



MODUL PRAKTIKUM PEMROGRAMAN MOBILE

**2023
2024**



LABORATORIUM TERPADU
FAKULTAS ILMU KOMPUTER
TEKNIK INFORMATIKA
UNIVERSITAS MUSLIM INDONESIA

TIM PENYUSUN

- Lutfi Budi Ilmawan, S.Kom., M.Cs., MTA.

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Kuasa, yang telah memberikan rahmat-Nya sehingga Modul Praktikum **Pemrograman Mobile** untuk mahasiswa/i Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Muslim Indonesia ini dapat diselesaikan dengan sebaik-baiknya.

Modul praktikum ini dibuat sebagai pedoman dalam melakukan kegiatan praktikum **Pemrograman Mobile** yang merupakan kegiatan penunjang mata kuliah pada Program Studi Teknik Informatika. Modul praktikum ini diharapkan dapat membantu mahasiswa/i dalam mempersiapkan dan melaksanakan praktikum dengan lebih baik, terarah, dan terencana. Pada setiap topik telah ditetapkan capaian pembelajaran mata kuliah pelaksanaan praktikum dan semua kegiatan yang harus dilakukan oleh mahasiswa/i serta teori singkat untuk memperdalam pemahaman mahasiswa/i mengenai materi yang dibahas.

Penyusun menyakini bahwa dalam pembuatan Modul Praktikum **Pemrograman Mobile** ini masih jauh dari sempurna. Oleh karena itu penyusun mengharapkan kritik dan saran yang membangun guna penyempurnaan modul praktikum ini dimasa yang akan datang.

Akhir kata, penyusun mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung.

Makassar, September 2023

Tim Penyusun

TATA TERTIB PELAKSANAAN PRAKTIKUM

Tata Tertib Pelaksanaan Praktikum pada Laboratorium Terpadu Fakultas Ilmu Komputer UMI adalah sebagai berikut:

1. Seluruh Pengguna laboratorium harus dalam keadaan sehat tidak menunjukkan gejala sakit (batuk, hidung tersumbat, dan suhu badan diatas 37°C).
2. Praktikan hanya diizinkan melaksanakan praktikum apabila :
 - a. Pria
 - Berpakaian rapi memakai kemeja putih polos;
 - Menggunakan celana kain berwarna hitam bukan dari bahan jeans/semi jeans;
 - Rambut rapi dan tidak panjang;
 - b. Wanita
 - Berpakaian rapi memakai kemeja tunik putih polos (tidak transparan)
 - Memakai Jilbab Segitiga Hitam (bukan pasmina) dan menutupi dada.
 - Menggunakan Rok Panjang berwarna hitam yang tidak terbelah dan tidak span serta bukan dari bahan jeans/semi jeans;
3. Ketika memasuki dan selama berada dalam ruangan, praktikan diwajibkan :
 - Tenang, tertib, dan sopan;
 - Tidak mengganggu praktikan lain yang sedang melaksanakan praktikum;
 - Tidak diperbolehkan merokok, membawa makanan / minuman senjata tajam dan senjata api ke dalam ruangan praktikum;
 - Tidak diperbolehkan membawa *handphone* ke meja praktikum dan *handphone* dalam mode senyap;
 - Tidak diperbolehkan membawa media penyimpanan eksternal atau *flashdisk* ke meja praktikum tanpa seizin Dosen Pengampu atau Asisten;
4. Dilarang membawa, mengambil, serta memindahkan perangkat yang digunakan pada saat praktikum tanpa instruksi dari Dosen Pengampu atau Asisten.
5. Toleransi keterlambatan praktikan maksimal 5 menit.
6. Praktikan berada diarea laboratorium dengan mengikuti jadwal yang telah ditentukan oleh Kepala Laboratorium.
7. Penggunaan fasilitas Laboratorium menyesuaikan dengan kapasitas ruang Laboratorium.

- 8 Segala pelanggaran yang dilakukan oleh praktikan akan berakibat pada penutupan dan penghentian penggunaan seluruh fasilitas laboratorium dan ditindak sesuai dengan aturan yang berlaku.

SANKSI-SANKSI

Sanksi terhadap pelanggaran **TATA TERTIB**:

Dosen Pengampu dan Asisten laboratorium berhak menjatuhkan sanksi, sesuai dengan aturan yang berlaku di Laboratorium Terpadu Fakultas Ilmu Komputer UMI apabila :

1. Praktikan merusak peralatan praktikum (*Personal Computer*) secara sengaja, maka praktikan bertanggung jawab untuk mengganti kerusakan tersebut.
2. Praktikan tidak mematuhi dan mentaati aturan praktikum maka tidak diperkenankan mengikuti praktikum.

Pelanggaran point lainnya dikenakan sanksi teguran, dikeluarkan/dicoret namanya dalam kegiatan praktikum (mengulang mata kuliah sesuai dengan semester berjalan) sampai sanksi akademik.



Makassar, September 2023
Kepala Laboratorium Terpadu,

Redya
Ir. Abdul Rachman Manga', S.Kom., M.T., MTA., MCF

DAFTAR ISI

KATA PENGANTAR	2
TATA TERTIB PELAKSANAAN PRAKTIKUM.....	3
DAFTAR ISI.....	5
MODUL 1 : BAGIAN 1 – EXPLICIT INTENT	6
MODUL 1 : BAGIAN 2 – IMPLICIT INTENT	13
MODUL 2 – INPUT CONTROL	18
MODUL 3 – ANDROID SQLITE DATABASE	23
MODUL 4 – ANDROID & MYSQL DATABASE.....	33

MODUL 1 : BAGIAN 1 – EXPLICIT INTENT

A. Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu mengimplementasikan penggunaan Object Explicit Intent pada pembuatan Aplikasi Android

B. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu membuat object Intent sebagai explicit intent
2. Mahasiswa mampu mengimplementasikan passing data ke Activity lain melalui Intent Data dan Intent Extras
3. Mahasiswa mampu mengimplementasikan proses menerima kembali data dari activity yang telah dibuka melalui override method onActivityResult()

C. Teori Dasar

Intent adalah sebuah jembatan yang menghubungkan interaksi antar Activity di aplikasi Android. Intent juga dapat membawa dan mengirimkan data ke Activity lain, bahkan ke aplikasi lain (Gmail, Google Map dsb). Intent Explicit berfungsi melakukan perpindahan activity (halaman) ke activity (halaman) lainnya.

Pemanggilan Intent secara explicit dengan menentukan nama Activity tujuan. Intent membutuhkan dua parameter, yang pertama adalah Context (dapat didapatkan dari Activity atau Application) dan berikutnya adalah nama NamaActivity.class.

```
Intent intent = new Intent(this, NextActivity.class);
startActivity(intent);
```

D. Kegiatan Praktikum

1. Instrument

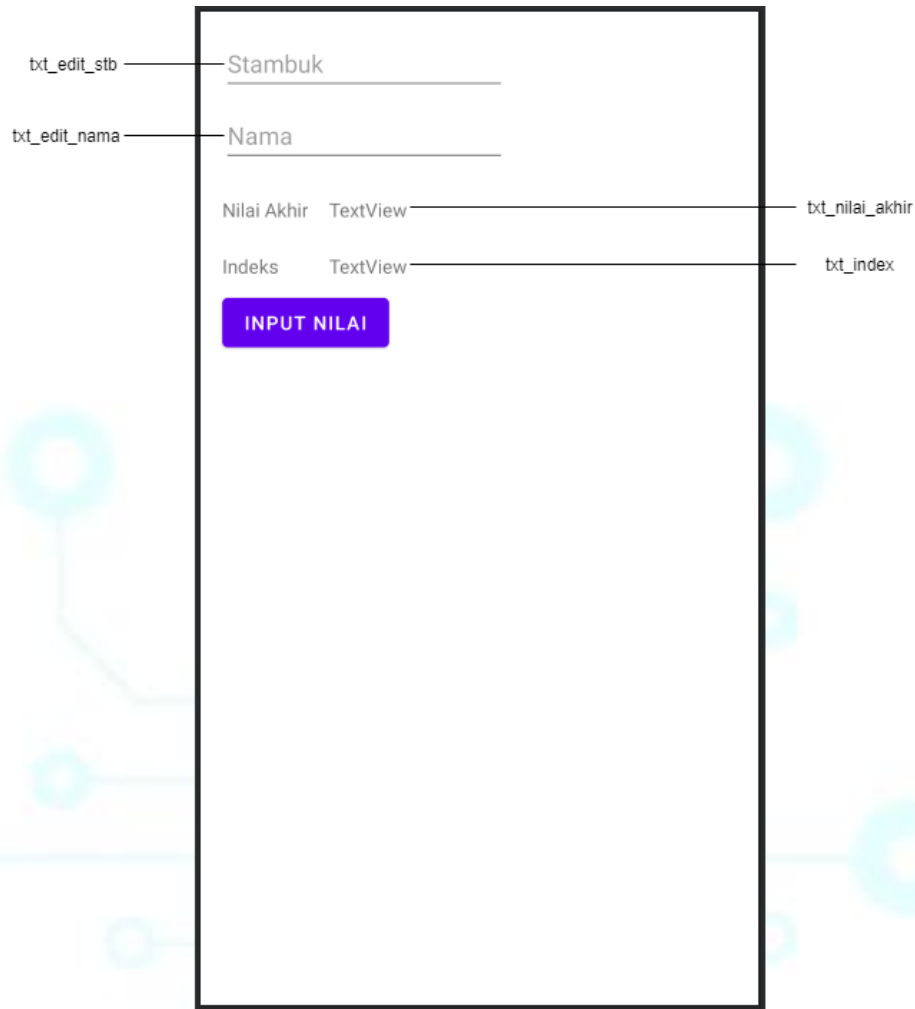
- a) Perangkat komputer
- b) Sistem Operasi Windows
- c) IDE Android Studio
- d) Emulator Android

2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain.
- e) Pastikan diri anda tidak menyentuh sumber listrik.

3. Studi Kasus

- a) Buka Android Studio, buat project baru dengan nama sesuai format berikut:
- b) Kelas-Prak1A-STB. Contoh: **A1-Prak1A-13020190001**
- c) Pilih Empty Activity
- d) Pada layout `activity_main.xml`, buat tampilannya sesuai pada Gambar 1, kemudian sesuaikan juga atribut id nya sesuai pada Gambar 1.



Gambar 1 Layout MainActivity (activity_main.xml)

- e) Setelah membuat layout `MainActivity` sesuai pada Gambar 1, buka file `MainActivity.java`, tambahkan Instance Variable/Object pada class `MainActivity` sbb:

```
private EditText txtStb, txtNama;
private TextView txtNilaiAkhir, txtIndeks;
static final String KEY_STB = "STB";
static final String KEY_NAMA = "NAMA";
static final String KEY_NILAI_TUGAS = "NILAI_TUGAS";
static final String KEY_NILAI_MID = "NILAI_MID";
static final String KEY_NILAI_FINAL = "NILAI_FINAL";
static final int RESULT_OK = 1;
static final int RESULT_CANCEL = 0;
private final int REQ_CODE_ACTIVITY2 = 0;
```


- f) Pasangkan object instance yang telah dibuat dengan layout dan lakukan inisialisasi pada beberapa objek, dengan menambahkan kode berikut pada method `onCreate()`, setelah statement `setContentView(R.layout.activity_main)` pada class `MainActivity`:

```
txtStb = findViewById(R.id.txt_edit_stb);
txtNama = findViewById(R.id.txt_edit_nama);
txtNilaiAkhir = findViewById(R.id.txt_nilai_akhir);
txtIndeks = findViewById(R.id.txt_indeks);

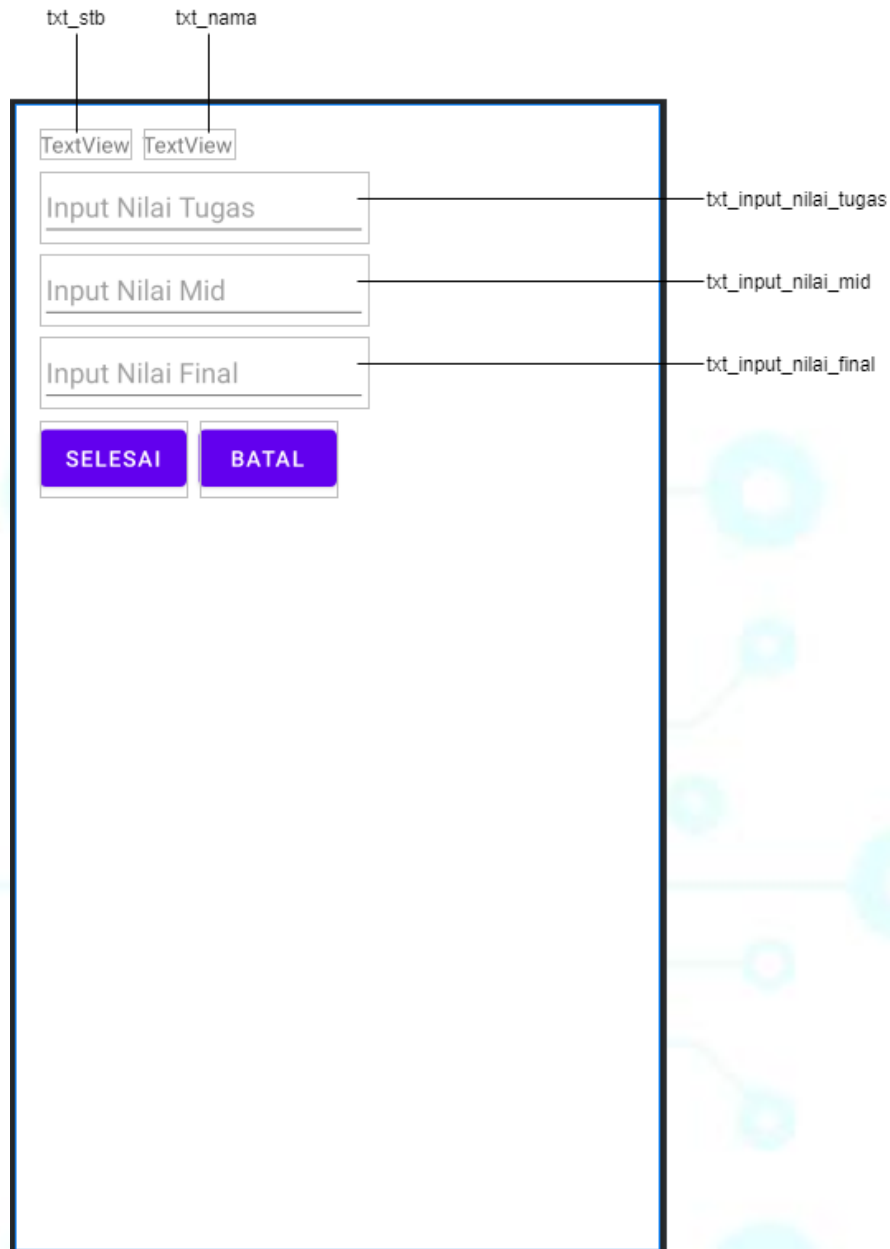
txtNilaiAkhir.setText(":");
txtIndeks.setText(":");
```

- g) Buat activity baru, klik menu File > New > Activity > Empty Activity, kemudian pada activity baru tersebut berikan nama: `Activity2`
- h) Pada class `MainActivity` buat sebuah method dengan nama `bukaActivity2()` untuk menangani event klik pada button `Input Nilai` yang terdapat pada layout `MainActivity`:

```
public void bukaActivity2(View view){
    Intent intent = new Intent( packageContext: this, Activity2.class);
    intent.putExtra(KEY_STB, txtStb.getText().toString());
    intent.putExtra(KEY_NAMA, txtNama.getText().toString());
    txtNilaiAkhir.setText(":");
    txtIndeks.setText(":");
    startActivityForResult(intent, REQ_CODE_ACTIVITY2);
}
```

Method di atas berfungsi untuk membuka `Activity2` kemudian menunggu kembali respon yang akan dikembalikan dari `Activity2`. Fungsi lainnya yaitu melakukan *passing* data ke `Activity2` melalui *Intent Extras*.

- i) Setelah method di atas dibuat, buka file layout `activity_main.xml`, klik button `Input Nilai`, kemudian set atribut `onClick = bukaActivity2`.
- j) Pada layout `activity_2.xml`, buat tampilannya sesuai pada Gambar 2, kemudian sesuaikan juga atribut id nya sesuai pada Gambar 2.



Gambar 2 Layout Activity2 (activity_2.xml)

- k) Setelah membuat layout `Activity2` sesuai pada Gambar 2, buka file `Activity2.java`, tambahkan Instance Variable/Object pada class `Activity2` sbb:

```
private TextView txtStb, txtNama;  
private EditText txtNilaiTugas, txtNilaiFinal, txtNilaiMid;
```

- l) Pasangkan object instance yang telah dibuat dengan layout dan lakukan inisialisasi pada beberapa objek, termasuk pengambilan data yang di-passing dari MainActivity melalui *Intent Extras*, dengan menambahkan kode berikut pada method `onCreate()`, setelah statement `setContentView(R.layout.activity_main)` pada class `Activity2`:

```

txtStb = findViewById(R.id.txt_stb);
txtNama = findViewById(R.id.txt_nama);
txtNilaiTugas = findViewById(R.id.txt_input_nilai_tugas);
txtNilaiMid = findViewById(R.id.txt_input_nilai_mid);
txtNilaiFinal = findViewById(R.id.txt_input_nilai_final);

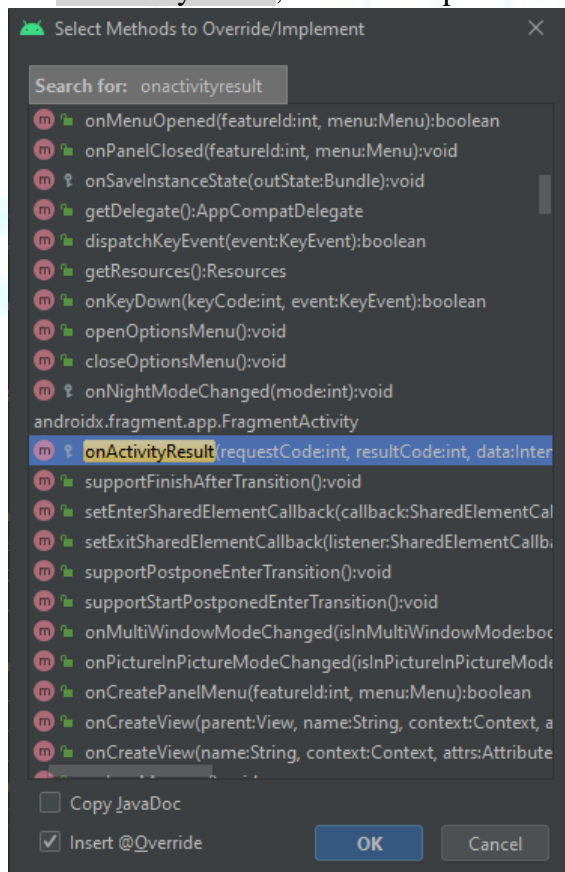
Intent intent = getIntent();
String stb, nama;

stb = intent.getStringExtra(MainActivity.KEY_STB);
nama = intent.getStringExtra(MainActivity.KEY_NAMA);

txtStb.setText(stb);
txtNama.setText(nama);

```

- m) Kembali buka MainActivity.java, lakukan *method overriding* untuk method `onActivityResult()` dari class `AppCompatActivity`, dengan menekan pada keyboard `ctrl+o`. Setelah daftar pilihan method yg akan di-*override* muncul, cari `onActivityResult`, setelah didapatkan klik OK.



- n) Method override yang telah dimasukkan akan tampak seperti pada kode di bawah:

```

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
}

```

- o) Pada method `onActivityResult()`, setelah statement `super.onActivityResult(requestCode, resultCode, data)` tambahkan kode berikut:

```

if (data == null) return;

if (requestCode == REQ_CODE_ACTIVITY2){
    if (resultCode == MainActivity.RESULT_OK ) {
        float nilaiAkhir;
        float nTgs, nMid, nFinal;
        nTgs = Float.parseFloat(data.getStringExtra(KEY_NILAI_TUGAS));
        nMid = Float.parseFloat(data.getStringExtra(KEY_NILAI_MID));
        nFinal = Float.parseFloat(data.getStringExtra(KEY_NILAI_FINAL));
        nilaiAkhir = (nTgs + nMid + nFinal) / 3;
        txtNilaiAkhir.setText(" " + nilaiAkhir);

        char indeks=' ';
        if (nilaiAkhir>=90 && nilaiAkhir<=100) indeks = 'A';
        else if (nilaiAkhir>=80 && nilaiAkhir<90) indeks = 'B';
        else if (nilaiAkhir>=70 && nilaiAkhir<80) indeks = 'C';
        else if (nilaiAkhir>=45 && nilaiAkhir<70) indeks = 'D';
        else if (nilaiAkhir<45) indeks = 'E';

        txtIndeks.setText(" " + indeks);
    }
    else if (resultCode == RESULT_CANCEL) {
        txtStb.setText("");
        txtNama.setText("");
        txtNilaiAkhir.setText(":");
        txtIndeks.setText(":");
        Toast.makeText( context: this, text: "Input NILai dibatalkan...", Toast.LENGTH_SHORT).show();
        txtStb.requestFocus();
    }
}
}

```

- p) Buka Activity2.java, pada class Activity2, tambahkan method `inputSelesai()` dan `inputBatal()`. Method ini untuk memberikan event klik pada masing-masing button `Selesai` dan `Batal`.

```

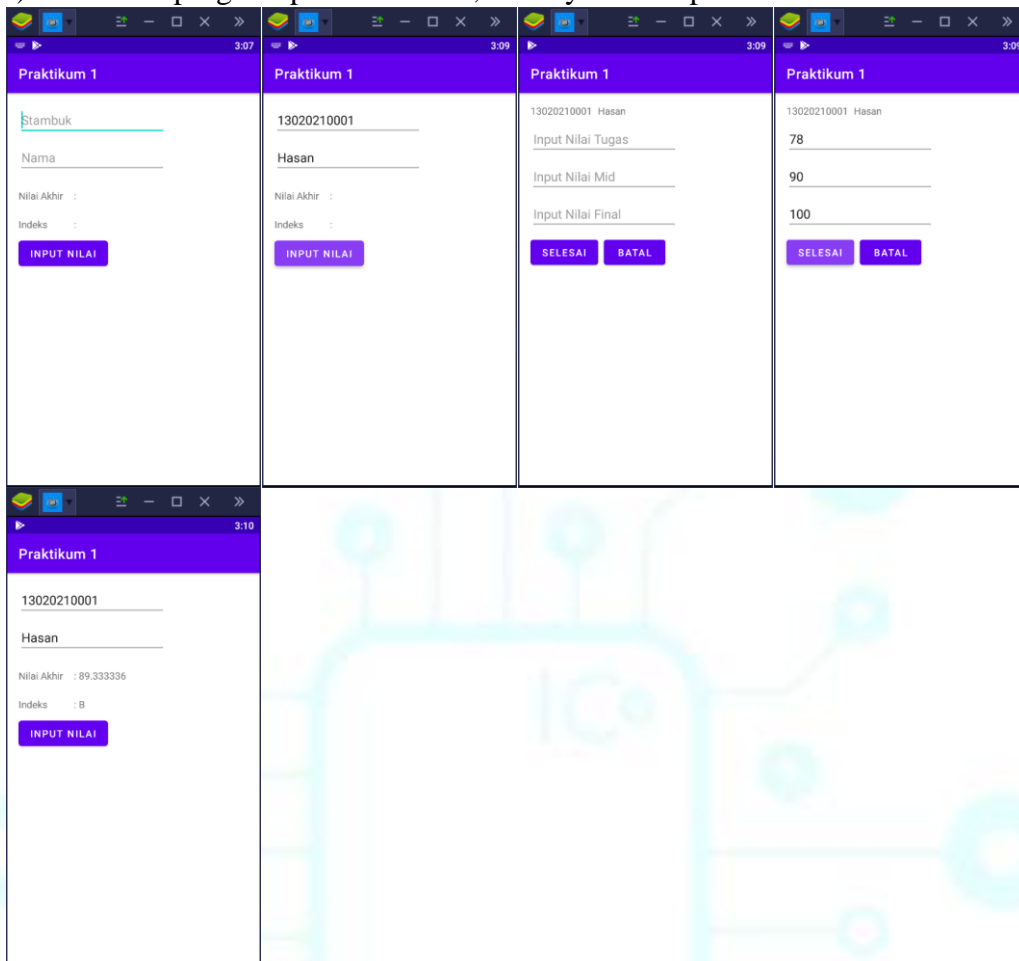
public void inputSelesai(View view){
    Intent intent = new Intent();
    intent.putExtra(MainActivity.KEY_NILAI_TUGAS, txtNilaiTugas.getText().toString());
    intent.putExtra(MainActivity.KEY_NILAI_MID, txtNilaiMid.getText().toString());
    intent.putExtra(MainActivity.KEY_NILAI_FINAL, txtNilaiFinal.getText().toString());
    setResult(MainActivity.RESULT_OK, intent);
    finish();
}

public void inputBatal(View view){
    Intent intent = new Intent();
    setResult(MainActivity.RESULT_CANCEL, intent);
    finish();
}

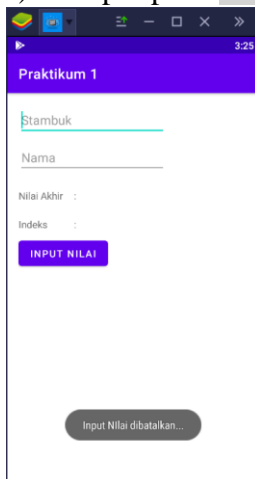
```

- q) Setelah kedua method di atas dibuat, buka file layout `activity_2.xml`, klik button `Selesai`, kemudian set atribut `onClick = inputSelesai`. Terakhir, klik button `Batal`, kemudian set atribut `onClick = inputBatal`.

r) Jalankan program pada emulator, hasilnya sesuai pada Gambar di bawah:



s) Output pada MainActivity ketika tombolnya Batal pada Activity2 diklik:



MODUL 1 : BAGIAN 2 – IMPLICIT INTENT**A. Capaian Pembelajaran Mata Kuliah (CPMK)**

Mahasiswa mampu mengimplementasikan penggunaan Object Implicit Intent pada pembuatan Aplikasi Android

B. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu membuat object Intent sebagai explicit intent
2. Mahasiswa mampu mengimplementasikan passing data ke Activity lain melalui Intent Data dan Intent Extras
3. Mahasiswa mampu mengimplementasikan proses menerima kembali data dari activity yang telah dibuka melalui override method onActivityResult()

C. Teori Dasar

Terkadang pada aplikasi Android yang dibangun, membutuhkan activity bawaan yang telah disediakan oleh Android. Pemanggilan Intent ini dilakukan secara implicit. Pada umumnya untuk melakukan ini, cukup dengan mendefinisikan aksi tertentu (dapat menggunakan konstanta pada Intent yang umumnya menggunakan prefiks ACTION). Aplikasi yang umum digunakan antara lain:

- Call
- Dialpad
- Contact
- Browser
- Call Log
- Gallery
- Camera

Sebagai contoh pemanggilan Gallery yang berisi gambar, dapat dilihat pada kode berikut.

```
Intent intent = new Intent(Intent.ACTION_PICK,  
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
```

D. Kegiatan Praktikum**1. Instrument**

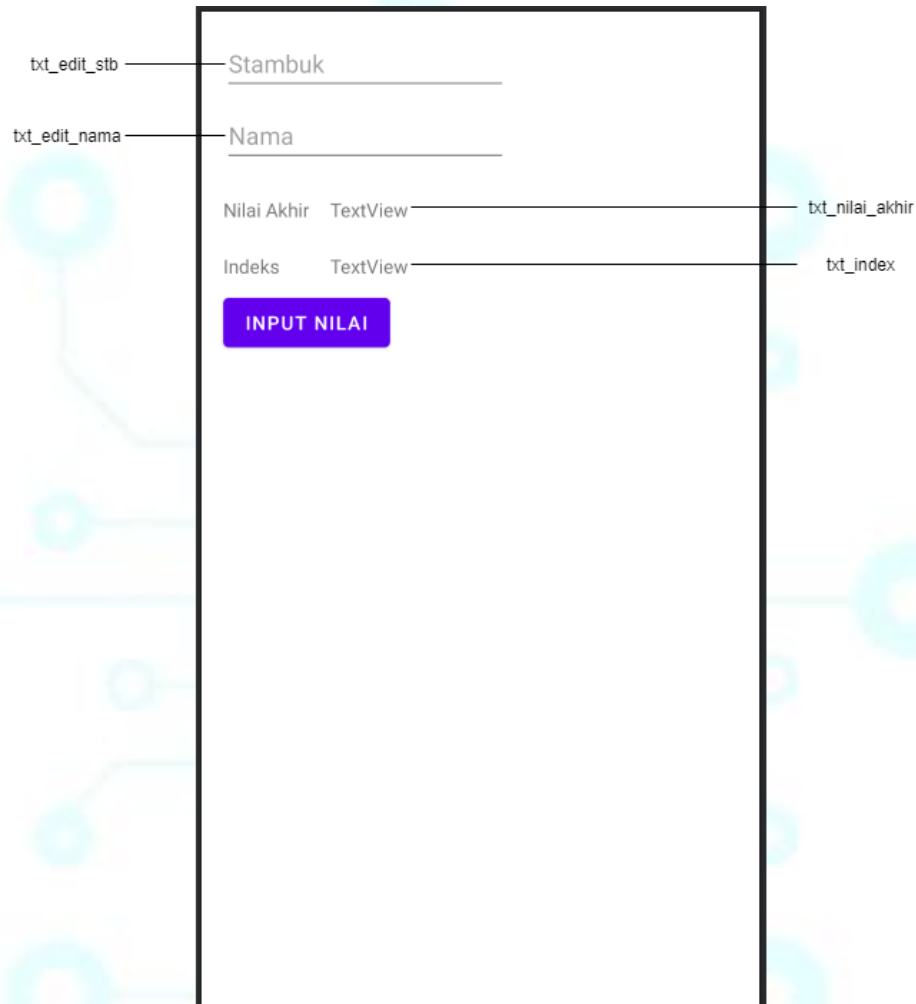
- a) Perangkat komputer
- b) Sistem Operasi Windows
- c) IDE Android Studio
- d) Emulator Android

2. Prosedur

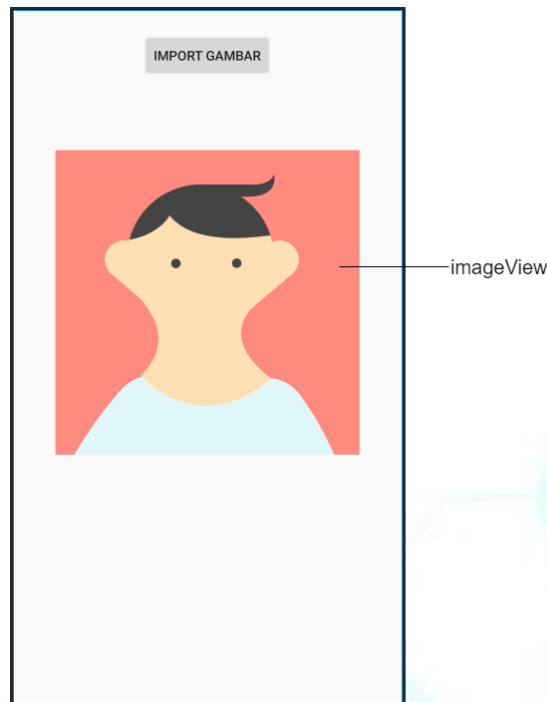
- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain.
- e) Pastikan diri anda tidak menyentuh sumber listrik.

3. Studi Kasus

- Buka Android Studio, buat project baru dengan nama sesuai format berikut: Kelas-Prak1B-STB. Contoh: **A1-Prak1B-13020190001**
- Pilih Empty Activity
- Pada layout `activity_main.xml`, buat tampilannya sesuai pada Gambar 1, kemudian sesuaikan juga atribut id nya sesuai pada Gambar 3.



Gambar 1 Layout MainActivity (activity_main.xml)



Gambar 3 Layout MainActivity (activity_main.xml)

- d. Pada objek `imageView`, atur atribut `scaleType = fitCenter`.
- e. Setelah membuat layout `MainActivity` sesuai pada Gambar 1, buka file `MainActivity.java`, tambahkan Instance Variable/Object pada class `MainActivity` sbb:

```
private ImageView imageView;
private final int REQUEST_CODE_PERMISSION_BACA_STORAGE = 1;
private final int REQUEST_CODE_ACTIVITY_GALLERY = 1;
```

- f. Pasangkan object instance yang telah dibuat dengan layout, dengan menambahkan kode berikut pada method `onCreate()`, setelah statement `setContentView(R.layout.activity_main)` pada class `MainActivity`:

```
imageView = findViewById(R.id.imageView);
```

- g. Buat method dengan nama `bukaGallery()`, method ini berfungsi untuk memulai activity dari aplikasi lain yang dapat mengambil gambar dari *storage*, kemudian untuk *request code activity*-nya = 1.

```
private void bukaGallery(){
    Uri uri = MediaStore.Images.Media.EXTERNAL_CONTENT_URI;
    Intent intent = new Intent(Intent.ACTION_PICK, uri);
    startActivityForResult(intent, REQUEST_CODE_ACTIVITY_GALLERY);
}
```

- h. Tambahkan method dengan nama `buttonBukaGalleryClick()` untuk event klik pada button `Import Gambar`:

```
public void buttonBukaGalleryClick(View view){
    final int PERMISSION_BACA_EXT_STORAGE = ContextCompat.checkSelfPermission( context: this, Manifest.permission.READ_EXTERNAL_STORAGE);
    if ( PERMISSION_BACA_EXT_STORAGE != PackageManager.PERMISSION_GRANTED){
        ActivityCompat.requestPermissions( activity: this, new String[] {Manifest.permission.READ_EXTERNAL_STORAGE}, REQUEST_CODE_PERMISSION_BACA_STORAGE);
        return;
    }
    bukaGallery();
}
```


Pada method ini, method `bukaGellery()` akan di-*invoke*. Tapi sebelum itu, terlebih dahulu dilakukan pengecekan *permission* untuk membaca *file* di luar dari *storage* aplikasi yang kita miliki. Jika belum diizinkan, maka *request permission* `READ_EXTERNAL_STORAGE` akan dilakukan dengan *request code* = 1 dan method `bukaGallery()` tidak tereksekusi.

- i. Lakukan *method overriding* untuk method `onRequestPermissionsResult()`, kemudian pada method tersebut tambahkan baris kode berikut setelah `super.onRequestPermissionsResult(requestCode, permissions, grantResults)` :

```
if (requestCode == REQUEST_CODE_PERMISSION_BACA_STORAGE){
    if (grantResults[0] == PackageManager.PERMISSION_GRANTED){
        bukaGallery();
    }
}
```

Baris kode ini berfungsi menerima respon (*callback*) dari *request permission* yang dilakukan sebelumnya. Jika *permission* diizinkan, maka method `bukaGallery()` akan dieksekusi.

- j. Lakukan *method overriding* untuk method `onActivityResult()`, kemudian pada method tersebut tambahkan baris kode berikut setelah `super.onActivityResult(requestCode, resultCode, data)` :

```
if (requestCode == REQUEST_CODE_ACTIVITY_GALLERY){
    Cursor cursor;
    String col = "_data";
    String[] projection = {col};
    cursor = getApplicationContext().getContentResolver().query(data.getData(), projection, selection: null, selectionArgs: null, sortOrder: null);
    cursor.moveToFirst();
    int index = cursor.getColumnIndexOrThrow(col);
    String pathGambar = cursor.getString(index);
    Bitmap gambar = BitmapFactory.decodeFile(pathGambar);
    imageView.setImageBitmap(gambar);
}
```

Baris kode ini berfungsi untuk menampilkan objek `bitmap` ke `imageView` yang telah dimasukkan sebelumnya pada *layout* `activity_main.xml`. Objek `bitmap` ini berasal dari *activity* yang dibuka pada method `bukaGallery()`.

LEMBAR EVALUASI PRAKTIKUM

Evaluasi Praktikum 1 :

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) ≤ 40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) ≥ 86
1.						
2.						
3.						
4.						

Catatan Asisten :

Dosen : _____

Asisten 1 : _____

Asisten 2 : _____

TINGKAT KEBERHASILAN

MODUL 2 – INPUT CONTROL

A. Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu mengimplementasikan penggunaan berbagai input control pada Android

B. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu mengimplementasikan penggunaan object dari class EditText.
2. Mahasiswa mampu mengimplementasikan penggunaan object dari class TextView.
3. Mahasiswa mampu mengimplementasikan penggunaan object dari class CheckBox.
4. Mahasiswa mampu mengimplementasikan penggunaan object dari class RadioGroup dan RadioButton.
5. Mahasiswa mampu mengimplementasikan penggunaan object dari class Spinner.

C. Teori Dasar

a. EditText

- Set value: **setText(String arg);**
`editTextName.setText(string);`
- Get value: **getText();**
`String string = editTextName.getText().toString();`
- Event change: **addTextChangedListener(TextWatcher arg);**

b. CheckBox

- Set Value: **setChecked(boolean arg);**
`checkBoxName.setChecked(true);`
- Get Value: **isChecked();**
`boolean bool = checkBoxName.isSelected();`

c. RadioGroup & RadioButton

- Get value:

```
switch(radioGroupName.getCheckedRadioButtonId()){
    case R.id.radioButton1Id:
        //statement(s)
        break;
    case R.id.radioButton2Id:
        //statement(s)
        break;
    ...
}
```
- Event : **setOnCheckedChangeListener(OnCheckedChangeListener arg)**

d. Spinner

- Mengisi Item pada Spinner:
 - 1) Buat objek **ArrayList/Array** dengan [generic] type String, isi objek tsb dengan item yang akan dimasukkan pada Spinner.

- 2) Buat objek **ArrayAdapter**. Beri nilai pada objek tsb: **new ArrayAdapter<>(this, android.R.layout.simple_spinner_item, objArrayList)**
- 3) Eksekusi method **setDropDownViewResource()** dengan argument **android.R.layout.simple_spinner_dropdown_item** untuk objek **ArrayAdapter**.
- 4) Set adapter pada spinner objek: **namaSpinner.setAdapter(objArrayAdapter);**
- 5) Contoh:

```
String kontenSpinner[] = {"satu", "dua", "tiga"};
ArrayAdapter<CharSequence> adapter = new ArrayAdapter<>
    (this, android.R.layout.simple_spinner_item, kontenSpinner);
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinner.setAdapter(adapter);
```

- Event saat item dipilih: **setOnItemSelectedListener(AdapterView.OnItemSelectedListener arg);**
- Mengambil nilai index: **namaSpinner.getSelectedItemPosition();**
- Mengambil nilai item: **namaSpinner.getSelectedItem();**
- Memilih berdasarkan index: **namaSpinner.setSelection(int index);**

D. Kegiatan Praktikum

1. Instrument

- a) Perangkat komputer
- b) Sistem Operasi Windows
- c) IDE Android Studio
- d) Emulator Android

2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain.
- e) Pastikan diri anda tidak menyentuh sumber listrik.

3. Studi Kasus

- a. Buat project baru dengan nama sesuai format berikut:
Kelas-Prak2-STB. Contoh: **A1-Prak2-13020190001**
- b. Pada layout **activity_main.xml**, buat tampilannya sesuai pada Gambar 1.

The screenshot shows the MainActivity layout with the following components labeled:

- EditText**: Points to the input field for "Masukkan Stambuk".
- Spinner**: Points to the dropdown menu for "-Pilih Angkatan-".
- RadioGroup**: Points to the group of radio buttons for "Program Studi".
- RadioButton**: Points to the "Teknik Informatika" radio button.
- CheckBox**: Points to the group of checkboxes for "Minat".
- Button**: Points to the "TAMPILKAN RINGKASAN" button.

Gambar 1 Layout MainActivity (activity_main.xml)

- c. Untuk spinner angkatan berisi item seperti yang tampak pada gambar di bawah:

The spinner dropdown menu displays the following items:

- Pilih Angkatan-
- 2021
- 2020
- 2019
- 2018
- 2017
- 2016
- 2015
- 2014

- d. Ketika tombol Tampil Ringkasan diklik, maka activity kedua akan terbuka untuk menampilkan ringkasan informasi sesuai dengan data yang diinputkan pada MainActivity.

Praktikum 2

13120180002

Burhan

2018

Program Studi

☐ Teknik Informatika ☒ Sistem Informasi

Minat

☒ Badan Eksekutif Mahasiswa

☒ Penulisan Karya Ilmiah

☐ Kewirausahaan

☒ Kesenian

☒ Jurnalistik

☐ Olahraga

TAMPILKAN RINGKASAN

Praktikum 2

RINGKASAN DATA

Stambuk	13120180002
Nama	Burhan
Angkatan	2018
Program Studi	Sistem Informasi
Minat	- Badan Eksekutif Mahasiswa - Penulisan Karya Ilmiah - Kesenian - Jurnalistik

TUTUP

LEMBAR EVALUASI PRAKTIKUM

Evaluasi Praktikum 2 :

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) ≤ 40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) ≥ 86
1.						
2.						
3.						
4.						

Catatan Asisten :

Dosen : _____

Asisten 1 : _____

Asisten 2 : _____

TINGKAT KEBERHASILAN

MODUL 3 – ANDROID SQLITE DATABASE

1. Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu mengimplementasikan SQLite database pada Android

2. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu mengimplementasikan pembuatan class untuk Database Helper yang merupakan turunan dari class SQLiteOpenHelper.
2. Mahasiswa mampu mengimplementasikan pendefinisian abstract method dari class SQLiteOpenHelper.
3. Mahasiswa mampu mengimplementasikan pembuatan POJO Class sesuai dengan entitas yang digunakan.
4. Mahasiswa mampu mengimplementasikan CRUD pada SQLiteDatabase.

3. Teori Dasar

SQLite adalah database SQL opensource yang menyimpan data ke file teks di perangkat. Android hadir dengan implementasi database SQLite bawaan.

SQLite mendukung semua fitur database relasional. Untuk mengakses database ini, Anda tidak perlu membuat koneksi apa pun seperti JDBC, ODBC, dll.

SQLite pada dasarnya adalah Relational Database Management System (RDBMS), sama seperti SQL. Ini merupakan perpustakaan dalam proses sumber terbuka yang mandiri, tanpa server, memiliki konfigurasi nol, dan mesin database SQL transaksional. Di sini, konfigurasi nol berarti tidak seperti sistem manajemen basis data lainnya, tidak perlu dikonfigurasi pada perangkat. Lite di sini di SQLite adalah dalam hal pengaturannya, administrasi database, dan semua sumber daya yang diperlukan.

1. SQLite mendukung tiga jenis data berikut:
 - Jenis Teks – untuk menyimpan data tipe string atau karakter.
 - Tipe Integer – untuk menyimpan tipe data integer.
 - Tipe Nyata – untuk menyimpan nilai panjang.
2. Untuk menggunakan SQLite di aplikasi Android, kita dapat menggunakan paket android.database.sqlite. Paket ini berisi semua API untuk menggunakan SQLite.
3. SQLiteOpenHelper adalah class yang berguna untuk membuat database dan mengelolanya. Dua konstruktor kelas SQLiteOpenHelper adalah:
4. SQLiteOpenHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version_no) – Ini menciptakan objek untuk membuat, membuka, dan mengelola database.
5. SQLiteOpenHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version_no, DatabaseErrorHandler errorHandler) – Objeknya menentukan pengendali kesalahan bersama dengan membuat, membuka, dan mengelola database.
6. SQLiteDatabase adalah kelas yang memiliki metode untuk melakukan operasi seperti membuat, memperbarui, menghapus, dll.

SQLite Database

1. Database – Package

Database package adalah `android.database.sqlite` yang berisi kelas-kelas untuk mengelola database Anda sendiri

Fungsi lain dari database package untuk melakukan pekerjaan :

- `openDatabase(String path, SQLiteDatabase.CursorFactory factory, int flags, DatabaseErrorHandler errorHandler)`
- `openDatabase(String path, SQLiteDatabase.CursorFactory factory, int flags)`
- `openOrCreateDatabase(String path, SQLiteDatabase.CursorFactory factory)`
- `openOrCreateDatabase(File file, SQLiteDatabase.CursorFactory factory)`

2. Database – Insert

Kita dapat membuat tabel atau memasukkan data ke dalam tabel menggunakan metode `execSQL` yang didefinisikan dalam kelas `SQLiteDatabase`. Sintaksnya diberikan di bawah ini:

```
mydatabase.execSQL("CREATE TABLE IF NOT EXISTS TutorialsPoint(Username
VARCHAR,Password VARCHAR);");
```

```
mydatabase.execSQL("INSERT INTO TutorialsPoint VALUES('admin','admin');");
```

Ini akan memasukkan beberapa nilai ke dalam tabel di database. Metode lain yang juga melakukan pekerjaan yang sama tetapi mengambil beberapa parameter tambahan diberikan di bawah ini:

```
execSQL(String sql, Object[] bindArgs)
```

3. Database – Fetching

Kita dapat mengambil apapun dari database menggunakan objek dari kelas `Cursor`. Kami akan memanggil metode kelas ini yang disebut `rawQuery` dan itu akan mengembalikan hasil dengan kursor menunjuk ke tabel. Kita dapat menggerakkan kursor ke depan dan mengambil data. Contoh codenya :

```
Cursor resultSet = mydatabase.rawQuery("Select * from TutorialsPoint",null);
resultSet.moveToFirst();
String username = resultSet.getString(0);
String password = resultSet.getString(1);
```

4. Database – helper class

Untuk mengelola semua operasi yang berhubungan dengan database, telah disediakan kelas helper yang disebut `SQLiteOpenHelper`. Kelola pembuatan dan pembaruan basis data secara otomatis. Sintaksnya diberikan di bawah ini

```
public class DBHelper extends SQLiteOpenHelper {
    public DBHelper(){
        super(context,DATABASE_NAME,null,1);
    }
    public void onCreate(SQLiteDatabase db) {}
    public void onUpgrade(SQLiteDatabase database, int oldVersion, int newVersion) {}
}
```

5. Kegiatan Praktikum

1. Instrument

- Perangkat komputer
- Sistem Operasi Windows
- IDE Android Studio
- Emulator Android

2. Prosedur

- Baca dan pahami semua tahapan praktikum dengan cermat.
- Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain.
- Pastikan diri anda tidak menyentuh sumber listrik.

3. Studi Kasus

- Buat project baru dengan nama sesuai format berikut:
Kelas-Prak3-STB. Contoh: **A1-Prak3-13020190001**
- Buat class baru dengan nama **Mahasiswa** (Klik kanan package > new > Java Class), pada class Mahasiswa buat 3 field: stb, nama, dan angkatan. Sesuai pada gambar di bawah:

```
public class Mahasiswa {
    private String stb;
    private String nama;
    private int angkatan;

    public Mahasiswa(String stb, String nama, int angkatan) {
        this.stb = stb;
        this.nama = nama;
        this.angkatan = angkatan;
    }

    public String getStb() {
        return stb;
    }

    public String getNama() {
        return nama;
    }

    public int getAngkatan() {
        return angkatan;
    }
}
```

- Buat class baru dengan nama **DbHelper** (Klik kanan package > new > Java Class), pada class DbHelper lakukan extends dari **abstract class SQLiteOpenHelper**. Definisikan abstract method yang disediakan oleh class **SQLiteOpenHelper**. Buat konstruktor untuk mengeksekusi konstruktor dari class yang di-extend. Pada class **SQLiteOpenHelper**, konstruktornya memiliki 4 parameter, yaitu object dari class: Context (context aplikasi/activity), String (untuk nama database), CursorFactory (untuk cursor object), int (untuk versi database).

```

public class DbHelper extends SQLiteOpenHelper {

    private ArrayList<Mahasiswa> arrListMhs = new ArrayList<>();

    public DbHelper(Context context){
        super(context, "db_mhs", null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
        String sql = "create table tb_mhs(stb text(11) primary key, nama text(50), angkatan text(4))";
        sqLiteDatabase.execSQL(sql);
    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {

    }

    public void insertData(SQLiteDatabase db, Mahasiswa mhs){
        ContentValues cv = new ContentValues();
        cv.put("stb", mhs.getStb());
        cv.put("nama", mhs.getNama());
        cv.put("angkatan", mhs.getAngkatan());
        db.insert("tb_mhs", null, cv);
    }

    public ArrayList<Mahasiswa> getArrListMhs(SQLiteDatabase db){
        arrListMhs.clear();
        Cursor cursor = db.rawQuery("select * from tb_mhs", null);
        if (cursor.getCount() > 0){
            cursor.moveToFirst();
            do{
                arrListMhs.add(new Mahasiswa(cursor.getString(0), cursor.getString(1), cursor.getInt(2)));
            }while(cursor.moveToNext());
        }
        return arrListMhs;
    }

    public void hapusData(SQLiteDatabase db, String stb){
        db.delete("tb_mhs", "stb=?", new String[]{stb});
    }

    public void editData(SQLiteDatabase db, Mahasiswa mhs, String stb){
        ContentValues cv = new ContentValues();
        cv.put("stb", mhs.getStb());
        cv.put("nama", mhs.getNama());
        cv.put("angkatan", mhs.getAngkatan());
        db.update("tb_mhs", cv, "stb=?", new String[]{stb});
    }

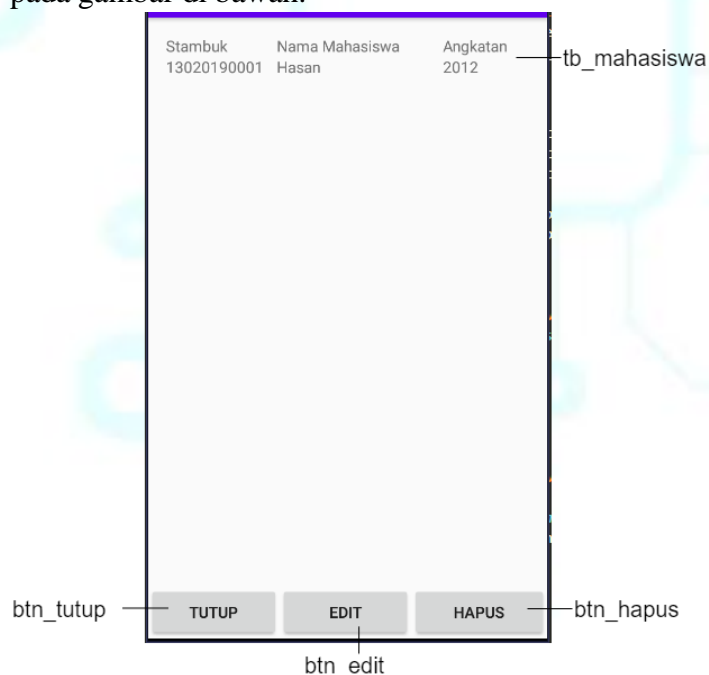
}

```

- d. Pada layout `activity_main.xml`, buat tampilannya sesuai pada gambar di bawah, sesuaikan dengan id yang telah diberikan.



- e. Buat *Activity* baru dengan nama **TampilActivity**, sesuaikan layout dan id-nya sesuai pada gambar di bawah.



- f. Buat *instance object/variable* pada class **MainActivity** sesuai pada potongan kode di bawah:

```
private EditText txtStb, txtNama, txtAngkatan;
private Button btnSimpan, btnTampil;
private DBHelper dbHelper;
private Mahasiswa mhs;
private Intent intentEdit;
```

- g. Pada method `onCreate()` dalam class `MainActivity` masukkan kode berikut untuk melakukan inisialisasi terhadap instance object yg dibuat sebelumnya, serta untuk memberikan event klik pada button yang terdapat pada layout.

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

    dbHelper = new DbHelper(this);

    txtStb = findViewById(R.id.txt_stb);
    txtNama = findViewById(R.id.txt_nama);
    txtAngkatan = findViewById(R.id.txt_angkatan);

    btnSimpan = findViewById(R.id.btn_simpan);
    btnTampil = findViewById(R.id.btn_tampil);

    btnSimpan.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (intentEdit == null) simpanData();
            else editData();
        }
    });

    btnTampil.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            intentEdit = null;
            Intent intent = new Intent(getApplicationContext(), TampilActivity.class);
            startActivityForResult(intent, 1);
            dbHelper.close();
        }
    });
}
```

- h. Buat method untuk mengosongkan inputan pada `MainActivity`. Pada class `MainActivity` tambahkan method `clearText()`:

```
private void clearText(){
    txtStb.setText("");
    txtNama.setText("");
    txtAngkatan.setText("");
    intentEdit = null;
    txtStb.requestFocus();
}
```

- i. Buat method dengan nama `simpanData()` pada class `MainActivity` untuk menyimpan data dalam tabel `tb_mhs` pada database `db_mhs`, dengan mengeksekusi method `insertData()` yang dibuat sebelumnya pada class `DbHelper`.

```
private void simpanData(){
    mhs = new Mahasiswa(
        txtStb.getText().toString(),
        txtNama.getText().toString(),
        Integer.parseInt(txtAngkatan.getText().toString())
    );

    dbHelper.insertData(dbHelper.getWritableDatabase(), mhs);

    Toast.makeText(this, "Data tersimpan...", Toast.LENGTH_LONG).show();
    clearText();
}
```

- j. Buat method dengan nama `editData()` pada class `MainActivity` untuk mengubah data dalam tabel `tb_mhs` pada database `db_mhs` dengan mengeksekusi method `editData()` yang dibuat sebelumnya pada class `DbHelper`.

```
private void editData(){
    mhs = new Mahasiswa(
        txtStb.getText().toString(),
        txtNama.getText().toString(),
        Integer.parseInt(txtAngkatan.getText().toString())
    );
    dbHelper.editData(dbHelper.getWritableDatabase(), mhs, intentEdit.getStringExtra("stb"));

    Toast.makeText(this, "Edit Data berhasil...", Toast.LENGTH_LONG).show();
    clearText();
}
```

- k. Lakukan *override* terhadap method `onActivityResult()` pada class `MainActivity`.

```
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == 1){
        intentEdit = data;
        txtStb.setText(data.getStringExtra("stb"));
        txtNama.setText(data.getStringExtra("nama"));
        txtAngkatan.setText(String.valueOf(data.getIntExtra("angkatan", 0)));
    }
}
```

- l. Buka `TampilActivity.java`, buat *instance object/variable* pada class `TampilActivity` sesuai pada potongan kode di bawah:

```
private DbHelper dbHelper;
private TableLayout tbMhs;
private TableRow tr;
private TextView col1, col2, col3;
private Button btnTutup, btnEdit, btnHapus;
private ArrayList<Mahasiswa> arrListMhs = new ArrayList<>();
private String stb, nama;
private int angkatan;
```


- m. Pada method `OnCreate()` dalam class `TampilActivity` masukkan kode berikut untuk melakukan inisialisasi terhadap instance object yg dibuat sebelumnya, serta untuk memberikan event klik pada button yang terdapat pada layout.

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

    dbHelper = new DbHelper(this);

    tbMhs = findViewById(R.id.tb_mahasiswa);
    btnTutup = findViewById(R.id.btn_tutup);
    btnEdit = findViewById(R.id.btn_edit);
    btnHapus = findViewById(R.id.btn_hapus);

    tampilTabelMhs();

    btnHapus.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            dbHelper.hapusData(dbHelper.getWritableDatabase(), stb);
            tampilTabelMhs();
        }
    });

    btnEdit.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent();
            intent.putExtra("stb", stb);
            intent.putExtra("nama", nama);
            intent.putExtra("angkatan", angkatan);
            dbHelper.close();
            setResult(1, intent);
            finish();
        }
    });

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

- n. Buat method dengan nama `tampilTabelMhs()` pada class `TampilActivity` untuk menampilkan data dari tabel `tb_mhs` pada database `db_mhs`, dengan mengeksekusi method `getArrListMhs()` yang dibuat sebelumnya pada class `DbHelper`. Data berupa object `ArrayList Mahasiswa` akan dikembalikan oleh method `getArrListMhs()`, kemudian isi `ArrayList` tersebut ditampilkan ke dalam object `tbMhs`.

```
public void tampilTabelMhs(){
    tbMhs.removeAllViews();

    arrListMhs = dbHelper.getArrListMhs(dbHelper.getWritableDatabase());

    tr = new TableRow(this);
    col1 = new TextView(this);
    col2 = new TextView(this);
    col3 = new TextView(this);

    col1.setText("Stambuk");
    col2.setText("Nama Mahasiswa");
    col3.setText("Angkatan");

    col1.setWidth(200);
    col2.setWidth(300);
    col3.setWidth(150);

    tr.addView(col1);
    tr.addView(col2);
    tr.addView(col3);

    tbMhs.addView(tr);

    for(final Mahasiswa mhs: arrListMhs){
        tr = new TableRow(this);
        col1 = new TextView(this);
        col2 = new TextView(this);
        col3 = new TextView(this);

        col1.setText(mhs.getStb());
        col2.setText(mhs.getNama());
        col3.setText(String.valueOf(mhs.getAngkatan()));

        col1.setWidth(200);
        col2.setWidth(300);
        col3.setWidth(150);

        tr.addView(col1);
        tr.addView(col2);
        tr.addView(col3);

        tbMhs.addView(tr);

        tr.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                for(int i=0; i<tbMhs.getChildCount(); i++){
                    stb = mhs.getStb();
                    nama = mhs.getNama();
                    angkatan = mhs.getAngkatan();
                    if(tbMhs.getChildAt(i) == view)
                        tbMhs.getChildAt(i).setBackgroundColor(Color.LTGRAY);
                    else
                        tbMhs.getChildAt(i).setBackgroundColor(Color.WHITE);
                }
            }
        });
    }
}
```

LEMBAR EVALUASI PRAKTIKUM

Evaluasi Praktikum 3 :

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) ≤ 40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) ≥ 86
1.						
2.						
3.						
4.						

Catatan Asisten :

Dosen : _____

Asisten 1 : _____

Asisten 2 : _____

TINGKAT KEBERHASILAN

MODUL 4 – ANDROID & MYSQL DATABASE

1. Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu mengimplementasikan penggunaan DBMS MySQL sebagai penyimpanan untuk Android Client.

2. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu mengimplementasikan pembuatan object untuk melakukan request dan menerima response dari web service server menggunakan Library Volley.
2. Mahasiswa mampu mengimplementasikan DDL pada MySQL.
3. Mahasiswa mampu mengimplementasikan pembuatan web service berbasis REST menggunakan PHP untuk pengkoneksian ke MySQL.
4. Mahasiswa mampu mengimplementasikan CRUD pada MySQL Database melalui PHP Web Service.

3. Teori Dasar

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Untuk membuat aplikasi android studio agar terhubung dengan *database*, kita membutuhkan yang namanya REST API (*Representational State Transfer*) yaitu suatu arsitektur komunikasi data menggunakan protokol HTTP untuk menerima dan merespon (*request* dan *response*) data. Dengan kata lain, API berperan sebagai jembatan yang menghubungkan aplikasi dengan aplikasi lainnya.

Umumnya ada 4 metode HTTP yaitu :

GET

POST

PUT

DELETE

4. Kegiatan Praktikum

1. Instrument

- a) Perangkat komputer
- b) Sistem Operasi Windows
- c) IDE Android Studio
- d) Emulator Android
- e) Web Server (Apache)
- f) PHP

2. Prosedur

- Baca dan pahami semua tahapan praktikum dengan cermat.
- Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain.
- Pastikan diri anda tidak menyentuh sumber listrik.

3. Studi Kasus

- Jalankan service yang dibutuhkan:
 - Apache Web Server
 - MySQL
- Buat sebuah database pada MySQL dengan nama `db_akademik`.
- Dalam database `db_akademik` buat tabel `tb_mhs` dengan stuktur sebagai berikut:

Field	Type	Null	Key	Default	Extra
stb	char(11)	NO	PRI	HULL	
nama	varchar(50)	YES		HULL	
angkatan	year	YES		HULL	

- Buat web service sebagai perantara antara MySQL dan Aplikasi Android yang akan dibuat. Web service bertugas menerima request dari Android Client, kemudian melakukan pemrosesan pada sisi server (melakukan query ke database MySQL berdasarkan request) dan mengembalikan hasilnya (response) kembali ke Android Client. Pertama-tama, buat folder untuk web service, jika menggunakan paket aplikasi Xampp, buat folder pada `c:\xampp\htdocs\`, dengan nama `praktikum4`.

Buka Sublime-Text, klik File > Open Folder, buka folder `praktikum4` yang telah dibuat sebelumnya. Buat file dengan nama `Koneksi.php` pada folder `praktikum4`. Klik kanan `praktikum4`, klik new file, masukkan nama `Koneksi.php`. Isi file `Koneksi.php`:

```
<?php
    $userName = "root";
    $password = "root";
    $database = "db_akademik";
    $con = new mysqli("localhost", $userName, $password, $database);
?>
```

Sesuaikan username dan password dari MySQL yang Anda gunakan.

Selanjutnya dalam folder `praktikum4` buat file dengan nama `Mahasiswa.php`, isi filenya:

```
<?php
class Mahasiswa{
    function __construct($stb, $nama, $angkatan){
        $this->stb = $stb;
        $this->nama = $nama;
        $this->angkatan = $angkatan;
    }
}
```

Selanjutnya dalam folder **praktikum4** buat file dengan nama **InsertData.php**, isi filenya:

```
<?php

include 'Koneksi.php';

$hasil = array();

$_POST = json_decode(file_get_contents('php://input'), true);
$stb    = $_POST["stb"];
$nama   = $_POST["nama"];
$angkatan = $_POST["angkatan"];

$ps = $con->stmt_init();
$ps->prepare("insert into tb_mhs values (?, ?, ?)");
$ps->bind_param("ssi", $stb, $nama, $angkatan);
if($ps->execute())
    $hasil['hasil'] = 1;
else
    $hasil['hasil'] = 0;

$ps->close();
$con->close();

echo json_encode($hasil);

?>
```

Selanjutnya dalam folder **praktikum4** buat file dengan nama **TampilData.php**, isi filenya:

```
<?php
include 'Koneksi.php';
include 'Mahasiswa.php';

$hasil = array();
$result = $con->query("select * from tb_mhs");

$i=0;
while($record = $result->fetch_assoc()){
    $hasil[$i] = new Mahasiswa($record["stb"], $record["nama"], $record["angkatan"]);
    $i++;
}

echo json_encode($hasil);

?>
```

Selanjutnya dalam folder praktikum4 buat file dengan nama EditData.php, isi filenya:

```
<?php

include 'Koneksi.php';

$hasil = array();

$_POST = json_decode(file_get_contents('php://input'), true);
$stbLama = $_POST["stb_lama"];
$stb = $_POST["stb"];
$nama = $_POST["nama"];
$angkatan = $_POST["angkatan"];

$ps = $con->stmt_init();
$ps->prepare("update tb_mhs set stb=?, nama=?, angkatan=? where stb=?");
$ps->bind_param("ssis", $stb, $nama, $angkatan, $stbLama);
if($ps->execute())
    $hasil['hasil'] = 1;
else
    $hasil['hasil'] = 0;

$ps->close();
$con->close();

echo json_encode($hasil);

?>
```

Selanjutnya dalam folder praktikum4 buat file dengan nama HapusData.php, isi filenya:

```
<?php

include 'Koneksi.php';

$hasil = array();

$_POST = json_decode(file_get_contents('php://input'), true);
$stb = $_POST["stb"];

$ps = $con->stmt_init();
$ps->prepare("delete from tb_mhs where stb=?");
$ps->bind_param("s", $stb);
if($ps->execute())
    $hasil['hasil'] = 1;
else
    $hasil['hasil'] = 0;

$ps->close();
$con->close();

echo json_encode($hasil);

?>
```

- e. Selanjutnya pembuatan Android Client, buka Android Studio, buat project baru dengan nama sesuai format berikut:
Kelas-Prak4-STB. Contoh: **A1-Prak4-13020190001**
- f. Pada android studio, buka manifests, tambahkan dalam tag <manifest>:

<uses-permission android:name="android.permission.INTERNET" />

- g. Dalam package yang sama dengan `MainActivity.java`, buat class baru dengan nama `RestHelper`.

Dalam package yang sama dengan `MainActivity.java`, buat interface baru dengan nama `RestCallbackMahasiswa`.

Dalam package yang sama dengan `MainActivity.java`, buat class baru dengan nama `Mahasiswa`.

- h. Buka `Mahasiswa.java`, Class `Mahasiswa` berisi:

```
public class Mahasiswa {
    private String stb;
    private String nama;
    private int angkatan;

    private JSONObject jsonObject;
    private Mahasiswa mhs;

    public Mahasiswa(String stb, String nama, int angkatan) {
        this.stb = stb;
        this.nama = nama;
        this.angkatan = angkatan;
    }

    public String getStb() {
        return stb;
    }

    public String getNama() {
        return nama;
    }

    public int getAngkatan() {
        return angkatan;
    }

    public JSONObject toJSON() throws JSONException {
        jsonObject = new JSONObject();
        jsonObject.put("stb", stb);
        jsonObject.put("nama", nama);
        jsonObject.put("angkatan", angkatan);
        return jsonObject;
    }
}
```

- i. Buka `RestCallbackMahasiswa.java`, dalam interface tersebut, tambahkan method abstract sbb:

```
void requestDataMhsSuccess(ArrayList<Mahasiswa> arrayList);
```

- j. Buka `RestHelper.java`, buat instance variable pada class `RestHelper` sbb:

```
private Context context;
private RequestQueue requestQueue;
private final int REQ_METHOD = Request.Method.POST;
private final String URL = "http://192.168.1.6/praktikum4/";
private ArrayList<Mahasiswa> arrListMhs = new ArrayList<>();
```

Untuk object URL, sesuaikan dengan IP Address dari komputer yang Anda gunakan. Untuk melihat IP Address, buka command prompt, eksekusi perintah: `ipconfig`.

Buat konstruktor dari class **RestHelper**:

```
public RestHelper(Context context){
    this.context = context;
    requestQueue = Volley.newRequestQueue(context);
}
```

Buat method **tampilPesan()** dalam class **RestHelper**: untuk menampilkan Toast object:

```
private void tampilPesan(Context context, String teks){
    Toast.makeText(context, teks, Toast.LENGTH_LONG).show();
}
```

Buat method **insertData()** dalam class **RestHelper**: untuk melakukan request insert data ke server:

```
public void insertData(JSONObject jsonObjectMhs){
    JSONObjectRequest jsonObjectRequest;
    Response.Listener<JSONObject> jsonObjectListener;
    Response.ErrorListener errorListener;

    jsonObjectListener = new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {
            try {
                if (response.getInt("hasil") == 1){
                    tampilPesan(context, "Data berhasil disimpan...");
                }
                else{
                    tampilPesan(context, "Data gagal tersimpan...");
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    };

    errorListener = new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            tampilPesan(context, error.toString());
        }
    };

    jsonObjectRequest = new JSONObjectRequest(
        REQ_METHOD,
        URL + "InsertData.php",
        jsonObjectMhs,
        jsonObjectListener,
        errorListener
    );

    requestQueue.add(jsonObjectRequest);
}
```

Buat method `getDataMhs()` dalam class `RestHelper`: untuk mengambil data dari `tb_mhs` pada server, data yang diambil kemudian dimasukkan ke dalam objek `ArrayList Mahasiswa`:

```
public void getDataMhs(final RestCallbackMahasiswa callbackMahasiswa){
    arrListMhs.clear();

    JsonRequest jsonArrayRequest;
    Response.Listener<JSONArray> jsonArrayListener;
    Response.ErrorListener errorListener;

    jsonArrayListener = new Response.Listener<JSONArray>() {
        @Override
        public void onResponse(JSONArray response) {
            for(int i=0; i<response.length(); i++){
                try {
                    arrListMhs.add(new Mahasiswa(
                        response.getJSONObject(i).getString("stb"),
                        response.getJSONObject(i).getString("nama"),
                        response.getJSONObject(i).getInt("angkatan")
                    ));
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
            callbackMahasiswa.requestDataMhsSuccess(arrListMhs);
        }
    };

    errorListener = new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            tampilPesan(context, error.toString());
        }
    };

    jsonArrayRequest = new JsonRequest(
        REQ_METHOD,
        URL + "TampilData.php",
        null,
        jsonArrayListener,
        errorListener
    );

    requestQueue.add(jsonArrayRequest);
}
```

Buat method `editData()` dalam class `RestHelper`: untuk melakukan edit data pada record `tb_mhs` pada server.

```
public void editData(String stb, Mahasiswa mhs) {
    JsonObjectRequest jsonObjectRequest;
    JSONObject jsonObject;
    Response.Listener<JSONObject> jsonObjectListener;
    Response.ErrorListener errorListener;

    jsonObject = new JSONObject();
    try {
        jsonObject.put("stb_lama", stb);
        jsonObject.put("stb", mhs.getStb());
        jsonObject.put("nama", mhs.getNama());
        jsonObject.put("angkatan", mhs.getAngkatan());
    } catch (JSONException e) {
        e.printStackTrace();
    }
    System.out.println("data json:" + jsonObject.toString());
    jsonObjectListener = new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {
            try {
                if (response.getInt("hasil") == 1){
                    tampilPesan(context, "Data berhasil disimpan...");
                }
                else{
                    tampilPesan(context, "Data gagal tersimpan...");
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    };

    errorListener = new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            tampilPesan(context, error.toString());
        }
    };

    jsonObjectRequest = new JsonObjectRequest(
        REQ_METHOD,
        URL + "EditData.php",
        jsonObject,
        jsonObjectListener,
        errorListener
    );

    requestQueue.add(jsonObjectRequest);
}
```

Buat method `hapusData()` dalam class `RestHelper`: untuk melakukan hapus record pada `tb_mhs` pada server berdasarkan isi dari parameter `stb`:

```
public void hapusData(String stb, final RestCallbackMahasiswa callbackMahasiswa){
    JsonObjectRequest jsonObjectRequest;
    JSONObject jsonObject;
    Response.Listener<JSONObject> jsonObjectListener;
    Response.ErrorListener errorListener;

    jsonObject = new JSONObject();
    try {
        jsonObject.put("stb", stb);
    } catch (JSONException e) {
        e.printStackTrace();
    }
    System.out.println("data json:" + jsonObject.toString());
    jsonObjectListener = new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {
            try {
                if (response.getInt("hasil") == 1){
                    tampilPesan(context, "Data berhasil dihapus...");
                }
                else{
                    tampilPesan(context, "Data gagal dihapus...");
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
            getDataMhs(callbackMahasiswa);
        }
    };

    errorListener = new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            tampilPesan(context, error.toString());
        }
    };

    jsonObjectRequest = new JsonObjectRequest(
        REQ_METHOD,
        URL + "HapusData.php",
        jsonObject,
        jsonObjectListener,
        errorListener
    );

    requestQueue.add(jsonObjectRequest);
}
```

- k. Pada layout `activity_main.xml`, buat tampilannya sesuai pada gambar di bawah, sesuaikan dengan id yang telah diberikan.

Stambuk	txt_stb
Nama Mahasiswa	txt_nama
Angkatan	txt_angkatan
<div>SIMPAN</div> <div>TAMPIL</div>	

Buat *Activity* baru dengan nama `TampilDataActivity`, sesuaikan layout dan id-nya sesuai pada gambar di bawah.

Stambuk	Nama Mahasiswa	Angkatan	tbl_mhs
13020190001	Hasan Basri	2019	
<div>EDIT</div>		<div>HAPUS</div>	

1. Buat *instance object/variable* pada class **MainActivity** sesuai pada potongan kode di bawah:

```
private EditText txtStb, txtNama, txtAngkatan;
private RestHelper restHelper;
private Mahasiswa mhs;
private Intent intentEdit;
```

Pada method **OnCreate()** dalam class **MainActivity** masukkan kode berikut untuk melakukan inisialisasi terhadap *instance object* yg dibuat sebelumnya.

```
restHelper = new RestHelper(this);
intentEdit = null;

txtStb      = findViewById(R.id.txt_stb);
txtNama     = findViewById(R.id.txt_nama);
txtAngkatan = findViewById(R.id.txt_angkatan);
```

Buat method untuk mengosongkan inputan pada **MainActivity**. Pada class **MainActivity** tambahkan method **clearData()**:

```
private void clearData() {
    txtStb.setText("");
    txtNama.setText("");
    txtAngkatan.setText("");
    intentEdit = null;
    txtStb.requestFocus();
}
```

Buat method untuk event klik pada **Button Simpan** dengan nama **btnSimpanClick()** pada **MainActivity** untuk menyimpan data dalam tabel **tb_mhs** :

```
public void btnSimpanClick(View view) {
    mhs = new Mahasiswa(
        txtStb.getText().toString(),
        txtNama.getText().toString(),
        Integer.parseInt(txtAngkatan.getText().toString())
    );

    try {
        if (intentEdit == null)
            restHelper.insertData(mhs.toJSON());
        else
            restHelper.editData(intentEdit.getStringExtra("stb"), new Mahasiswa(
                txtStb.getText().toString(),
                txtNama.getText().toString(),
                Integer.parseInt(txtAngkatan.getText().toString())
            ));
    } catch (JSONException e) {
        e.printStackTrace();
    }
    clearData();
}
```

Buat method untuk event klik pada Button Tampil dengan nama `btnTampilDataClick()` pada `MainActivity` untuk membuka `TampilDataActivity`.

```
public void btnTampilDataClick(View view){
    intentEdit = null;
    Intent intent = new Intent(this, TampilDataActivity.class);
    startActivityForResult(intent, 1);
}
```

Lakukan *override* terhadap method `onActivityResult()` pada class `MainActivity`.

```
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == 1){
        intentEdit = data;
        txtStb.setText(data.getStringExtra("stb"));
        txtNama.setText(data.getStringExtra("nama"));
        txtAngkatan.setText(data.getStringExtra("angkatan"));
    }
}
```

- m. Buat *instance object/variable* pada class `TampilDataActivity` sesuai pada potongan kode di bawah:

```
private TableLayout tblMhs;
private TableRow tr;
private TextView col1, col2, col3;
private RestHelper restHelper;
private String stb, nama, angkatan;
```

Pada method `OnCreate()` dalam class `TampilDataActivity` masukkan kode berikut untuk melakukan inisialisasi terhadap instance object yg dibuat sebelumnya.

```
restHelper = new RestHelper(this);
tblMhs = findViewById(R.id.tbl_mhs);
tampilData();
```


Buat method dengan nama `tampilTblMhs()` pada class `TampilDataActivity` untuk menampilkan data dari objek `ArrayList Mahasiswa` ke dalam tabel `tblMhs`.

```
private void tampilTblMhs(ArrayList<Mahasiswa> arrListMhs) {
    tblMhs.removeAllViews();

    tr = new TableRow(this);
    col1 = new TextView(this);
    col2 = new TextView(this);
    col3 = new TextView(this);

    col1.setText("Stambuk");
    col2.setText("Nama Mahasiswa");
    col3.setText("Angkatan");

    col1.setWidth(200);
    col2.setWidth(300);
    col3.setWidth(150);

    tr.addView(col1);
    tr.addView(col2);
    tr.addView(col3);

    tblMhs.addView(tr);

    for(final Mahasiswa mhs: arrListMhs){
        tr = new TableRow(this);
        col1 = new TextView(this);
        col2 = new TextView(this);
        col3 = new TextView(this);

        col1.setText(mhs.getStb());
        col2.setText(mhs.getNama());
        col3.setText(String.valueOf(mhs.getAngkatan()));

        col1.setWidth(200);
        col2.setWidth(300);
        col3.setWidth(150);

        tr.addView(col1);
        tr.addView(col2);
        tr.addView(col3);

        tblMhs.addView(tr);

        tr.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                for(int i=0; i<tblMhs.getChildCount(); i++){
                    stb = mhs.getStb();
                    nama = mhs.getNama();
                    angkatan = String.valueOf(mhs.getAngkatan());
                    if (tblMhs.getChildAt(i) == view)
                        tblMhs.getChildAt(i).setBackgroundColor(Color.LTGRAY);
                    else
                        tblMhs.getChildAt(i).setBackgroundColor(Color.WHITE);
                }
            }
        });
    }
}
```

- n. Buat method dengan nama `tampilData()` pada class `TampilDataActivity` untuk melakukan *request* dan memasukkan hasil *response*-nya dari web service ke objek `ArrayList` Mahasiswa.

```
private void tampilData() {
    restHelper.getDataMhs(new RestCallbackMahasiswa() {
        @Override
        public void requestDataMhsSuccess(ArrayList<Mahasiswa> arrayList) {
            tampilTblMhs(arrayList);
        }
    });
}
```

Buat method untuk event klik pada `Button Edit` dengan nama `btnEditClick()` pada `TampilDataActivity` untuk melakukan proses edit pada `MainActivity`.

```
public void btnEditClick(View view) {
    Intent intent = new Intent();
    intent.putExtra("stb", stb);
    intent.putExtra("nama", nama);
    intent.putExtra("angkatan", angkatan);
    setResult(1, intent);
    finish();
}
```

Buat method untuk event klik pada `Button Hapus` dengan nama `btnHapusClick()` pada `TampilDataActivity` untuk melakukan proses penghapusan *record*.

```
public void btnHapusClick(View view) {
    if (stb == null) return;
    restHelper.hapusData(stb, new RestCallbackMahasiswa() {
        @Override
        public void requestDataMhsSuccess(ArrayList<Mahasiswa> arrayList) {
            tampilTblMhs(arrayList);
        }
    });
}
```

Jalankan Aplikasi pada emulator

LEMBAR EVALUASI PRAKTIKUM

Evaluasi Praktikum 4 :

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) ≤ 40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) ≥ 86
1.						
2.						
3.						
4.						

Catatan Asisten :

Dosen : _____

Asisten 1 : _____

Asisten 2 : _____

TINGKAT KEBERHASILAN