android
c. SKPL-NF-003 Sistem yang dibangun menggunakan smartphone.
d. SKPL-NF-004 Sistem dibangun dengan spesifikasi hardware yang memenuhi standar minimum kebutuhan seperti:
    a) Komputer dengan VGA on board 512 MB
    b) Layar dengan resolusi 1024x600
    c) Komputer dengan ram 2 GB
    d) Perangkat android dengan ram 512
    e) Device android dengan resolusi 320x480

2. Organisational Requirement

Berikut adalah kebutuhan non fungsional berdasarkan klasifikasi Organisational Requirement yaitu SKPL-NF-005 Desain user interface pada perangkat lunak yang akan dibangun mengacu kepada design guidelines android dari google.

3. Analisis Pengguna

Analisis pengguna sistem dimaksudkan untuk mengetahui siapa saja pengguna yang terlibat dalam menjalankan sistem, sehingga dapat diketahui

tingkat dan pemahaman pengguna. Berikut pengguna sistem, dijelaskan pada Tabel 3.3 dibawah ini.

Tabel 3.3 Analisis Pengguna

No	Pengguna	Karakteristik
1.	Masyarakat	a. Dapat mengoperasikan *smartphone* b. Memiliki pengalaman menggunakan internet c. Mengerti pengaturan GPS d. Mengerti pengaturan internet

Spesifikasi pengguna yang terlibat dalam sistem yang akan dibangun dapat dideskripsikan sebagai berikut:

1. Masyarakat yang menjadi pengguna aplikasi.

Tabel 3.4 Analisis Pengguna Masyarakat

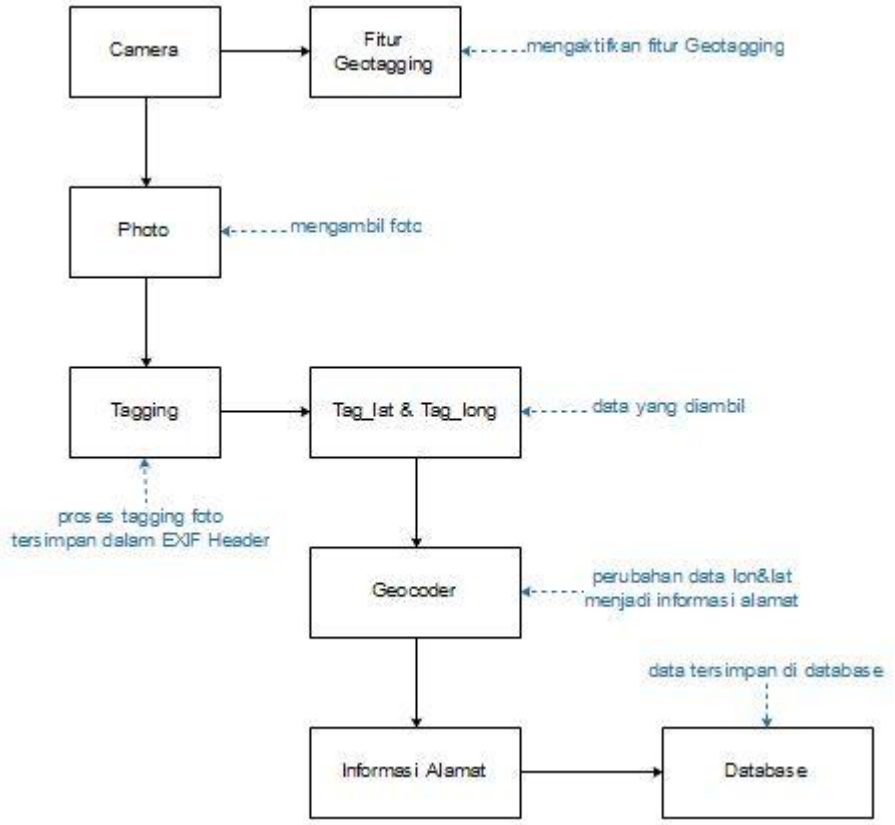
Pengguna	Masyarakat
Tanggung Jawab	Pengaturan profil pengguna, mengupload kondisi jalan sekitar (foto), memberikan rating/ penilaian.
Hak Akses	Melihat informasi kondisi jalan sekitar, membuat route ke lokasi tujuan dari lokasi awal, melihat informasi tempat usaha, wisata di sekitar jalan route tujuan, mengupload/melihat photo tagging kondisi jalan sekitar, pengaturan profil pengguna, melakukan pencarian tempat-tempat usaha dan wisata, melihat rating tempat usaha
Tingkat Keterampilan	a. Memiliki kemampuan dalam mengoperasikan *smartphone* b. Memiliki kemampuan dalam menggunakan internet
Pengalaman	a. Pernah mengakses internet b. Pernah mengoperasikan *smartphone*

Setelah dilakukan analisis, dapat disimpulkan bahwa pengguna pada sistem yang akan dibangun hanya masyarakat dan telah memenuhi syarat untuk menjalankan aplikasi yang akan dibangun ini.

3.1.5 Analisis Data

3.1.5.1 Proses Pengambilan Data Geotagging

Aplikasi ini memanfaatkan teknologi geotagging yang sudah tertanam di dalam camera smartphone android, data geotagging yang diambil adalah tag_latitude dan tag_longitude. Tag tersebut akan digunakan sebagai data masukan untuk proses geocoder. Hasil dari geocoder adalah berupa informasi alamat tempat pengambilan foto. Dibawah ini merupakan gambar proses pengambilan data geotagging sebagai masukan geocoder hingga penyimpanan data ke database.



Gambar 3.2 Alur Pengambilan Data Geotagging

1. Dibawah ini merupakan coding untuk menampilkan *Action Button* untuk memanggil method getCamera().

```
FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        //intent = new Intent(MainActivity.this, LaporanActivity.class);
        //startActivity(intent);
        getCamera();
    }
});
```

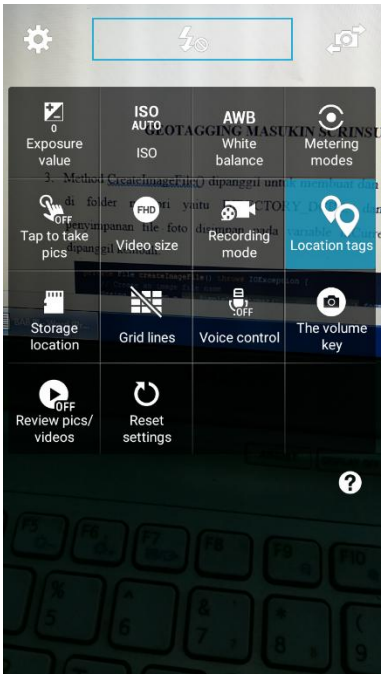
Gambar 3.3 Source Code untuk memanggil method getCamera()

2. Method getCamera() digunakan untuk memanggil fitur camera pada smartphone android agar user dapat melakukan pemotretan pelaporan.

```
public void getCamera() {
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    // Ensure that there's a camera activity to handle the intent
    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
        // Create the File where the photo should go
        File photoFile = null;
        try {
            photoFile = createImageFile();
        } catch (IOException ex) {
            // Error occurred while creating the File
            ex.printStackTrace();
        }
        // Continue only if the File was successfully created
        if (photoFile != null) {
            takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT,
                    Uri.fromFile(photoFile));
            startActivityForResult(takePictureIntent, REQUEST_CAMERA);
        }
    }
}
```

Gambar 3.4 Source Code method getCamera()

Bila fitur camera sudah muncul maka lakukan penyetingan fitur geotagging pada kamera. Seperti pada gambar dibawah ini.



Gambar 3.5 Mengaktifkan fitur geotagging

3. Method CreateImageFile() dipanggil untuk membuat dan menyimpan file foto di folder memori yaitu DIRECTORY_DCIM, dan alamat direktori penyimpanan file foto disimpan pada variable mCurrentPhotoPath untuk dipanggil kembali.

```
private File createImageFile() throws IOException {
    // Create an image file name
    String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
    String imageFileName = "IMG_" + timeStamp + "_";
    File storageDir = Environment.getExternalStoragePublicDirectory(
            Environment.DIRECTORY_DCIM);

    File image = File.createTempFile(
            imageFileName,  /* prefix */
            ".jpg",         /* suffix */
            storageDir      /* directory */
    );

    // Save a file: path for use with ACTION_VIEW intents
    mCurrentPhotoPath = image.getAbsolutePath();

    Log.e("Path", mCurrentPhotoPath);
    return image;
}
```

Gambar 3.6 Souce Code method createImageFile()

4. Method OnActivityResult() selanjutnya akan memanggil LaporanActivity dan disertai dengan variable mCurrentPhotoPath dipanggil untuk ditampilkan di LaporanActivity.

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == REQUEST_CAMERA){
        if (resultCode == RESULT_OK) {
            Intent i = new Intent(getApplicationContext(), LaporanActivity.class);
            i.putExtra("mCurrentPhotoPath", mCurrentPhotoPath);
            //i.putExtra("tag_datetime", tag_datetime);
            startActivity(i);
        }
    }
}
```

Gambar 3.7 Source Code method OnActivityResult()

5. Method setPic() berada di LaporanActivity method ini berfungsi untuk menampilkan foto yang sudah diambil user ke tampilan Posting Laporan.

```
private void setPic() {
    // Get the dimensions of the View
    File imgFile = new File(mCurrentPhotoPath);
    if(imgFile.exists())
    {
        Bitmap myBitmap = BitmapFactory.decodeFile(imgFile.getAbsolutePath());
        picture = (ImageView) findViewById(R.id.imgLaporan);
        picture.setImageBitmap(myBitmap);
    }
}
```

Gambar 3.8 Source Code Method setPic()

6. Method getTag() dipanggil untuk mengambil tagging waktu, latitude dan longitude yang terdapat dalam ExifInterface foto. mCurrentPhoto berisi alamat direktori penyimpanan foto yang di ambil user sebagai bahan pelaporan.

```
private void getTag() {
    //untuk mengambil tagging waktu, lotitude, dan longitude dalam foto
    try {
        exif = new ExifInterface(mCurrentPhotoPath);
        tag_datetime = exif.getAttribute(ExifInterface.TAG_DATETIME);
        String attrLATITUDE = exif.getAttribute(ExifInterface.TAG_GPS_LATITUDE);
        String attrLATITUDE_REF = exif.getAttribute(ExifInterface.TAG_GPS_LATITUDE_REF);
        String attrLONGITUDE = exif.getAttribute(ExifInterface.TAG_GPS_LONGITUDE);
        String attrLONGITUDE_REF = exif.getAttribute(ExifInterface.TAG_GPS_LONGITUDE_REF);
        if ((attrLATITUDE != null)
                && (attrLATITUDE_REF != null)
                && (attrLONGITUDE != null)
                && (attrLONGITUDE_REF != null)) {
            if (attrLATITUDE_REF.equals("N")) {
                latitude = convertToDegree(attrLATITUDE);
                tag_lat = String.valueOf(latitude);
            } else {
                latitude = 0 - convertToDegree(attrLATITUDE);
                tag_lat = String.valueOf(latitude);
            }

            if (attrLONGITUDE_REF.equals("E")) {
                longitude = convertToDegree(attrLONGITUDE);
                tag_long = String.valueOf(longitude);
            } else {
                longitude = 0 - convertToDegree(attrLONGITUDE);
                tag_long = String.valueOf(longitude);
            }
        }
```

Gambar 3.9 Source Code method getTag()

7. Method convertToDegree digunakan untuk mengkonversi tag_latitude dan tag_longitude menjadi masukan yang dapat diterima oleh Geocoder. Karena hasil tagging ExifHeader berupa koordinat derajat lintang.

```
private Double convertToDegree(String stringDMS) {
    Float result = null;
    String[] DMS = stringDMS.split(",", 3);

    String[] stringD = DMS[0].split("/", 2);
    Double D0 = new Double(stringD[0]);
    Double D1 = new Double(stringD[1]);
    Double FloatD = D0 / D1;

    String[] stringM = DMS[1].split("/", 2);
    Double M0 = new Double(stringM[0]);
    Double M1 = new Double(stringM[1]);
    Double FloatM = M0 / M1;

    String[] stringS = DMS[2].split("/", 2);
    Double S0 = new Double(stringS[0]);
    Double S1 = new Double(stringS[1]);
    Double FloatS = S0 / S1;

    result = new Float(FloatD + (FloatM / 60) + (FloatS / 3600));

    return Double.valueOf(result);
}
```

Gambar 3.10 Source Code method convertToDegree()

8. Dibawah ini coding untuk menampilkan alamat hasil geocoder. Apabila method getTag() tidak dapat menemukan tag_latitude dan tag_longitude maka inputan geocoder akan didapat dari GPS Android. Namun, apabila method getTag() berhasil menemukan tag_latitude dan tag_longitude maka selanjutnya inputan geocoder akan berasal dari Geotagging Foto yang sebelumnya telah di konversi.

```
        GPSTracker tracker = new GPSTracker(this);
        if (!tracker.canGetLocation()) {
            tracker.showSettingsAlert();
        }
        latitude = tracker.getLatitude();
        longitude = tracker.getLongitude();

        Geocoder geocoder = new Geocoder(this, Locale.getDefault());
        List<Address> addresses = null;
        try {
            if (tag_lat ==  null) {
                addresses = geocoder.getFromLocation(latitude, longitude, 1);
                address = addresses.get(0).getAddressLine(0);
                aq.id(R.id.tvLokasi).text("Lokasi anda saat ini di sekitar " + address);
                lat_user = String.valueOf(addresses.get(0).getLatitude());
                long_user = String.valueOf(addresses.get(0).getLongitude());
            } else {
                //untuk geocoder
                addresses = geocoder.getFromLocation(Double.parseDouble(tag_lat), Double.parseDouble(tag_long), 1);
                address = addresses.get(0).getAddressLine(0);
                lat_user = tag_lat;
                long_user = tag_long;
                aq.id(R.id.tvLokasi).text("Lokasi anda saat ini di sekitar " + address);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

Gambar 3.11 Source Code menampilkan hasil geocoder

9. Foto yang berada di mCurrentPhotoPath dirubah menjadi array agar dapat di kirim ke database.

```
Bitmap bm = BitmapFactory.decodeFile(mCurrentPhotoPath);
ByteArrayOutputStream bao = new ByteArrayOutputStream();
bm.compress(Bitmap.CompressFormat.JPEG, 50, bao);
byte[] ba = bao.toByteArray();
ba1 = Base64.encodeToString(ba, Base64.DEFAULT);
getTag();
```

Gambar 3.12 Souce Code merubah foto menjadi bentuk Array

10. Perintah dibawah ini berfungsi untuk mengirim data laporan ke API.

```
String s_deskrpisi = aq.id(R.id.edDeskripsi).getText().toString();
String url = AppConstans.APIKIRIMLAPORAN;
Map<String,String> params = new HashMap<>();
params.put("id_pengguna",id_pengguna);
params.put("lokasi",address);
params.put("deskripsi",s_deskrpisi);
params.put("waktu_laporan",tag_datetime);
params.put("lat_laporan",lat_user);
params.put("long_laporan",long_user);
params.put("base64", ba1);
params.put("img",System.currentTimeMillis() + ".jpg");

APIManager.Register(this, new OnApiListener() {
    @Override
    public void onApiStartLoad(String url) {
        showProgressDialog();
    }

    @Override
    public void onApiSuccessLoad(String url, String message, Object object, int resultNumber) {
        hideProgressDialog();
        dialogExit();
    }

    @Override
    public void onApiErrorLoad(String url, String message) {
        hideProgressDialog();
        dialogExit();
    }
}, url, params);
}
```

Gambar 3.13 Source Code untuk mengirim data ke API

11. AppConstants merupakan class yang menampung daftar alamat API.

```
public class AppConstans {

    public static final String TAG="swap";
    public static final boolean isLogging=true;

    public static final String APIHost="http://www.aslamabdulatif.web.id/adminswap/public/";

    public static final String keyConfigStatus="STATUS";
    public static final String keyData="DATA";
    public static final String keyMessage="MESSAGE";

    public static final String keyStatusSuccess="SUCCESS";

    public static final String APIBUATAKUN = APIHost+"buat_akun.php";
    public static final String APILISTLAPORAN = APIHost+"list_laporan.php";
    public static final String APIKIRIMLAPORAN = APIHost+"kirim_laporan.php";
    public static final String APIKIRIMLOKASI = APIHost+"kirim_lokasi.php";
    public static final String APILISTLOKASI = APIHost+"list_lokasi.php";
    public static final String APIGETTWITTER = APIHost+"get_twitter.php";
    public static final String APICATEGORY = APIHost+"get_category.php";
    public static final String APIDETAIL = APIHost+"get_detail.php";
    public static final String APILISTDISTANCE = APIHost+"list_lokasi_bydistance.php";
    public static final String APILAPORANKU = APIHost+"list_laporanku.php";
    public static final String APIEDIT = APIHost+"edit_profile.php";
    public static final String APILUPA = APIHost+"lupa_password.php";
```

Gambar 3.14 Source Code AppConstants

12. Pada Kirim_Laporan.php API menerima data dari android dan menginputkan data ke database.

```php
<?php
include ("koneksi.php");
$id_pengguna= $_POST['id_pengguna'];
$lokasi= $_POST['lokasi'];
$waktu_laporan= $_POST['waktu_laporan'];
$lat_laporan= $_POST['lat_laporan'];
$long_laporan= $_POST['long_laporan'];
$deskripsi= $_POST['deskripsi'];
$hash = md5( rand(0,1000) );
$imsrc = base64_decode($_POST['base64']);
$img= $_POST['img'];
header('Content-Type: bitmap; charset=utf-8');
    /*Lokasi penyimpanan gambar yang diupload*/
$path = "image/$img";
$handle = fopen($path, 'wb');
$numbytes = fwrite($handle, $imsrc);

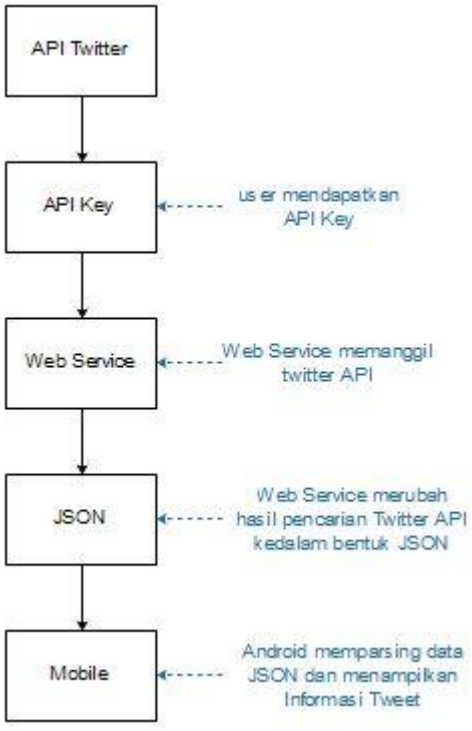
$q = mysql_query("INSERT INTO `t_laporan`(`id_pengguna`, `img_laporan`, `deskripsi_laporan`, `waktu_laporan`, `lokasi_laporan`, `lat_laporan`, `long_laporan`) VALUES
('$id_pengguna',CONCAT('http://www.originalvikingpersibmerchandise.com/swap_api/image/','$img'),'$deskripsi', '$waktu_laporan', '$lokasi', '$lat_laporan',
'$long_laporan') ");

?>
```

Gambar 3.15 Source Code mengirim data dari android ke Database

3.1.5.2 Proses Pengambilan Data Twitter

Aplikasi ini menggunakan teknik grab tweet dengan menggunakan Twitter API. Pertama, API Twitter memberikan API Key sebagai verifikasi. Lalu Web Service memanggil Twitter API dan merubah hasil pencarian dalam bentuk JSON. Setelah itu Android memparsing data JSON tersebut dan menampilkan informasi tweet. Dibawah ini merupakan gambar proses pengambilan data Twitter API.



Gambar 3.16 Alur Pengambilan Data Twitter

1. Mendaftar pada API Twitter (apps.twitter.com) untuk mendapatkan consumerkey, consumersecret, accesstoken dan accesstokensecret , lalu setelah itu key-key tersebut di inputkan pada webservice untuk autentikasi agar dapat mengakses API.
2. Data yang di ambil adalah data dari Search API, hasil pencarian yang ditampilkan berdasarkan keyword pencarian. Untuk penelitian ini keyword yang digunakan yaitu "Jalan Rusak Purwakarta". Selanjutnya hasil pencarian akan diparsing menggunakan format JSON.

```php
<?php
session_start();
require_once("twitteroauth.php"); //Path to twitteroauth library

$twitteruser = "@aaslamm";
$notweets = 30;
$consumerkey = "PSapGC4fPsmJfFgtQGWhIeY5y";
$consumersecret = "57ePAeqknf8yeeNPi4Q5eQIlCr0E4r2OfAfHQrgtZjWfx6xf8X";
$accesstoken = "497619382-nzeXkhvrM9blmxpVL7H6fGUULsbR16IEMpPiMzte";
$accesstokensecret = "89i59qo5d6NJiA9aTm3UACYFK1klxExbN99mqrQkIGJDV";

function getConnectionWithAccessToken($cons_key, $cons_secret, $oauth_token, $oauth_token_secret) {
  $connection = new TwitterOAuth($cons_key, $cons_secret, $oauth_token, $oauth_token_secret);
  return $connection;
}

$connection = getConnectionWithAccessToken($consumerkey, $consumersecret, $accesstoken, $accesstokensecret);

$tweets = $connection->get("https://api.twitter.com/1.1/search/tweets.json?q=jalan%20rusak%20purwakarta%20filter%3Aimages&include_entities=true");

echo json_encode($tweets);
?>
```

Gambar 3.17 Source Code Twitter API

3. Method getData() dipanggil untuk mengambil data yang telah di parsing dalam format JSON dari webservice melalui class AppConstants.

```java
public void getData(){
    String url = AppConstans.APIGETTWITTER;
    Map<String, String> params = new HashMap<>();
    //params.put("lat_user", strlat);
    //params.put("long_user", strlon);
    APIManager.loadTwitterList(this, new OnApiListener() {
        @Override
        public void onApiStartLoad(String url) { showProgressDialog(); }

        @Override
        public void onApiSuccessLoad(String url, String message, Object object, int resultNumber) {
            hideProgressDialog();
            tweetItems = (ArrayList<TweetItem>) object;
            setList();
        }

        @Override
        public void onApiErrorLoad(String url, String message) {
            hideProgressDialog();
            showToast(message);
        }
    }, url, params);
}
```

Gambar 3.18 Source Code getData() Twitter

4. Class TweetItem dipanggil untuk menampung data array yang ada di JSON hasil pengambilan method getData().

```
public class TweetItem implements Serializable {

    public String id;
    public String created_at;
    public String text;
    public String media_url;
    public JSONObject entities;
    public JSONObject c;
    public JSONArray media;

    public TweetItem(JSONObject object) throws JSONException {
        id = object.getString("id");
        created_at = object.getString("created_at");
        text = object.getString("text");
        media = object.getJSONObject("entities").getJSONArray("media");
        //media = entities.getJSONArray("media");
        for (int i = 0; i < media.length(); i++) {
            c = media.getJSONObject(i);
            media_url = c.getString("media_url");
        }
    }
}
```

Gambar 3.19 Source Code Class TweetItem

5. Method getView() berada pada class TweetAdapter, method ini dipanggil untuk menginputkan array yang ditampung oleh class tweetItem kedalam layout tampilan tweetActivity.

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {

    final ViewHolder holder;
    if (convertView == null) {
        convertView = li.inflate(R.layout.list_tweet, null);
        holder = new ViewHolder(convertView);
        convertView.setTag(holder);
    } else {
        holder = (ViewHolder) convertView.getTag();
    }

    AQuery aq = new AQuery(convertView);
    aq.id(holder.img).image(mItems.get(position).media_url, true, true, 200, 0);
    aq.id(holder.tvDeskripsi).text(mItems.get(position).text);

    return convertView;
}
```

Gambar 3.20 Source Code method getView()

6. Method setList() dipanggil untuk menampilkan array yang sudah diinputkan pada layout tampilan TweetActivity untuk akhirnya tampil di layar android.

```
private void setList() {
    TweetAdapter tweetAdapter = new TweetAdapter(this,tweetItems);
    listView.setAdapter(tweetAdapter);
    listView.setExpanded(true);
}
```

Gambar 3.21 Source Code method setList()

3.1.5.3 Struktur Data JSON

Analisis data yang digunakan dalam penelitian ini adalah Object Relational Mapping (ORM). Object relational mapping melakukan pemetaan terhadap tabel-tabel pada basis data relasional dengan suatu class entitas yang ada pada bahasa pemrograman berorientasi objek.

Pemetaan yang dilakukan ORM akan membutuhkan suatu jembatan berupa format data JSON yang dapat menghubungkan ORM dengan database fisik yang ada pada server. Struktur format data JSON yang digunakan pada penelitian ini terbagi menjadi tiga yaitu data objek, data array dan data null.

a. Data Objek

Format pada Tabel 3.5 Struktur JSON Data Objek digunakan ketika data yang diterima dari server merupakan data tunggal atau berupa satu objek

Tabel 3.5 Struktur JSON Data Objek

Struktur Data Objek

{
“key” : “value”
}

b. Data Array

Format pada Tabel 3.6 Struktur JSON Data Array digunakan ketika data yang diterima dari server merupakan data tunggal atau berupa satu objek

Tabel 3.6 Struktur JSON Data Array

Struktur Data Array

{
"result" : "value",
"item" : [{ "key" : "value" },{ "key" : "value" }]
}

c. Data Null

Format pada Tabel 3.7 Struktur JSON Data Null digunakan ketika data yang diterima dari server merupakan data tunggal atau berupa satu objek

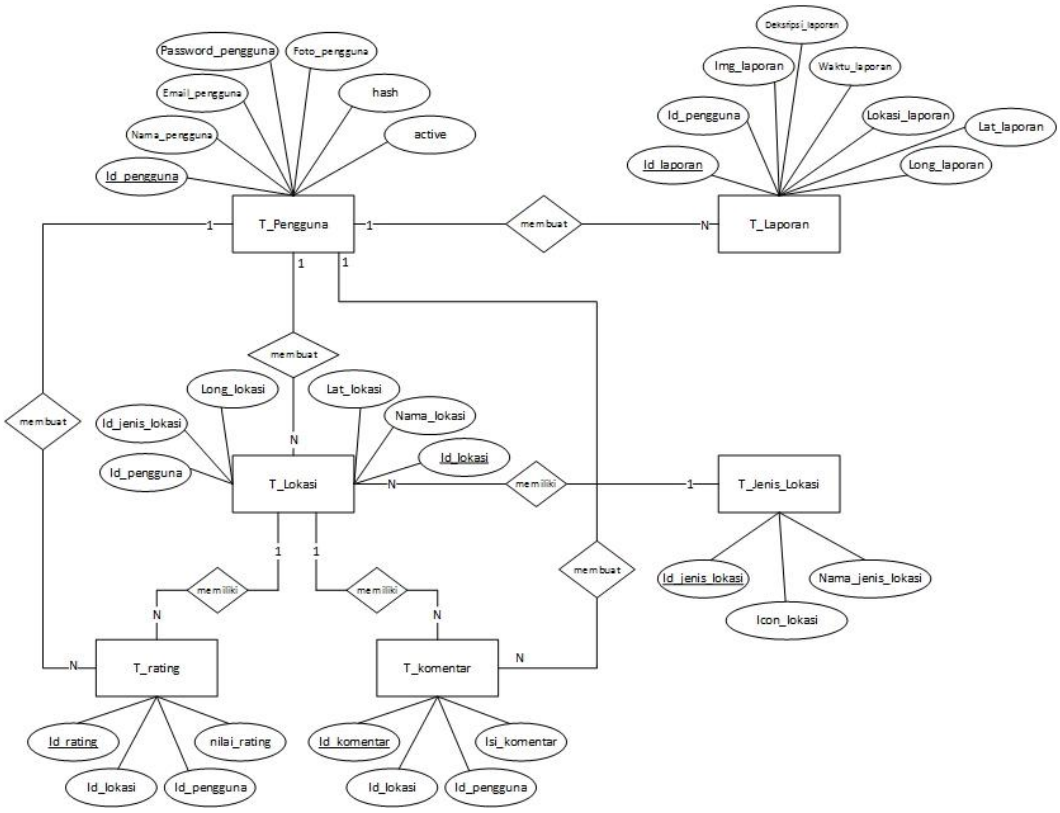
Tabel 3.7 Struktur JSON Data Null

Struktur Data Null

{
"result" : "value",
"message" : "value",
"status_code":value
}

3.1.5.4 Entity Relation Diagram (ERD)

Dibawah ini merupakan Diagram Entitas dari sistem yang dibangun :

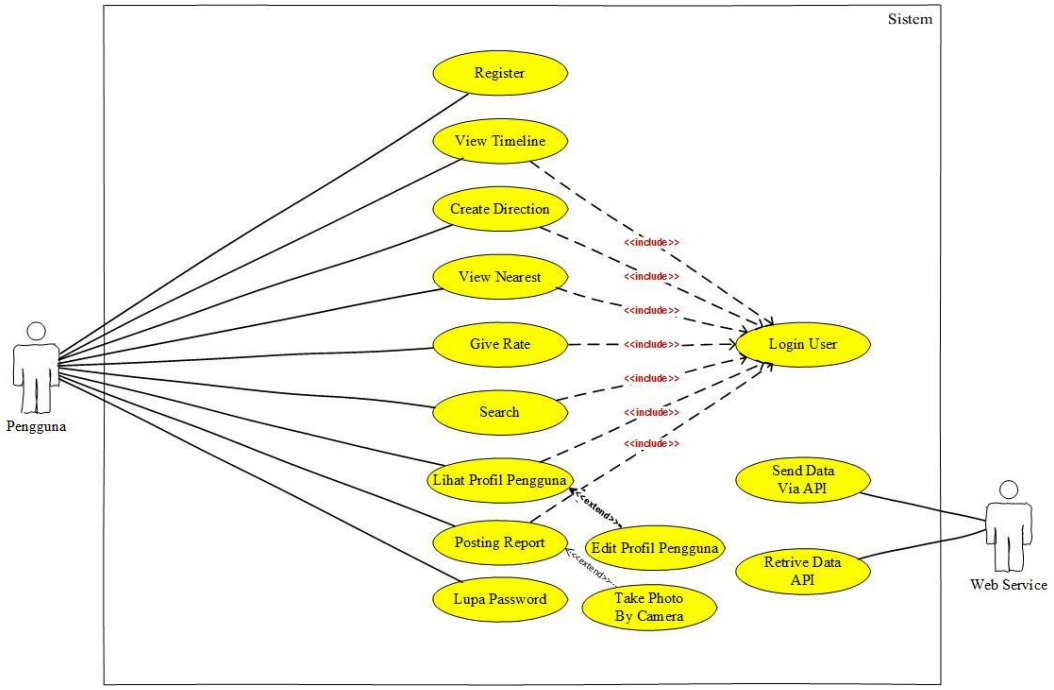


Gambar 3.22 Entity Relation Diagram

3.1.6 Analisis Kebutuhan Fungsional

1. Diagram Use Case

Diagram Use Case merupakan pemodelan untuk menggambarkan kelakuan (behavior) perangkat lunak yang akan dibuat. Berikut adalah Gambar 3.2 Diagram Use Case. Garis keterhubungan yang dimiliki use case Send data via API dan Retrieve data via API tidak digambarkan untuk mempermudah pembacaan diagram.



Gambar 3.23 Use Case Diagram

2. Definisi Aktor

Definsi Aktor diisi dengan daftar actor dan deskripsi role untuk actor tersebut. Deskripsi role harus menjelaskan wewenang pada role tersebut dalam perangkat lunak. Bisa dibuat dalam bentuk Tabel 3.8 Definisi Aktor berikut.

Tabel 3.8 Definisi Aktor

No	Actor	Deskripsi
1	Pengguna	Aktor dengan role ini memiliki wewenang untuk menggunakan fungsionalitas pada level user interface.
2	Web Service	Actor dengan role ini memiliki wewenang sebagai penyedia API / jembatan sumber data dan berjalan secara *background*.

3. Definisi Use Case

Pada Tabel 3.9 Definisi Use Case berisi daftar use case dan deskripsi singkat mengenai use case tersebut.

Tabel 3.9 Definisi Use Case

No	Use Case	Deskripsi
1	Register	Sistem menampilkan form Register untuk pengguna yang terdiri dari Nama Lengkap, E-Mail dan Password
2	Login	Sistem menampilkan form login untuk pengguna yang terdiri dari E-Mail dan Password
3	View Timeline	Sistem menampilkan berita-berita perkembangan kondisi jalan
4	Create Direction	Sistem menampilkan form Create Route untuk pengguna yang terdiri dari posisi awal dan posisi yang akan dituju
5	View Nearest	Sistem menampilkan pemetaan tempat usaha dan wisata yang memiliki tempat terdekat dari user. Dan memposting tempat usaha baru yang belum terdaftar.
6	Posting Report	Sistem menampilkan form untuk melakukan upload/share kondisi jalan sekitar (foto) yang selanjutnya berada di timeline user dan diberitakan pada user lain.
7	Searching	Sistem menampilkan form untuk melakukan pencarian Pelaku Usaha dengan beberapa Keyword yang dijadikan kategori.
8	Lihat Profile Pengguna	Sistem menampilkan detail profil pengguna
9	Edit Profile Pengguna	Sistem menampilkan form untuk melakukkan pengaturan profil pengguna diantaranya Nama, upload profil picture, email dan pasword
10	Give Rate	Sistem menampilkan form untuk melakukan rating pada tempat usaha dan wisata.
11	Take Photo By Camera	Sistem memanggil fungsi camera yang ada pada device pengguna
12	Lupa Password	Sistem menampilkan menu lupa password, user menginputkan email terdaftar.
13	Send Data Via API	Sistem mengirim data melalui API ke server
14	Retrieve Data API	Sistem menerima data melalui API dari server

4. Skenario Use Case

Skenario yang dibuat ditujukan untuk beberapa use case utama, yang menggambarkan urutan interaksi actor dengan use case tersebut, dari awal sampai akhir.

a. Skenario Use Case Register

Berikut adalah Tabel 3.10 Requirement A.1 yang menjelaskan kebutuhan dari use case register dan Tabel 3.11 Use Case Skenario Register yang menjelaskan skenario dari use case Register.

Tabel 3.10 Requirement A.1

Requirement A.1
Data Register terdiri dari foto, nama lengkap, email dan password.

Tabel 3.11 Use Case Skenario Register

Use Case Name	Register	
Related Requirements	Requirement A.1	
Goal Context	Melakukan Register kedalam perangkat lunak dan terdata di database	
Preconditions	Activity Form Register perangkat lunak ditampilkan	
Successful End Condition	Menampilkan pesan proses Register berhasil	
Failed End Condition	Menampilkan pesan kesalahan input data dan Register gagal	
Primary Actor	Pengguna	
Trigger	Pengguna menekan tombol Register pernagkat lunak	
Main Flow	Step	Action
	1	User menekan tombol “Buat Akun Baru”
	2	Sistem menampilkan form Register perangkat lunak
	3	Pengguna mengisi data Register
	4	Pengguna menekan tombol Register
	5	Sistem melakukan validasi field Register
	6	User mendapatkan email verifikasi
	7	Sistem menampilkan pesan Register berhasil
Extension	Step	Branching Action
	3.1	Nama lengkap, nip dan password kosong
	3.2	Nama lengkap, nip dan password terisi

b. Skenario Use Case Login User

Berikut adalah Tabel Requirement A.2 yang menjelaskan kebutuhan dari use case login dan Tabel 3.13 Use Case Skenario Login yang menjelaskan skenario dari use case login.

Tabel 3.12 Requirement A.2

Requirement A.2
Data login terdiri e-mail dan password.

Tabel 3.13 Use Case Skenario Login

Use Case Name	Login	
Related Requirements	Requirement A.2	
Goal Context	Melakukan login kedalam perangkat lunak	
Preconditions	Activity Form login perangkat lunak ditampilkan	
Successful End Condition	Menampilkan tampilan utama perangkat lunak	
Failed End Condition	Menampilkan pesan kesalahan login	
Primary Actor	Pengguna	
Trigger	Pengguna membuka aplikasi	
Main Flow	Step	Action
	1	Sistem menampilkan form login perangkat lunak
	2	Pengguna mengisi data login
	3	Pengguna menekan tombol login
	4	Sistem melakukan validasi field login
	5	Sistem melakukan verifikasi data login
	6	Sistem menampilkan tampilan utama perangkat lunak
Extension	Step	Branching Action
	4.1	Email dan password kosong
	4.2	Email dan password terisi
	5.1	Email dan password valid
	5.2	Email dan password tidak valid.

c. Skenario Use Case View Timeline

Berikut adalah Tabel 3.14 Use Case Skenario View Timeline yang menjelaskan skenario dari use case view kondisi jalan

Tabel 3.14 Use Case Skenario Timeline

Use Case Name	View Timeline	
Related Requirements	-	
Goal Context	Timeline infromasi kondisi jalan sekitar	
Preconditions	Activity View Timeline ditampilkan	
Successful End Condition	Informasi kondisi jalan sekitar	
Failed End Condition	-	
Primary Actor	Pengguna	
Trigger	Pengguna membuka aplikasi	
Main Flow	Step	Action
	1	Pengguna membuka aplikasi swap
	2	Sistem menampilkan tampilan utama
	3	Sistem meminta data dari server
	4	Sistem menampilkan data timeline informasi kondisi jalan
Extension	Step	Branching Action
	4.1	Sistem gagal menampilkan timeline infomasi kondisi jalan

d. Skenario Use Case Create Direction

Berikut adalah Tabel 3.15 Use Case Skenario Create Direction yang menjelaskan skenario dari use case membuat rute perjalanan menuju lokasi yang dituju.

Tabel 3.15 Use Case View Direction

Use Case Name	Direction	
Related Requirements	-	
Goal Context	Menentukan route perjalanan yang dituju	
Preconditions	Activity Direction ditampilkan	
Successful End Condition	Menampilkan route perjalanan yang dituju	
Failed End Condition	Gagal menampilkan route perjalanan menuju tujuan	
Primary Actor	Pengguna	
Trigger	Pengguna menekan tombol Direction perangkat lunak	
Main Flow	Step	Action

	1	Pengguna menekan tombol Direction
	2	Sistem menampilkan Menu Direction (Starting Point dan Destination)
	3	Pengguna menentukan Starting Point, Destination dan Moda Kendaraan yang digunakan
	4	Sistem mencari lokasi yang dituju
	5	Sistem menampilkan rute menuju tempat tujuan
Extension	Step	Branching Action
	4.1	Sistem gagal mencari lokasi yang dituju
	5.1	Sistem gagal menampilkan rute menuju tempat tujuan

e. Skenario Use Case View Nearest

Berikut adalah Tabel 3.16 Use Case Skenario View Nearest yang menjelaskan skenario dari use case view nearest.

Tabel 3.16 Use Case View Nearest

Use Case Name	Nearest	
Related Requirements	-	
Goal Context	Melihat pelaku usaha dan hambatan di jalanan yang berada di dekat pengguna	
Preconditions	Activity Nearest ditampilkan	
Successful End Condition	Menampilkan posisi pointer pelaku-pelaku usaha dan hambatan-hambatan di jalanan	
Failed End Condition	Gagal menampilkan posisi pointer pelaku-pelaku usaha dan hambatan-hambatan di jalanan	
Primary Actor	Pengguna	
Trigger	Pengguna menekan tombol Nearest	
Main Flow	Step	Action
	1	Pengguna menekan tombol Nearest
	2	Sistem menampilkan tampilan Activity Nearest
	3	Sistem meminta data dari server
	4	Sistem menampilkan posisi peta pelaku-pelaku usaha dan hambatan-hambatan di jalanan
Extension	Step	Branching Action

	3.1	Sistem gagal menampilkan peta lokasi posisi pelaku usaha dan hambatan di jalanan

f. Skenario Use Case Posting Report

Berikut adalah Tabel 3.17 Use Case Skenario Posting Report kondisi jalan yang menjelaskan skenario dari use case Share Posting Report.

Tabel 3.17 Use Case Skenario Posting Report

Use Case Name	Posting Report	
Related Requirements	-	
Goal Context	Melakukan pemotoan dan melaporkan kondisi jalan	
Preconditions	Activity Posting Report perangkat lunak ditampilkan	
Successful End Condition	Membagikan hasil foto, informasi kondisi jalan dan marker lokasi pada peta	
Failed End Condition	Gagal membagikan hasil foto, informasi kondisi jalan dan marker lokasi pada peta	
Primary Actor	Pengguna	
Trigger	Pengguna memilih tombol Post Laporan Kondisi Jalan perangkat lunak.	
Main Flow	Step	Action
	1	Pengguna menekan tombol Post Report laporan Kondisi Jalan
	2	Sistem menampilkan form Deskripsi dan Pengambilan Foto
	3	Pengguna melakukan pemotoan dan mengisi deskripsi foto
	4	Sistem mengambil gambar dari camera
	5	Pengguna menekan tombol “Posting Laporan”
	6	Sistem menampilkan hasil foto, informasi kondisi jalan dan marker lokasi pada peta dan timeline
Extension	Step	Branching Action
	6.1	Sistem gagal mempositng ke timeline
	6.2	Sistem berhasil memposting ke timeline

g. Skenario Use Case Search

Berikut adalah Tabel 3.18 merupakan requirement/kebutuhan dari Use Case Skenario Search Pelaku Usaha yang dijelaskan skenario dari use case Search pada Tabel 3.19.

Tabel 3.18 Requirement A.3

Requirement A.3
Search Pelaku Usaha dapat dicari dengan Nama, Alamat dan Rating Pelaku Usaha.

Tabel 3.19 Use Case Skenario Search

Use Case Name	Search	
Related Requirements	Requirement A.3	
Goal Context	Mencari Pemetaan Pelaku Usaha	
Preconditions	Activity Search pelaku usaha perangkat lunak ditampilkan	
Successful End Condition	Mencari pemetaan pelaku usaha	
Failed End Condition	Gagal melakukan pencarian pemetaan pelaku usaha	
Primary Actor	Pengguna	
Trigger	Pengguna memilih tombol Search	
Main Flow	Step	Action
	1	Pengguna memilih tombol Search Pelaku Usaha perangkat lunak.
	2	Sistem menampilkan form kebutuhan pencarian.
	3	Pengguna menginputkan seperti pada Tabel 3.18 Requirement A.3 dan pengguna menekan tombol Search
	4	Sistem mencari pelaku usaha yang dicari
	5	Sistem menampilkan informasi pelaku usaha yang dicari
Extension	Step	Branching Action
	4.1	Sistem berhasil menemukan informasi pelaku usaha yang dicari
	4.2	Sistem gagal menemukan informasi pelaku usaha yang dicari

h. Skenario Use Case Lihat Profile Pengguna

Berikut adalah Tabel 3.20 Use Case Skenario Lihat Profil Pengguna yang menjelaskan skenario dari use case Lihat Profil Pengguna.

Tabel 3.20 Use Case Skenario Lihat Profil Pengguna

Use Case Name	Lihat Profil Pengguna	
Related Requirements	-	
Goal Context	Melihat detail Profil Pengguna	
Preconditions	Activity Lihat Profil Pengguna perangkat lunak ditampilkan	
Successful End Condition	Profil pengguna ditampilkan	
Failed End Condition	Profil pengguna gagal ditampilkan	
Primary Actor	Pengguna	
Trigger	Pengguna memilih tombol edit profil user	
Main Flow	Step	Action
	1	Pengguna memilih menu lihat profil pengguna
	2	Sistem menampilkan halaman lihat profile pengguna
	3	Sistem menampilkan foto, nama dan fasilitas edit

		profil pengguna dan list laporan dari pengguna
Extension	Step	Branching Action
	3.1	Sistem menampilkan data profil pengguna.
	3.2	Sistem gagal menampilkan data profil pengguna.

i. Skenario Use Case Edit Profile Pengguna

Berikut adalah Tabel 3.21 Use Case Skenario Edit Profil Pengguna yang menjelaskan skenario dari use case Edit Profil Pengguna.

Tabel 3.21 Use Case Skenario Edit Profil User

Use Case Name	Edit Profil	
Related Requirements	-	
Goal Context	Melakukan perubahan Profil User	
Preconditions	Activity Setting Profil User perangkat lunak ditampilkan	
Successful End Condition	Profil User berhasil di set	
Failed End Condition	Profil User gagal di set	
Primary Actor	Pengguna	
Trigger	Pengguna memilih tombol edit profil user	
Main Flow	Step	Action
	1	Pengguna memilih menu edit profile
	2	Sistem menampilkan halaman edit profile
	3	Sistem menampilkan form Nama, Alamat dan fasilitas upload profil picture, email dan pasword
	4	Pengguna menginputkan data edit profile dan melakukan pengyimpanan
	5	Sistem memasukan data ke database
Extension	Step	Branching Action
	4.1	masukan tidak diterima.
	4.2	masukan diterima

j. Skenario Use Case Give Rate

Berikut adalah Tabel 3.22 Use Case Skenario Give Rate yang menjelaskan skenario dari use case give rate.

Tabel 3.22 Use Case Skenario Give Rate

Use Case Name	Give Rate	
Related Requirements	-	
Goal Context	Melakukan Rating Pelaku Usaha	
Preconditions	Activity Rating Pelaku Usaha perangkat lunak ditampilkan	
Successful End Condition	Berhasil melakukan rating pelaku usaha	
Failed End Condition	Gagal melakukan rating pelaku usaha	
Primary Actor	Pengguna	
Trigger	Pengguna memilih tombol Give Rate	
Main Flow	Step	Action

	1	Pengguna memilih tombol Rating Pelaku Usaha
	2	Sistem menampilkan halaman Rating Pelaku Usaha
	3	Sistem menampilkan Nama, Alamat, Detail pelaku usaha dan Form Penilaian Rating
	4	Pengguna menginputkan nilai rating
	5	Sistem menyimpan hasil rating
	6	Sistem menampilkan hasil rating
Extension	Step	Branching Action
	4.1	Masukan rating gagal di simpan
	4.2	Masukan rating diterima
	5.1	Sistem gagal menshare rating
	5.2	Sistem berhasil menshare rating

k. Skenario Use Case Take Photo By Camera

Berikut adalah Tabel 3.23 Use Case Skenario Take Photo By Camera yang menjelaskan skenario dari use case Take Photo By Camera.

Tabel 3.23 Use Case Skenario Take Photo By Camera

Use Case Name	Take Photo by Camera	
Related Requirements	-	
Goal Context	Melakukan pengambilan take photo	
Preconditions	Form Posting Report ditampilkan	
Successful End Condition	Photo berhasil diambil	
Failed End Condition	Photo tidak berhasil diambil	
Primary Actor	Pengguna	
Trigger	Pengguna memilih diambil	
Included Case	-	
Main Flow	Step	Action
	1	Sistem menampilkan activity camera
	2	Pengguna memilih take photo
	3	Sistem mengambil gambar dari camera
Extension	Step	Branching Action
	3.1	Sistem gagal mengambil gambar
	3.2	Sistem berhasil mengambil gambar.

l. Skenario Use Case Lupa Password

Berikut adalah Tabel 3.24 Requirement A.1 yang menjelaskan kebutuhan dari use case lupa password dan Tabel 3.25 Use Case Skenario Lupa Password yang menjelaskan skenario dari use case lupa password.

Tabel 3.24 Requirement A.4

Requirement A.4
Data lupa password yang dibutuhkan adalah email terdaftar

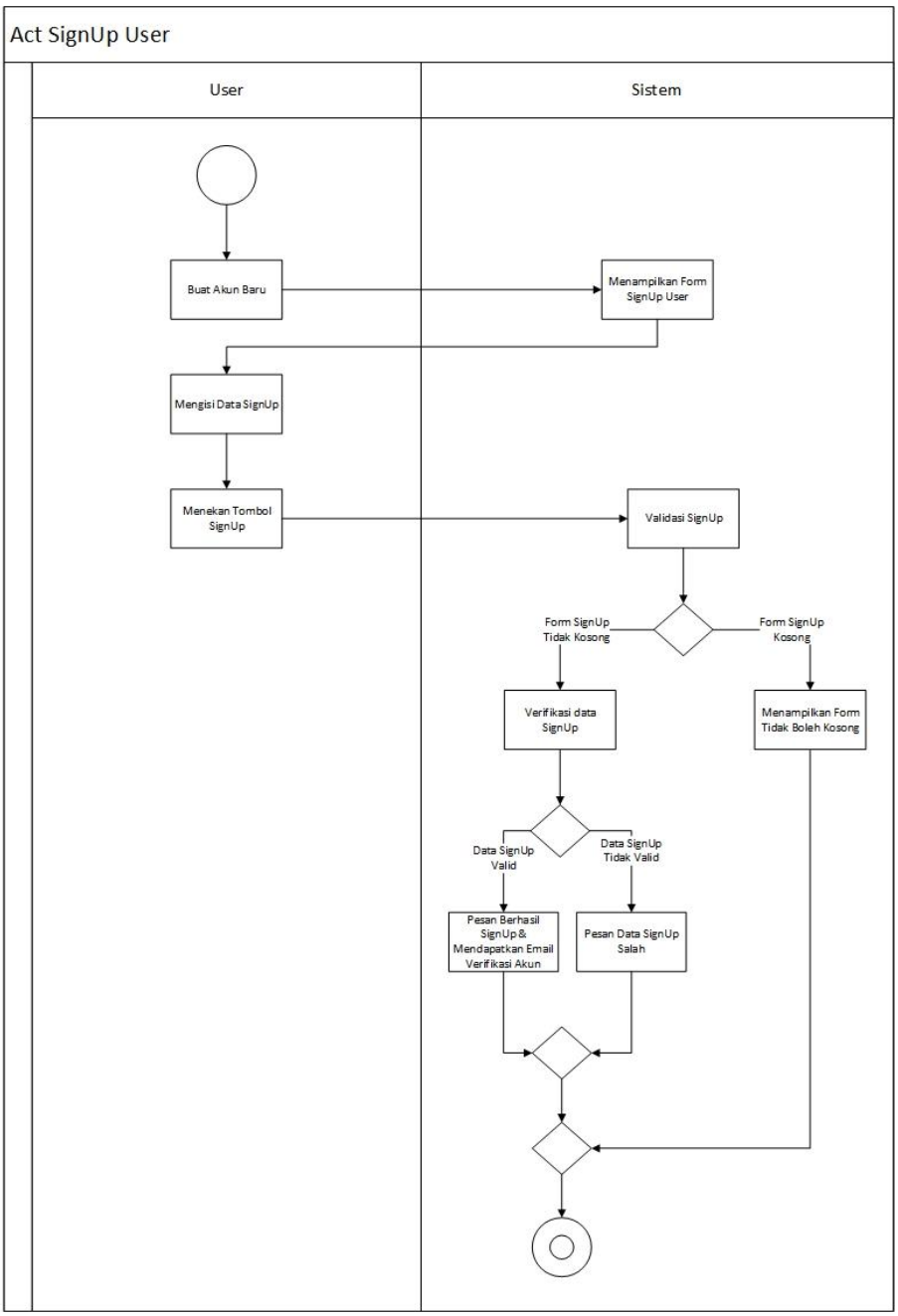
Berikut adalah Tabel 3.24 Use Case Skenario Lupa Password yang menjelaskan skenario dari use case lupa password.

Tabel 3.25 Use Case Skenario Lupa Password

Use Case Name	Lupa Password	
Related Requirements	-	
Goal Context	Password berhasil direcovery	
Preconditions	Form Login ditampilkan	
Successful End Condition	Password berhasil dikirim ke email terdaftar	
Failed End Condition	Password gagal dikirim ke email terdaftar	
Primary Actor	Pengguna	
Trigger	Pengguna memilih send	
Included Case	-	
Main Flow	Step	Action
	1	Pengguna memilih menu lupa password
	2	Sistem menampilkan form lupa password
	3	Pengguna menginputkan email yang terdaftar
	4	Sistem mengirim email recovery
Extension	Step	Branching Action
	3.1	Sistem gagal mengirim email recovery
	3.2	Sistem berhasil mengirim email recovery

5. Activity diagram menjelaskan aliran kerja aktor pada perangkat lunak yang akan dibangun.
   a. Activity Diagram Register User

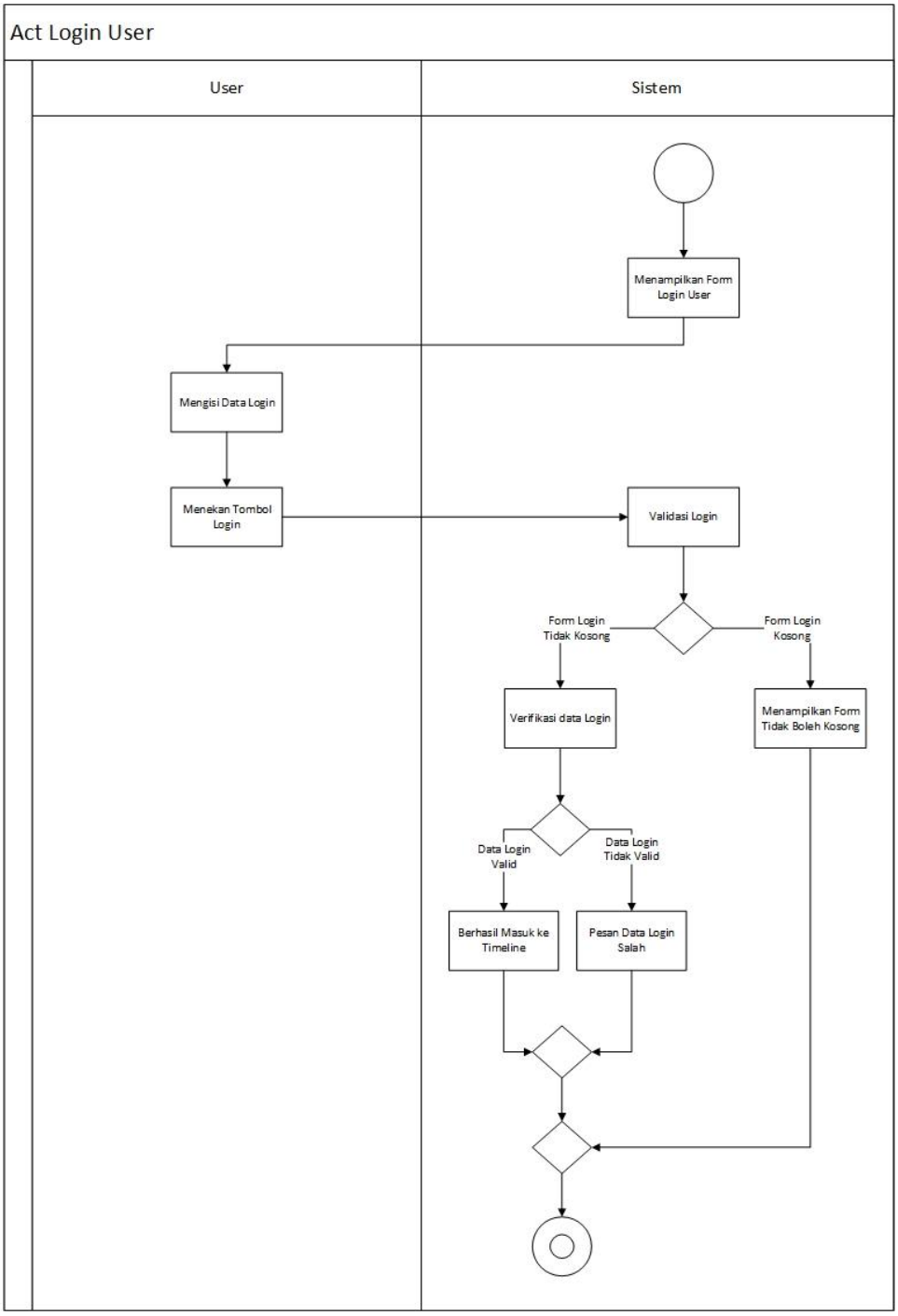
      Berikut adalah Gambar 3.24 Activity Diagram Register User yang menggambarkan activity diagram Register User.



Gambar 3.24 Activity Diagram Register User

b. Activity Diagram Login

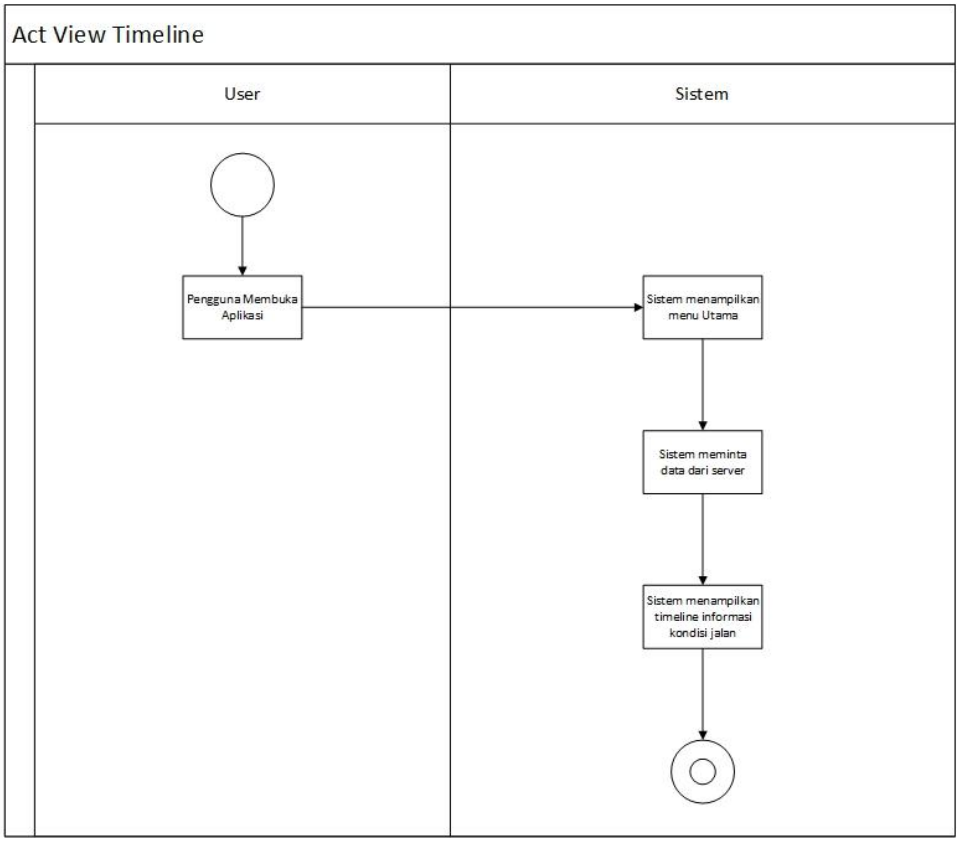
Berikut adalah Gambar 3.25 Activity Diagram Login yang menggambarkan activity diagram login.



Gambar 3.25 Activity Diagram Login

c. Activity Diagram View Timeline

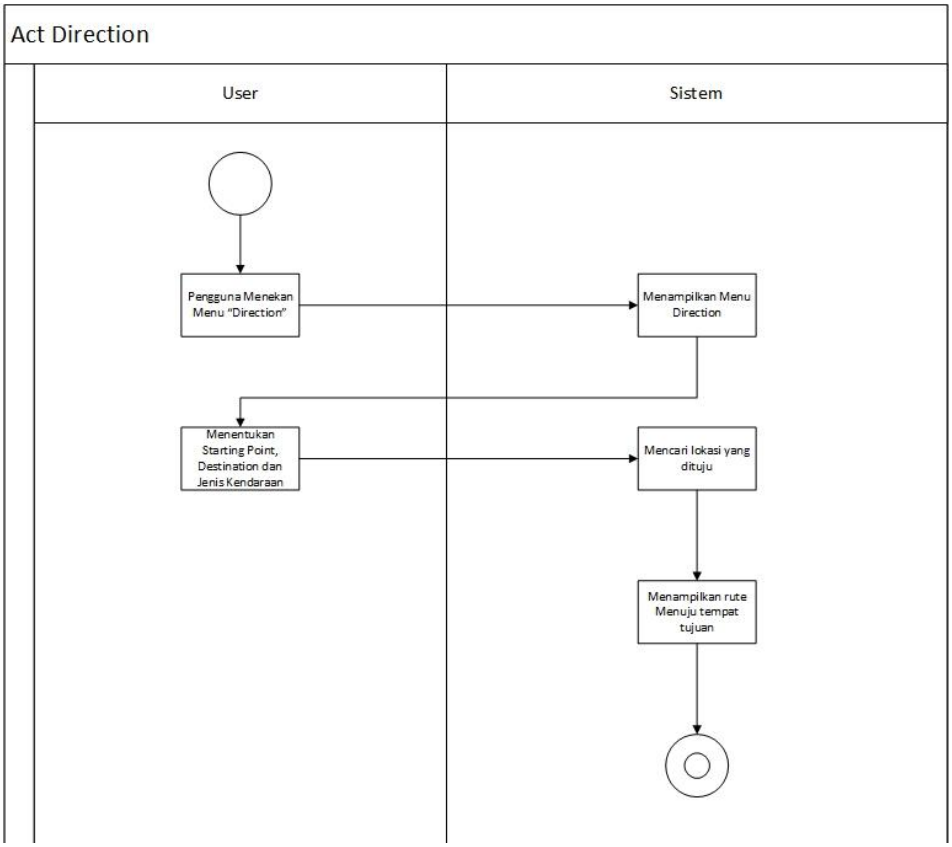
Berikut adalah Gambar 3.26 Activity Diagram View Timeline yang menggambarkan activity diagram view Timeline.



Gambar 3.26 Activity Diagram View Timeline

d. Activity Diagram Create Direction

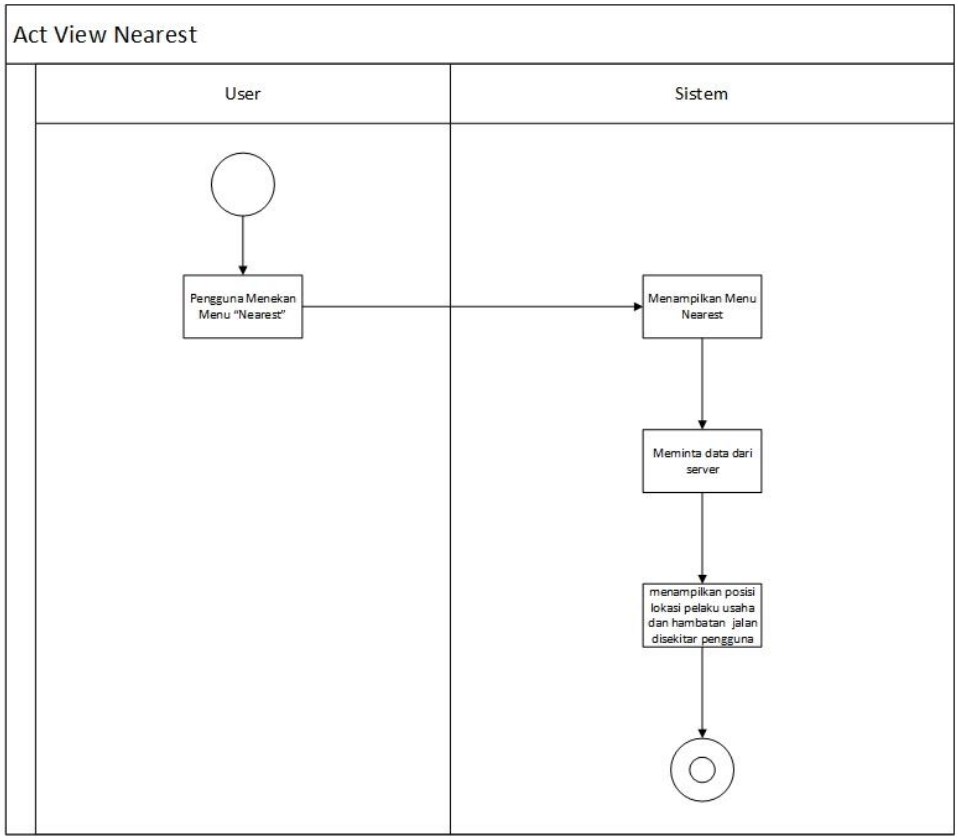
Berikut adalah Gambar 3.27 Activity Diagram Create Direction yang menggambarkan activity diagram create direction.



Gambar 3.27 Activity Diagram Create Direction

e. Activity Diagram View Nearest

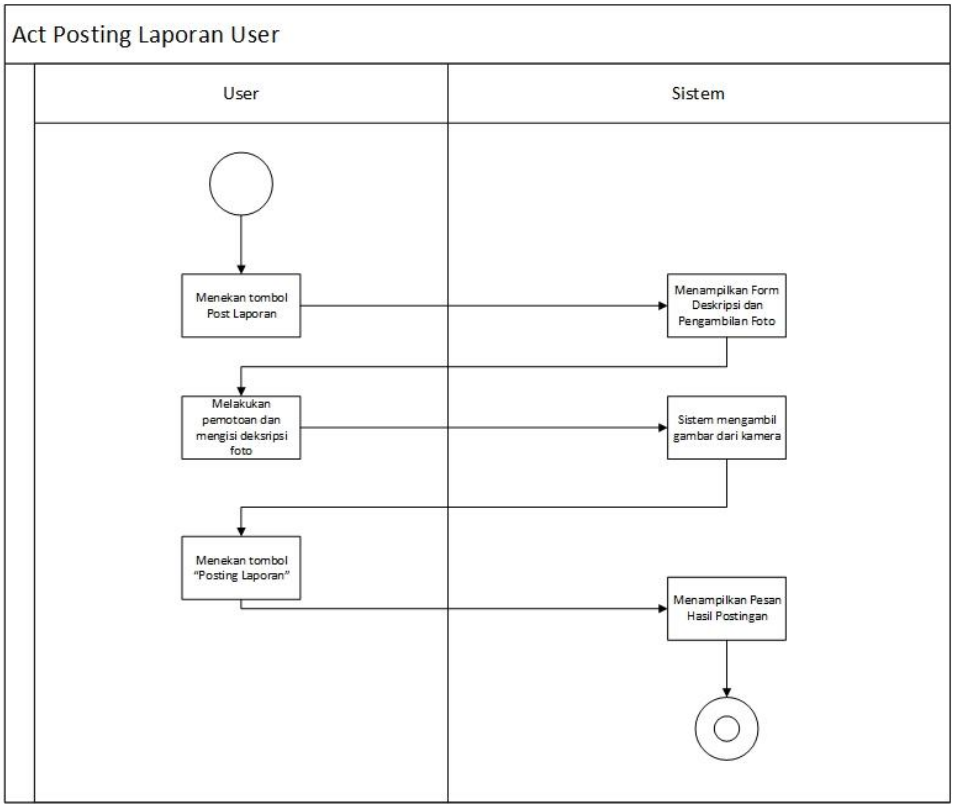
Berikut adalah Gambar 3.28 Activity Diagram View Nearest yang menggambarkan activity diagram view nearest.



Gambar 3.28 Activity Diagram View Nearest

f. Activity Diagram Posting Laporan

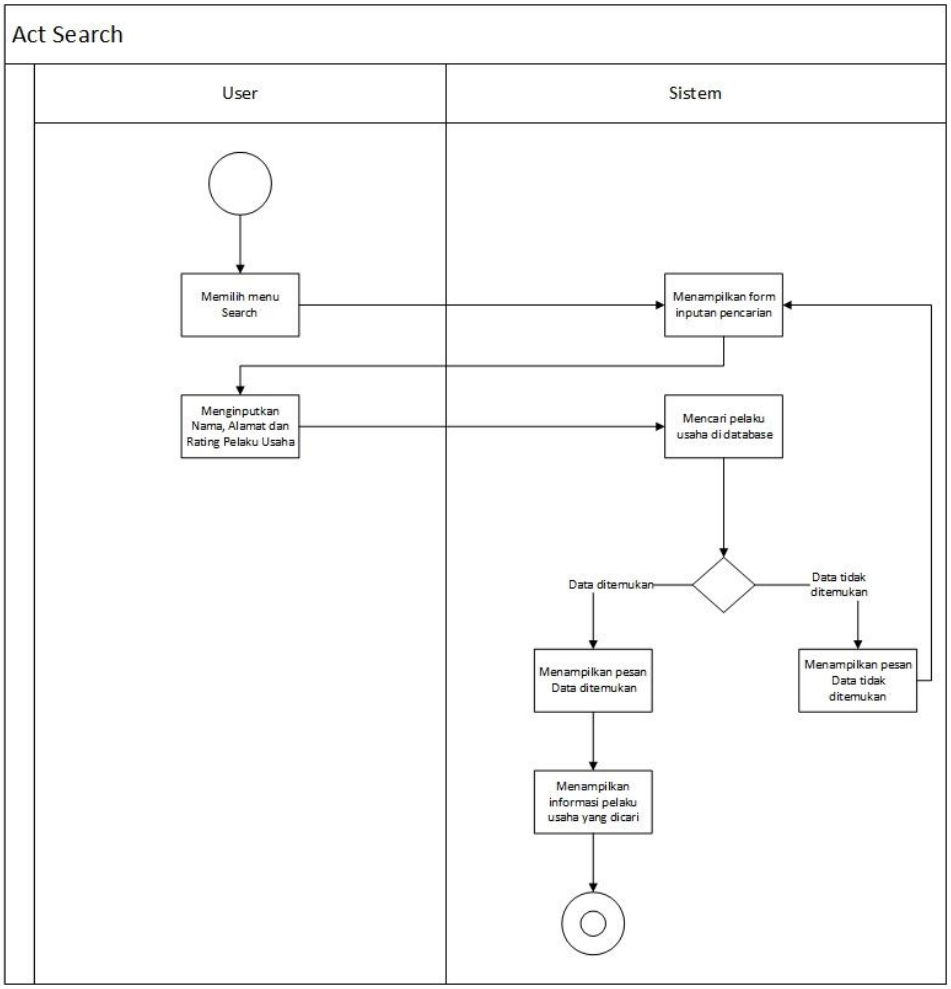
Berikut adalah Gambar 3.29 Activity Diagram Posting Laporan yang menggambarkan activity diagram Posting Laporan.



Gambar 3.29 Activity Diagram Posting Laporan

g. Activity Diagram Search

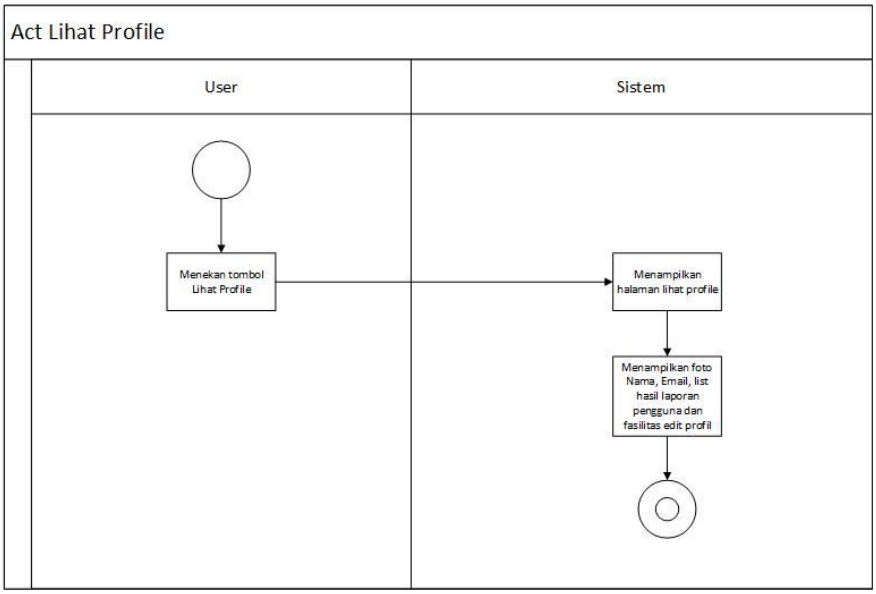
Berikut adalah Gambar 3.30 Activity Diagram Posting Laporan yang menggambarkan activity diagram search.



Gambar 3.30 Activity Diagram Search

h. Activity Diagram Lihat Profile

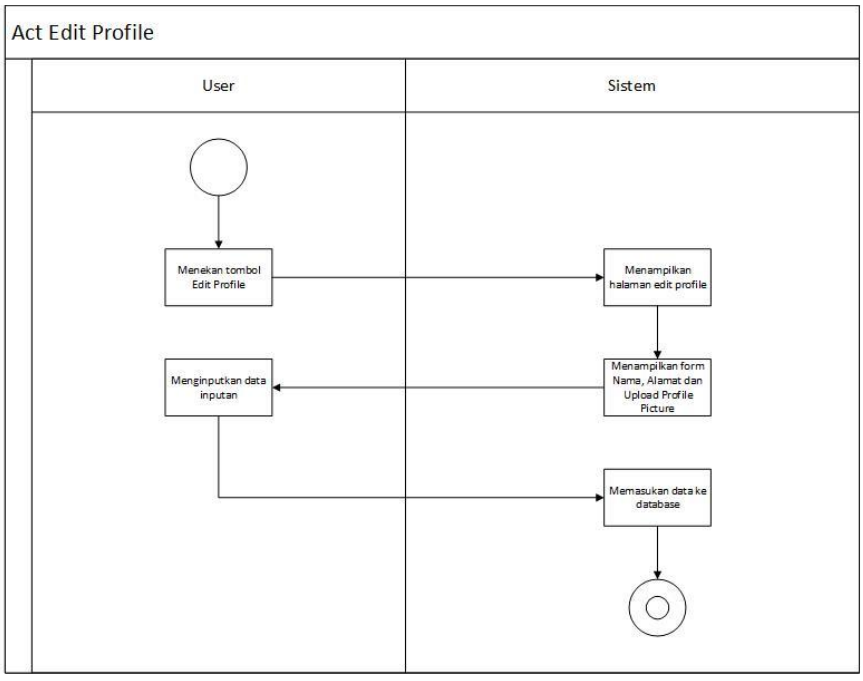
Berikut adalah Gambar 3.31 Activity Diagram Lihat Profile yang menggambarkan activity diagram lihat profile.



Gambar 3.31 Activity Diagram Lihat Profile

i. Activity Diagram Edit Profile

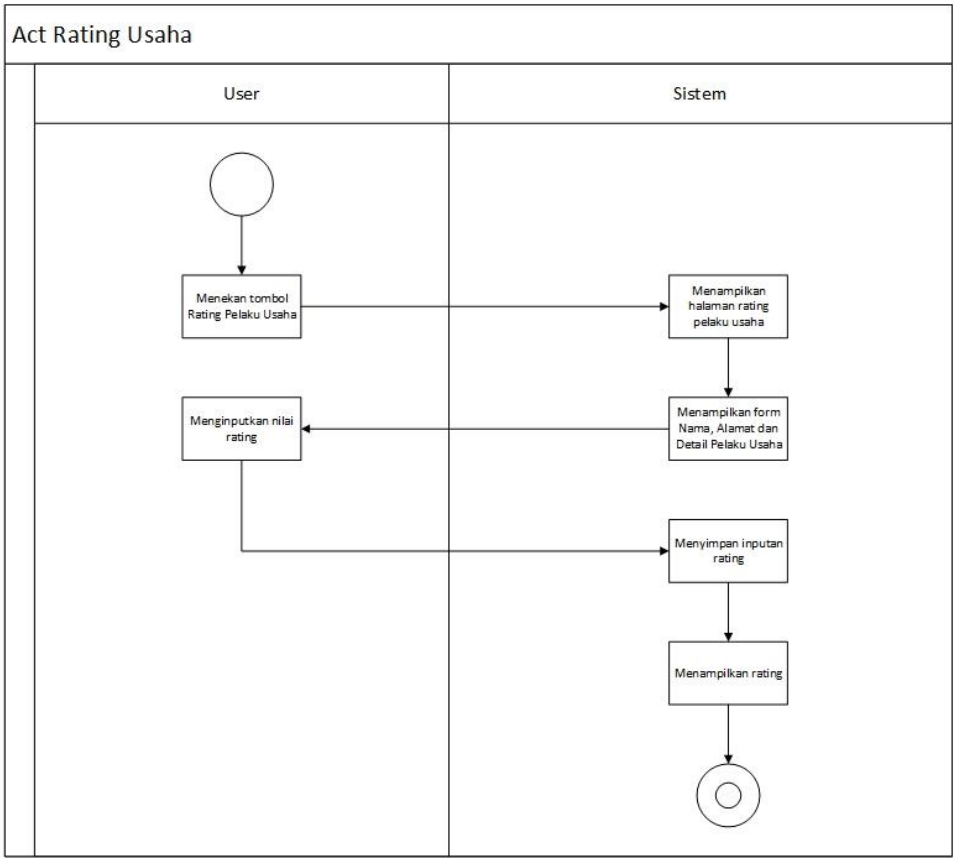
Berikut adalah Gambar 3.32 Activity Diagram Edit Profile yang menggambarkan activity diagram edit profile.



Gambar 3.32 Activity Diagram Edit Profile

j. Activity Diagram Rating Usaha

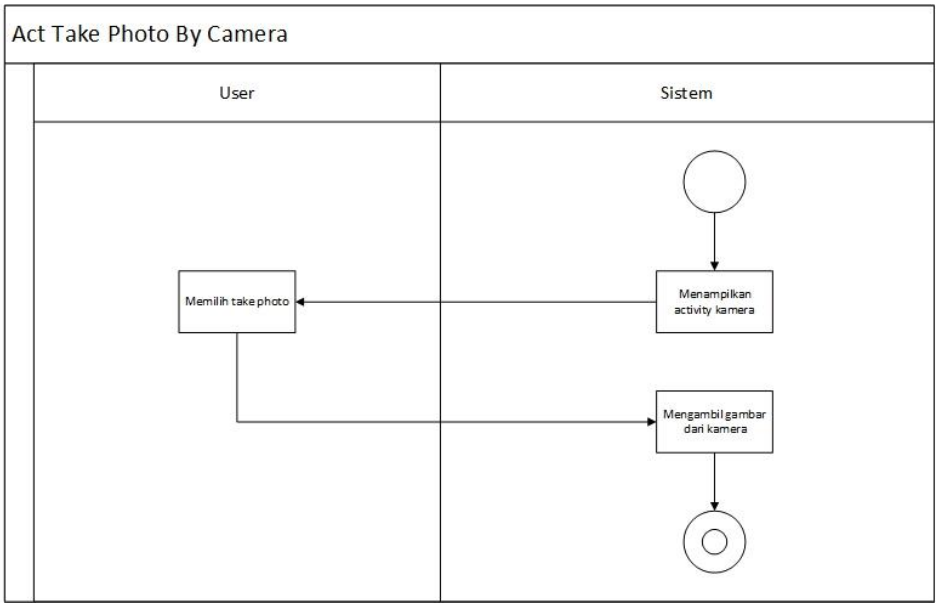
Berikut adalah Gambar 3.33 Activity Diagram Rating Usaha yang menggambarkan activity diagram rating usaha.



Gambar 3.33 Activity Diagram Rating Usaha

k. Activity Diagram Take Photo By Camera

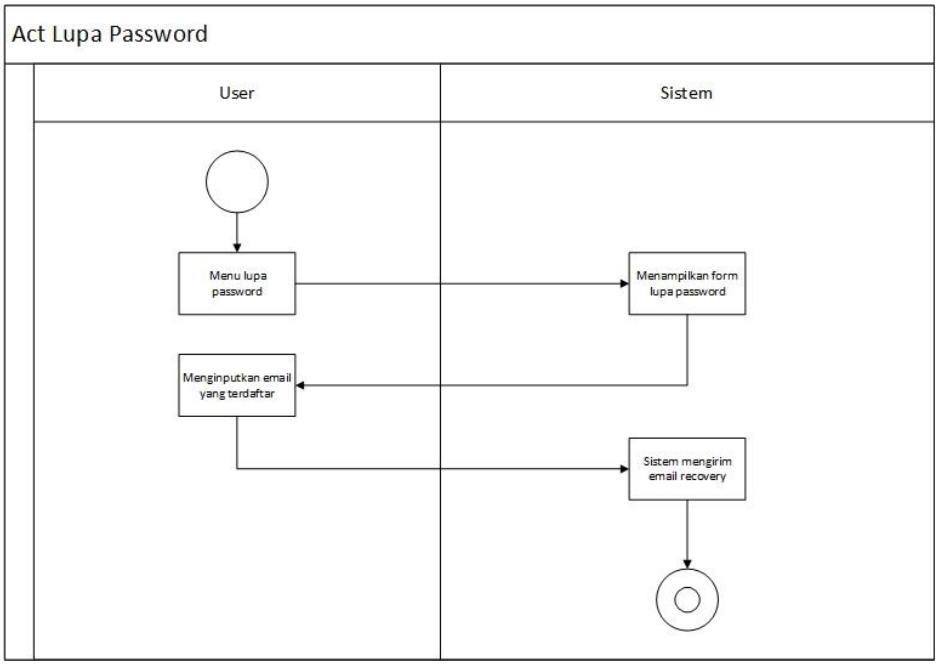
Berikut adalah Gambar 3.34 Activity Diagram Take Photo by Camera yang menggambarkan activity diagram take photo by camera.



Gambar 3.34 Activity Diagram Take Photo by Camera

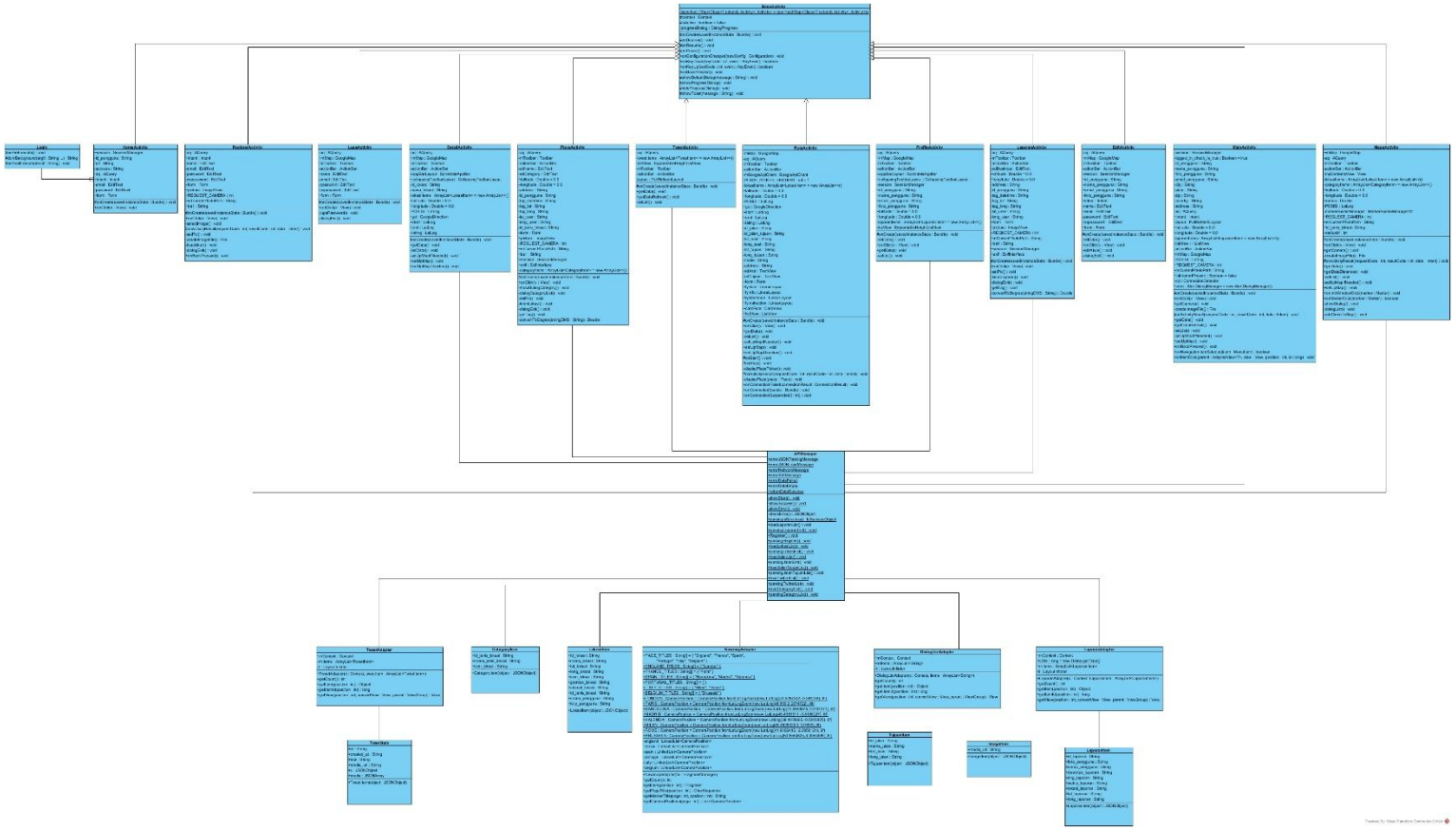
l. Activity Diagram Lupa Password

Berikut adalah Gambar 3.35Activity Diagram Lupa Password yang menggambarkan activity diagram lupa password.



Gambar 3.35 Activity Diagram Lupa Password

6. Diagram Kelas



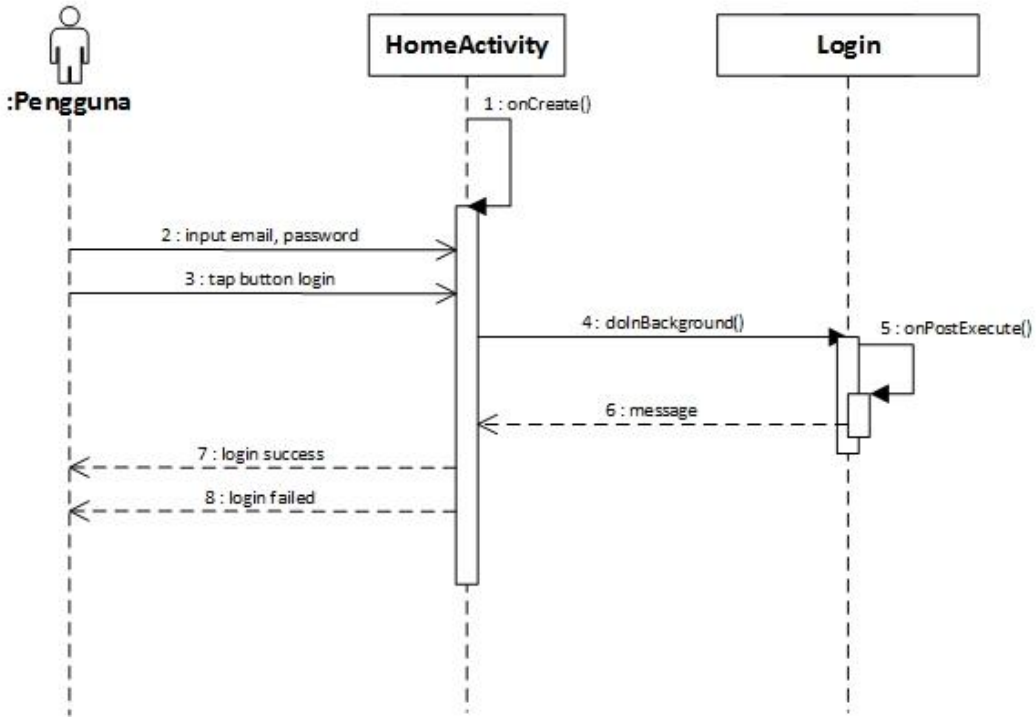
Gambar 3.36 Diagram Kelas

7. Diagram Sequence

Sequence diagram digunakan untuk menggambarkan arus pekerjaan, pesan yang di sampaikan dan bagaimana elemen-elemen di dalamnya bekerja sama dari waktu ke waktu untuk mencapai suatu hasil.

a. Sequence Diagram Login User

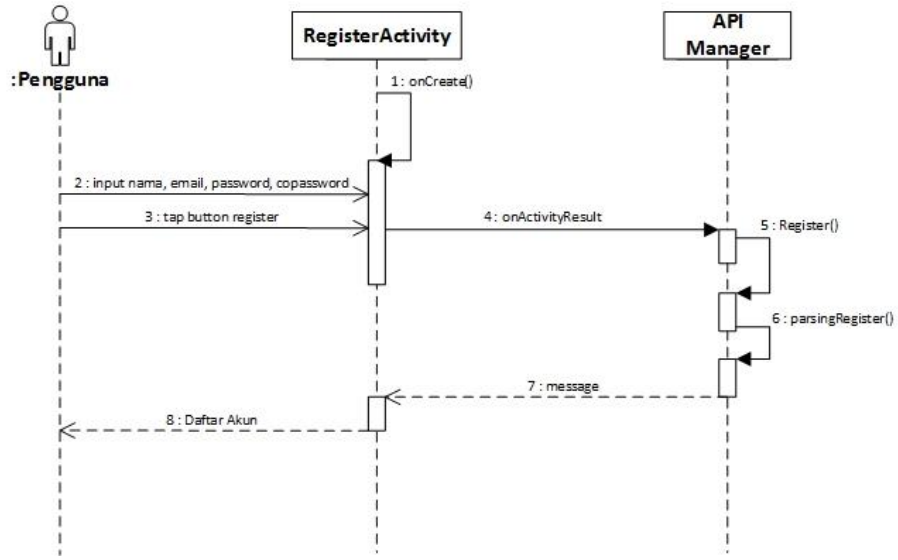
Berikut adalah Gambar 3.37 Sequence Diagram Login User yang menggambarkan sequence diagram login user.



Gambar 3.37 Sequence Diagram Login User

b. Sequence Diagram Register User

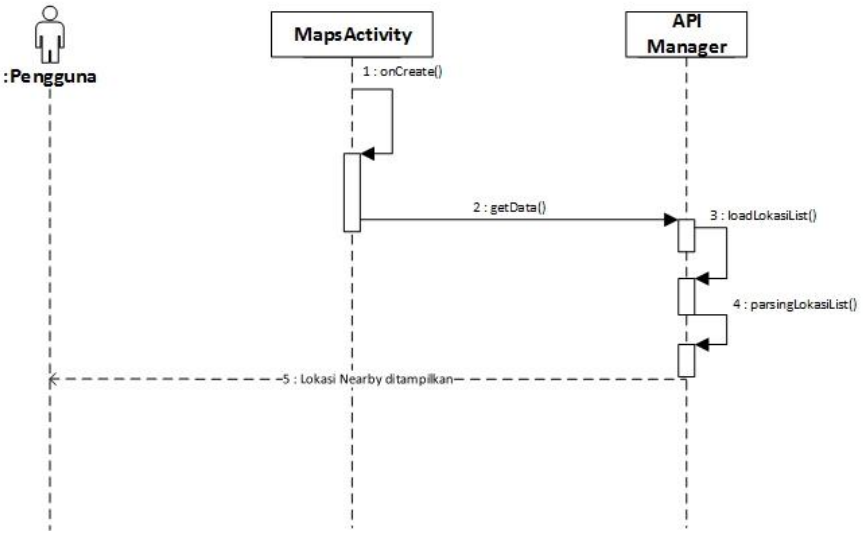
Berikut adalah Gambar 3.38 Sequence Diagram Register User yang menggambarkan sequence diagram register user.



Gambar 3.38 Sequence Diagram Register User

c. Sequence Diagram View Nearest

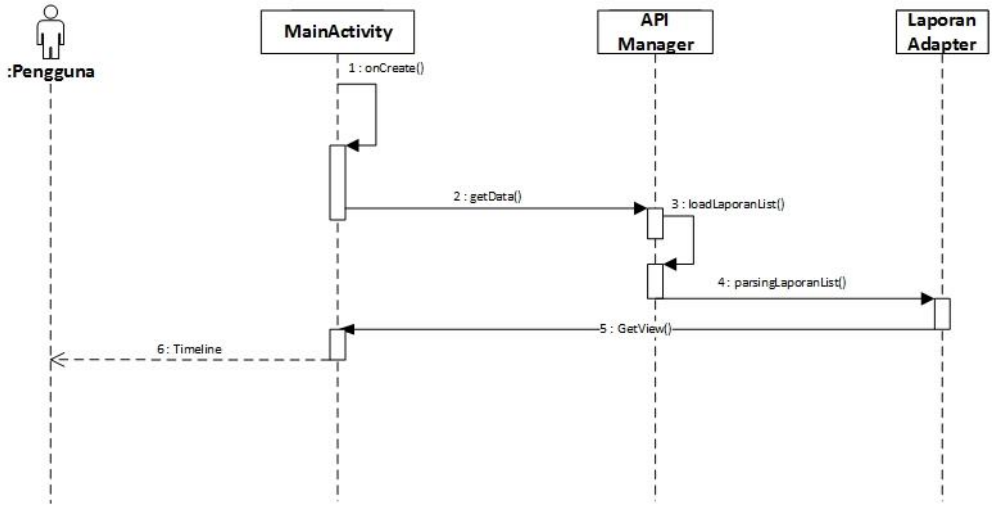
Berikut adalah Gambar 3.39 Sequence Diagram View Nearest yang menggambarkan sequence diagram view nearest.



Gambar 3.39 Sequence Diagram View Nearest

d. Sequence Diagram View Timeline

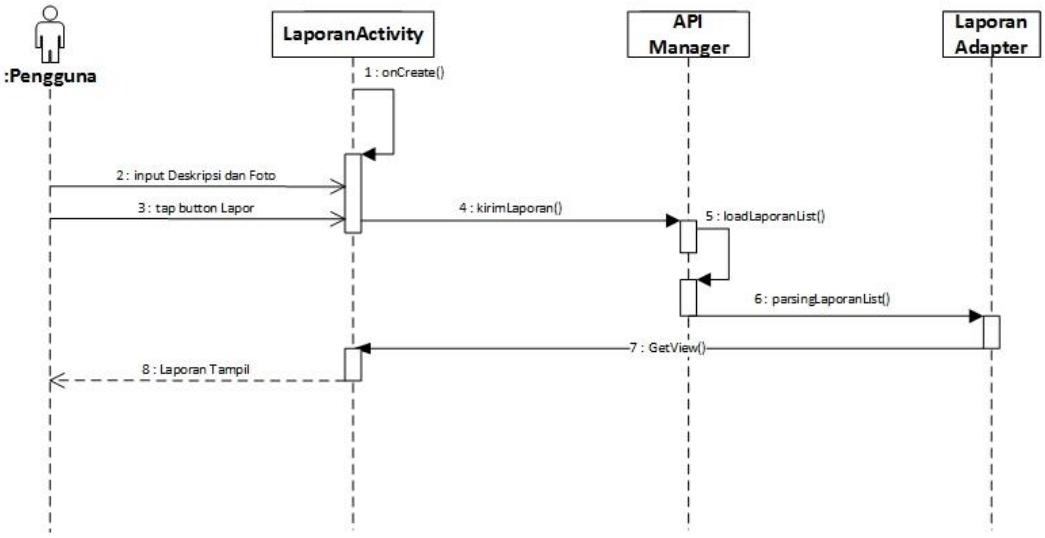
Berikut adalah Gambar 3.40 Sequence Diagram View Timeline yang menggambarkan sequence diagram view timeline.



Gambar 3.40 Sequence Diagram View Timeline

e. Sequence Diagram Posting Report

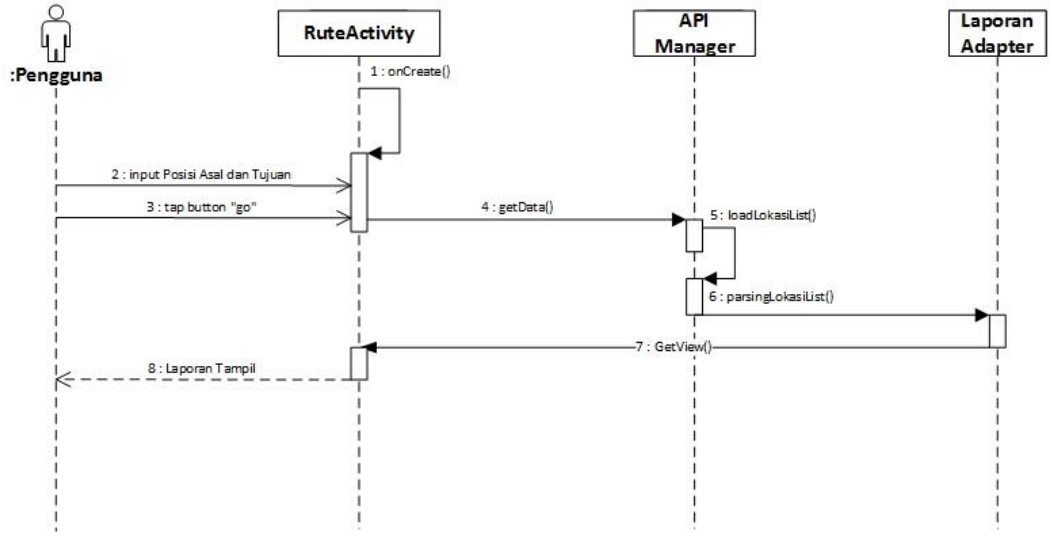
Berikut adalah Gambar 3.41 Sequence Diagram Posting Report yang menggambarkan sequence diagram posting report.



Gambar 3.41 Sequence Diagram Posting Report

f. Sequence Diagram Create Direction

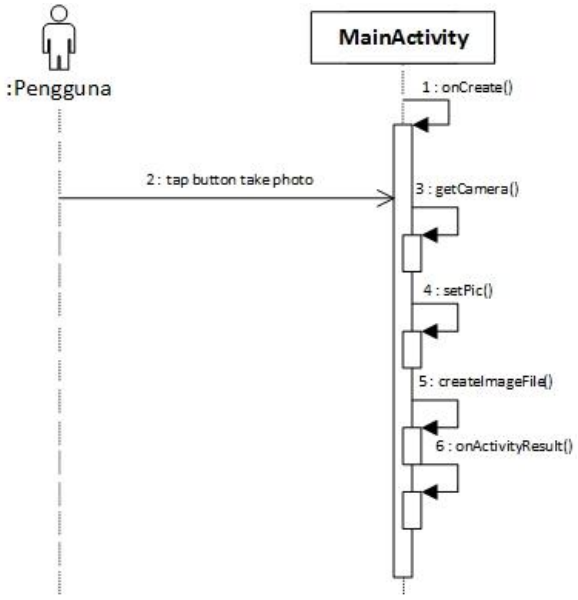
Berikut adalah Gambar 3.42 Sequence Diagram Create Direction yang menggambarkan sequence diagram create direction.



Gambar 3.42 Sequence Diagram Create Direction

g. Sequence Diagram Take Photo By Camera

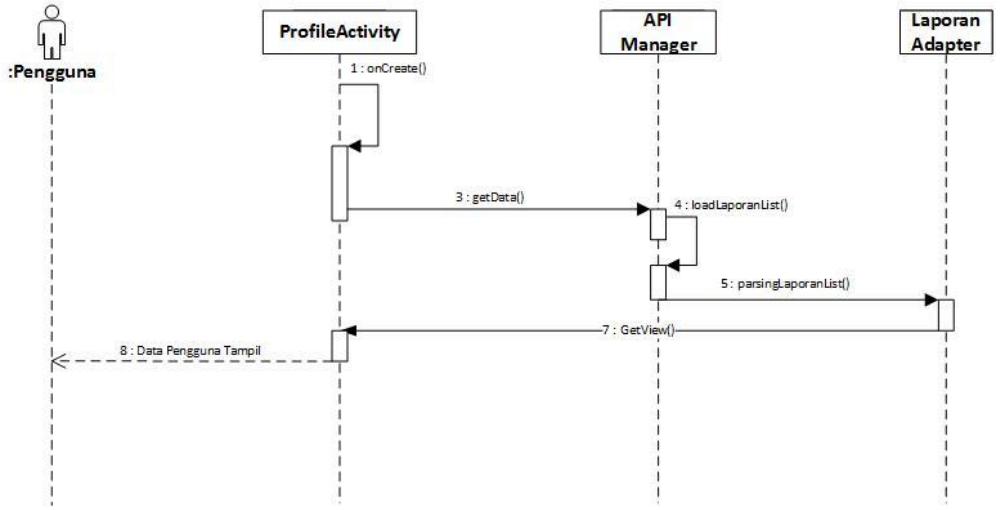
Berikut adalah Gambar 3.43 Sequence Diagram Take Photo By Camera yang menggambarkan sequence diagram take photo by camera.



Gambar 3.43 Sequence Diagram Take Photo By Camera

h. Sequence Diagram Lihat Profi Pengguna

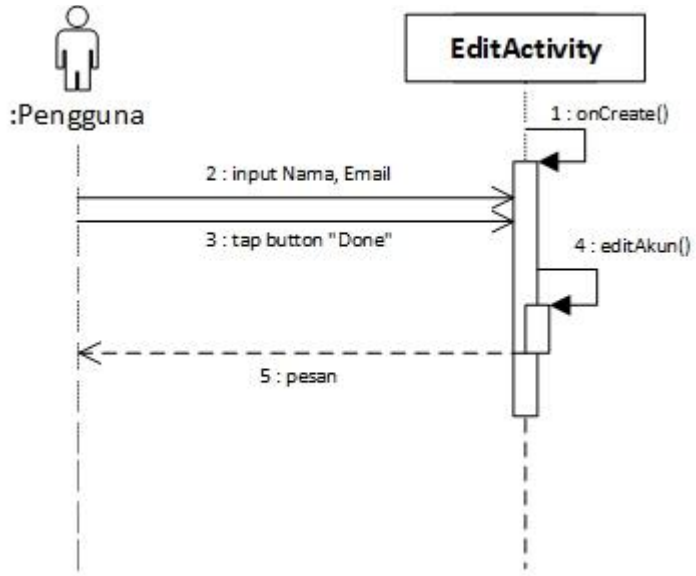
Berikut adalah Gambar 3.44 Sequence Diagram Lihat Profil Pengguna yang menggambarkan sequence diagram lihat profil pengguna.



Gambar 3.44 Sequence Diagram Lihat Profil Pengguna

i. Sequence Diagram Edit Profil Pengguna

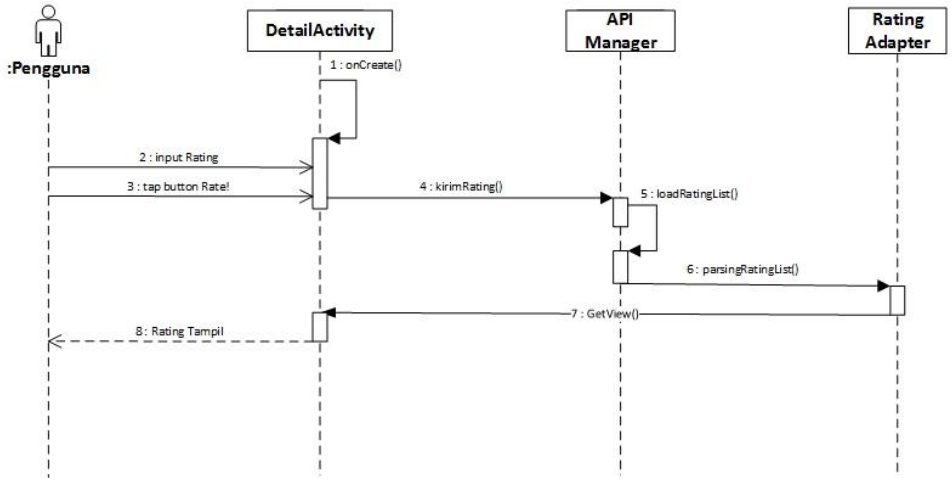
Berikut adalah Gambar 3.45 Sequence Diagram Edit Profil Pengguna yang menggambarkan sequence diagram edit profil pengguna.



Gambar 3.45 Sequence Diagram Edit Profil Pengguna

j. Sequence Diagram Give Rate

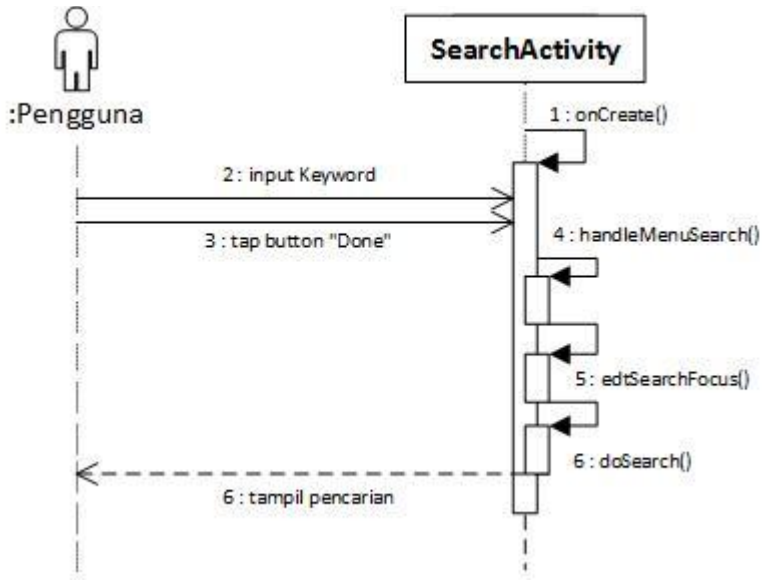
Berikut adalah Gambar 3.46 Sequence Diagram Give Rate yang menggambarkan sequence diagram give rating.



Gambar 3.46 Sequence Diagram Give Rate

k. Sequence Diagram Search

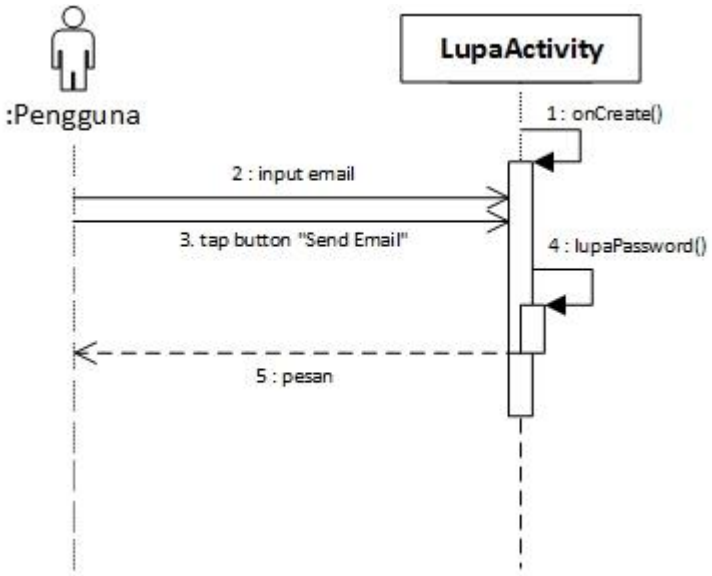
Berikut adalah Gambar 3.47 Sequence Diagram Search yang menggambarkan sequence diagram search.



Gambar 3.47 Sequence Diagram Search

l. Sequence Diagram Lupa Password

Berikut adalah Gambar 3.48 Sequence Diagram Lupa Password yang menggambarkan sequence diagram lupa password.



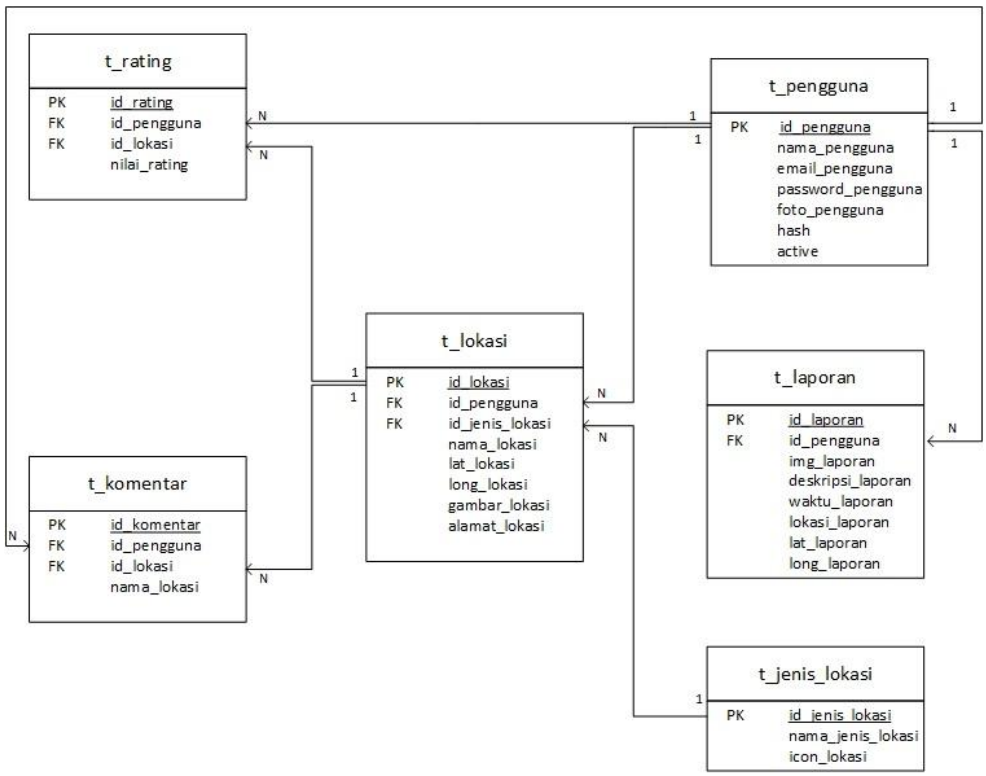
Gambar 3.48 Sequence Diagram Lupa Password

3.1.7 Perancangan Sistem

3.1.7.1 Perancangan Data

1. Diagram Relasi

   Dibawah ini merupakan Gambar 3.49 Diagram Relasi dari perangkat lunak yang dibangun.



Gambar 3.49 Diagram Relasi

2. Struktur Tabel

Berikut adalah struktur table dari perangkat lunak yang dibangun

a. Tabel t_pengguna

Berikut adalah Tabel 3.26 Struktur Tabel t_pengguna yang mendeskripsikan tabel pengguna

Nama tabel : t_pengguna

Tempat penyimpanan : harddisk

Tabel 3.26 Struktur Tabel t_pengguna

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
ID_PENGGUNA	Int	11	Primary key	not null,auto increment

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
NAMA_PENGGUNA	Varchar	100		not null
EMAIL_PENGGUNA	Varchar	100		not null
PASSWORD_PENGGUNA	Varchar	100		not null
FOTO_PENGGUNA	Varchar	100		not null
HASH	Varchar	32		not null
ACTIVE	Int	1		not null

b. Tabel t_lokasi

Berikut adalah Tabel 3.27 Struktur Tabel t_lokasi yang mendeskripsikan tabel lokasi.
Nama tabel : t_lokasi
Tempat penyimpanan : harddisk

Tabel 3.27 Struktur Tabel t_lokasi

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
ID_LOKASI	Int	11	Primary key	not null, auto increment
NAMA_LOKASI	Varchar	100		not null
LAT_LOKASI	Double			not null
LONG_LOKASI	Double			not null
GAMBAR_LOKASI	Varchar	100		not null
ALAMAT_LOKASI	Varchar	100		Not null
ID_JENIS_LOKASI	Int	11		not null
ID_PENGGUNA	Int	11		Not null

c. Tabel t_jenis_lokasi

Berikut adalah Tabel 3.28 Struktur Tabel t_jenis_lokasi yang mendeskripsikan tabel jenis lokasi.
Nama tabel : t_jenis_lokasi
Tempat penyimpanan : harddisk

Tabel 3.28 Struktur Tabel t_jenis_lokasi

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
ID_JENIS_LOKASI	Int	11	Primary key	not null, auto increment
NAMA_JENIS_LOKASI	Varchar	50		not null
ICON_LOKASI	Varchar	100		Not null

d. Tabel t_laporan

Berikut adalah Tabel 3.28 Struktur Tabel t_laporan yang mendeskripsikan tabel laporan.
Nama tabel : t_laporan
Tempat penyimpanan : harddisk

Tabel 3.28 Struktur Tabel t_laporan

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
ID_LAPORAN	Int	11	Primary key	not null, auto increment
ID_PENGGUNA	Int	11		not null
IMG_LAPORAN	Varchar	100		not null
DESKRIPSI_LAPORAN	Varchar	200		not null
WAKTU_LAPORAN	Varchar	50		not null
LOKASI_LAPORAN	Varchar	200		not null
LAT_LAPORAN	Double			not null
LONG_LAPORAN	Double			not null

e. Tabel t_komentar

Berikut adalah Tabel 3.29 Struktur Tabel t_komentar yang mendeskripsikan tabel komentar.
Nama tabel : t_komentar
Tempat penyimpanan : harddisk

Tabel 3.29 Struktur Tabel t_laporan

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
ID_KOMENTAR	Int	11	Primary key	not null, auto increment
ID_PENGGUNA	Int	11		not null
ID_LOKASI	Int	11		not null
ISI_KOMENTAR	Varchar	200		not null

f. Tabel t_rating

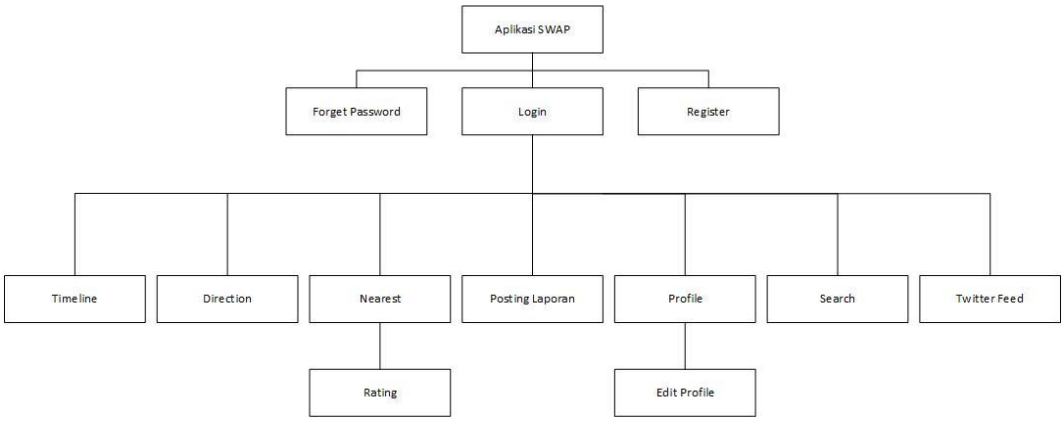
Berikut adalah Tabel 3.30 Struktur Tabel t_rating yang mendeskripsikan tabel komentar.
Nama tabel : t_rating
Tempat penyimpanan : harddisk

Tabel 3.30 Struktur Tabel t_laporan

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
ID_RATING	Int	11	Primary key	not null, auto increment
ID_PENGGUNA	Int	11		not null
ID_LOKASI	Int	11		not null
NILAI_RATING	Int			not null

3.1.7.2 Perancangan Arsitektur Menu

Perancangan arsitektur menyediakan software engineer satu gambaran dari stuktur menu program atau blueprint dari perangkat lunak yang akan dibuat. Tujuan perancangan ini adalah untuk membangun struktur program secara moduler dan menggambarkan hubungan kendali diantara modul program. Berikut adalah gambaran perancangan arsitektural menu. Dibawah ini merupakan Struktur Menu Aplikasi SWAP (Safety Way Guide App Purwakarta).



Gambar 3.50 Perancangan Arsitektur Menu

3.1.7.3 Perancangan Antarmuka

1. Perancangan Antarmuka Login User

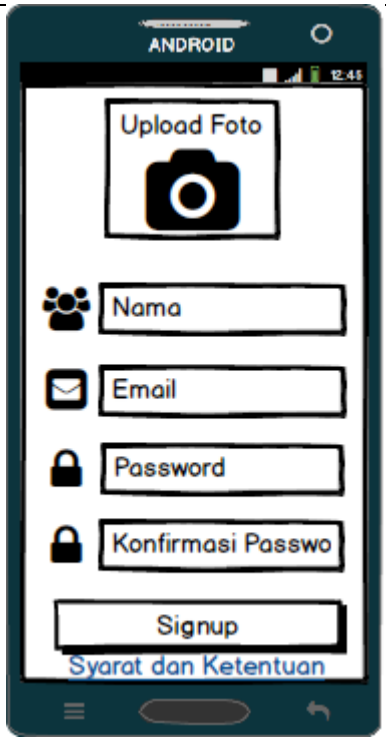
   Berikut adalah Gambar 3.51 Perancangan Antarmuka Login User yang mengambarkan rancangan antarmuka login user.

<table>
<tr><td colspan="2">M01 – Login User</td></tr>
<tr><td>
</td><td>1. Input Email dan Password untuk melakukan login.
2. Tap tombol SignIn untuk masuk ke akun inputan dan akan muncul tampilan M05.
3. Bila Email dan Password kosong maka akan menampilkan pesan P01.
4. Bila Email dan Password salah maka akan menampilkan pesan P02.
5. Tap tombol “Lupa Password” bila lupa password maka akan tampil M03.
6. Tap tombol “Buat Akun Baru?” untuk mendaftar sebagai user baru, dan akan tampil menu M02.</td></tr>
<tr><td colspan="2">Ukuran Layar Menyesuaikan
Ukuran Font dan Warna disesuaikan
Jenis Font Roboto</td></tr>
</table>

Gambar 3.51 Perancangan Antarmuka Login User

2. Perancangan Antarmuka Register User

Berikut adalah Gambar 3.52 Perancangan Antarmuka Register User yang mengambarkan rancangan antarmuka register user.

<table>
<tr><td colspan="2">M02 – Register User</td></tr>
<tr><td>
</td><td>1. Input “Upload Foto” untuk mengisi foto profil user
2. Input Nama, Email dan Password untuk melakukan registrasi
3. Bila Nama, Email dan Password kosong maka akan menampilkan pesan P03.
4. Tap tombol “Signup” untuk men-submit inputan Nama, Email, Password, dan akan muncul pesan P04 lalu kembali ke tampilan M01.</td></tr>
<tr><td colspan="2">Ukuran Layar Menyesuaikan
Ukuran Font dan Warna disesuaikan
Jenis Font Roboto</td></tr>
</table>

Gambar 3.52 Perancangan Antarmuka Register User

3. Perancangan Antarmuka Lupa Password

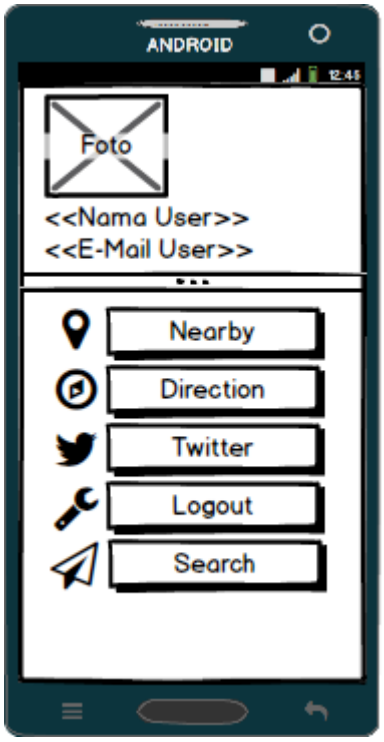
Berikut adalah Gambar 3.53 Perancangan Antarmuka Lupa Password yang mengambarkan rancangan antarmuka Lupa Password.

<table>
<tr><td colspan="2">M03 – Lupa Password</td></tr>
<tr><td>
</td><td>1. Input “Email Address” yang sudah terdaftar.
2. User menekan tombol “Send” lalu akan tampil pesan P06 bila form email kosong (tidak berisi) dan akan muncul pesan P07 apabila reset password telah terkirim ke email inputan.</td></tr>
<tr><td colspan="2">Ukuran Layar Menyesuaikan
Ukuran Font dan Warna disesuaikan
Jenis Font Roboto</td></tr>
</table>

Gambar 3.53 Perancangan Antarmuka Lupa Password

4. Perancangan Antarmuka Menu

Berikut adalah Gambar 3.54 Perancangan Antarmuka Menu yang mengambarkan rancangan antarmuka Menu.

<table>
<tr><td colspan="2">M04 – View Menu</td></tr>
<tr><td>
</td><td>1. Tap “Foto Profile” untuk masuk ke menu M011.
2. Tap Button “Nearby” untuk masuk ke menu M08.
3. Tap Button “Direction” untuk masuk ke menu M07.
4. Tap button “Twitter” untuk masuk ke menu M013.
5. Tap button “Logout” untuk melakukan proses logout.
6. Tap button “Search” untuk masuk ke menu M09.</td></tr>
<tr><td colspan="2">Ukuran Layar Menyesuaikan
Ukuran Font dan Warna disesuaikan
Jenis Font Roboto</td></tr>
</table>

Gambar 3.54 Perancangan Antarmuka View Menu

5. Perancangan Antarmuka View Timeline

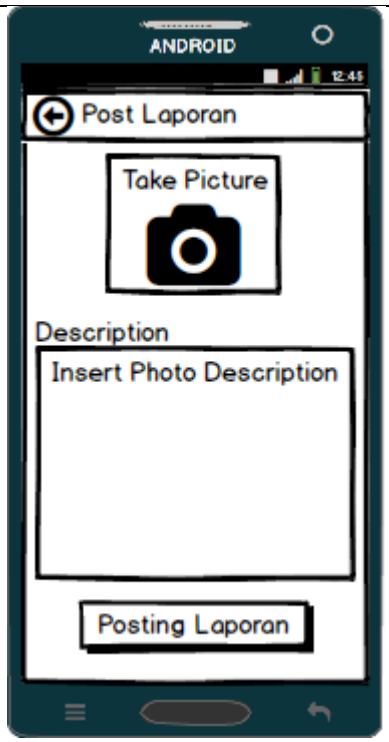
Berikut adalah Gambar 3.55 Perancangan Antarmuka View Timeline yang mengambarkan rancangan antarmuka view timeline.

<table>
<tr><td colspan="2">M05 – View Timeline</td></tr>
<tr><td>
</td><td>1. Tombol “☰” untuk menuju pada tampilan M04.
2. Menampilkan foto profil, nama user dan posisi yang didapaat dari GPS Geotagging.
3. Menampilkan informasi posting pelaporan yang di share dari user lain dan twitter.
4. Bila akan melakukan Posting Pelaporan makan Tap tombol “Post” dan akan tampil menu M05.</td></tr>
<tr><td colspan="2">Ukuran Layar Menyesuaikan
Ukuran Font dan Warna disesuaikan
Jenis Font Roboto</td></tr>
</table>

Gambar 3.55 Perancangan Antarmuka View Timeline

6. Perancangan Antarmuka Posting Report

Berikut adalah Gambar 3.56 Perancangan Antarmuka Posting Report yang mengambarkan rancangan antarmuka view timeline.

<table>
<tr><td colspan="2">M06 – Posting Report</td></tr>
<tr><td>
</td><td>1. Tap logo “←” untuk kembali ke M05.
2. Tap “Logo Kamera” untuk melakukan pemotretan lokasi yang akan dilaporkan.
3. User mengisi deskripsi kondisi lokasi yang difoto untuk lebih mendeskripsikan.
4. Tap “Post Report” untuk men-share inputan. Dan bila berhasil akan tampil pesan P05, dan kembali ke tampilan M05.</td></tr>
<tr><td colspan="2">Ukuran Layar Menyesuaikan
Ukuran Font dan Warna disesuaikan
Jenis Font Roboto</td></tr>
</table>

Gambar 3.56 Perancangan Antarmuka Posting Report

7. Perancangan Antarmuka Create Direction

Berikut adalah Gambar 3.57 Perancangan Antarmuka Create Direction yang mengambarkan rancangan antarmuka create direction.

M07 – Create Direction



1. Tap logo "←" untuk kembali ke M05.
2. Tap pilihan "FROM" untuk memilih jalan posisi utama user.
3. Tap pilihan "TO" untuk memilih jalan tujuan user.
4. Tap Button "Go!" untuk mendapatkan direction ke tempat tujuan.

Ukuran Layar Menyesuaikan
Ukuran Font dan Warna disesuaikan
Jenis Font Roboto

Gambar 3.57 Perancangan Antarmuka Create Direction

8. Perancangan Antarmuka View Nearest

Berikut adalah Gambar 3.58 Perancangan Antarmuka View Nearest yang mengambarkan rancangan antarmuka create direction.

<table>
<tr><td colspan="2">M08 – View Nearest</td></tr>
<tr><td>
</td><td>1. Tap logo “←” untuk kembali ke M05.
2. “Filter By Category” akan menampilkan tempat-tempat berdasarkan kategori.
3. Akan menampilkan tempat-tempat berdasarkan jarak dari user.
4. Tap pointer maka akan muncul tampilan M010.</td></tr>
<tr><td colspan="2">Ukuran Layar Menyesuaikan
Ukuran Font dan Warna disesuaikan
Jenis Font Roboto</td></tr>
</table>

Gambar 3.58 Perancangan Antarmuka View Nearest

9. Perancangan Antarmuka Search

Berikut adalah Gambar 3.59 Perancangan Antarmuka Search yang mengambarkan rancangan antarmuka search.

<table>
<tr><td colspan="2">M09 – Search</td></tr>
<tr><td>
</td><td>1. Tap logo “←” untuk kembali ke M05.
2. Input “Nama, Alamat” pelaku usaha untuk menjadi keyword pencarian.
3. Tap “→” untuk melakukan pencarian, akan muncul pesan P08 bila keyword pencarian belum terisi dan pesan P11 bila lokasi tidak ditemukan..</td></tr>
<tr><td colspan="2">Ukuran Layar Menyesuaikan
Ukuran Font dan Warna disesuaikan
Jenis Font Roboto</td></tr>
</table>

Gambar 3.59 Perancangan Antarmuka Search

10. Perancangan Antarmuka Give Rate

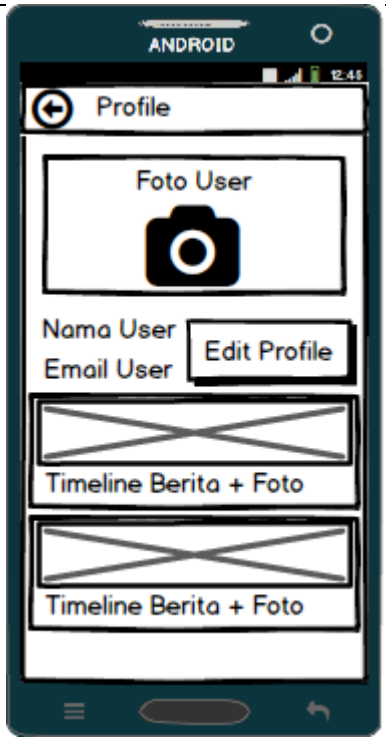
Berikut adalah Gambar 3.60 Perancangan Antarmuka Give Rate yang mengambarkan rancangan antarmuka rating.

<table>
<tr><td colspan="2">M010 – Rating</td></tr>
<tr><td>
</td><td>1. Tap logo “←” untuk kembali ke M08.
2. Menampilkan profil usaha dan alamat tempat pelaku usaha berada.
3. “Give Rate” untuk memberikan rating kepada pelaku usaha.
4. Tap “Rate!” untuk melakukan penyimpanan rating yang telah di inputkan dan akan muncul pesan P10.</td></tr>
<tr><td colspan="2">Ukuran Layar Menyesuaikan
Ukuran Font dan Warna disesuaikan
Jenis Font Roboto</td></tr>
</table>

Gambar 3.60 Perancangan Antarmuka Give Rate

11. Perancangan Antarmuka Lihat Profile Pengguna

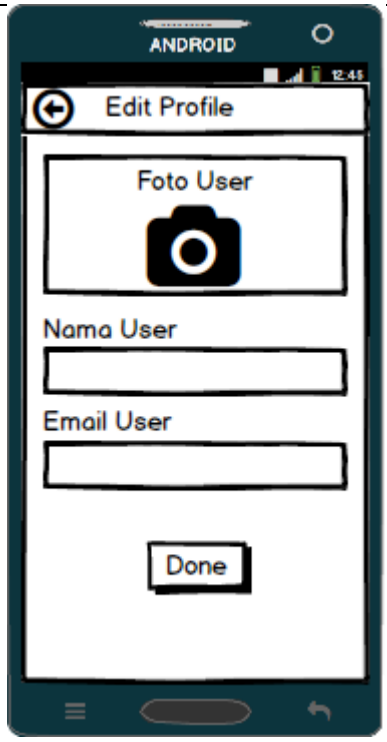
Berikut adalah Gambar 3.61 Perancangan Antarmuka Lihat Profile Pengguna yang mengambarkan rancangan antarmuka setting profile.

<table>
<tr><td colspan="2">M011 – Setting Profile</td></tr>
<tr><td>
</td><td>1. Tap logo “←” untuk kembali ke M05.
2. Tap “Edit Profile” maka akan tampil M012.
3. Pada menu ini terdapat postingan yang telah user bersangkutan laporkan.</td></tr>
<tr><td colspan="2">Ukuran Layar Menyesuaikan
Ukuran Font dan Warna disesuaikan
Jenis Font Roboto</td></tr>
</table>

Gambar 3.61 Perancangan Antarmuka Lihat Profile Pengguna

12. Perancangan Antarmuka Edit Profile Pengguna

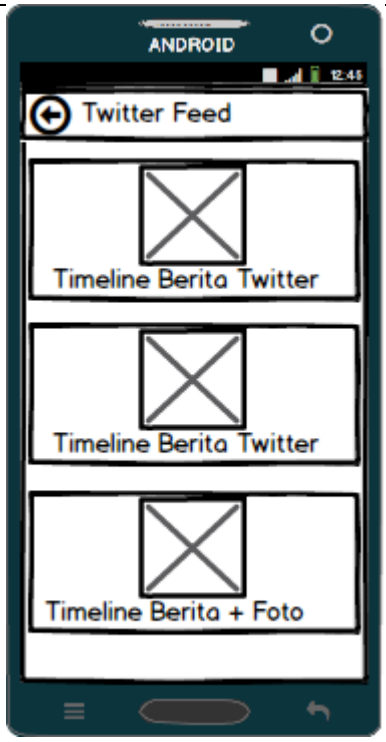
Berikut adalah Gambar 3.62 Perancangan Antarmuka Edit Profile PEngguna yang mengambarkan rancangan antarmuka setting profile.

<table>
<tr><td colspan="2">M012 – Setting Profile</td></tr>
<tr><td>
</td><td>1. Tap logo “←” untuk kembali ke M011.
2. Text field “Nama User” berguna untuk meng-edit nama user begitupun Email User.</td></tr>
<tr><td colspan="2">Ukuran Layar Menyesuaikan
Ukuran Font dan Warna disesuaikan
Jenis Font Roboto</td></tr>
</table>

Gambar 3.62 Perancangan Antarmuka Edit Profile

13. Perancangan Antarmuka Twitter Feed

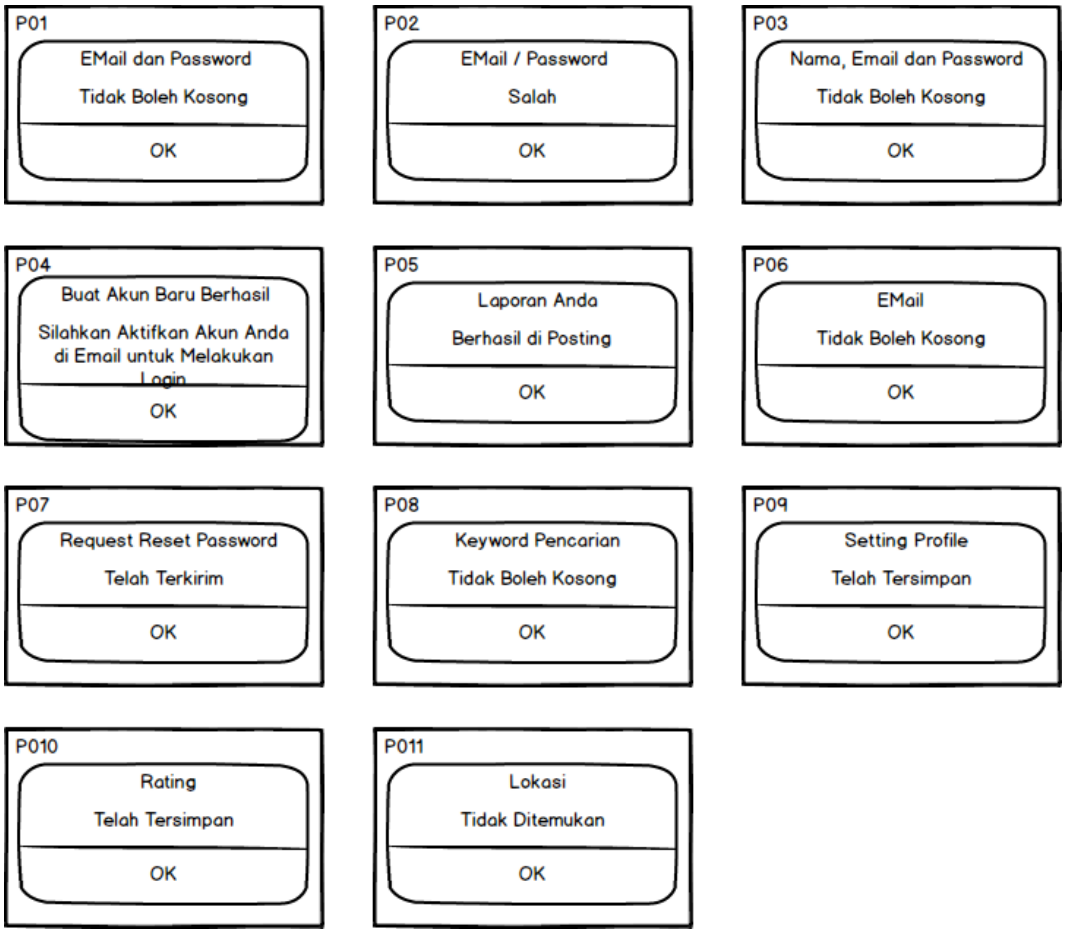
Berikut adalah Gambar 3.63 Perancangan Antarmuka Setting Profile yang mengambarkan rancangan antarmuka setting profile.

<table>
<tr><td colspan="2">M013 – Setting Profile</td></tr>
<tr><td>
</td><td>1. Tap logo “←” untuk kembali ke M011.
2. Menu ini berisi feed dari twitter. Dengan kata kunci filtering nya #Jalan #Rusak #Purwakarta.</td></tr>
<tr><td colspan="2">Ukuran Layar Menyesuaikan
Ukuran Font dan Warna disesuaikan
Jenis Font Roboto</td></tr>
</table>

Gambar 3.63 Perancangan Antarmuka Twitter Feed

6.1.7.4 Perancangan Pesan

Dibawah ini merupakan perancangan pesan pada perangkat lunak yang dibangun.

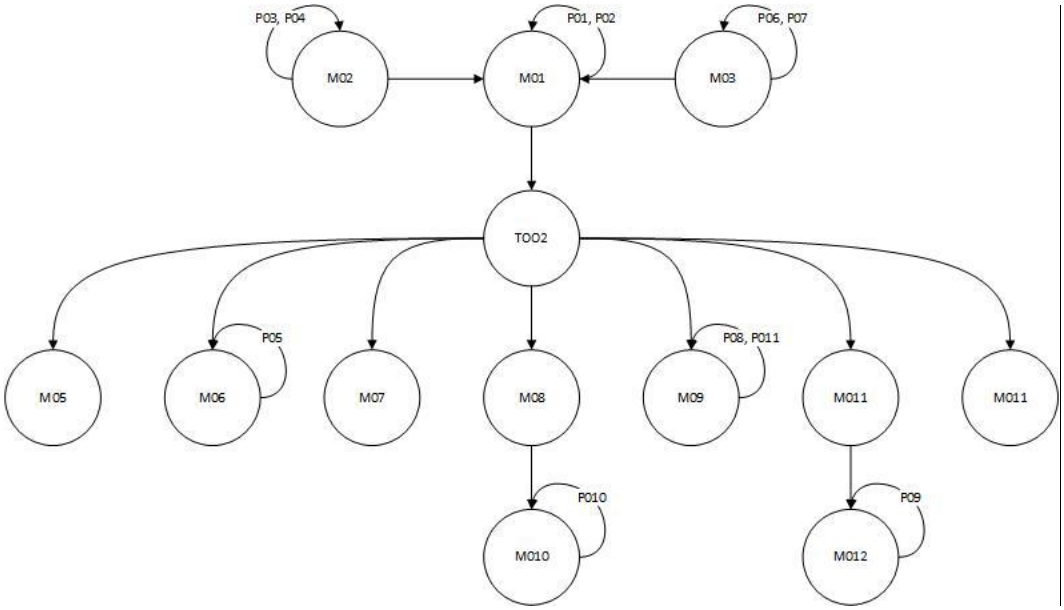


Gambar 3.64 Perancangan Pesan

6.1.7.5 Jaringan Semantik

Jaringan Semantik merupakan gambaran pengetahuan grafis yang menunjukkan hubungan antar objek (lingkaran) dan garis yang menggambarkan formasi antarobjek tersebut. Adapun jaringan semantik untuk perangkat yang dibangun sebagai berikut.

.



.

Gambar 3.65 Jaringan Semantik