**PEMROGRAMAN MOBILE**

**APLIKASI LAUNDRY**

Dosen pengampu: Nurfiah. S.ST, M.Kom

****

**RAFANO ALESANDER (2211533016)**

**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

**2024**

DATABASE SQLITE

1. **Model** 
   * **Model Pelanggan**

**package** com.example.aplikasilaundry.model;

**public** **class** **ModelPelanggan** {

String id, nama, email, hp;

**public** String **getId**(){

**return** id;

}

**public** **void** **setId**(String id) {

this.id = id;

}

**public** String **getNama**() {

**return** nama;

}

**public** **void** **setNama**(String nama) {

this.nama = nama;

}

**public** String **getEmail**() {

**return** email;

}

**public** **void** **setEmail**(String email) {

this.email = email;

}

**public** String **getHp**() {

**return** hp;

}

**public** **void** **setHp**(String hp) {

this.hp = hp;

}

}

* + **Model Layanan**

**package** com.example.aplikasilaundry.model;

**public** **class** **ModelLayanan** {

String id, tipe, harga;

// Method Getter and Setter

**public** String **getId**() {

**return** id;

}

**public** **void** **setId**(String id) {

this.id = id;

}

**public** String **getTipe**() {

**return** tipe;

}

**public** **void** **setTipe**(String tipe) { // Ubah Tipe menjadi tipe

this.tipe = tipe;

}

**public** String **getHarga**() {

**return** harga;

}

**public** **void** **setHarga**(String harga) {

this.harga = harga;

}

1. **SQLiteHelper**

Untuk mengelola database SQLite pada alikasi android.

**package** com.example.aplikasilaundry.database;

**import** android.content.ContentValues;

**import** android.content.Context;

**import** android.database.Cursor;

**import** android.database.sqlite.SQLiteDatabase;

**import** android.database.sqlite.SQLiteOpenHelper;

**import** com.example.aplikasilaundry.model.ModelLayanan;

**import** com.example.aplikasilaundry.model.ModelPelanggan;

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** **SQLiteHelper** **extends** **SQLiteOpenHelper** {

**public** **static** String DATABASE\_NAME = "my\_laundry.db";

**public** **static** **final** int DATABASE\_VERSION = 2;

**public** **static** **final** String TABLE\_PELANGGAN = "pelanggan";

**public** **static** **final** String KEY\_PELANGGAN\_ID = "pelanggan\_id";

**public** **static** **final** String KEY\_PELANGGAN\_NAMA = "nama";

**public** **static** **final** String KEY\_PELANGGAN\_EMAIL = "email";

**public** **static** **final** String KEY\_PELANGGAN\_HP = "hp";

**public** **static** **final** String TABLE\_LAYANAN = "layanan";

**public** **static** **final** String KEY\_LAYANAN\_ID = "layanan\_id";

**public** **static** **final** String KEY\_LAYANAN\_TIPE = "tipeLayanan";

**public** **static** **final** String KEY\_LAYANAN\_HARGA = "harga";

**private** **static** **final** String CREATE\_TABLE\_PELANGGAN = "CREATE TABLE " +

TABLE\_PELANGGAN + "("

+ KEY\_PELANGGAN\_ID + " TEXT, " + KEY\_PELANGGAN\_NAMA + " TEXT, "+

KEY\_PELANGGAN\_EMAIL + " TEXT, "+KEY\_PELANGGAN\_HP +" TEXT )";

**private** **static** **final** String CREATE\_TABLE\_LAYANAN = "CREATE TABLE " +

TABLE\_LAYANAN + "("

+ KEY\_LAYANAN\_ID + " TEXT, " + KEY\_LAYANAN\_TIPE + " TEXT, " +

KEY\_LAYANAN\_HARGA + " TEXT )";

**public** **SQLiteHelper**(Context context){

super(context, DATABASE\_NAME, null, DATABASE\_VERSION);

}

@Override

**public** **void** **onCreate**(SQLiteDatabase db) {

db.execSQL(CREATE\_TABLE\_PELANGGAN);

db.execSQL(CREATE\_TABLE\_LAYANAN);

}

@Override

**public** **void** **onUpgrade**(SQLiteDatabase db, int oldVersion, int newVersion)

{

db.execSQL("DROP TABLE IF EXISTS " + TABLE\_PELANGGAN);

db.execSQL("DROP TABLE IF EXISTS " + TABLE\_LAYANAN);

onCreate(db);

}

**public** boolean **insertPelanggan**(ModelPelanggan mp){

SQLiteDatabase database = this.getReadableDatabase();

ContentValues contentValues = **new** **ContentValues**();

contentValues.put(KEY\_PELANGGAN\_ID, mp.getId());

contentValues.put(KEY\_PELANGGAN\_NAMA, mp.getNama());

contentValues.put(KEY\_PELANGGAN\_EMAIL, mp.getEmail());

contentValues.put(KEY\_PELANGGAN\_HP, mp.getHp());

long id = database.insert(TABLE\_PELANGGAN, null, contentValues);

database.close();

**if** (id != -1){

**return** true;

}**else**{

**return** false;

}

}

**public** List<ModelPelanggan> **getPelanggan**(){

List<ModelPelanggan> pel = **new** **ArrayList**<ModelPelanggan>();

String query = "SELECT \* FROM " + TABLE\_PELANGGAN;

SQLiteDatabase db = this.getWritableDatabase();

Cursor cursor = db.rawQuery(query, null);

**if** (cursor.moveToFirst()){

**do**{

ModelPelanggan k = **new** **ModelPelanggan**();

k.setId(cursor.getString(0));

k.setNama(cursor.getString(1));

k.setEmail(cursor.getString(2));

k.setHp(cursor.getString(3));

pel.add(k);

}**while** (cursor.moveToNext());

}

**return** pel;

}

// Fungsi untuk memperbarui pelanggan

**public** boolean **updatePelanggan**(String id, String name, String email, String hp) {

SQLiteDatabase db = this.getWritableDatabase();

ContentValues contentValues = **new** **ContentValues**();

contentValues.put(KEY\_PELANGGAN\_NAMA, name);

contentValues.put(KEY\_PELANGGAN\_EMAIL, email);

contentValues.put(KEY\_PELANGGAN\_HP, hp);

// Update the row where the id matches

int result = db.update(TABLE\_PELANGGAN, contentValues, KEY\_PELANGGAN\_ID + " = ?", **new** **String**[]{id});

db.close();

**return** result > 0; // Mengembalikan true jika update berhasil

}

// Fungsi untuk menghapus pelanggan

**public** boolean **deletePelanggan**(String id) {

SQLiteDatabase db = this.getWritableDatabase();

int result = db.delete(TABLE\_PELANGGAN, KEY\_PELANGGAN\_ID + " = ?", **new** **String**[]{id});

db.close();

**return** result > 0; // Mengembalikan true jika penghapusan berhasil

}

**public** boolean **insertLayanan**(ModelLayanan ml){

SQLiteDatabase database = this.getReadableDatabase();

ContentValues contentValues = **new** **ContentValues**();

contentValues.put(KEY\_LAYANAN\_ID, ml.getId());

contentValues.put(KEY\_LAYANAN\_TIPE, ml.getTipe());

contentValues.put(KEY\_LAYANAN\_HARGA, ml.getHarga());

long id = database.insert(TABLE\_LAYANAN, null, contentValues);

database.close();

**if** (id != -1){

**return** true;

}**else**{

**return** false;

}

}

**public** List<ModelLayanan> **getLayanan**() {

List<ModelLayanan> lay = **new** **ArrayList**<ModelLayanan>();

String query = "SELECT \* FROM " + TABLE\_LAYANAN;

SQLiteDatabase db = this.getWritableDatabase();

Cursor cursor = db.rawQuery(query, null);

**if** (cursor.moveToFirst()){

**do** {

ModelLayanan k = **new** **ModelLayanan**();

k.setId(cursor.getString(0));

k.setTipe(cursor.getString(1));

k.setHarga(cursor.getString(2));

lay.add(k);

} **while** (cursor.moveToNext());

}

**return** lay;

}

**public** boolean **updateLayanan**(String id, String tipe, String harga) {

SQLiteDatabase db = this.getWritableDatabase();

ContentValues contentValues = **new** **ContentValues**();

contentValues.put(KEY\_LAYANAN\_TIPE, tipe);

contentValues.put(KEY\_LAYANAN\_HARGA, harga);

int result = db.update(TABLE\_LAYANAN, contentValues, KEY\_LAYANAN\_ID + " = ?", **new** **String**[]{id});

db.close();

**return** result > 0;

}

**public** boolean **deleteLayanan**(String id) {

SQLiteDatabase db = this.getWritableDatabase();

int result = db.delete(TABLE\_LAYANAN, KEY\_LAYANAN\_ID + " = ?", **new** **String**[]{id});

db.close();

**return** result > 0;

}

}

1. **Adapter**
   * **Adapter Pelanggan**

**package** com.example.aplikasilaundry.adapter;

**import** android.content.Context;

**import** android.view.LayoutInflater;

**import** android.view.View;

**import** android.view.ViewGroup;

**import** android.widget.TextView;

**import** androidx.recyclerview.widget.RecyclerView;

**import** com.example.aplikasilaundry.model.ModelPelanggan;

**import** com.example.aplikasilaundry.R;

**import** java.util.List;

**public** **class** **AdapterPelanggan** **extends** **RecyclerView**.Adapter<AdapterPelanggan.ViewHolder> {

**private** **static** **final** String TAG = AdapterPelanggan.class.getSimpleName();

**private** Context context;

**private** List<ModelPelanggan> list;

**private** View.OnClickListener onItemClicked;

**public** **void** **setOnItemClickListener**(View.OnClickListener itemClickListener)

{

onItemClicked = itemClickListener;

}

**public** **AdapterPelanggan**(Context context, List<ModelPelanggan> list)

{

this.context = context;

this.list = list;

}

@Override

**public** ViewHolder **onCreateViewHolder**(ViewGroup parent, int viewType)

{

View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item\_pelanggan, parent, false);

**return** **new** **ViewHolder**(view);

}

**public** **void** **onBindViewHolder**(ViewHolder holder, int position)

{

ModelPelanggan item = list.get(position);

holder.tvNama.setText(item.getNama());

holder.tvHp.setText(item.getHp());

}

**public** int **getItemCount**()

{

**return** list.size();

}

**public** **void** **clear**()

{

int size = this.list.size();

this.list.clear();

notifyItemRangeRemoved(0, size);

}

**public** **class** **ViewHolder** **extends** **RecyclerView**.ViewHolder

{

TextView tvNama, tvHp;

**public** **ViewHolder**(View itemView)

{

super(itemView);

tvNama = (TextView) itemView.findViewById(R.id.tvItemPelNama);

tvHp = (TextView) itemView.findViewById(R.id.tvItemPelHp);

itemView.setTag(this);

itemView.setOnClickListener(onItemClicked);

}

}

}

* + **Adapter Layanan**

**package** com.example.aplikasilaundry.adapter;

**import** android.content.Context;

**import** android.view.LayoutInflater;

**import** android.view.View;

**import** android.view.ViewGroup;

**import** android.widget.TextView;

**import** androidx.annotation.NonNull;

**import** androidx.recyclerview.widget.RecyclerView;

**import** com.example.aplikasilaundry.R;

**import** com.example.aplikasilaundry.model.ModelLayanan;

**import** java.util.List;

**public** **class** **AdapterLayanan** **extends** **RecyclerView**.Adapter<AdapterLayanan.LayananViewHolder> {

**private** **static** **final** String TAG = AdapterLayanan.class.getSimpleName();

**private** Context context;

**private** List<ModelLayanan> list;

**private** View.OnClickListener onItemClicked;

**public** **void** **setOnItemClickListener**(View.OnClickListener

itemClickListener) {

onItemClicked = itemClickListener;

}

// Konstruktor AdapterLayanan

**public** **AdapterLayanan**(Context context, List<ModelLayanan> layananList) {

this.context = context;

this.list = layananList;

}

@NonNull

@Override

**public** LayananViewHolder **onCreateViewHolder**(@NonNull ViewGroup parent, int viewType) {

// Inflate layout untuk item layanan

View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item\_layanan, parent, false);

**return** **new** **LayananViewHolder**(view);

}

@Override

**public** **void** **onBindViewHolder**(@NonNull LayananViewHolder holder, int position) {

// Menghubungkan data layanan dengan tampilan

ModelLayanan layanan = list.get(position);

holder.txtName.setText(layanan.getTipe());

holder.txtPrice.setText(String.valueOf(layanan.getHarga()));

}

@Override

**public** int **getItemCount**() {

// Mengembalikan jumlah item dalam daftar layanan

**return** list.size();

}

**public** **void** **clear**() {

int size = this.list.size();

this.list.clear();

notifyItemRangeRemoved(0, size);

}

// ViewHolder untuk AdapterLayanan

**public** **class** **LayananViewHolder** **extends** **RecyclerView**.ViewHolder {

TextView txtName, txtPrice;

**public** **LayananViewHolder**(@NonNull View itemView) {

super(itemView);

txtName = (TextView) itemView.findViewById(R.id.tvItemLayananName);

txtPrice = (TextView) itemView.findViewById(R.id.tvItemHarga);

itemView.setTag(this);

itemView.setOnClickListener(onItemClicked);

}

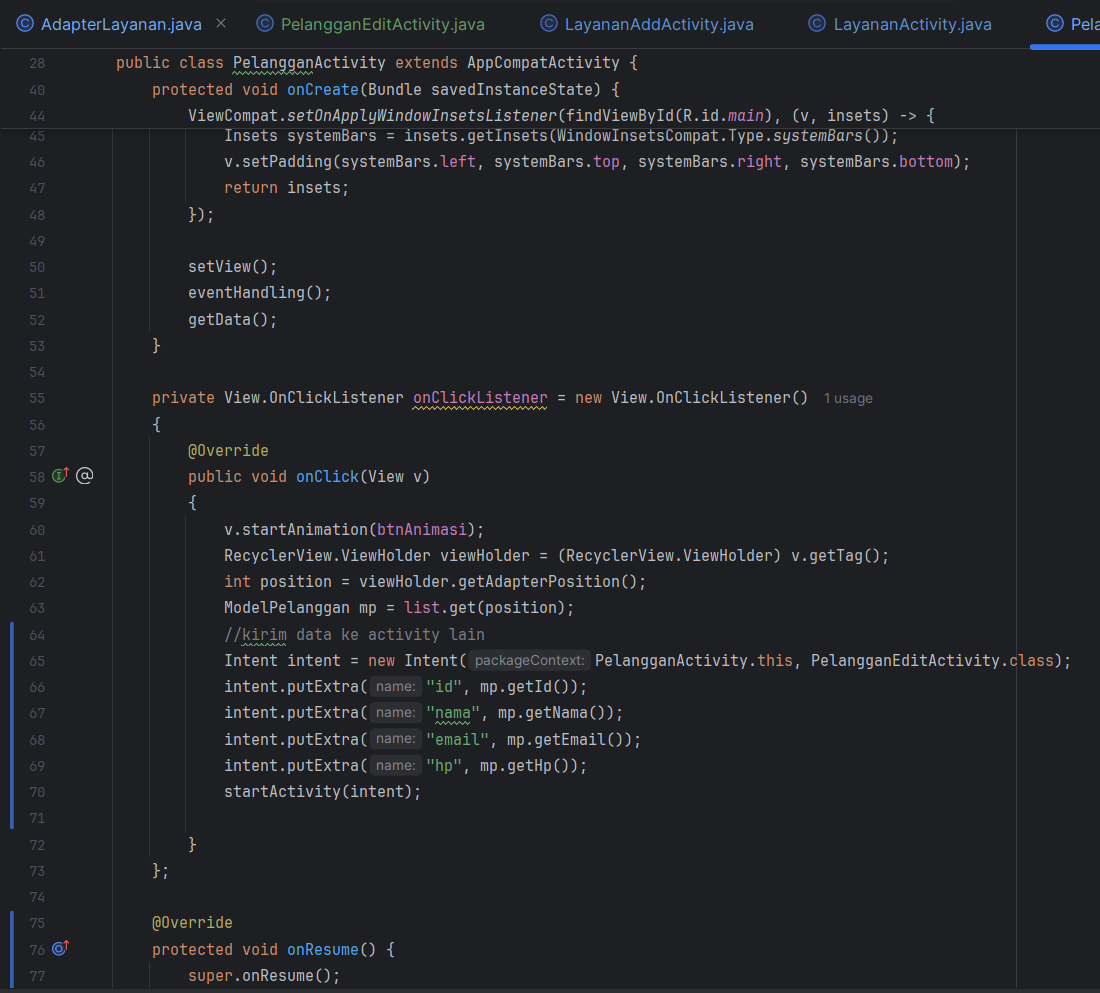
}

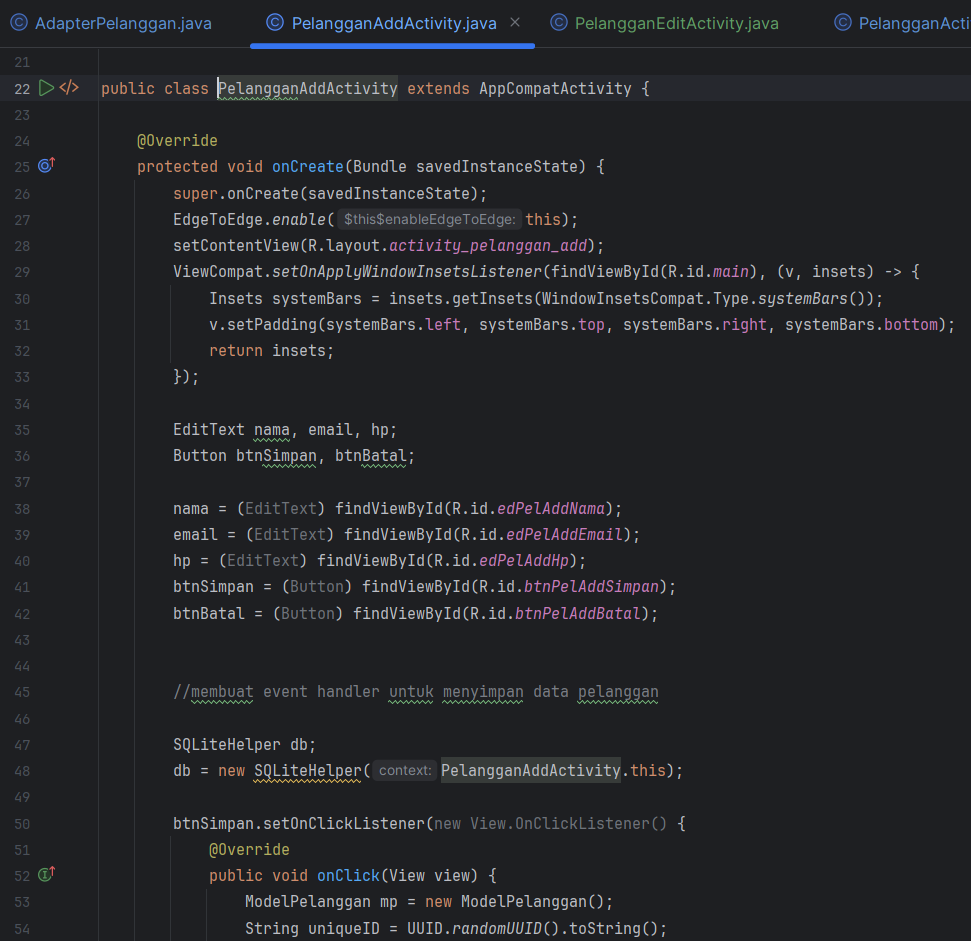
}

1. **Update Pelanggan**

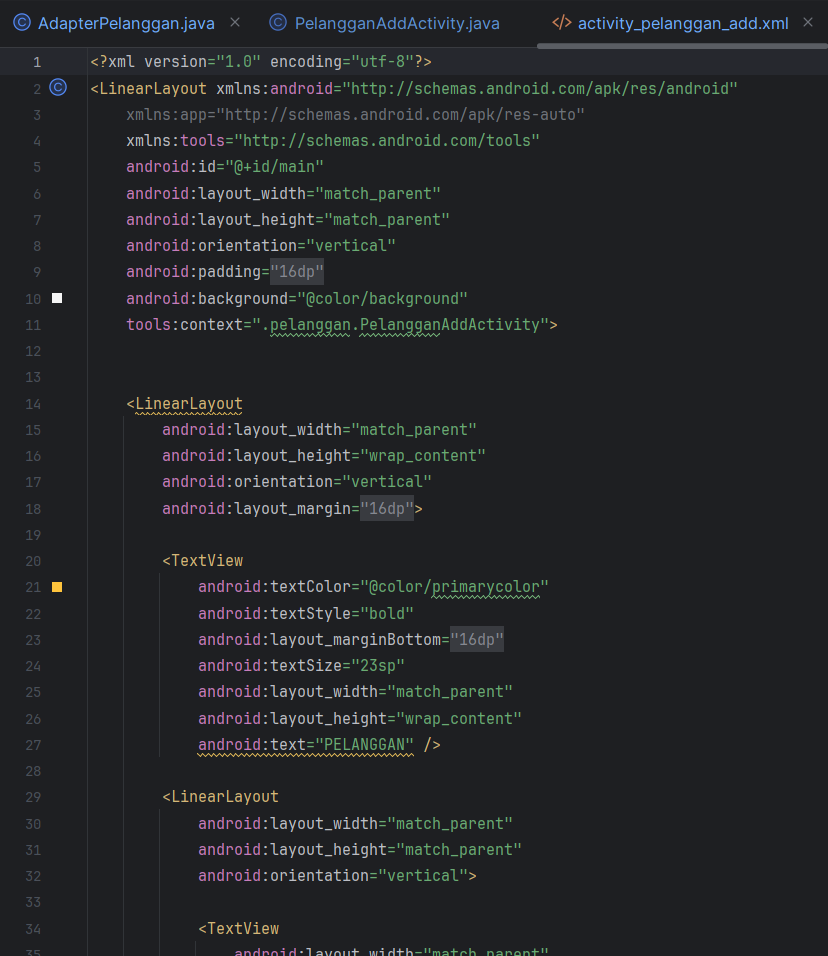
Berikut adalah update yang diberikan untuk Package Pelanggan dengan implementasi database SQLite.

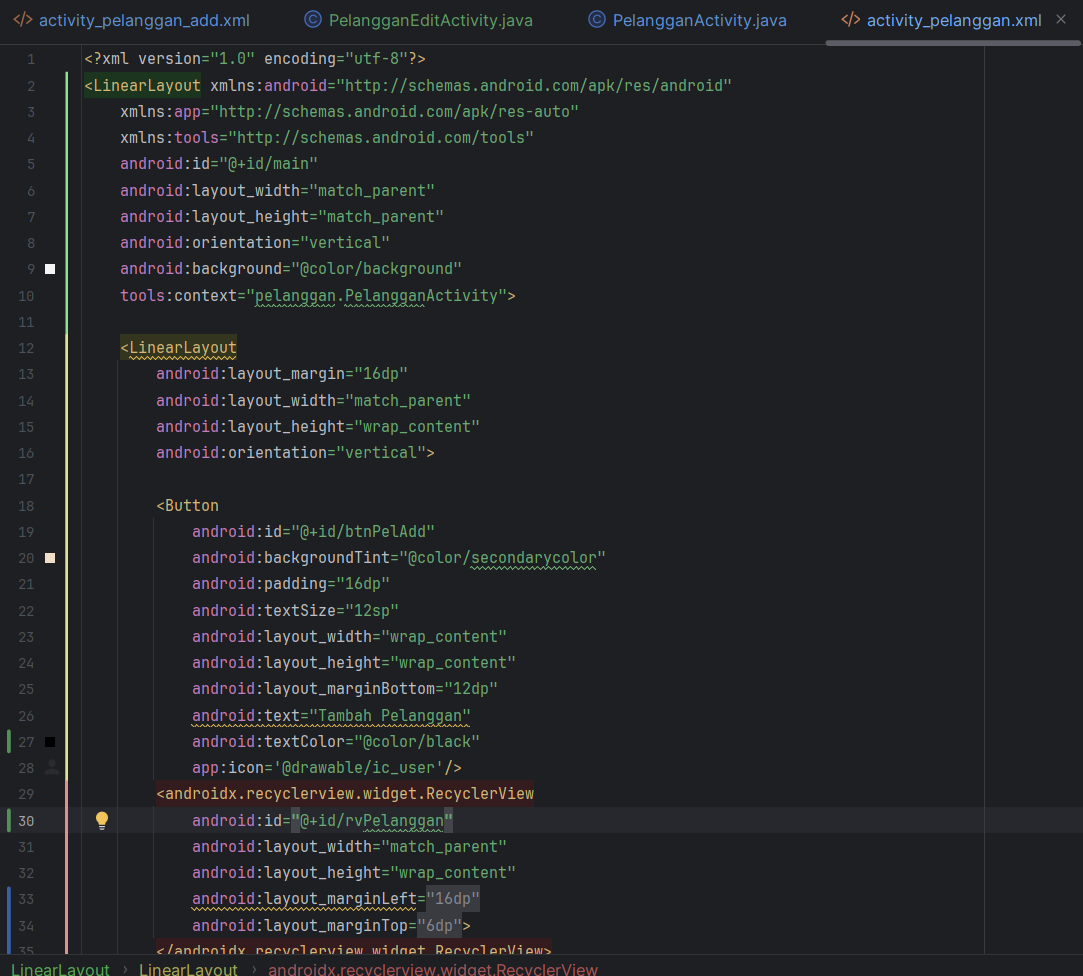
* + Java



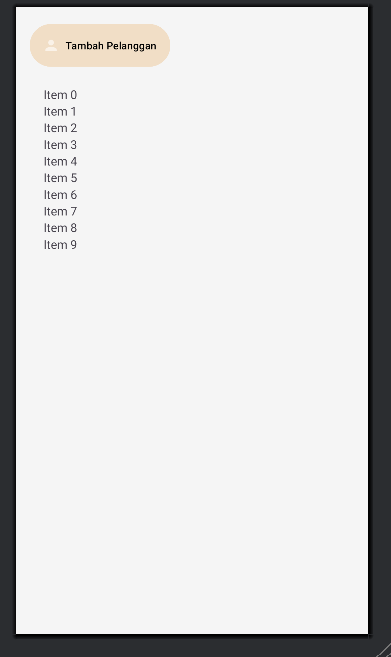
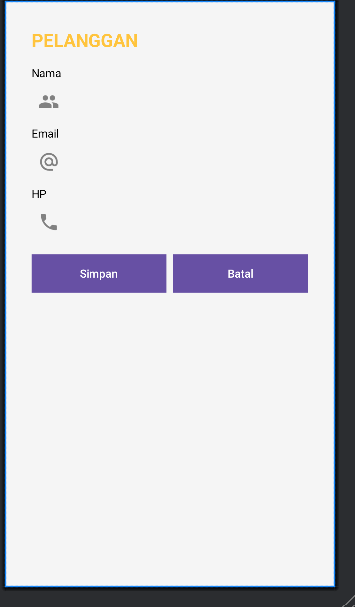


* + Xml





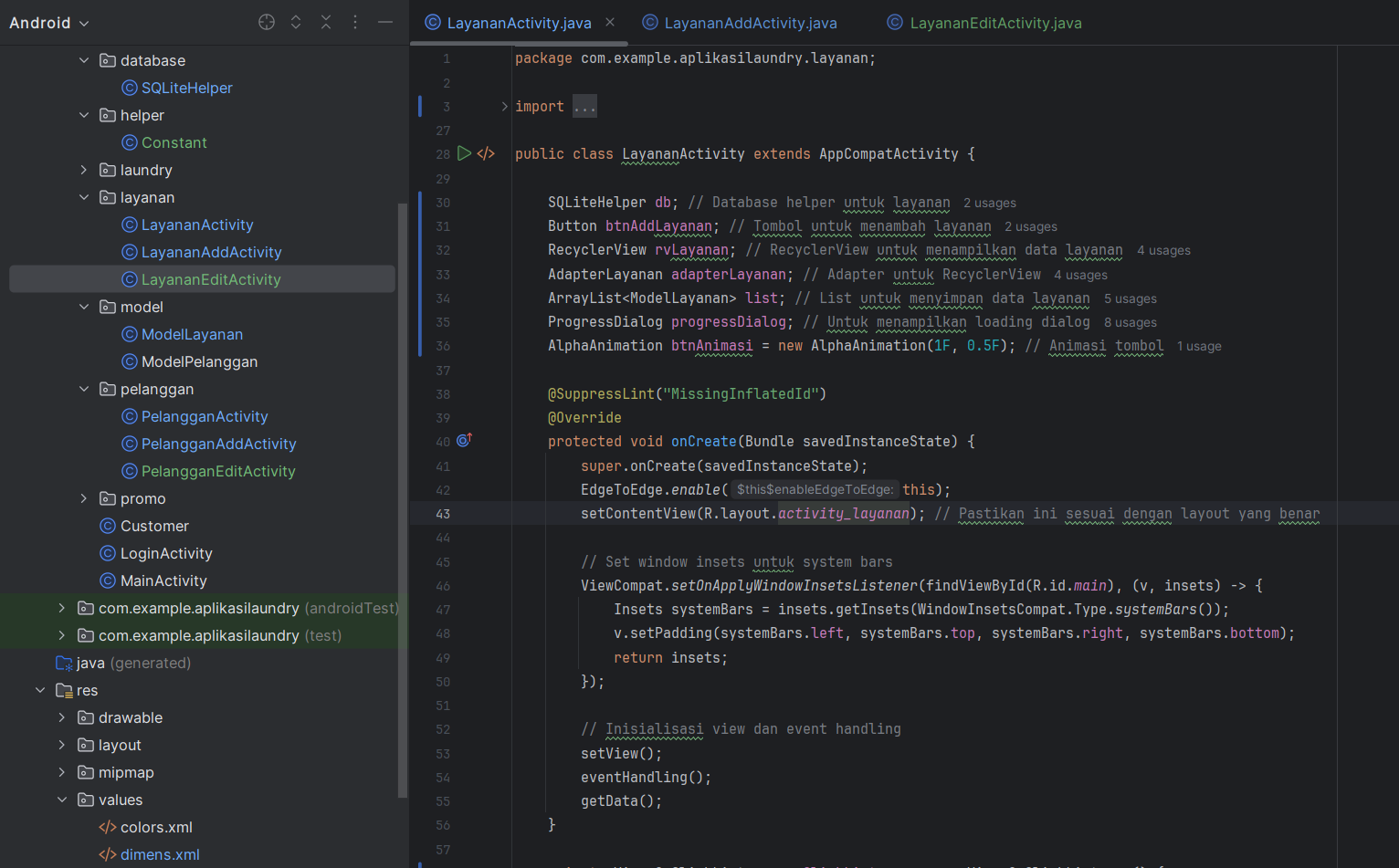
* + Tampilan

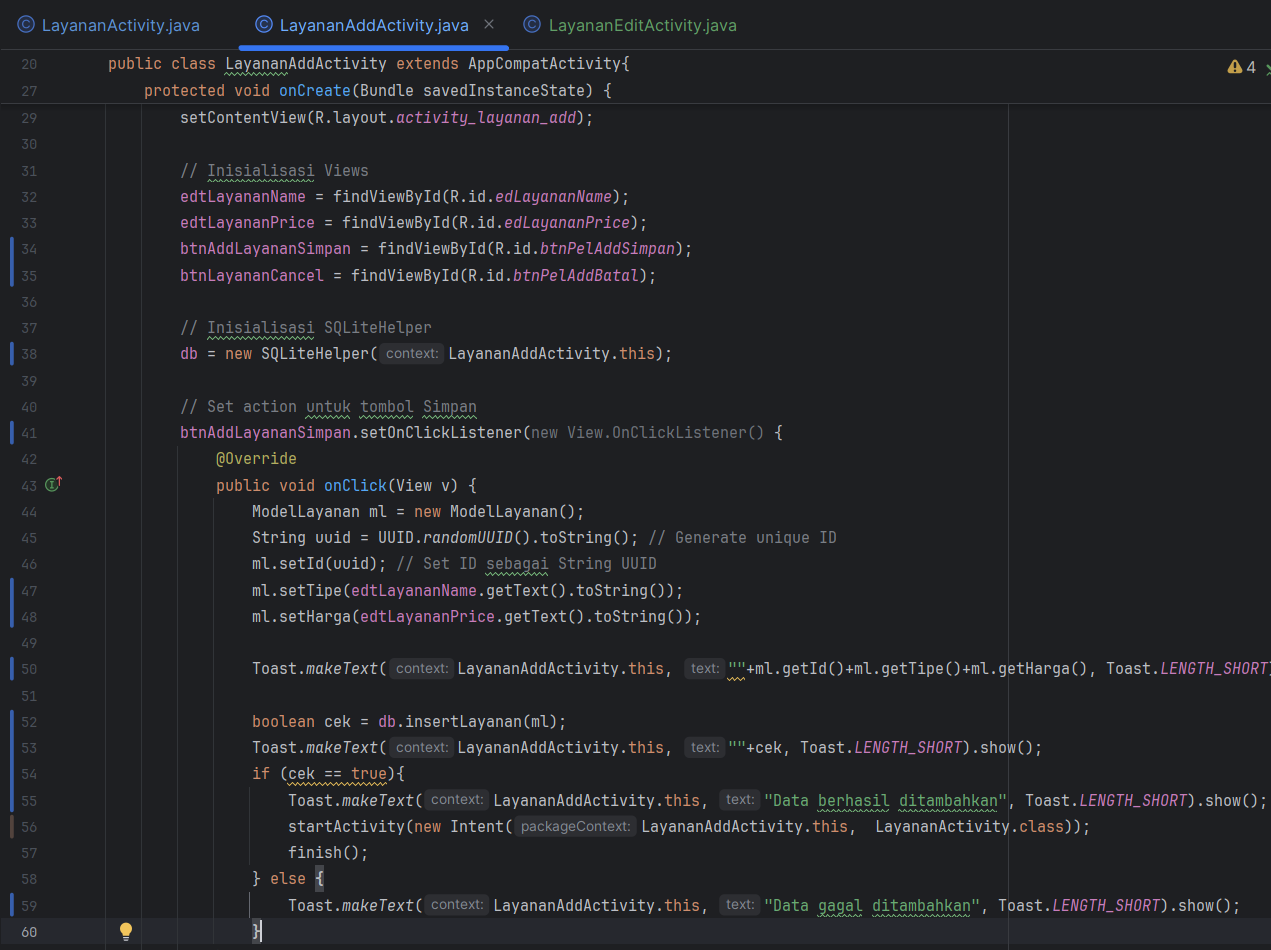
 

1. **Update Layanan**

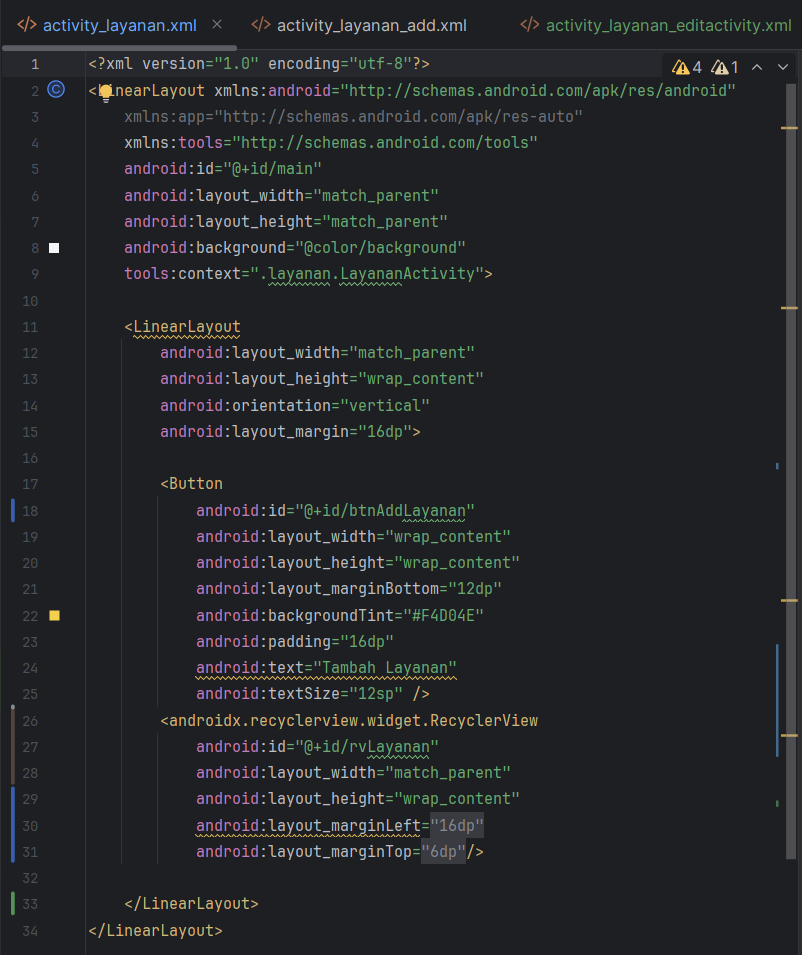
Berikut adalah update yang diberikan untuk Package Layanan dengan implementasi database SQLite.

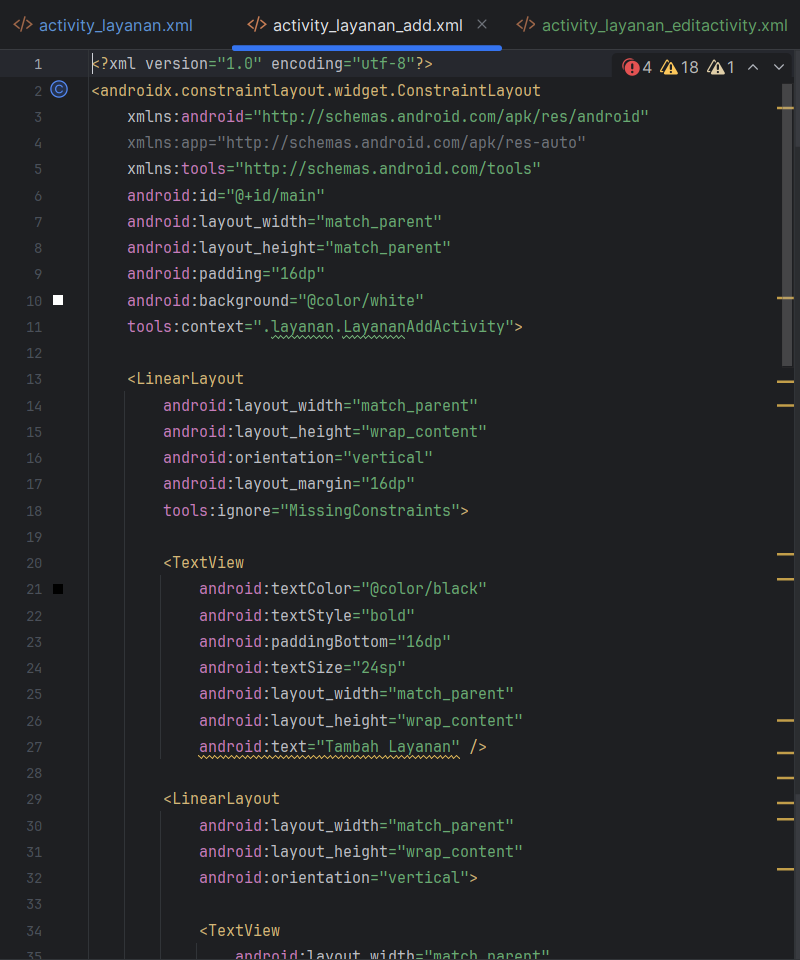
* + Java



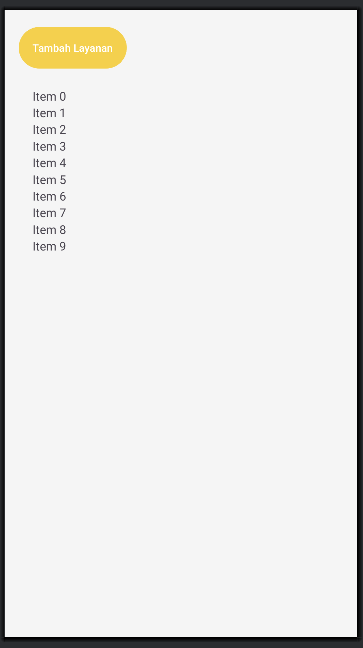
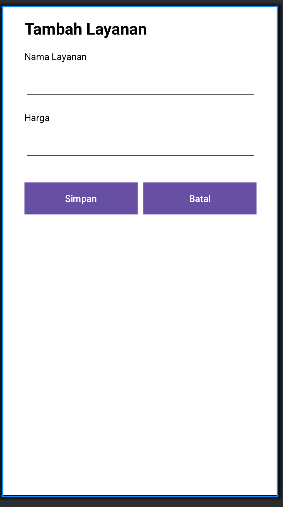


* + Xml





* + Tampilan

1. **Manifest**

File AndroidManifest .xml memberikan izin kepada aplikasi Android untuk membaca data dari penyimpanan eksternal perangkat

