

BAB IV

IMPLEMENTASI SISTEM

4.1 Implementasi Basis Data

Perangkat lunak (*software*) yang digunakan dalam mengimplementasikan basis data dalam pembuatan sistem ini adalah *SQLite Studio* dengan bahasa pemrograman *Java* dan *Kotlin*. Berikut ini akan dibahas tabel-tabel yang digunakan menjadi basis data sistem :

4.1.1 Implementasi Tabel Admin

Tabel admin memiliki 2 *field* yaitu *username* dan *password*. Tabel admin berguna untuk menyimpan data admin.



Name	Data type	Primary Key	Foreign Key	Unsigned	Check	Not NULL	Default value
1 username	VARCHAR(255)					Yes	NULL
2 password	VARCHAR(255)					Yes	NULL

Gambar 4.1 Implementasi Tabel Admin

4.1.2 Implementasi Tabel Rumah Makan

Tabel rumah makan memiliki 14 *field* yaitu *Id* sebagai *primary key*, *compoundcode*, *formattedaddress*, *foto1*, *foto2*, *foto3*, *globalcode*, *lat*, *lng*, *name*, *jam_buka*, *rating*, *reference*, *userratingtotal*.

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not Null	Collate	Default value
1 ID	INTEGER	✓						
2 COMPANYCODE	TEXT							
3 FORMATTEDADDRESS	TEXT							
4 ZODIAC	TEXT							
5 ZODIAC	TEXT							
6 FURUS	TEXT							
7 ALIBALCODE	TEXT							
8 LAT	DOUBLE							
9 LONG	DOUBLE							
10 NAME	TEXT							
11 JAWA_BURU	TEXT							
12 KATRANG	DOUBLE							
13 KATRANG2	TEXT							
14 SUBPARTNERID	INTEGER							

Gambar 4.2 Implementasi Tabel Rumah Makan

4.1.3 Implementasi Tabel Daftar Menu

Tabel daftar menu memiliki 4 *field* yaitu *Id* sebagai *primary key*, harga, nama, rumah_makan_id.

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not Null	Collate	Default value
1 ID	INTEGER	✓						
2 HARGA	TEXT							
3 NAME	TEXT							
4 RUMAH_MAKAN_ID	INTEGER							

Gambar 4.3 Implementasi Tabel Daftar Menu

4.2 Implementasi Sistem

4.2.1 Tampilan Menu Utama User

Menu utama user merupakan halaman yang berisi informasi mengenai aplikasi berupa gambar (Makanan Jawa, Padang dan Madura, dan di bagian sebelah kiri terdapat *side* Menu ☰ yang dapat dilihat pada Gambar 4.5 serta di sebelah kanan terdapat *pop up* login admin ⋮ yang

dapat dilihat pada Gambar 4.12 . Tampilan menu utama user dapat di lihat pada Gambar 4.4.



Gambar 4.4 Tampilan Menu Utama User

Tampilan halaman menu utama user dibuat dengan menggunakan *coding* sebagai berikut :

```
DrawerLayout drawer = findViewById(R.id.drawer_layout);
ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
    this, drawer, toolbar, R.string.navigation_drawer_open,
    R.string.navigation_drawer_close);
drawer.addDrawerListener(toggle);
toggle.syncState();
NavigationView navigationView = findViewById(R.id.nav_view);
navigationView.setNavigationItemSelectedListener(this);
<androidx.drawerlayout.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/drawer_layout"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:fitsSystemWindows="true"
tools:openDrawer="start" >
<include
    layout="@layout/app_bar_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
<com.google.android.material.navigation.NavigationView
    android:id="@+id/nav_view"
```

```

        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header_main"
        app:menu="@menu/activity_main_drawer" />
</androidx.drawerlayout.widget.DrawerLayout>
<menu
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/action_login"
        android:orderInCategory="100"
        android:title="@string/action_settings"
        app:showAsAction="never" />
</menu>

```

4.2.2 Tampilan *Side Menu*

Side menu berisi navigasi aplikasi (*latitude, longitude, accuracy, lokasi user*), menu utama, rumah makan padang, rumah makan madura, rumah makan jawa, daftar rumah makan dan opsi keluar. Tampilan *side menu* dapat dilihat pada gambar 4.5.



Gambar 4.5 Tampilan *Side Menu*

Tampilan *side menu* dibuat dengan menggunakan *coding* sebagai berikut :

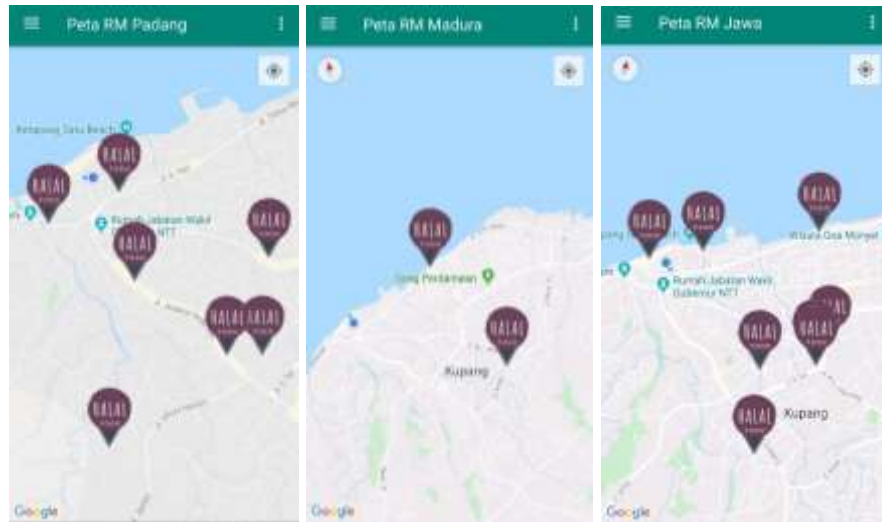
```

<group android:checkableBehavior="single">
  <item
    android:id="@+id/nav_main"
    android:icon="@drawable/ic_menu_gallery"
    android:title="Menu Utama" />
  <item
    android:id="@+id/nav_padang"
    android:icon="@drawable/ic_menu_slideshow"
    android:title="RM Padang" />
  <item
    android:id="@+id/nav_madura"
    android:icon="@drawable/ic_menu_slideshow"
    android:title="RM Madura" />
  <item
    android:id="@+id/nav_jawa"
    android:icon="@drawable/ic_menu_slideshow"
    android:title="RM Jawa" />
  <item
    android:id="@+id/nav_all"
    android:icon="@drawable/ic_menu_slideshow"
    android:title="Daftar Rumah Makan" />
</group>
<item android:title="Ops" >
  <menu>
    <item
      android:id="@+id/nav_exit"
      android:icon="@drawable/ic_menu_send"
      android:title="Keluar" />
    </item>
  </menu>
</item>

```

4.2.3 Tampilan Menu RM (Padang, Madura dan Jawa)

Menu RM Padang, Madura dan Jawa merupakan halaman yang menampilkan peta digital kota Kupang yang sudah tersedia tanda lokasi rumah makan halal Padang, Madura dan Jawa. Tampilan menu rumah makan (padang, madura dan jawa) dapat dilihat pada gambar 4.6.



Gambar 4.6 Tampilan Menu RM Padang, Madura dan Jawa

Tampilan halaman menu rumah makan padang, madura dan jawa dibuat dengan menggunakan *coding* sebagai berikut :

```
Select.from(RumahMakan.class)
    .where (
        Condition.prop("GLOBALCODE").eq(mParam1)
    ).list();
System.out.println("list = " + list.size());
```

4.2.4 Tampilan Menu Daftar Rumah Makan

Menu daftar rumah makan merupakan halaman yang menampilkan semua daftar rumah makan halal di kota Kupang. Tampilan menu daftar rumah makan dapat dilihat pada gambar 4.7.



Gambar 4.7 Tampilan Menu Daftar Rumah Makan

Tampilan halaman menu daftar rumah makan dibuat dengan menggunakan *coding* sebagai berikut :

```
List<RumahMakan> rumahMakans =
RumahMakan.listAll(RumahMakan.class);
recyclerView.setAdapter(new RumahMakanAdapter(rumahMakans,
mListener));
```

4.2.5 Tampilan Info Rumah Makan

Info rumah makan merupakan halaman yang menampilkan informasi rumah makan. Informasi tersebut berupa lokasi dan jalur rumah makan, alamat, *rating*, jam buka serta foto rumah makan. Tampilan halaman info rumah makan dapat dilihat pada gambar 4.8.



Gambar 4.8 Tampilan Info Rumah Makan


Tampilan halaman info rumah makan dibuat dengan menggunakan *coding* sebagai berikut :

```

textAlamat.setText(rumahMakan.getFormatted_address());
textRating.setText(rumahMakan.getRating().toString());
textKode.setText(rumahMakan.getCompound_code());
textNamaRM.setText(rumahMakan.getName());
switch (rumahMakan.getGlobal_code()) {
    case "RMjawa.json":
        textJenis.setText("Jawa");
        break;
    case "RMmadura.json":
        textJenis.setText("Madura");
        break;
    case "Rmpadang.json":
        textJenis.setText("Padang");
        break;
}

```

4.2.6 Tampilan Buka Lokasi Google Map

Tampilan fitur  ini tersedia pada gambar 4.8. Fitur ini akan menampilkan titik lokasi rumah makan yang dipilih pada peta digital. Tampilan buka lokasi google map dapat dilihat pada gambar 4.9.



Gambar 4.9 Tampilan Buka Lokasi Google Map

Tampilan fitur buka lokasi google map dibuat dengan menggunakan *coding* sebagai berikut :

```
private void OpenMap() {
    Uri.Builder builder = new Uri.Builder();
    builder.scheme("https")
        .authority("www.google.com")
        .appendPath("maps")
        .appendPath("search")
        .appendPath("")
        .appendQueryParameter("api", "1")
        .appendQueryParameter("query", rumahMakan.getName());
    //
    intent.getStringExtra("id");
    String key = builder.build().toString();
    Intent intent = new Intent(Intent.ACTION_VIEW,
    Uri.parse(key));
    startActivity(intent);
}
```

4.2.7 Tampilan Buka Jalur Google Map

Tampilan fitur  ini tersedia pada gambar 4.8. Fitur ini akan menampilkan jalur pada peta digital yang dapat ditempuh *user* dari

lokasi asal ke rumah makan yang sudah dipilih. Tampilan buka jalur google map dapat dilihat pada gambar 4.10.



Gambar 4.10 Buka Jalur Google Map

Tampilan fitur buka jalur google map dibuat dengan menggunakan *coding* sebagai berikut :

```
private void OpenJalur() {
    final String format = String.format(
        Locale.ENGLISH,
        "http://maps.google.com/maps?saddr=%f,%f&daddr=%f,%f",
        myLat,
        myLng,
        rumahMakan.getLat(),
        rumahMakan.getLng()
    );
    System.out.println("format = " + format);
    Intent intent = new Intent(Intent.ACTION_VIEW,
        Uri.parse(format));
    intent.setPackage("com.google.android.apps.maps");
    startActivity(intent);
}
```

4.2.8 Tampilan Daftar Menu Rumah Makan

Daftar menu rumah makan merupakan halaman yang menampilkan daftar menu makanan beserta harganya. Tampilan daftar menu dapat dilihat pada gambar 4.11.



Gambar 4.11 Tampilan Daftar Menu Rumah Makan

Tampilan halaman daftar menu dibuat dengan menggunakan *coding* sebagai berikut :

```
List<DaftarMenu> list = Select.from(DaftarMenu.class)
    .where(
        Condition.prop("Rumah_Makan_ID").eq(idRM),
        Condition.prop("NAMA").notEq("0")
    ).list();
System.out.println("list.size() = " + list.size());
recyclerView.setAdapter(new DaftarMenuAdapter(list,
mListener));
```

4.2.9 Tampilan Pop Up Login Admin

Pop up login admin merupakan halaman yang berisi *field* *username* dan *password* yang akan diisi oleh pengguna agar dapat masuk

pada tampilan admin. Tampilan *pop up login* admin dapat dilihat pada gambar 4.12.



Gambar 4.12 Tampilan *Pop Up Login* Admin

Tampilan halaman *pop up login* admin dibuat dengan menggunakan *coding* sebagai berikut :

```
final AlertDialog dialog = new
AlertDialog.Builder(MainActivity.this)
    .setView(prompt)
    .setTitle("Admin Mode")
    .setMessage("Masukkan Username dan Password")
    .setPositiveButton("Login", null) //Set to null. We
override the onclick
    .setNegativeButton("Batal", null)
    .create();
final EditText txtPassword =
prompt.findViewById(R.id.txtPassword);
final EditText txtUsername =
prompt.findViewById(R.id.txtUsername);
dialog.setOnShowListener({
    String pass = txtPassword.getText().toString();
    String user = txtUsername.getText().toString();
    boolean login = Login(user, pass);
    if (login) {
        navUserSub.setText("Admin Mode");
        txtPassword.setText("");
        txtUsername.setText("");
        dialog.dismiss();
        setAdmin(true);
    }
});
```

```

    }
    else {
        txtPassword.setError("Salah password");
        txtUsername.setError("Salah username");
    }
});
dialog.show();

```

4.2.10 Tampilan Menu Utama Admin

Menu utama admin merupakan halaman yang berisi informasi mengenai daftar rumah makan serta menambahkan data. Tampilan menu utama admin dapat di lihat pada Gambar 4.13.



Gambar 4.13 Tampilan Menu Utama Admin

Tampilan halaman menu utama admin dibuat dengan menggunakan *coding* sebagai berikut :

```

View view = inflater.inflate(R.layout.fragment_rumahmakan_list,
container, false);
// Set the adapter
if (view instanceof RecyclerView) {
    Context context = view.getContext();
    RecyclerView recyclerView = (RecyclerView) view;
    if (mColumnCount <= 1) {
        recyclerView.setLayoutManager(new
LinearLayoutManager(context));

```

```

    } else {
        recyclerView.setLayoutManager(new
GridLayoutManager(context, mColumnCount));
    }
    List<RumahMakan> rumahMakans =
RumahMakan.listAll(RumahMakan.class);
    System.out.println("rumahMakans.size() = " +
rumahMakans.size());
    recyclerView.setAdapter(new RumahMakanAdapter(rumahMakans,
mListener));
}
return view;

```

4.2.11 Tampilan *Pop Up* Aksi Admin

Pop up aksi admin merupakan halaman yang berisikan pilihan aksi admin seperti Lihat, Edit dan Hapus rumah makan. Tampilan *pop up* aksi admin dapat dilihat pada gambar 4.14.



Gambar 4.14 Tampilan *Pop Up* Aksi Admin

Tampilan halaman *pop up* aksi admin dibuat dengan menggunakan *coding* sebagai berikut :

```

AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setTitle("Pilih Aksi :"+item.getName());
builder.setItems(colors, new DialogInterface.OnClickListener()

```

```

{
  @Override
  public void onClick(DialogInterface dialog, int which) {
    // the user clicked on colors[which]
    System.out.println("which = " + which);
    switch (which) {
      case 0:
        System.out.println("Lihat");
        System.out.println("item = " + item);
        startActivity(myIntent);
        break;
      case 1:
        System.out.println("Edit");
        EditRumahMakan(item);
        break;
      case 2:
        System.out.println("Hapus");
        HapusRumahMakan(item);
        break;
      case 3:
        System.out.println("Batal");
        break;
    }
  }
});
builder.setCancelable(false);
builder.show();

```

4.2.12 Tampilan Menu Aksi Lihat

Menu aksi lihat merupakan halaman yang digunakan admin untuk melihat data-data yang berkaitan dengan rumah makan. Tampilan menu aksi lihat dapat dilihat pada gambar 4.15.



Gambar 4.15 Tampilan Menu Aksi Lihat

Tampilan halaman menu aksi lihat dibuat dengan menggunakan *coding* sebagai berikut :

```

textAlamat.setText(rumahMakan.getFormatted_address());
textRating.setText(rumahMakan.getRating().toString());
textKode.setText(rumahMakan.getCompound_code());
textNamaRM.setText(rumahMakan.getName());
    switch (rumahMakan.getGlobal_code()) {
        case "RMjawa.json":
            textJenis.setText("Jawa");
            break;
        case "RMmadura.json":
            textJenis.setText("Madura");
            break;
        case "Rmpadang.json":
            textJenis.setText("Padang");
            break;
    }
layout.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        //this will log the page number that was click
        Log.i("TAG", "This page was clicked: " + position);

        if (mAdmin)
            SimpanGambar(position);
    }
});

```


4.2.13 Tampilan Menu Aksi Edit

Menu aksi edit merupakan halaman yang digunakan admin untuk mengedit data-data yang berkaitan dengan rumah makan. Tampilan menu aksi edit dapat dilihat pada gambar 4.16.



Gambar 4.16 Tampilan Menu Aksi Edit

Tampilan halaman menu aksi edit dibuat dengan menggunakan *coding* sebagai berikut :

```
item.setFormatted_address(RumahMakanAlamat.getText().toString());
item.setName(RumahMakanNama.getText().toString());
item.setCompound_code(RumahMakanKode.getText().toString());
double value = 0d;
if (MyUtil.isValidRating(ratingStr)) {
    value = Double.parseDouble(ratingStr);
} else {
    RumahMakanRating.setError("Rating tidak valid");
    RumahMakanRating.requestFocus();
    return;
}
item.setRating(value);
final int selectedItemPosition =
spinner.getSelectedItemPosition();
System.out.println("spinner = " + spinner.getSelectedItem());
switch (selectedItemPosition) {
    case 1:
        item.setGlobal_code("RMjawa.json");
        break;
```

```

    case 0:
        item.setGlobal_code("RMmadura.json");
        break;
    case 2:
        item.setGlobal_code("Rmpadang.json");
        break;
    default:
        System.out.println("Fucking wrong in here");
        break;
}
item.save();

```

4.2.14 Tampilan Menu Aksi Hapus

Menu aksi hapus merupakan halaman yang digunakan admin untuk menghapus data rumah makan yang ada pada sistem. Tampilan menu aksi hapus dapat dilihat pada gambar 4.17.



Gambar 4.17 Tampilan Menu Aksi Hapus

Tampilan halaman menu aksi hapus dibuat dengan menggunakan *coding* sebagai berikut :

```

item.delete();
dialog.dismiss();
Toast
    .makeText(getApplicationContext(), "Berhasil menghapus Rumah
Makan", Toast.LENGTH_LONG)
    .show();
ResetDataFragment();

```

4.2.15 Tampilan Menu Tambah Rumah Makan

Menu tambah rumah makan merupakan halaman yang digunakan admin untuk menambahkan rumah makan serta data-data yang berkaitan seperti nama rumah makan, alamat, jam buka, *rating* dan jenis rumah makan. Tampilan menu tambah rumah makan dapat dilihat pada gambar 4.18.



Gambar 4.18 Tambah Rumah Makan

Tampilan halaman tambah rumah makan dibuat dengan menggunakan *coding* sebagai berikut :

```
RumahMakan rumahMakan = new RumahMakan();  
rumahMakan.setFormatted_address(RumahMakanAlamat.getText().toString());  
rumahMakan.setLng(position.longitude);  
rumahMakan.setLat(position.latitude);  
rumahMakan.setName(RumahMakanNama.getText().toString());  
rumahMakan.setCompound_code(RumahMakanKode.getText().toString());  
rumahMakan.setRating(value);  
rumahMakan.save();
```

4.2.16 Tampilan Daftar Menu Admin

Daftar menu admin merupakan halaman yang digunakan admin untuk melihat dan menambahkan menu makanan apa saja yang terdapat pada sebuah rumah makan. Tampilan daftar menu makanan admin dapat dilihat pada gambar 4.19.



Gambar 4.19 Tampilan Daftar Menu Admin

Tampilan halaman daftar menu admin dibuat dengan menggunakan *coding* sebagai berikut :

```
System.out.println("idRM = " + idRM);  
List<DaftarMenu> list = Select.from(DaftarMenu.class)  
    .where(  
        Condition.prop("Rumah_Makan_ID").eq(idRM),  
        Condition.prop("NAMA").notEq("0")  
    ).list();
```

4.2.17 Tampilan Aksi Add Menu

Aksi *add* menu merupakan halaman yang digunakan admin untuk menambahkan menu baru pada daftar menu dengan mengisi *field* nama dan harga. Tampilan aksi *add* menu dapat dilihat pada gambar 4.20.



Gambar 4.20 Tampilan Aksi Add Menu

Tampilan halaman aksi *add* menu dibuat dengan menggunakan *coding* sebagai berikut :

```
DaftarMenu daftarMenu = new DaftarMenu(  
    txtHargaMenu.getText().toString(),  
    txtNamaMenu.getText().toString(),  
    rumahMakan.getId()  
);  
daftarMenu.save();
```

4.2.18 Tampilan Aksi Edit Menu

Aksi edit menu merupakan halaman yang digunakan admin untuk memperbaharui data menu yang sudah ada pada daftar menu rumah makan. Tampilan aksi edit menu dapat dilihat pada gambar 4.21.



Gambar 4.21 Tampilan Aksi Edit Menu

Tampilan halaman aksi edit menu dibuat dengan menggunakan *coding* sebagai berikut :

```

if (!isValidHarga(hargaStr)) {
    txtHargaMenu.setError("Harga tidak valid");
    txtHargaMenu.requestFocus();
    return;
}
item.setNama(txtNamaMenu.getText().toString());
item.setHarga(txtHargaMenu.getText().toString());
System.out.println("Update.daftarMenu = " + item);
item.save();

```

4.2.19 Tampilan Aksi Hapus Menu

Aksi hapus menu merupakan halaman yang digunakan admin untuk menghapus menu dari daftar menu rumah makan. Tampilan aksi hapus menu dapat dilihat pada gambar 4.22.



Gambar 4.22 Tampilan Aksi Hapus Menu

Tampilan halaman aksi hapus menu dibuat dengan menggunakan *coding* sebagai berikut :

```
AlertDialog HapusDialog =new
AlertDialog.Builder(RumahMakanActivity.this)
    .setTitle("Hapus")
    .setMessage("Anda Yakin untuk Hapus")
    .setPositiveButton("Hapus", {
        //your deleting code
        item.delete();
        dialog.dismiss();
        switchToMenu();
    });
```