

BAB 4

IMPLEMENTASI DAN PEMBAHASAN

Berdasarkan analisis dan perancangan yang telah dilakukan pada bab sebelumnya, maka dalam bab ini akan diimplementasikan “Aplikasi *Location Based Service (LBS)* Pemetaan Faskes Berbasis *Android*”.

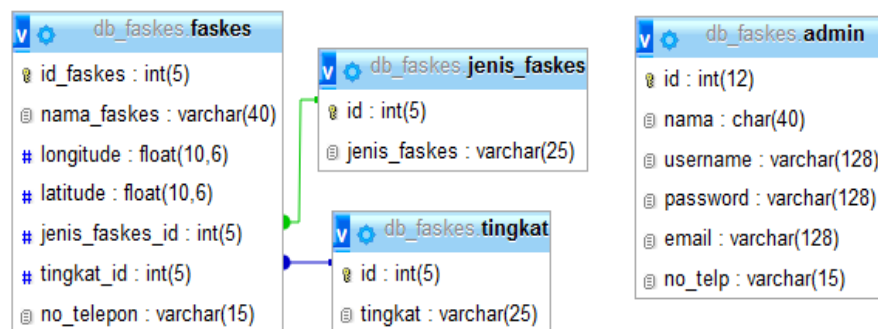
4.1. Implementasi dan Uji Coba Sistem

Implementasi adalah penerapan menjadi program aplikasi berdasarkan hasil analisis dan juga perancangan yang telah dibuat sebelumnya kedalam suatu bahasa pemrograman.

4.1.1. Implementasi Sistem *Web* untuk Admin

Dalam mengimplementasikan sistem untuk admin dilakukan dengan *Framework Yii*, yaitu dengan operasi *CRUD* (*Create, Read, Update, Delete*) yang diimplementasikan dengan metode *Active Record (AR)*.

Gambar 4.1 merupakan Desain *Database* yang akan digunakan *CRUD Generator* untuk menciptakan *model, view, dan controller*.



Gambar 4.1 Desain *Database*

4.1.2. Implementasi Sistem *Android* untuk User

1. *Parsing* data (*server*)

Untuk mengirimkan data dari *database server* ke aplikasi dengan mem-*parsing* terlebih dahulu dengan *syntax PHP* kemudian diubah ke dalam bentuk *JSON*.

```
while ($row = mysql_fetch_array($result)) {
    $faskes = array();
    $faskes["id_faskes"] = $row["id_faskes"];
    $faskes["nama_faskes"] = $row["nama_faskes"];
    $faskes["longitude"] = $row["longitude"];
    $faskes["latitude"] = $row["latitude"];
    $faskes["jenis_faskes"] = $row["jenis_faskes"];
    $faskes["tingkat_id"] = $row["tingkat_id"];
    $faskes["no_darurat"] = $row["no_darurat"];
    $faskes["jarak"] = 0;
    array_push($response["faskes"], $faskes);
}
$response["success"] = 1;
echo json_encode($response);
```

Potongan program diatas adalah sebuah perulangan untuk mengambil data dari *database* dan memasukkannya kedalam *array* (\$faskes) kemudian mengubahnya dalam bentuk *json* dengan *method* *json_encode()*.

2. *Parsing* Data (*Client*)

```
public JSONObject makeHttpRequest(String url, String method,
                                List<NameValuePair> params)
{
    try {
        if (method == "POST") {
            DefaultHttpClient httpClient = new DefaultHttpClient();
            HttpPost httpPost = new HttpPost(url);
            httpPost.setEntity(new UrlEncodedFormEntity(params));
            HttpResponse httpResponse = httpClient.execute(httpPost);
            HttpEntity httpEntity = httpResponse.getEntity();
            is = httpEntity.getContent();
        }else if (method == "GET") {
            DefaultHttpClient httpClient = new DefaultHttpClient();
            String paramString = URLEncodedUtils
                .format(params, "utf-8");
            url += "?" + paramString;
            HttpGet httpGet = new HttpGet(url);
```

```

    HttpResponse httpResponse = httpClient.execute(httpGet);
    HttpEntity httpEntity = httpResponse.getEntity();
    is = httpEntity.getContent();
}

```

Potongan program diatas adalah sebuah method `JSONObject()` dari kelas `JSONParser` yang berfungsi untuk mendapatkan data *json* dari *url* dengan membuat HTTP POST dan GET *method*.

3. Hitung Jarak

```

double latitude1 = Double.parseDouble(latitude);
double longitude1 = Double.parseDouble(longitude);
double userLatitude = gps.getLatitude();
double userLongitude = gps.getLongitude();
Location.distanceBetween(userLatitude, userLongitude,
                        latitude1, longitude1, dist);

jarak = (df.format(dist[0] / 1000)) + " KM";

```

Potongan program diatas adalah program untuk menghitung jarak dengan menggunakan *method* `distanceBetween()`, yaitu dengan menghitung jarak antara dua titik.

4. *Sorting* Data

```

if (min > dist[0] && !Arrays.asList(x).contains(id)) {
min = dist[0];
y = id;
map = new HashMap<String, String>();
map.put(TAG_ID, id);
map.put(TAG_NAMA, nama);
map.put(TAG_LONGITUDE, longitude);
map.put(TAG_LATITUDE, latitude);
map.put(TAG_JENIS_FASKES, jenis);
map.put(TAG_TINGKAT_ID, tingkat);
map.put(TAG_NO_DARURAT, no_darurat);
map.put(TAG_JARAK, jarak);
}

```

Potongan program diatas adalah sebuah pernyataan yang digunakan dalam melakukan *sorting*. Pernyataan ini dilakukan dengan mencari data dengan jarak terdekat (`min > dist[0]`) dan data dengan id yang belum ada pada *array* `x[]`

(!Arrays.asList(x).contains(id)). Pernyataan ini dilakukan didalam perulangan dalam perulangan, yaitu menentukan jarak terdekat dan menyimpannya kedalam *array*. kemudian mencari lagi jarak terdekat, jika jarak terdekat sudah terdapat pada *array* maka jarak terdekat selanjutnya yang kemudian disimpan, dst.

5. Mendapatkan rute

```
LatLng coordinates = new LatLng(userLat, userLong);
mMap.animateCamera(CameraUpdateFactory
    .newLatLngZoom(coordinates, 11));
mMap.addMarker(new MarkerOptions(
    .icon(BitmapDescriptorFactory
    .defaultMarker(BitmapDescriptorFactory.HUE_RED))
    .position(fromPosition).title("Start"));
mMap.addMarker(new MarkerOptions()
    .icon(BitmapDescriptorFactory
    .defaultMarker(BitmapDescriptorFactory.HUE_BLUE))
    .position(toPosition).title("End"));
getDirectionMap(fromPosition, toPosition);
```

Potongan program diatas adalah program untuk menampilkan rute. Rute ditampilkan dengan *marker* merah("Start") dan biru("End").

6. Menampilkan *Marker*

```
markerOptions = new MarkerOptions()
    .icon(BitmapDescriptorFactory
    .fromResource(R.drawable.hospital));
```

Potongan program diatas adalah program untuk menampilkan *marker*.

7. Menelpon

```
Intent callIntent = new Intent(Intent.ACTION_CALL);
callIntent.setData(Uri.parse("tel:" + phonenumber));
startActivity(callIntent);
```

Potongan program diatas adalah program untuk menjalankan panggilan dalam aplikasi telepon.

4.1.3. Uji Coba Sistem *Web* untuk Admin

Pada pengujian sistem ini digunakan perangkat lunak berupa *webservice* yaitu *XAMPP* dan *Emulator Genymotion* dengan *Virtual Device Google Nexus 5 – 5.0.0 – API 21 – 1080x1920*. Pengujian harus terhubung ke *internet* dan Akses lokasi untuk menjalankan aplikasi ini, karena peta yang digunakan mengambil dari *Google Maps*.

1. Admin bisa melakukan *login*

Gambar 4.2 merupakan tampilan dari Halaman Login.

Gambar 4.2 Halaman *Login*

2. Admin bisa mengelola data faskes

Gambar 4.3 merupakan tampilan dari Halaman Kelola Data Faskes.

NO	Nama Faskes	Longitude	Latitude	Jenis Faskes	Tingkat	No Telepon	
1	RSU Nur Hidayah	110.385857	-7.886203	Rumah Sakit	Pertama	+622747472941	 
2	RSK Bedah Patmasuri	110.357635	-7.829988	Rumah Sakit	Pertama	(0274) 372021	 
3	Rumah Sakit Permata Husada	110.405914	-7.865651	Rumah Sakit	Pertama	+62 274 441212	 

Gambar 4.3 Halaman Kelola Data Faskes

a. Tambah Data

Gambar 4.4 merupakan tampilan dari Halaman Tambah Faskes.

The screenshot shows a web application interface for creating a new 'Faskes' (Health Service) entry. The navigation bar at the top contains 'Home', 'Faskes', 'Admin', and 'Logout (admin)'. Below the navigation bar, the breadcrumb trail is 'Home > Faskes > Create'. The main heading is 'Create Faskes'. A note below the heading states 'Fields with * are required.' The form contains the following fields: 'Nama Faskes *' with the value 'RSUD Bantul', 'Longitude *' with the value '110.3354615', 'Latitude *' with the value '-7.892496', 'Jenis Faskes *' with a dropdown menu set to 'Rumah Sakit', 'Tingkat *' with a dropdown menu set to 'Pertama', and 'No Telepon *' with the value '+62 274 367381'. A 'Create' button is located at the bottom of the form.

Gambar 4.4 Halaman Tambah Faskes

b. Ubah Data

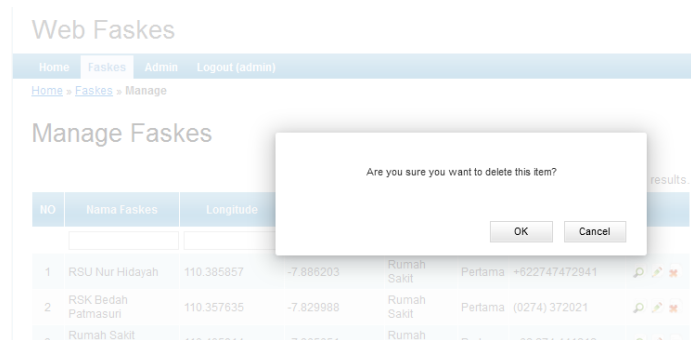
Gambar 4.5 merupakan tampilan dari Halaman Ubah Faskes.

The screenshot shows a web application interface for updating an existing 'Faskes' entry. The navigation bar at the top contains 'Home', 'Faskes', 'Admin', and 'Logout (admin)'. Below the navigation bar, the breadcrumb trail is 'Home > Faskes > Update'. The main heading is 'Update Faskes 3'. A note below the heading states 'Fields with * are required.' The form contains the following fields: 'Nama Faskes *' with the value 'RSU Nur Hidayah', 'Longitude *' with the value '110.385857', 'Latitude *' with the value '-7.886203', 'Jenis Faskes *' with a dropdown menu set to 'Rumah Sakit', 'Tingkat *' with a dropdown menu set to 'Pertama', and 'No Telepon *' with the value '+622747472941'. A 'Save' button is located at the bottom of the form.

Gambar 4.5 Halaman Ubah Faskes

c. Hapus Data

Gambar 4.6 merupakan tampilan dari Halaman Hapus Faskes.

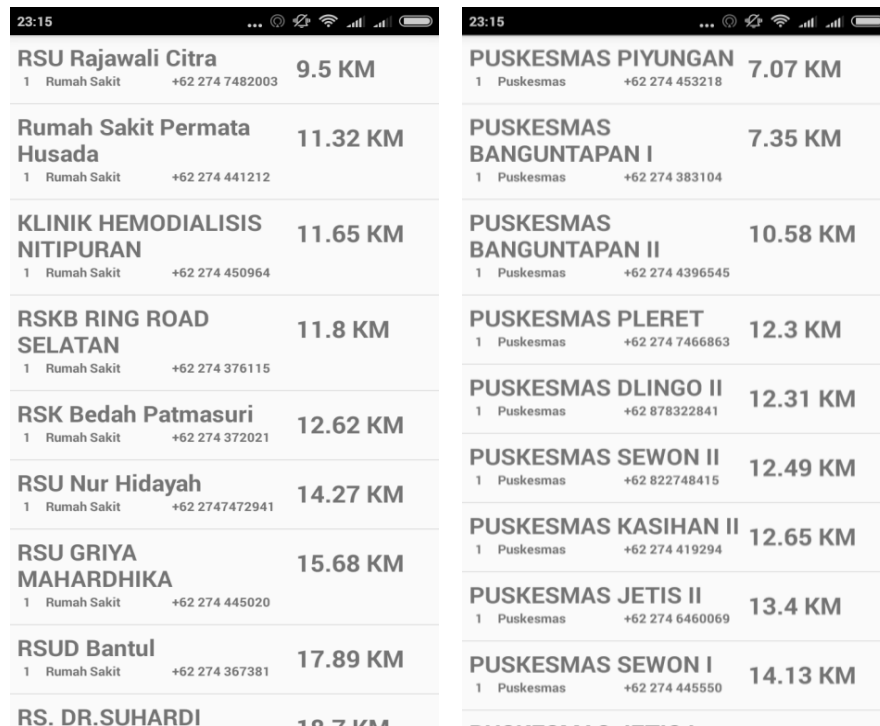


Gambar 4.6 Halaman Hapus Faskes

4.1.4. Uji Coba Sistem *Android* untuk *User*

1. *User* dapat melihat daftar faskes

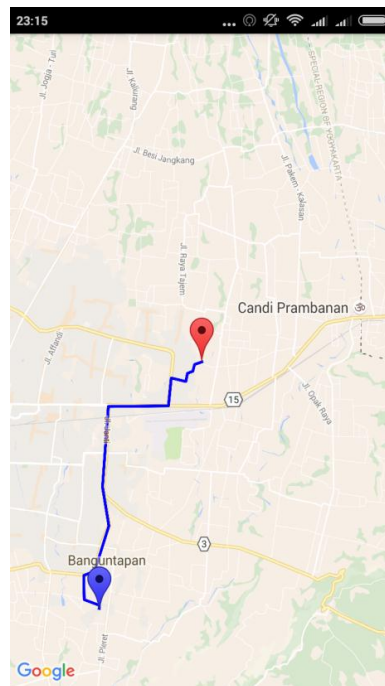
Gambar 4.7 merupakan tampilan dari Menu Faskes.



Gambar 4.7 Daftar Faskes

2. *User* mendapatkan navigasi rute

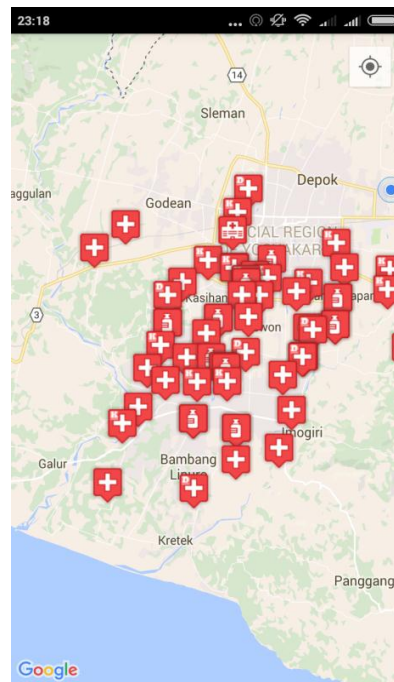
Gambar 4.8 merupakan tampilan setelah memilih salah satu *list* di Menu Faskes.



Gambar 4.8 Rute Faskes

3. *User* dapat melihat penyebaran faskes

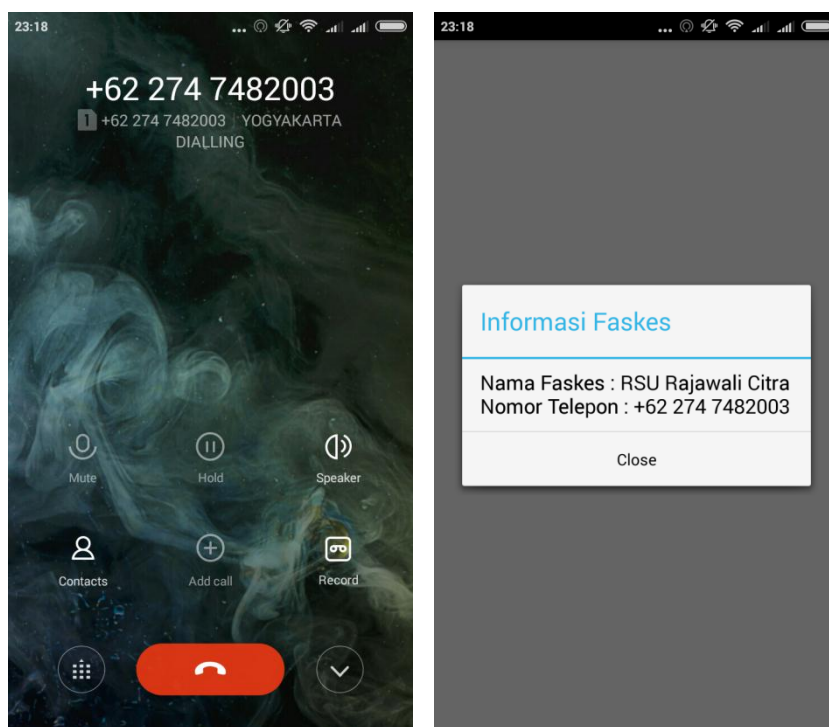
Gambar 4.9 merupakan tampilan dari Menu Pemetaan Faskes.



Gambar 4.9 Pemetaan Faskes

4. *User* dapat melakukan panggilan darurat ke faskes terdekat

Gambar 4.10 merupakan Menu Telepon Darurat dan Dialog Informasi Faskes.



Gambar 4.10 Tombol Darurat

4.2. Pembahasan

4.2.1. Pembahasan Sistem *Web* untuk Admin

1. Halaman *Login*

Pada sistem yang digunakan untuk admin, admin harus melakukan *login* terlebih dahulu sebelum masuk ke halaman *web*.

2. Halaman Kelola Data Faskes

Halaman Kelola Data Faskes akan menampilkan daftar faskes yang disampingnya terdapat tombol *view*, *ubah*, dan *delete*.

3. Halaman Tambah Data Faskes

Halaman Tambah Data Faskes berisi *form* untuk menambahkan data faskes ke *database*. *Form* ini mempunyai *field-field* yang semuanya harus diisi.

4.2.2. Pembahasan Sistem *Android* untuk *User*

1. Menu Faskes

Menu faskes memiliki 5 kategori, yaitu Rumah Sakit, Puskesmas, Klinik, Dokter Praktik, dan Apotik yang menampilkan daftar faskes dari kategori yang dipilih beserta jaraknya yang telah diurutakan berdasarkan jarak terdekat.

Dari daftar tersebut aplikasi akan menampilkan rute *user* ke faskes yang dipilih. Marker warna merah adalah posisi *user* dan marker warna biru adalah faskes yang dipilih.

2. Menu Pemetaan Faskes

Menu pemetaan faskes akan menampilkan penyebaran faskes di daerah Bantul dengan menggunakan *marker* yang berbeda dalam setiap jenis faskes.

3. Menu Tombol Darurat

Tombol darurat akan memanggil nomor telepon faskes dengan jarak terdekat. Menu ini juga akan menampilkan Dialog Informasi ketika panggilan sudah selesai. Dialog ini berisi informasi nama faskes yang telah dipanggil beserta Nomor Telepon.