

MODUL PRAKTIKUM PEMOGRAMAN BERGERAK

PEMBELAJARAN DARING KOLABORATIF (PDK)



TIM Penyusun :

KRISTINA ANNATASIA BR SITEPU
HERU SAPUTRA

PERTEMUAN 11

PENYIMPANAN DATA

11.1 TUJUAN PEMBELAJARAN :

- Mahasiswa dapat memahami jenis penyimpanan data dalam pengembangan aplikasi Android.
- Mahasiswa dapat memahami kelebihan dan kekurangan dari masing-masing jenis penyimpanan data.
- Mahasiswa mampu memilih jenis penyimpanan data yang tepat sesuai dengan kebutuhan aplikasi Android yang dikembangkan.
- Mahasiswa mampu menerapkan penyimpanan data dalam aplikasi Android.

11.2 ALAT DAN BAHAN :

- a. Laptop/PC
- b. Android Studio
- c. Sistem Operasi Windows

11.3 MATERI

1. Penyimpanan Data

Penyimpanan data pada aplikasi Android sangat penting karena memungkinkan aplikasi untuk menyimpan informasi secara permanen atau sementara. Penyimpanan ini memungkinkan aplikasi untuk mempertahankan keadaan dan informasi penting yang digunakan untuk meningkatkan pengalaman pengguna. Selain itu, penyimpanan data juga berfungsi untuk memastikan bahwa data tetap tersedia meskipun aplikasi ditutup atau perangkat dimulai ulang. Pemilihan media penyimpanan yang tepat sangat mempengaruhi kinerja aplikasi dan pengalaman pengguna secara keseluruhan.

Pemilihan media penyimpanan yang tepat dapat mempengaruhi kinerja dan pengalaman pengguna. Ada beberapa jenis media penyimpanan data di Android, masing-masing dengan kelebihan dan kekurangannya.

Macam-Macam Penyimpanan Data

1. Shared Preferences

Deskripsi: Digunakan untuk menyimpan data dalam bentuk pasangan kunci-nilai (key-value pairs). Biasanya digunakan untuk preferensi atau pengaturan pengguna.

Kelebihan:

- Mudah digunakan untuk data sederhana.
- Data disimpan dalam format yang ringan dan cepat diakses.
- Tidak memerlukan banyak ruang penyimpanan.

Kekurangan:

- Hanya cocok untuk data sederhana (misalnya: preferensi pengguna, status login).
- Tidak cocok untuk menyimpan data dalam jumlah besar atau data yang kompleks.

2. SQLite Database

Deskripsi: SQLite adalah database lokal yang memungkinkan penyimpanan data terstruktur dalam bentuk tabel. Cocok untuk aplikasi yang memerlukan penyimpanan data dalam jumlah besar atau yang membutuhkan query kompleks.

Kelebihan:

- Mendukung penyimpanan data dalam jumlah besar dan terstruktur.
- Mendukung operasi query SQL yang kompleks.
- Aplikasi dapat mengakses database secara langsung tanpa membutuhkan server.

Kekurangan:

- Membutuhkan pengelolaan yang lebih rumit dibandingkan Shared Preferences.
- Tidak cocok untuk data yang tidak terstruktur atau data dalam jumlah kecil.

3. MySQL Database

Deskripsi: MySQL adalah database server berbasis cloud yang digunakan untuk aplikasi yang memerlukan penyimpanan data terpusat dan akses dari beberapa perangkat.

Kelebihan:

- Dapat digunakan untuk aplikasi dengan kebutuhan penyimpanan data yang besar dan terdistribusi.
- Mendukung query SQL yang kompleks dan dapat diakses oleh banyak pengguna di berbagai perangkat.
- Dapat diintegrasikan dengan server untuk data yang bisa diakses dari banyak perangkat.

Kekurangan:

- Memerlukan koneksi internet dan pengelolaan server.
- Tidak cocok untuk aplikasi yang membutuhkan penyimpanan lokal atau aplikasi offline.

4. Internal Storage

Deskripsi: Digunakan untuk menyimpan data di dalam perangkat secara langsung. Hanya aplikasi yang memiliki izin yang dapat mengakses data ini.

Kelebihan:

- Data aman dan tidak bisa diakses aplikasi lain.
- Tidak memerlukan izin akses dari pengguna.

Kekurangan:

- Ruang penyimpanan terbatas.

5. External Storage

Deskripsi: Memungkinkan aplikasi untuk menyimpan data di penyimpanan eksternal perangkat (misalnya, kartu SD atau penyimpanan yang lebih besar).

Kelebihan:

- Kapasitas penyimpanan yang lebih besar.
- Dapat diakses oleh aplikasi lain dengan izin pengguna.

Kekurangan:

- Membutuhkan izin akses dari pengguna.
- Data dapat lebih rentan terhadap penghapusan atau kerusakan.
- Memilih Media Penyimpanan yang Tepat untuk Aplikasi yang Dibuat

Pemilihan media penyimpanan yang tepat sangat bergantung pada jenis dan kebutuhan aplikasi yang akan dibuat. Berikut adalah beberapa panduan:

- ✓ Jika aplikasi hanya membutuhkan penyimpanan pengaturan pengguna atau data sederhana (misalnya: preferensi pengguna, status login), Shared Preferences adalah pilihan terbaik.
- ✓ Jika aplikasi membutuhkan penyimpanan data yang terstruktur dan mendukung query kompleks, seperti daftar produk atau data pengguna dalam bentuk tabel, maka SQLite Database lebih sesuai.
- ✓ Jika aplikasi membutuhkan penyimpanan data yang lebih besar dan terdistribusi, seperti data yang disinkronkan dengan server, MySQL (atau database server lainnya) adalah pilihan yang lebih baik.
- ✓ Jika aplikasi menyimpan data yang lebih besar, seperti gambar atau video, dan ingin data tersebut dapat diakses oleh aplikasi lain, maka External Storage lebih tepat.
- ✓ Untuk data pribadi yang hanya digunakan oleh aplikasi itu sendiri, Internal Storage adalah pilihan yang baik karena lebih aman.

Pengenalan Media Penyimpanan

a. Shared Preferences

Shared Preferences adalah cara penyimpanan data berupa pasangan kunci-nilai yang sederhana dan efisien. Biasanya digunakan untuk menyimpan pengaturan aplikasi atau preferensi pengguna. Data disimpan dalam file XML yang dapat diakses dengan cepat.

Contoh Penggunaan:

```
// Menyimpan data
SharedPreferences sharedPref = getSharedPreferences("userPrefs", MODE_PRIVATE);
SharedPreferences.Editor editor = sharedPref.edit();
editor.putString("username", "user123");
editor.putInt("age", 30);
editor.apply();

// Mengambil data
String username = sharedPref.getString("username", "default");
int age = sharedPref.getInt("age", 0);
```

b. SQLite

SQLite adalah database lokal yang disimpan dalam perangkat. SQLite cocok untuk aplikasi yang membutuhkan penyimpanan data terstruktur dalam jumlah besar, seperti daftar produk, data pengguna, atau catatan. SQLite mendukung query SQL untuk melakukan operasi seperti pencarian, penyortiran, dan pembaruan data.

Contoh Penggunaan:

```
// Membuat atau membuka database
SQLiteDatabase db = openOrCreateDatabase("myDatabase", MODE_PRIVATE, null);

// Membuat tabel
db.execSQL("CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY, name TEXT, age INTEGER)");

// Menyimpan data
```

```
ContentValues values = new ContentValues();
values.put("name", "John Doe");
values.put("age", 25);
db.insert("users", null, values);

// Mengambil data
Cursor cursor = db.rawQuery("SELECT * FROM users", null);
if (cursor.moveToFirst()) {
    String name = cursor.getString(cursor.getColumnIndex("name"));
    int age = cursor.getInt(cursor.getColumnIndex("age"));
}
cursor.close();
```

c. MySQL

MySQL adalah database server yang digunakan untuk menyimpan data yang disentralisasi dan diakses melalui jaringan. Biasanya digunakan dalam aplikasi berbasis web atau aplikasi yang memerlukan penyimpanan data terpusat dan dapat diakses oleh beberapa perangkat atau pengguna.

Cara Kerja:

- Aplikasi Android dapat terhubung ke MySQL melalui API web atau server.
- Pengguna perlu mengatur API server untuk mengakses data MySQL, misalnya dengan menggunakan PHP, Node.js, atau framework lainnya.

Contoh Penggunaan:

Untuk menghubungkan aplikasi Android dengan MySQL, Anda perlu menyiapkan server backend yang mengelola query SQL dan mengirimkan data ke aplikasi Android menggunakan HTTP request.

Kesimpulan :

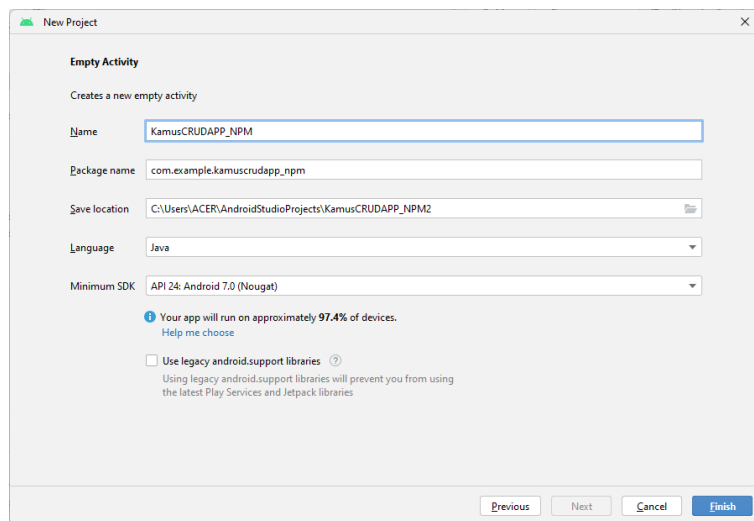
- ❖ Shared Preferences adalah pilihan yang tepat untuk data sederhana, seperti pengaturan pengguna.
- ❖ SQLite cocok untuk aplikasi yang membutuhkan penyimpanan data terstruktur, dengan kemampuan query yang lebih kompleks.

- ❖ MySQL digunakan untuk aplikasi yang membutuhkan penyimpanan data terpusat dan dapat diakses secara jaringan.

Pemilihan media penyimpanan sangat tergantung pada jenis aplikasi dan kebutuhan pengolahan data, dengan mempertimbangkan faktor-faktor seperti ukuran data, keamanan, dan aksesibilitas.

11.4 PRAKTIKUM

- 1) Buatlah project baru dengan nama KamusCRUDAPP_NPM



- 2) Buat Database dan Model Kamus

Untuk membuat database SQLite, kita perlu membuat class untuk mengatur struktur database dan juga model data untuk kamus.

- a. Buat class KamusEntry.java pada package utama

```
package com.example.kamuscudapp_npm;
public class KamusEntry {
    private int id;
    private String bahasaIndonesia;
    private String bahasaInggris;
    private String bahasaKorea;

    public KamusEntry(int id, String bahasaIndonesia, String
bahasaInggris, String bahasaKorea) {
        this.id = id;
        this.bahasaIndonesia = bahasaIndonesia;
        this.bahasaInggris = bahasaInggris;
        this.bahasaKorea = bahasaKorea;
    }
}
```

```

    public int getId() { return id; }
    public String getBahasaIndonesia() { return
bahasaIndonesia; }
    public String getBahasaInggris() { return bahasaInggris;
}
    public String getBahasaKorea() { return bahasaKorea; }

    public void setId(int id) { this.id = id; }
    public void setBahasaIndonesia(String bahasaIndonesia) {
this.bahasaIndonesia = bahasaIndonesia; }
    public void setBahasaInggris(String bahasaInggris) {
this.bahasaInggris = bahasaInggris; }
    public void setBahasaKorea(String bahasaKorea) {
this.bahasaKorea = bahasaKorea; }
}

```

b. Tambahkan DatabaseHelper.java untuk mengelola CRUD di database SQLite.

```

package com.example.kamuscudapp_npm;

import android.annotation.SuppressLint;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DatabaseHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "Kamus.db";
    private static final int DATABASE_VERSION = 1;
    private static final String TABLE_NAME = "kamus";

    // Kolom dalam tabel
    private static final String COLUMN_ID = "id";
    private static final String COLUMN_INDONESIA =
"bahasa_indonesia";
    private static final String COLUMN_INGGRIS =
"bahasa_inggris";
    private static final String COLUMN_KOREA =
"bahasa_korea";

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null,
DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        // Membuat tabel kamus
        String createTable = "CREATE TABLE " + TABLE_NAME +
" (" +

```



```

        COLUMN_ID + " INTEGER PRIMARY KEY
AUTOINCREMENT, " +
        COLUMN_INDONESIA + " TEXT, " +
        COLUMN_INGGRIS + " TEXT, " +
        COLUMN_KOREA + " TEXT)";
        db.execSQL(createTable);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion,
int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }

    // Tambah Data
    public boolean addEntry(String indo, String inggris,
String korea) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(COLUMN_INDONESIA, indo);
        values.put(COLUMN_INGGRIS, inggris);
        values.put(COLUMN_KOREA, korea);

        long result = db.insert(TABLE_NAME, null, values);
        db.close();
        return result != -1;
    }

    // Update Data Berdasarkan Bahasa Indonesia
    public boolean updateEntry(String indo, String inggris,
String korea) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(COLUMN_INGGRIS, inggris);
        values.put(COLUMN_KOREA, korea);

        int result = db.update(TABLE_NAME, values,
COLUMN_INDONESIA + " = ?", new String[]{indo});
        db.close();
        return result > 0;
    }

    // Hapus Data Berdasarkan Bahasa Indonesia
    public boolean deleteEntry(String indo) {
        SQLiteDatabase db = this.getWritableDatabase();
        int result = db.delete(TABLE_NAME, COLUMN_INDONESIA
+ " = ?", new String[]{indo});
        db.close();
        return result > 0;
    }

```

```

        // Add the getEntry method to fetch a specific entry by
        its Indonesian word
        public KamusEntry getEntry(String indoWord) {
            SQLiteDatabase db = this.getReadableDatabase();

            // Define the query to search for the Indonesian
            word in the database
            String query = "SELECT * FROM " + TABLE_NAME + "
            WHERE " + COLUMN_INDONESIA + " = ?";

            // Execute the query
            Cursor cursor = db.rawQuery(query, new
            String[]{indoWord});

            // Check if a record was returned
            if (cursor != null && cursor.moveToFirst()) {
                // Extract data from cursor
                @SuppressWarnings("Range") int id =
            cursor.getInt(cursor.getColumnIndex(COLUMN_ID));
                @SuppressWarnings("Range") String indonesia =
            cursor.getString(cursor.getColumnIndex(COLUMN_INDONESIA));
                @SuppressWarnings("Range") String inggris =
            cursor.getString(cursor.getColumnIndex(COLUMN_INGGRIS));
                @SuppressWarnings("Range") String korea =
            cursor.getString(cursor.getColumnIndex(COLUMN_KOREA));

                // Close cursor and database
                cursor.close();
                db.close();

                // Return the KamusEntry object
                return new KamusEntry(id, indonesia, inggris,
            korea);
            }

            // Close cursor and database if no entry was found
            cursor.close();
            db.close();

            // Return null if no entry was found
            return null;
        }
    }
}

```

3) Implementasi CRUD di MainActivity.java : untuk tampilan awal dan navigasi

```

package com.example.kamuscudapp_npm;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;

```

```

public class MainActivity extends AppCompatActivity {
    private Button btnTranslate, btnCrud;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

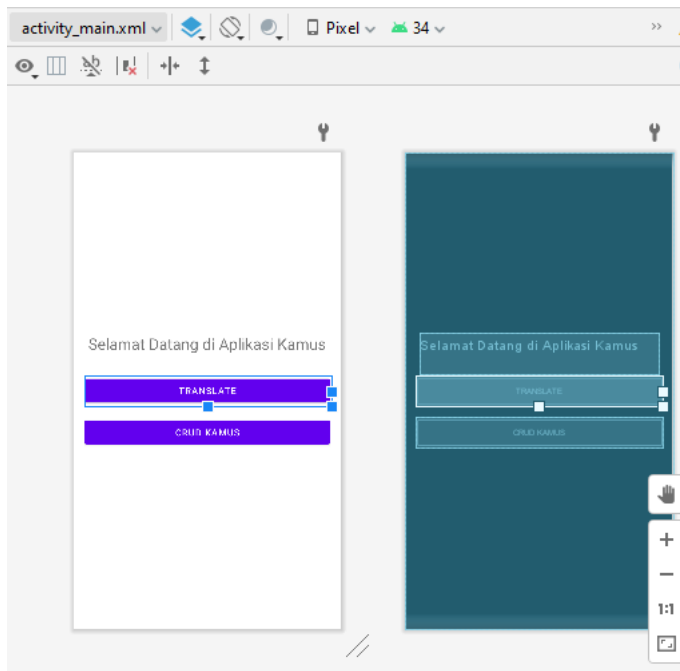
        btnTranslate = findViewById(R.id.btnTranslate);
        btnCrud = findViewById(R.id.btnCrud);

        btnTranslate.setOnClickListener(view -> {
            Intent intent = new Intent(MainActivity.this,
TranslateActivity.class);
            startActivity(intent);
        });

        btnCrud.setOnClickListener(view -> {
            Intent intent = new Intent(MainActivity.this,
CrudActivity.class);
            startActivity(intent);
        });
    }
}

```

- 4) Buat layout di activity_main.xml untuk memasukkan teks dan tombol.



```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"

```

```

        android:layout_height="match_parent"
        android:orientation="vertical"
        android:gravity="center"
        android:padding="16dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Selamat Datang di Aplikasi Kamus"
            android:textSize="24sp"
            android:gravity="center"
            android:paddingBottom="32dp" />

        <Button
            android:id="@+id/btnTranslate"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Translate" />

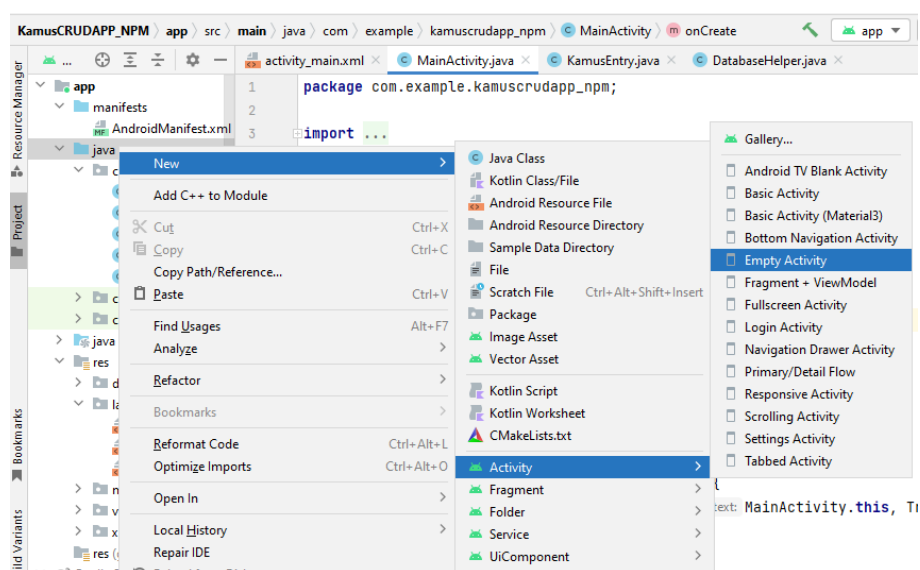
        <Button
            android:id="@+id/btnCrud"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="CRUD Kamus"
            android:layout_marginTop="16dp" />
    </LinearLayout>

```

5) Buat TranslateActivity untuk Fitur Terjemahan

TranslateActivity akan memungkinkan pengguna memasukkan kata dalam Bahasa Indonesia, Inggris, atau Korea, dan menerjemahkannya ke bahasa lain menggunakan data di database SQLite.

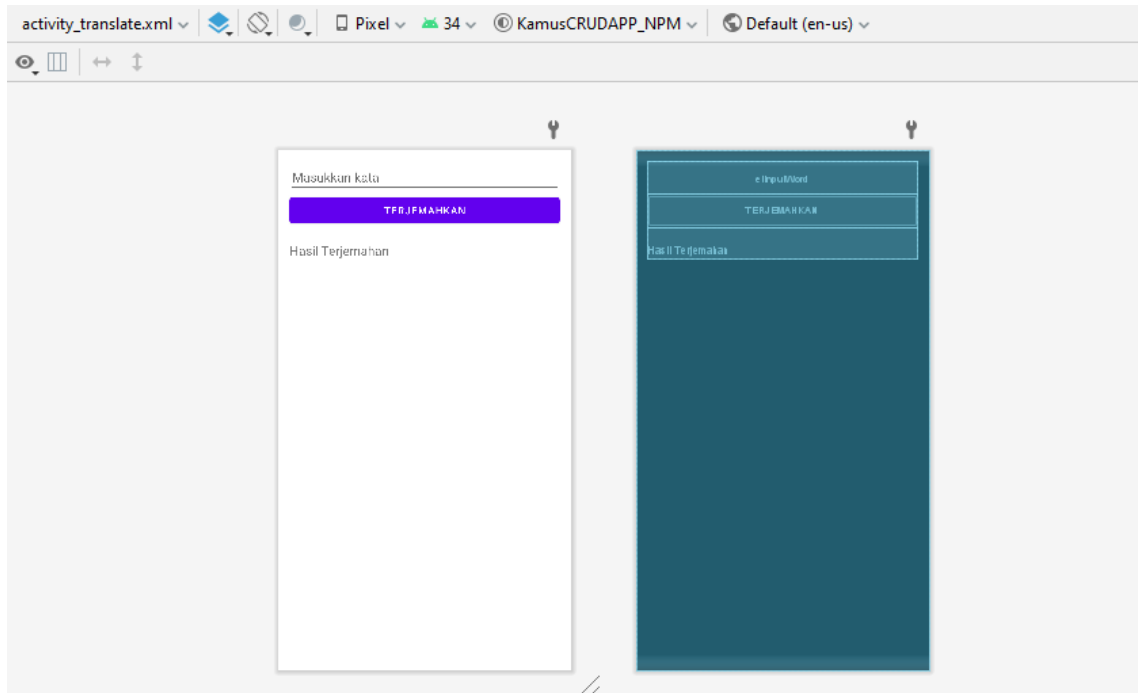
Klik kanan pada folder java → new → activity → Empty Activity.



Kemudian isikan pada activity name : TranslateActivity dan pada layout name : activity_translate → klik Oke.

activity_translate.xml

Tambahkan EditText untuk memasukkan kata dan Button untuk menerjemahkan.



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/etInputWord"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Masukkan kata" />

    <Button
        android:id="@+id/btnTranslateWord"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Terjemahkan" />

    <TextView
        android:id="@+id/tvTranslationResult"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
        android:text="Hasil Terjemahan"
        android:paddingTop="20dp"
        android:textSize="18sp" />
</LinearLayout>
```

TranslateActivity.java

Pada TranslateActivity, kita akan mengambil data dari database berdasarkan input dari pengguna dan menampilkan hasil terjemahan.

```
package com.example.kamuscudapp_npm;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class TranslateActivity extends AppCompatActivity {
    private EditText etInputWord;
    private Button btnTranslateWord;
    private TextView tvTranslationResult;
    private DatabaseHelper dbHelper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_translate);

        etInputWord = findViewById(R.id.etInputWord);
        btnTranslateWord =
            findViewById(R.id.btnTranslateWord);
        tvTranslationResult =
            findViewById(R.id.tvTranslationResult);

        dbHelper = new DatabaseHelper(this);

        btnTranslateWord.setOnClickListener(view -> {
            String inputWord =
                etInputWord.getText().toString().trim();
            if (!inputWord.isEmpty()) {
                KamusEntry entry =
                    dbHelper.getEntry(inputWord);

                if (entry != null) {
                    String result = "Bahasa Indonesia: " +
                        entry.getBahasaIndonesia() + "\n" +
                        "Bahasa Inggris: " +
                        entry.getBahasaInggris() + "\n" +
                        "Bahasa Korea: " +
```

```

entry.getBahasaKorea();
        tvTranslationResult.setText(result);
    } else {
        tvTranslationResult.setText("Kata tidak
ditemukan");
    }
    } else {
        tvTranslationResult.setText("Masukkan kata
yang ingin diterjemahkan");
    }
    });
}
}

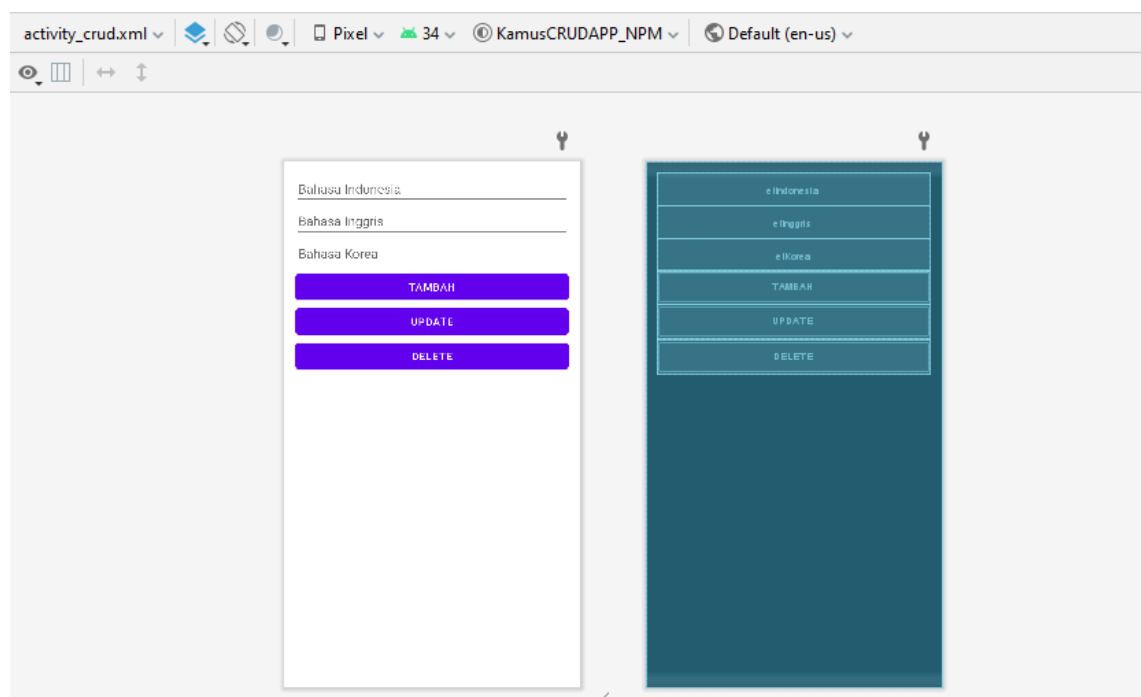
```

6) Buat CrudActivity untuk Fitur CRUD

CrudActivity akan menampilkan form untuk menambahkan, memperbarui, dan menghapus data kamus.

a) Layout activity_crud.xml

Tampilan ini akan memiliki EditText untuk input data kamus dan tombol untuk menambahkan, mengupdate, atau menghapus.



```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText

```

```

        android:id="@+id/etIndonesia"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Bahasa Indonesia" />

<EditText
    android:id="@+id/etInggris"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Bahasa Inggris" />

<EditText
    android:id="@+id/etKorea"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Bahasa Korea" />

<Button
    android:id="@+id/btnAddEntry"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Tambah" />

<Button
    android:id="@+id/btnUpdateEntry"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Update" />

<Button
    android:id="@+id/btnDeleteEntry"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Delete" />
</LinearLayout>

```

b) Kode CrudActivity.java

Implementasikan logika CRUD di dalam CrudActivity.

```

package com.example.kamuscrudapp_npm;

import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class CrudActivity extends AppCompatActivity {
    private EditText etIndonesia, etInggris, etKorea;
    private Button btnAddEntry, btnUpdateEntry,
    btnDeleteEntry;
    private DatabaseHelper dbHelper;

```



```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_crud);

    etIndonesia = findViewById(R.id.etIndonesia);
    etInggris = findViewById(R.id.etInggris);
    etKorea = findViewById(R.id.etKorea);

    btnAddEntry = findViewById(R.id.btnAddEntry);
    btnUpdateEntry = findViewById(R.id.btnUpdateEntry);
    btnDeleteEntry = findViewById(R.id.btnDeleteEntry);

    dbHelper = new DatabaseHelper(this);

    btnAddEntry.setOnClickListener(view -> {
        String indo = etIndonesia.getText().toString();
        String inggris = etInggris.getText().toString();
        String korea = etKorea.getText().toString();
        boolean isInserted = dbHelper.addEntry(indo,
inggris, korea);

        if (isInserted) {
            Toast.makeText(this, "Data berhasil
ditambahkan", Toast.LENGTH_SHORT).show();
            clearFields();
        } else {
            Toast.makeText(this, "Gagal menambahkan data",
Toast.LENGTH_SHORT).show();
        }
    });

    btnUpdateEntry.setOnClickListener(view -> {
        String indo = etIndonesia.getText().toString();
        String inggris = etInggris.getText().toString();
        String korea = etKorea.getText().toString();

        // Update using the three string parameters
        boolean isUpdated = dbHelper.updateEntry(indo,
inggris, korea);
        if (isUpdated) {
            Toast.makeText(CrudActivity.this, "Data
diperbarui", Toast.LENGTH_SHORT).show();
        }
    });

    btnDeleteEntry.setOnClickListener(view -> {
        int id =
Integer.parseInt(etIndonesia.getText().toString()); //
Assuming ID is entered in the Indonesian field
        boolean isDeleted =
dbHelper.deleteEntry(String.valueOf(id));
    });

```

```

        if (isDeleted) {
            Toast.makeText(CrudActivity.this, "Data
dihapus", Toast.LENGTH_SHORT).show();
        }
    });
}

private void clearFields() {
    etIndonesia.setText("");
    etInggris.setText("");
    etKorea.setText("");
}
}

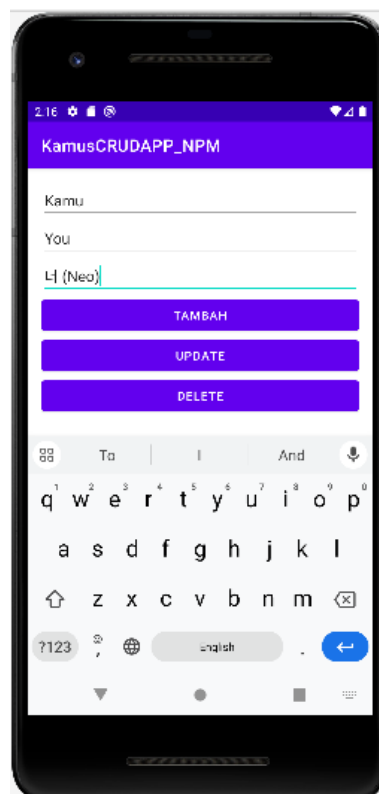
```

7) Menjalankan Aplikasi

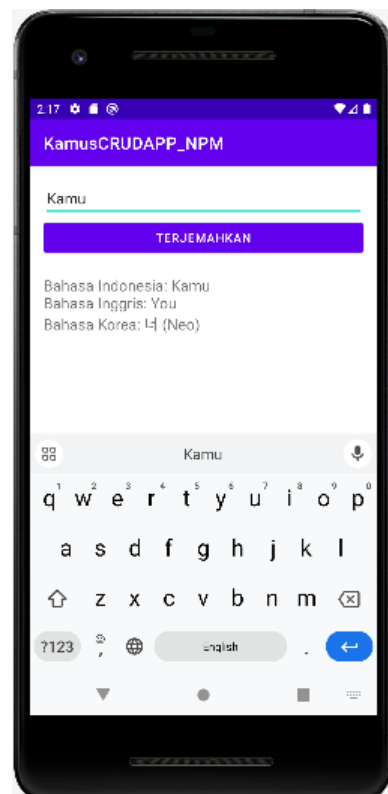
- Hubungkan perangkat Android atau buka Android Emulator.
- Klik tombol Run di Android Studio.
- Pilih perangkat yang ingin digunakan untuk menjalankan aplikasi.
- Aplikasi KAMUS CRUD akan ditampilkan.



Tampilan utama saat aplikasi dijalankan



Tampilan saat tombol CRUD Kamus di klik, Disini kita menambahkan mengubah, atau menghapus database dari kamus.



Tampilan saat tombol Translate di klik, Disini kita bisa mencari terjemahan dari kata yang telah kita masukkan ke database sebelumnya.

11.5 TUGAS

1. Demokan project tersebut hingga muncul di emulator!
2. Tambahkan minimal 10 kosakata baru ke dalam kamus yang sudah ada. Pastikan kosakata tersebut mencakup terjemahan untuk setiap kombinasi bahasa (Inggris-Indonesia, Indonesia-Korea, dll.).
3. Tugas dikumpul dalam format pdf dengan nama file : NPM_Tugaske?
4. Didalam file tersebu dituliskan :
 - NPM
 - Nama Lengkap
 - Program Studi
 - Kelas