

TUGAS PERTEMUAN 4
PRAKTIKUM APP MOBILE

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



Disusun Oleh :

Yusuf Fatha Mubina Alexander (2211533017)

Departemen Informatika
Fakultas Teknologi Informasi
Universitas Andalas
2024

A. PelangganActivity

- Java

Pada PelangganActivity.java kita menambahkan intent pada method onClickListener sehingga setelah kita mengklik pelanggan pada recycler view akan dialihkan ke Pelanggan Edit sehingga dapat melakukan update maupun delete dari data pelanggan.

```
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);
            Toast.makeText(PelangganActivity.this, "" +mp.getNama(),
Toast.LENGTH_SHORT).show();
        }
    };
```

B. PelangganEditActivity

- Java

Masukkan semua library yang akan dipakai untuk pembuatan activity edit pelanggan. Inisialisasi String id, nama, email, hp yang dimana datanya diambil dari pelangganActivity.

```
package com.example.laundryapps.pelanggan;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import com.example.laundryapps.R;
import com.example.laundryapps.database.SQLiteHelper;
import com.example.laundryapps.model.ModelPelanggan;

import java.util.UUID;

public class PelangganEditActivity extends AppCompatActivity {
    EditText etnama, etemail, ethp;
    Button btnSimpan, btnBatal, btnEdit;
    SQLiteHelper db;
    String id, nama, hp, email;
```

```

@SuppressLint("MissingInflatedId")
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable(this);
    setContentView(R.layout.activity_pelanggan_edit);

    etnama = (EditText) findViewById(R.id.edPelNama);
    etemail = (EditText) findViewById(R.id.edPelEmail);
    ethp = (EditText) findViewById(R.id.edPelAddHp);
    btnBatal = (Button) findViewById(R.id.btnPelAddBatal);
    btnEdit = (Button) findViewById(R.id.btnPelAddEdit);

    db = new SQLiteHelper(PelangganEditActivity.this);

    // menerima data mp dari Pelanggan_RecyclerViewAdapter
    id = getIntent().getStringExtra("id");
    nama = getIntent().getStringExtra("nama");
    email = getIntent().getStringExtra("email");
    hp = getIntent().getStringExtra("hp");

    /* supaya editText nya tidak kosong
     * atau apakah sebaiknya dibiarkan kosong?
     * */
    etnama.setText(nama);
    etemail.setText(email);
    ethp.setText(hp);

```

Mendeklarasi semua komponen, seperti EditText, Button, serta variabel-variabel data, yaitu nama, email, hp ke dalam adapterpelanggan pada recycleview. Lalu menambahkan setText pada 3 editText agar memberitahukan kepada pengguna data apa yang perlu dimasukkan.

```

btnEdit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        ModelPelanggan mp = new ModelPelanggan();
        String uniqueID = UUID.randomUUID().toString();
        mp.setNama(etnama.getText().toString());
        mp.setEmail(etemail.getText().toString());
        mp.setHp(ethp.getText().toString());

        Toast.makeText(PelangganEditActivity.this, "Nama: " + mp.getNama()
+ " Email: " + mp.getEmail() + " Hp: " + mp.getHp(),
Toast.LENGTH_SHORT).show();

        boolean cek = db.editPelanggan(mp, id);
        if (cek) {
            Toast.makeText(PelangganEditActivity.this, "Data berhasil
diubah", Toast.LENGTH_SHORT).show();
            startActivity(new Intent(PelangganEditActivity.this,
PelangganActivity.class));
            finish();
        } else {
            Toast.makeText(PelangganEditActivity.this, "Data gagal diubah",

```

```
Toast.LENGTH_SHORT).show();
    }
}
});
```

btnEdit disini memiliki fungsi untuk mengubah data yang sudah ada. Data yang bisa diganti seperti yang sudah terdapat pada model pelanggan, yaitu nama, email, hp. Kemudian setelah berhasil masukkan semua datanya, akan dikembalikan ke halaman atau activity pelanggan.

```
btnBatal.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});

}
```


Terakhir, untuk komponen btnBatal, ia memiliki fungsi untuk membatalkan opsi jika tidak jadi untuk mengubah data tersebut.

Output :


Untuk tampilan akan seperti ini

EDIT PELANGGAN


Nama



Email



HP



Edit Batal

C. LayananActivity

Pada layanan Activity sama hal nya dengan pelanggan Activity

- Java

```
public class LayananActivity extends AppCompatActivity {

    private SQLiteHelper2 db; // Database helper untuk layanan
    private Button btnAddLayanan; // Tombol untuk menambah
    layanan
    private RecyclerView rvLayanan; // RecyclerView untuk
    menampilkan data layanan
    private AdapterLayanan adapterLayanan; // Adapter untuk
    RecyclerView
    private ArrayList<ModelLayanan> list; // List untuk
    menyimpan data layanan
    private ProgressDialog progressDialog; // Untuk menampilkan
    loading dialog
    private AlphaAnimation btnAnimasi = new AlphaAnimation(1F,
    0.5F); // Animasi tombol

    @SuppressWarnings("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_layanan); // Pastikan
        ini sesuai dengan layout yang benar

        // Set window insets untuk system bars

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.mai
        n), (v, insets) -> {
            Insets systemBars =
            insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top,
            systemBars.right, systemBars.bottom);
            return insets;
        });

        // Inisialisasi view dan event handling
        setContentView();
        eventHandling();
        getData();
    }

    private void eventHandling() {
        btnAddLayanan.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(LayananActivity.this,
                LayananAddActivity.class));
            }
        });
    }

    private void getData() {
        list.clear(); // Bersihkan list sebelum mengambil data
        baru
        showMsg(); // Tampilkan loading dialog
    }
}
```

```

        try {
            List<ModelLayanan> p = db.getAllServices(); //
            Ambil data dari database
            if (p != null && p.size() > 0) {
                list.addAll(p); // Tambahkan data ke list
                adapterLayanan.notifyDataSetChanged(); //
                Beritahu adapter bahwa ada data baru
            } else {
                Toast.makeText(this, "Data tidak ditemukan",
                    Toast.LENGTH_SHORT).show();
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            progressDialog.dismiss(); // Sembunyikan loading
            dialog
        }
    }

    private void setView() {
        db = new SQLiteHelper2(this); // Inisialisasi
        SQLiteHelper untuk layanan
        progressDialog = new ProgressDialog(this); //
        Inisialisasi ProgressDialog
        btnAddLayanan = findViewById(R.id.btnPLAdd); // Temukan
        tombol tambah layanan dari layout
        rvLayanan = findViewById(R.id.rvLayanan); // Temukan
        RecyclerView dari layout
        list = new ArrayList<>(); // Inisialisasi list untuk
        menampung data layanan

        // Inisialisasi LinearLayoutManager untuk RecyclerView
        LinearLayoutManager llm = new
        LinearLayoutManager(this);
        rvLayanan.setLayoutManager(llm); // Set layout manager
        ke RecyclerView
        rvLayanan.setHasFixedSize(true); // Optimalkan ukuran
        RecyclerView

        // Inisialisasi adapter untuk RecyclerView
        adapterLayanan = new AdapterLayanan(this, list);
        rvLayanan.setAdapter(adapterLayanan); // Set adapter ke
        RecyclerView
    }

    private void showMsg() {
        if (progressDialog == null) {
            progressDialog = new ProgressDialog(this);
            progressDialog.setTitle("Informasi");
            progressDialog.setMessage("Loading Data...");
            progressDialog.setCancelable(false); // Tidak bisa
            dibatalkan dengan menekan di luar
        }
        progressDialog.show(); // Tampilkan dialog
    }
}

```

D. LayananAddActivity

- Java

Hal yang pertama dibuat disini adalah memasukkan library baik secara komponen, seperti widget, button, edittext maupun menghubungkan file java lainnya. Setelah itu buat deklarasi variabel EditText, Button, dan file databasenya.

```
• package com.example.laundryapps.layanan;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.laundryapps.R;
import com.example.laundryapps.database.SQLiteHelper2;
import com.example.laundryapps.model.ModelLayanan;

import java.util.UUID;

public class LayananAddActivity extends AppCompatActivity {

    EditText edtLayananName, edtLayananPrice; // Variabel untuk
    nama dan harga layanan
    Button btnAddLayanan, btnCancel;
    SQLiteHelper2 db;
```

Buat inisialisai terhadap semua nama variabel yang telah di deklarasi sebelumnya untuk menyesuaikan id yang ada pada file xml nya. Terlebih lagi mendeklarasi lagi untuk nama variabelnya.

```
@SuppressWarnings("MissingInflatedId")
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_layanan_add);

    // Inisialisasi Views
    edtLayananName = findViewById(R.id.edLayananName);
    edtLayananPrice = findViewById(R.id.edLayananPrice);
    btnAddLayanan = findViewById(R.id.btnPelAddSimpan);
    btnCancel = findViewById(R.id.btnPelAddBatal);

    // Inisialisasi SQLiteHelper
    db = new SQLiteHelper2(LayananAddActivity.this);
```

Mengatur aksi agar tombol simpan berfungsi dengan benar. Tombol ini akan generate ID untuk nama layanan yang akan ditambah serta atur harga yang tepat.

```
// Set action untuk tombol Simpan
btnAddLayanan.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
```

```

        Modellayanan ml = new Modellayanan();
        String uuid = UUID.randomUUID().toString(); // Generate unique ID
        ml.setId(uuid); // Set ID sebagai String UUID
        ml.setName(edtLayananName.getText().toString());

        // Parsing harga ke integer
        String priceText = edtLayananPrice.getText().toString();
        int price;

        try {
            price = Integer.parseInt(priceText); // Mengonversi input
menjadi integer
        } catch (NumberFormatException e) {
            Toast.makeText(LayananAddActivity.this, "Harga tidak valid",
Toast.LENGTH_SHORT).show();
            return; // Menghentikan eksekusi jika harga tidak valid
        }

        ml.setPrice(price); // Mengatur harga ke model layanan

        // Menyimpan ke database dan memeriksa apakah berhasil
        boolean cek = db.insertService(ml.getId(), ml.getName(),
ml.getPrice());
        if (cek) {
            Toast.makeText(LayananAddActivity.this, "Data berhasil
disimpan", Toast.LENGTH_SHORT).show();
            startActivity(new Intent(LayananAddActivity.this,
LayananActivity.class));
            finish();
        } else {
            Toast.makeText(LayananAddActivity.this, "Data gagal disimpan",
Toast.LENGTH_SHORT).show();
        }
    }
});

```

Setelah itu buat kondisi jika berhasil tersimpan di database

```

// Menyimpan ke database dan memeriksa apakah berhasil
        boolean cek = db.insertService(ml.getId(), ml.getName(),
ml.getPrice());
        if (cek) {
            Toast.makeText(LayananAddActivity.this, "Data berhasil
disimpan", Toast.LENGTH_SHORT).show();
            startActivity(new Intent(LayananAddActivity.this,
LayananActivity.class));
            finish();
        } else {
            Toast.makeText(LayananAddActivity.this, "Data gagal disimpan",
Toast.LENGTH_SHORT).show();
        }
    }
});

```

Terakhir membuat fungsi pada tombol batal dengan menutup activity layanan jika pengguna ingin membatalkan kegiatan penambahan layanan tersebut.

```

// Fungsi untuk tombol Batal
        btnCancel.setOnClickListener(new View.OnClickListener() {
            @Override

```



```
        public void onClick(View v) {  
            finish(); // Menutup activity jika tombol dibatalkan  
        }  
    });  
}  
}
```

Output :

Tambah Layanan

Nama Layanan

Harga

Simpan

Batal

E. SQLiteHelper

- Pelanggan
Data setelah di insert akan dikirimkan ke database dan dikirimkan kembali ke pelanggan activity.

```

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;
    }
}

```

- Layanan
Sama halnya dengan data pelanggan.

```

public boolean insertService(String id, String name, double price) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(KEY_SERVICE_ID, id); // Menyimpan ID layanan
    values.put(KEY_SERVICE_NAME, name);
    values.put(KEY_SERVICE_PRICE, price);

    long rowId = db.insert(TABLE_SERVICE, null, values);
    db.close();
    return rowId != -1; // Mengembalikan true jika insert berhasil
}

```