**PEMROGRAMAN MOBILE**

**APLIKASI LAUNDRY**

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

****

**RAFANO ALESANDER (2211533016)**

**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

**2024**

**FUNGSI EDIT DAN DELETE**

1. **PelangganActivity**

**private** View.OnClickListener onClickListener = **new** **View**.OnClickListener()

{

@Override

**public** **void** **onClick**(View v)

{

v.startAnimation(btnAnimasi);

RecyclerView.ViewHolder viewHolder = (RecyclerView.ViewHolder) v.getTag();

int position = viewHolder.getAdapterPosition();

ModelPelanggan mp = list.get(position);

//kirim data ke activity lain

Intent intent = **new** **Intent**(PelangganActivity.this, PelangganEditActivity.class);

intent.putExtra("id", mp.getId());

intent.putExtra("nama", mp.getNama());

intent.putExtra("email", mp.getEmail());

intent.putExtra("hp", mp.getHp());

startActivity(intent);

}

};

Penambahan intent untuk mengirim data dari PelangganActivity ke PelangganEditActivity

1. **PelanggganEditActivity**

**public** **class** **PelangganEditActivity** **extends** **AppCompatActivity** {

String id, nama, email, hp;

**private** EditText edPelEditNama, edPelEditEmail, edPelEditHp;

**private** Button btnPelEditSimpan, btnPelEditHapus, btnPelEditBatal;

**private** SQLiteHelper db;

@SuppressLint("MissingInflatedId")

@Override

**protected** **void** **onCreate**(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

EdgeToEdge.enable(this);

setContentView(R.layout.activity\_pelanggan\_edit);

ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {

Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());

v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);

**return** insets;

});

db = **new** **SQLiteHelper**(this);

id = getIntent().getStringExtra("id");

nama = getIntent().getStringExtra("nama");

email = getIntent().getStringExtra("email");

hp = getIntent().getStringExtra("hp");

Toast.makeText(this, "Id: "+id+"\nNama: "+nama+"\nEmail: "+email+"\nHp: "+hp, Toast.LENGTH\_SHORT).show();

// Initialize views

edPelEditNama = findViewById(R.id.edPelEditNama);

edPelEditEmail = findViewById(R.id.edPelEditEmail);

edPelEditHp = findViewById(R.id.edPelEditHp);

btnPelEditSimpan = findViewById(R.id.btnPelEditSimpan);

btnPelEditHapus = findViewById(R.id.btnPelEditHapus);

btnPelEditBatal = findViewById(R.id.btnPelEditBatal);

// Set the existing customer data to EditText fields

edPelEditNama.setText(nama);

edPelEditEmail.setText(email);

edPelEditHp.setText(hp);

// Set listeners for buttons

btnPelEditSimpan.setOnClickListener(v -> updateCustomer());

btnPelEditHapus.setOnClickListener(v -> deleteCustomer());

btnPelEditBatal.setOnClickListener(v -> cancelEdit());

}

// Update customer in the database

**private** **void** **updateCustomer**() {

String updatedName = edPelEditNama.getText().toString();

String updatedEmail = edPelEditEmail.getText().toString();

String updatedHp = edPelEditHp.getText().toString();

// Check if fields are empty

**if** (updatedName.isEmpty() || updatedEmail.isEmpty() || updatedHp.isEmpty()) {

Toast.makeText(this, "All fields must be filled", Toast.LENGTH\_SHORT).show();

**return**;

}

// Update the customer in the database

boolean isUpdated = db.updatePelanggan(id, updatedName, updatedEmail, updatedHp);

**if** (isUpdated) {

Toast.makeText(this, "Customer updated successfully", Toast.LENGTH\_SHORT).show();

PelangganActivity pa = **new** **PelangganActivity**();

pa.getData();

finish(); // Close the activity after successful update

} **else** {

Toast.makeText(this, "Failed to update customer", Toast.LENGTH\_SHORT).show();

}

}

// Delete customer from the database

**private** **void** **deleteCustomer**() {

// Confirmation dialog before deleting

**new** **AlertDialog**.Builder(this)

.setTitle("Delete Confirmation")

.setMessage("Are you sure you want to delete this customer?")

.setPositiveButton("Yes", (dialog, which) -> {

boolean isDeleted = db.deletePelanggan(id);

**if** (isDeleted) {

Toast.makeText(this, "Customer deleted successfully", Toast.LENGTH\_SHORT).show();

Intent intent = **new** **Intent**(this, PelangganActivity.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(intent);

finish(); // Close the current activity after deletion

} **else** {

Toast.makeText(this, "Failed to delete customer", Toast.LENGTH\_SHORT).show();

}

})

.setNegativeButton("No", null)

.show();

}

// Cancel edit and go back to the previous screen

**private** **void** **cancelEdit**() {

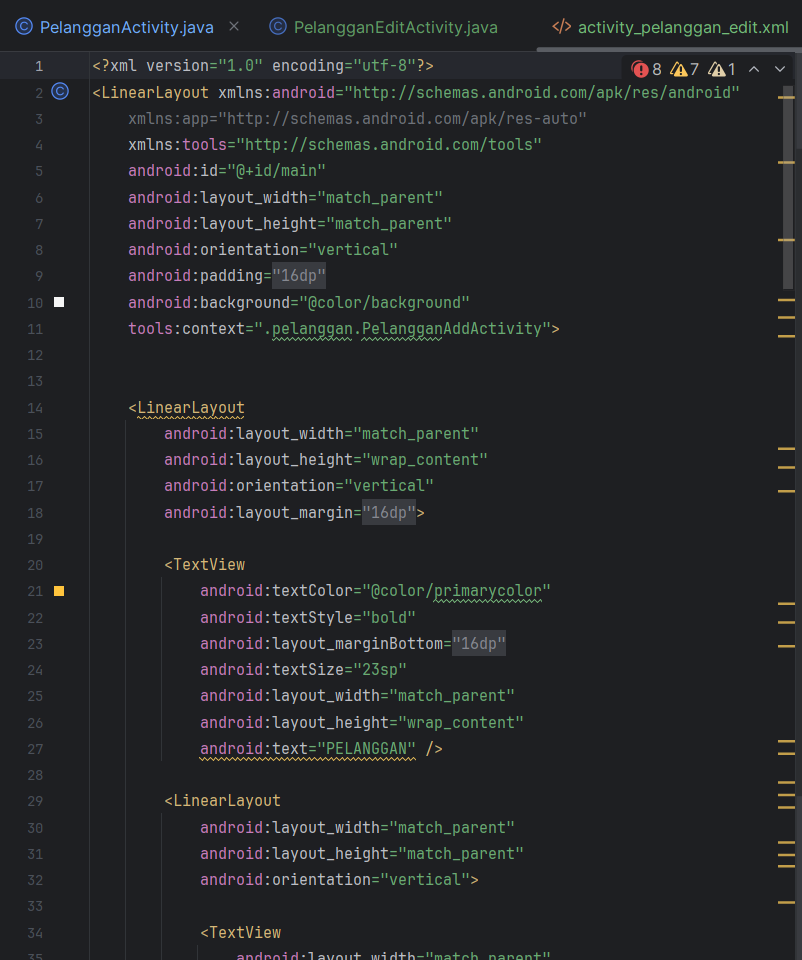
finish(); // Close the activity without making any changes

}

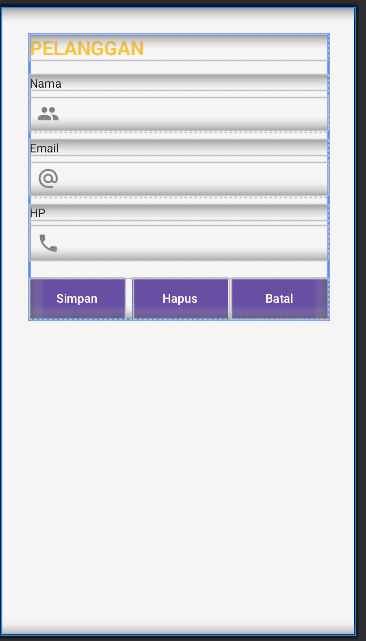
}

code pada class PelangganEditActivity bertujuan untuk mengelola data pelanggan, khususnya dalam mengedit dan menghapus informasi pelanggan yang tersimpan pada database SQLite. Pada aktivitas ini, data pelanggan (nama, email, dan nomor hp) diterima dari PelangganActivity dan ditampilkan dalam kolom input untuk diedit. Pengguna dapat memperbarui data dengan mengetikkan informasi baru dan menekan tombol "Simpan", yang kemudian akan memvalidasi apakah semua data telah diisi. Jika valid, data pelanggan diperbarui di database, dan pengguna kembali ke halaman utama pelanggan. Tombol "Hapus" akan menghapus data pelanggan. Jika pengguna mengonfirmasi dari database, dan PelangganActivity diperbarui untuk merefleksikan perubahan ini. Tombol "Batal" akan membatalkan pengeditan dan menutup halaman ini tanpa menyimpan perubahan.

* XML



* Tampilan



1. **LayananActivity**

**private** View.OnClickListener onClickListener = **new** **View**.OnClickListener() {

@Override

**public** **void** **onClick**(View v) {

v.startAnimation(btnAnimasi); // Menjalankan animasi tombol

RecyclerView.ViewHolder viewHolder = (RecyclerView.ViewHolder) v.getTag();

int position = viewHolder.getAdapterPosition();

ModelLayanan ml = list.get(position); // Ambil data pelanggan dari posisi yang diklik

// kirim data ke activity lain

Intent intent= **new** **Intent**(LayananActivity.this, LayananEditActivity.class);

intent.putExtra("id", ml.getId());

intent.putExtra("tipe", ml.getTipe());

intent.putExtra("harga", ml.getHarga());

startActivity(intent);

}

};

Penambahan intent untuk mengirim data dari PelangganActivity ke PelangganEditActivity

1. **LayananEditActivity**

**public** **class** **LayananEditActivity** **extends** **AppCompatActivity** {

String id, layanan, harga;

**private** EditText edLayEditLayanan, edLayEditHarga;

**private** Button btnLayEditSimpan, btnLayEditHapus, btnLayEditBatal;

**private** SQLiteHelper db;

@SuppressLint("MissingInflatedId")

@Override

**protected** **void** **onCreate**(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

EdgeToEdge.enable(this);

setContentView(R.layout.activity\_layanan\_editactivity);

ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {

Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());

v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);

**return** insets;

});

db = **new** **SQLiteHelper**(this);

id = getIntent().getStringExtra("id");

layanan = getIntent().getStringExtra("layanan");

harga = getIntent().getStringExtra("harga");

Toast.makeText(this, "Id: " + id + "\nLayanan: " + layanan + "\nHarga: " + harga, Toast.LENGTH\_SHORT).show();

// Initialize views

edLayEditLayanan = findViewById(R.id.edLayEditLayanan);

edLayEditHarga = findViewById(R.id.edLayEditHarga);

btnLayEditSimpan = findViewById(R.id.btnLayEditSimpan);

btnLayEditHapus = findViewById(R.id.btnLayEditHapus);

btnLayEditBatal = findViewById(R.id.btnLayEditBatal);

// Set the existing customer data to EditText fields

edLayEditLayanan.setText(layanan);

edLayEditHarga.setText(harga);

// Set listeners for buttons

btnLayEditSimpan.setOnClickListener(v -> updateLayanan());

btnLayEditHapus.setOnClickListener(v -> deleteLayanan());

btnLayEditBatal.setOnClickListener(v -> cancelEdit());

}

// Update customer in the database

**private** **void** **updateLayanan**() {

String updatedLayanan = edLayEditLayanan.getText().toString();

String updatedHarga = edLayEditHarga.getText().toString();

// Check if fields are empty

**if** (updatedLayanan.isEmpty() || updatedHarga.isEmpty()) {

Toast.makeText(this, "All fields must be filled", Toast.LENGTH\_SHORT).show();

**return**;

}

// Update the customer in the database

boolean isUpdated = db.updateLayanan(id, updatedLayanan, updatedHarga);

**if** (isUpdated) {

Toast.makeText(this, "Layanan updated successfully", Toast.LENGTH\_SHORT).show();

LayananActivity pa = **new** **LayananActivity**();

pa.getData();

finish(); // Close the activity after successful update

} **else** {

Toast.makeText(this, "Failed to update layanan", Toast.LENGTH\_SHORT).show();

}

}

// Delete customer from the database

**private** **void** **deleteLayanan**() {

// Confirmation dialog before deleting

**new** **AlertDialog**.Builder(this)

.setTitle("Delete Confirmation")

.setMessage("Are you sure you want to delete this customer?")

.setPositiveButton("Yes", (dialog, which) -> {

boolean isDeleted = db.deletePelanggan(id);

**if** (isDeleted) {

Toast.makeText(this, "Customer deleted successfully", Toast.LENGTH\_SHORT).show();

Intent intent = **new** **Intent**(this, PelangganActivity.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(intent);

finish(); // Close the current activity after deletion

} **else** {

Toast.makeText(this, "Failed to delete customer", Toast.LENGTH\_SHORT).show();

}

})

.setNegativeButton("No", null)

.show();

}

// Cancel edit and go back to the previous screen

**private** **void** **cancelEdit**() {

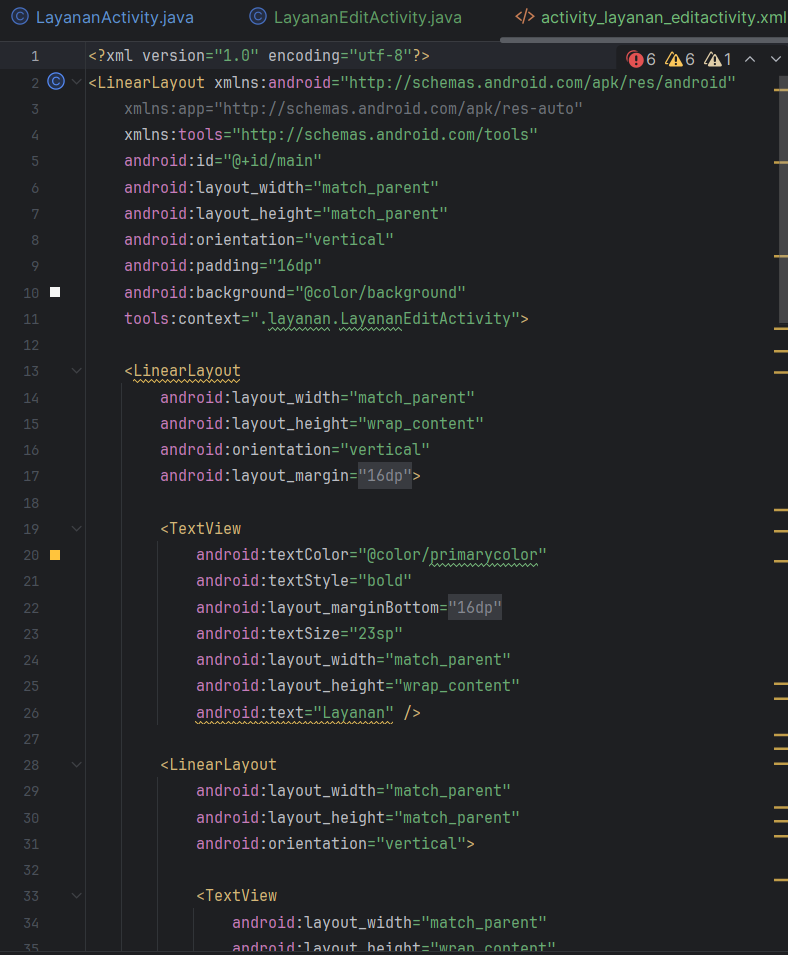
finish(); // Close the activity without making any changes

}

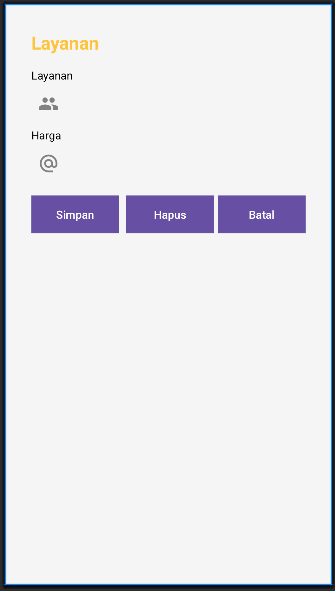
}

Kode pada class LayananEditActivity ini berfungsi untuk mengedit atau menghapus data layanan di dalam database lokal menggunakan SQLite. Pada onCreate(), diterapkan penyesuaian antarmuka dan tombol "Simpan," "Hapus," dan "Batal". Tombol "Simpan" memanggil metode updateLayanan(), yang memvalidasi input dan mengupdate data layanan jika semua kolom terisi hingga mengembalikan halaman ke LayananActivity untuk memperbarui tampilan. Tombol "Hapus" akan menghapus data layanan dan LayananActivity terbuka kembali untuk mencerminkan perubahan. Tombol "Batal" digunakan untuk menutup halaman tanpa menyimpan perubahan.

* XML



* Tampilan



1. **SQLiteHelper**

// 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 **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;

}

}