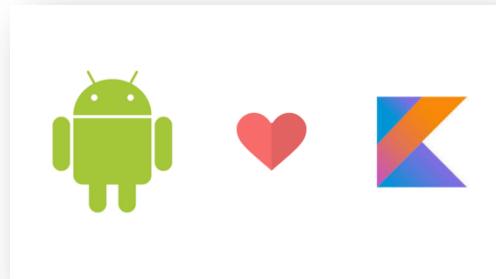


Android App Development Lanjutan dan Membuat RESTful API



Praktikum 5

Tujuan :

Mahasiswa mengenal dan mengimplementasikan :

- Shared Preferences
- Internal Storage
- Menampilkan Data Menggunakan Listview dan Recyclerview
- Membuat Navigation Drawer Menggunakan Fragment
- Membuat RESTful API pada Local Web Server
- Menguji Akses Layanan API pada Local Web Server

Hasil Praktikum :

- Contoh program praktikum.

Shared Preferences (Preferensi Bersama)

Preferensi bersama adalah kamus objek yang menggunakan mekanisme struktur pasangan *key* dan *value* untuk menyimpan data sederhana kedalam memori internal Android. Android menyimpan data preferensi bersama dalam memori internal sebagai file XML. Kita dapat menyimpan beberapa tipe data sederhana seperti String dan tipe dasar lainnya. Perhatikan sintaks preferensi bersama berikut untuk menyimpan data:

```
1 val pref = getPreferences(Context.MODE_PRIVATE)
2 val editor = pref.edit()
3 editor.putString("lastname", "Situmorang")
4 editor.putString("firstname", "Anton")
5 editor.apply()
```

Baris kode ke-1 adalah membuat objek preferensi bersama dengan memanggil method getPreferences(). Dalam method getPreferences() ini terdapat sebuah parameter pilihan mode Context.

Baris ke-2 untuk mendapatkan objek editor preferensi bersama dengan memanggil method edit().

Baris ke-3 dan ke-4 memasukkan data kedalam objek editor

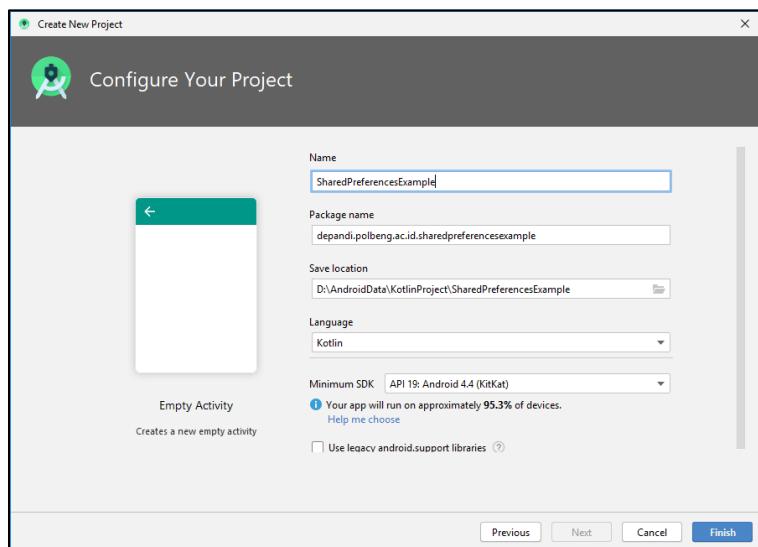
Baris ke-5 menyimpan data secara permanen, kita gunakan method commit atau apply pada objek editor.

Terdapat empat pilihan mode Context yaitu:

1. MODE_PRIVATE: adalah mode default, dimana file yang dibuat hanya bisa diakses oleh aplikasi pemanggil.
2. MODE_WORLD_READABLE: apapun aplikasi dapat membaca data preferensi bersama. Ini mungkin akan menjadi celah keamanan dari aplikasi. Sebaiknya gunakan content provider jika data dibagikan ke aplikasi yang lain.
3. MODE_WORLD_WRITEABLE: apapun aplikasi dapat merubah data preferensi bersama.
4. MODE_APPEND: mode ini akan menambahkan preferensi baru dari preferensi yang sudah ada.

Membuat Aplikasi Preferensi Bersama

1. Buatlah project baru dengan konfigurasi seperti berikut :



2. Setelah selesai membuat project baru, selanjutnya editlah file *strings.xml* pada folder *values*, tambahkan beberapa variabel dan nilai *string* sebagai berikut:

```
<resources>
    <string name="app_name">SharedPreferencesExample</string>
    <string name="save">save</string>
    <string name="load">load</string>
    <string name="first_name">First Name</string>
    <string name="last_name">Last Name</string>
    <string name="textview">TextView</string>
</resources>
```

3. Editlah file *activity_main.xml* sehingga kode *layout* menjadi seperti berikut :

```
<?xml version="1.0" encoding="utf-8" ?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/btnsave"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:text="@string/save"
        app:layout_constraintBaseline_toBaselineOf="@+id(btnload)"
        app:layout_constraintStart_toStartOf="parent" />

    <Button
        android:id="@+id	btnload"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="11dp"
        android:layout_marginTop="27dp"
        android:text="@string/load"
        app:layout_constraintEnd_toEndOf="@+id/txtlastname"
        app:layout_constraintTop_toBottomOf="@+id/txtlastname" />

    <EditText
        android:id="@+id/txtfirstname"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="36dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:hint="@string/first_name"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:autofillHints="" />

    <EditText
        android:id="@+id/txtlastname"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="11dp"
        android:layout_marginTop="27dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:hint="@string/last_name"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtfirstname" />
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:hint="@string/last_name"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtfirstname"
        android:autofillHints="" />

    <TextView
        android:id="@+id/txtoutput"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="183dp"
        android:layout_marginStart="16dp"
        android:text="@string/textview"
        android:textSize="36sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

4. Buka file sumber *MainActivity.kt* editlah kode program hingga menjadi seperti berikut :

```

package depandi.porbeng.ac.id.sharedpreferencesexample

import android.content.Context
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Toast
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val pref = getPreferences(Context.MODE_PRIVATE)
        btnsave.setOnClickListener {
            val editor = pref.edit()
            editor.putString("lastname", txtlastname.text.toString())
            editor.putString("firstname", txtfirstname.text.toString())
            editor.apply()
            Toast.makeText(this, "Saved data", Toast.LENGTH_LONG).show()
        }
        btnload.setOnClickListener {
            val mlastname = pref.getString("lastname", "")
            val mfirstname = pref.getString("firstname", "")
            val moutput = "$firstname $lastname"
            txtoutput.text = moutput
        }
    }

    override fun onResume() {
        super.onResume()
    }
}

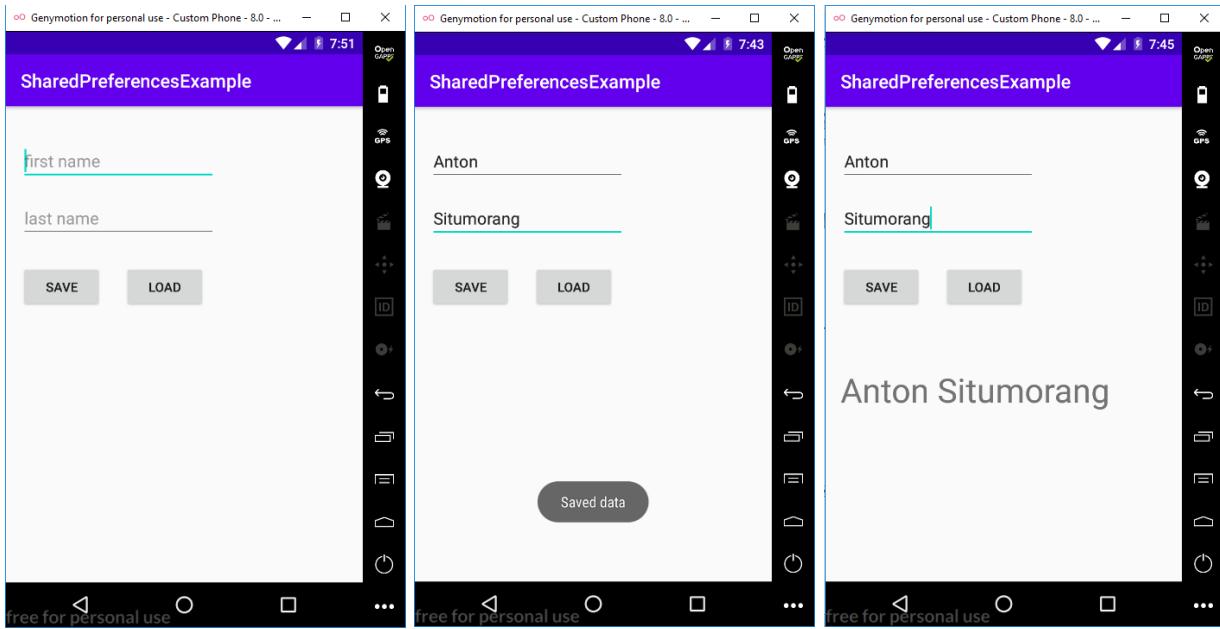
```

```

        txtfirstname.setText("")
        txtlastname.setText("")
        txtfirstname.setHint("first name")
        txtlastname.setHint("last name")
        txtoutput.setText("")
    }
}

```

5. Jalankan program di Emulator Android. Perhatikan hasilnya dan lakukan analisa terhadap kode layout *activity_main.xml* dan kode logika program *ActivityMain.kt* dilaporan praktikum.

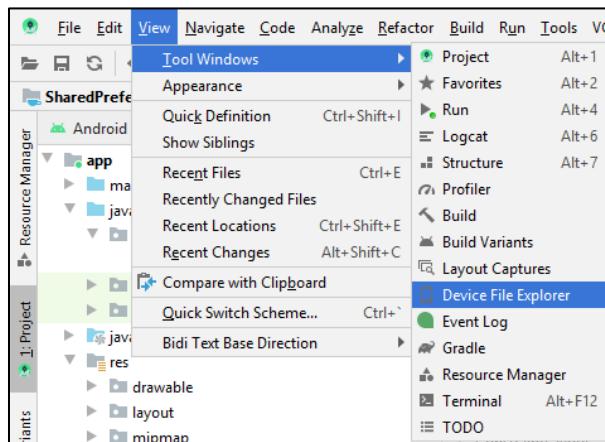


Tampilan Awal

Menyimpan Data

Menampilkan Data

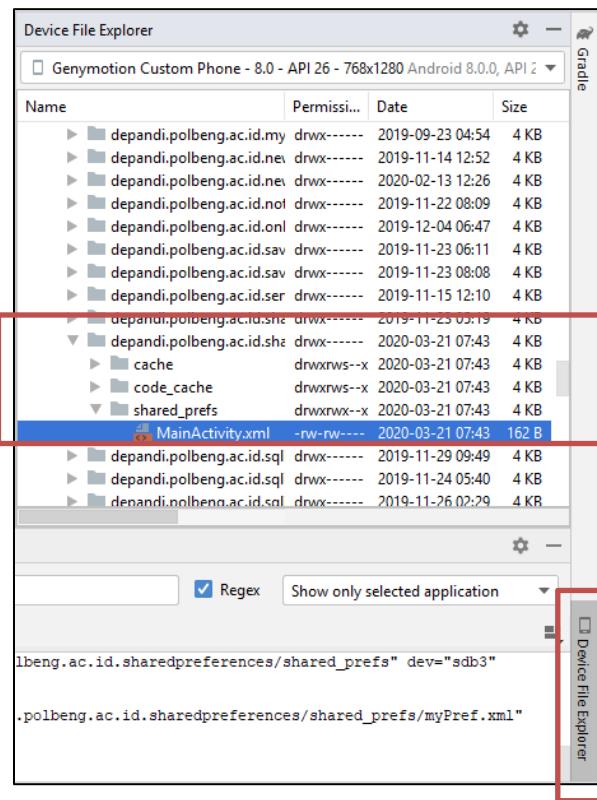
Untuk melihat file preferensi bersama yang telah dibuat pergilah ke tampilan Device File Explorer dengan cara klik **View → Tool Windows → Device File Explorer** atau dengan cara klik tulisan **Device File Explorer** paling kanan bawah Android Studio IDE.



Setelah muncul tampilan Device File Explorer selanjutnya carilah data *MainActivity.xml* dalam struktur folder *data* → *data* → *nama_lengkap_project_anda* → *shared_prefs* → *MainActivity.xml*.

Contoh nama lengkap project adalah seperti berikut :

depandi.polbeng.ac.id.sharedpreferencesexample

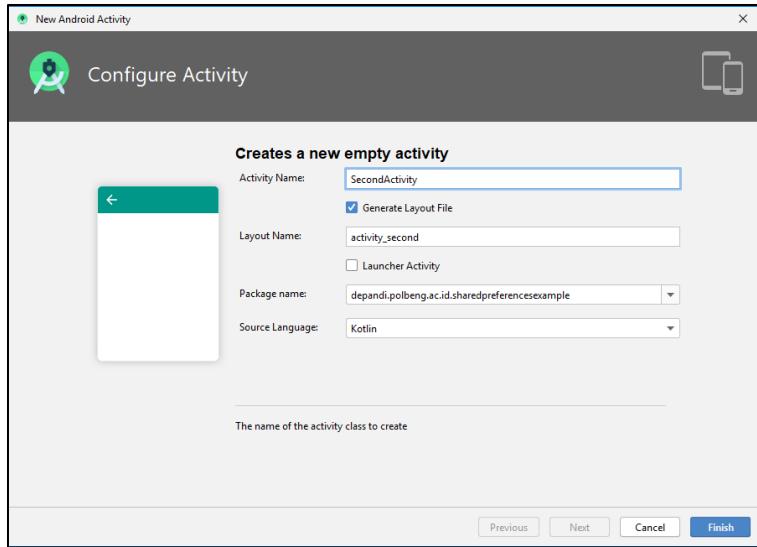


Selanjutnya klik kanan pada file MainActivity.xml pilih Save as... untuk menyimpan file preferensi bersama yang dibuat di folder lokal disk anda. Kemudian ditampilkan file yang disimpan tersebut kedalam editor teks anda seperti notepad++. Isi filenya adalah seperti berikut:

```
D:\MainActivity.xml - Notepad++
Berkas Edit Cari Tampilan Enkode Bahasa Pengaturan Alat Makro Jalankan Plugin Jendela ?
MainActivity.xml
1 <?xml version='1.0' encoding='utf-8' standalone='yes'?>
2 <map>
3   <string name="firstname">Anton</string>
4   <string name="lastname">Situdmorang</string>
5 </map>
6
length : 162 lines : 6 Ln : 4 Col : 48 Sel : 0 | 0 Unix (LF) UTF-8 INS
```

Membagi Data Preferensi Bersama Antar Activity

6. Buatlah activity baru dengan nama *SecondActivity* dan tipe Empty Activity pada project SharedPreferencesExample.



7. Editlah file strings.xml pada folder values, tambahkan beberapa variabel dan nilai string sebagai berikut:

```
<resources>
    <string name="app_name">SharedPreferencesExample</string>
    <string name="save">save</string>
    <string name="load">load</string>
    <string name="first_name">First Name</string>
    <string name="last_name">Last Name</string>
    <string name="textview">TextView</string>
    <string name="shared_preferences">Shared Preferences</string>
    <string name="next_activity">Next Activity</string>
</resources>
```

8. Editlah file activity_main.xml sehingga kode layout menjadi seperti berikut :

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

    <LinearLayout
        android:id="@+id/l1InputText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/tvJudul"
        android:layout_marginTop="20dp"
        android:orientation="vertical">

        <EditText
            android:id="@+id/txtfirstname"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/first_name"
            android:inputType="textPersonName"
            android:autofillHints="" />
```

```
<EditText
    android:id="@+id/txtlastname"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:hint="@string/last_name"
    android:inputType="textPersonName"
    android:autofillHints="" />
</LinearLayout>

<LinearLayout
    android:id="@+id/l1Button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/l1InputText"
    android:layout_marginTop="20dp"
    android:orientation="horizontal"
    android:weightSum="10">

    <Button
        android:id="@+id/btnsave"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="3"
        android:text="@string/save"/>

    <Button
        android:id="@+id(btnload"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="3"
        android:text="@string/load"/>

    <Button
        android:id="@+id/btnSecondActivity"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="4"
        android:text="@string/next_activity"/>
</LinearLayout>

<TextView
    android:id="@+id/tvJudul"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    android:text="@string/shared_preferences"
    android:textAlignment="center"
    android:textSize="30sp"
    android:textStyle="bold" />

<TextView
    android:id="@+id/txtoutput"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
```

```

        android:layout_marginBottom="20dp"
        android:text="@string/textview"
        android:textAlignment="center"
        android:textSize="36sp" />

</RelativeLayout>

```

9. Bukanlah file sumber *MainActivity.kt* editlah kode program hingga menjadi seperti berikut :

```

package depandi.polbeng.ac.id.sharedpreferencesexample

import android.content.Context
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Toast
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val filename = "$packageName TESTFILE"
        val pref = getSharedPreferences(filename, Context.MODE_PRIVATE)

        btnsave.setOnClickListener {
            val editor = pref.edit()
            editor.putString("lastname", txtlastname.text.toString())
            editor.putString("firstname", txtfirstname.text.toString())
            editor.apply()
            Toast.makeText(this, "Saved data", Toast.LENGTH_LONG).show()
        }

        btnload.setOnClickListener {
            val mlastname = pref.getString("lastname", "")
            val mfirstname = pref.getString("firstname", "")
            val moutput = "$firstname $lastname"
            txtoutput.text = moutput
        }

        btnSecondActivity.setOnClickListener{
            val intent = Intent(this@MainActivity,
SecondActivity::class.java)
            startActivity(intent)
        }
    }

    override fun onResume() {
        super.onResume()
        txtfirstname.setText("")
        txtlastname.setText("")
        txtoutput.setText("")
    }
}

```

10. Editlah file *activity_second.xml* sehingga kode *layout* menjadi seperti berikut :

```

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

    <TextView
        android:id="@+id/txtoutput"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="TextView"
        android:textSize="30sp"
        android:textAlignment="center"
        android:textStyle="bold"/>

    <Button
        android:id="@+id	btnloaddata"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Load Data"
        android:layout_marginTop="20dp"/>

</LinearLayout>

```

11. Buka file sumber *SecondActivity.kt* editlah kode program hingga menjadi seperti berikut :

```

package depandi.polbeng.ac.id.sharedpreferencesexample

import android.content.Context
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_second.*

class SecondActivity : AppCompatActivity() {

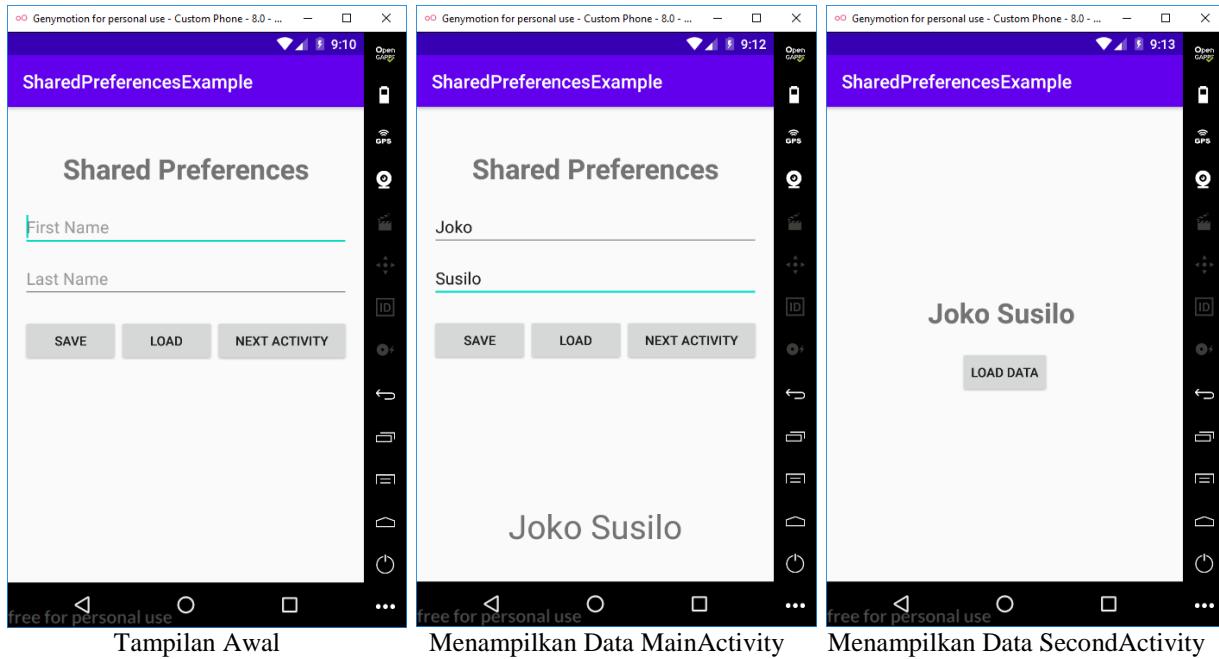
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_second)

        btnloaddata.setOnClickListener {
            val filename = "$packageName TESTFILE"
            val pref = getSharedPreferences(filename,
Context.MODE_PRIVATE)
            val mlastname = pref.getString("lastname", "")
            val mfirstname = pref.getString("firstname", "")
            txtoutput.text = "$firstname $lastname"
        }
    }

    override fun onResume() {
        super.onResume()
        txtoutput.text = "Click 'LOAD DATA'"
    }
}

```

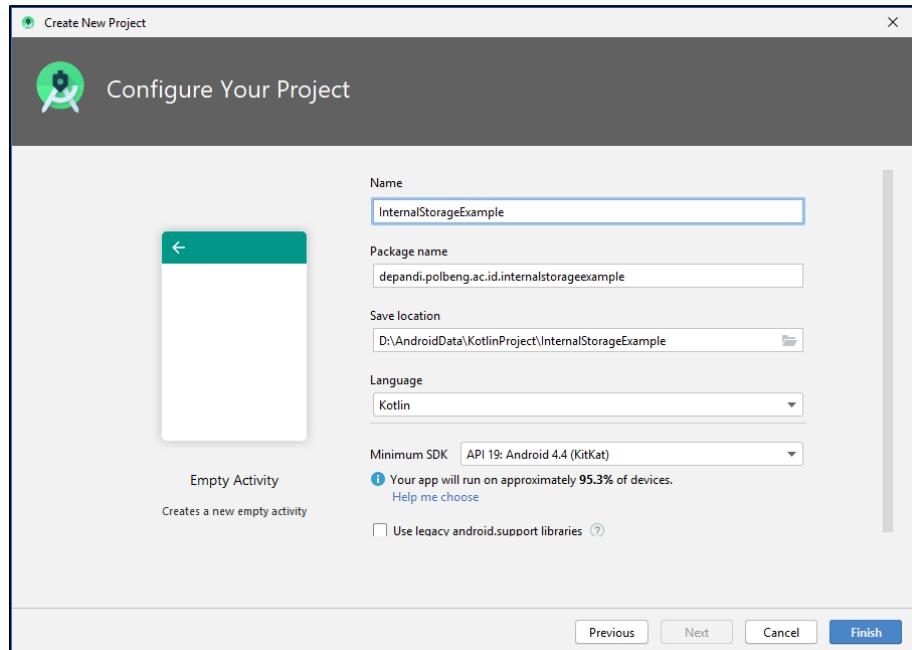
12. Jalankan program di Emulator Android. Perhatikan hasilnya dan lakukan analisa terhadap kode sumber *activity_main.xml*, *activity_second.xml* dan *ActivityMain.kt*, *ActivitySecond.kt* dilaporan praktikum.



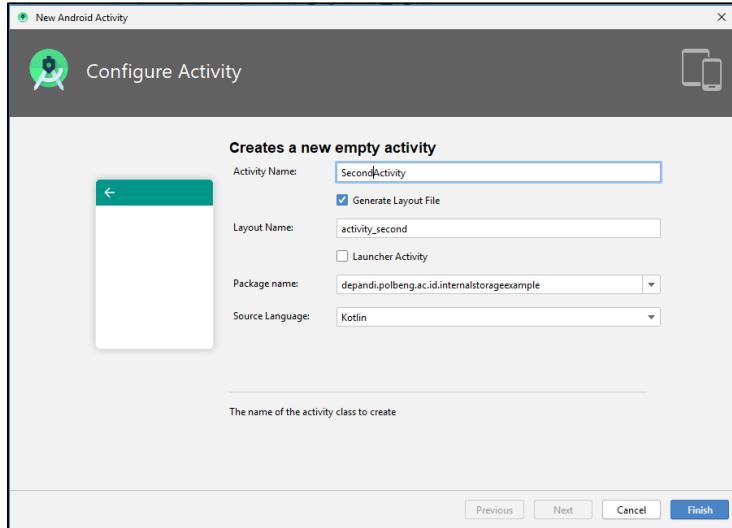
Internal Storage

Menyimpan data teks (.txt) pada Internal Storage

- Buatlah *project* baru dengan konfigurasi seperti berikut :



2. Setelah selesai membuat project baru, buatlah *activity* baru dengan nama SecondActivity dan tipe **Empty Activity**.



3. Selanjutnya editlah file *strings.xml* pada folder *values*, tambahkan beberapa variabel dan nilai *string* sebagai berikut:

```
<resources>
    <string name="app_name">InternalStorageExample</string>
    <string name="_2nd_activity">2nd Activity</string>
    <string name="input_text">Input text here...</string>
    <string name="_1st_activity">1st activity</string>
    <string name="textview">TextView</string>
</resources>
```

4. Editlah file *activity_main.xml* sehingga kode *layout* menjadi seperti berikut :

```
<?xml version="1.0" encoding="utf-8" ?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:padding="16dp">

    <Button
        android:id="@+id/btnsecondactivity"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:layout_marginTop="8dp"
        android:text="@string/_2nd_activity"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtinput"
        app:layout_constraintVertical_bias="0.963"
        tools:layout_editor_absoluteX="16dp" />
```

```

<EditText
    android:id="@+id/txtinput"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="34dp"
    android:ems="10"
    android:inputType="textMultiLine"
    android:hint="@string/input_text"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

5. Buka file sumber *MainActivity.kt* editlah kode program hingga menjadi seperti berikut :

```

package depandi.polbeng.ac.id.internalstorageexample

import android.content.Context
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.widget.Toast
import kotlinx.android.synthetic.main.activity_main.*
import java.io.FileNotFoundException
import java.io.IOException

class MainActivity : AppCompatActivity() {

    private val TAG = "MainActivity"

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        btnsecondactivity.setOnClickListener {
            startActivity(Intent(this, SecondActivity::class.java))
        }
    }

    private fun saveData() {
        val filename = "ourfile.txt"
        Thread(Runnable {
            try {
                val out = openFileOutput(filename, Context.MODE_PRIVATE)
                out.use {
                    out.write(txtinput.text.toString().toByteArray())
                }
                runOnUiThread(Runnable {
                    Toast.makeText(this, "Saved",
                    Toast.LENGTH_LONG).show()
                })
            } catch (ioe: IOException) {
                Log.w(TAG, "Error while saving ${filename} : ${ioe}")
            }
        }).start()
    }
}

```

```

    }

    override fun onPause() {
        super.onPause()
        saveData()
    }

    override fun onResume() {
        super.onResume()
        loadData()
    }

    private fun loadData() {
        val filename = "ourfile.txt"
        Thread(Runnable {
            try {
                val input = openFileInput(filename)
                input.use {
                    var buffer = StringBuilder()
                    var bytes_read = input.read()
                    while (bytes_read != -1) {
                        buffer.append(bytes_read.toChar())
                        bytes_read = input.read()
                    }
                    runOnUiThread(Runnable {
                        txtinput.setText(buffer.toString())
                    })
                }
            } catch (fnfe: FileNotFoundException) {
                Log.w(TAG, "file not found, occurs only once")
            } catch (ioe: IOException) {
                Log.w(TAG, "IOException : $ioe")
            }
        }).start()
    }
}
}

```

6. Editlah file *activity_second.xml* sehingga kode *layout* menjadi seperti berikut :

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SecondActivity"
    android:padding="16dp">

    <Button
        android:id="@+id/btnmainactivity"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginBottom="18dp"
        android:text="@string/_1st_activity"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

```

```

<TextView
    android:id="@+id/txtoutput"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="29dp"
    android:inputType="textMultiLine"
    android:text="@string/textview"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

7. Bukalah file sumber *SecondActivity.kt* editlah kode program hingga menjadi seperti berikut :

```

package depandi.polbeng.ac.id.internalstorageexample

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_second.*

class SecondActivity : AppCompatActivity() {

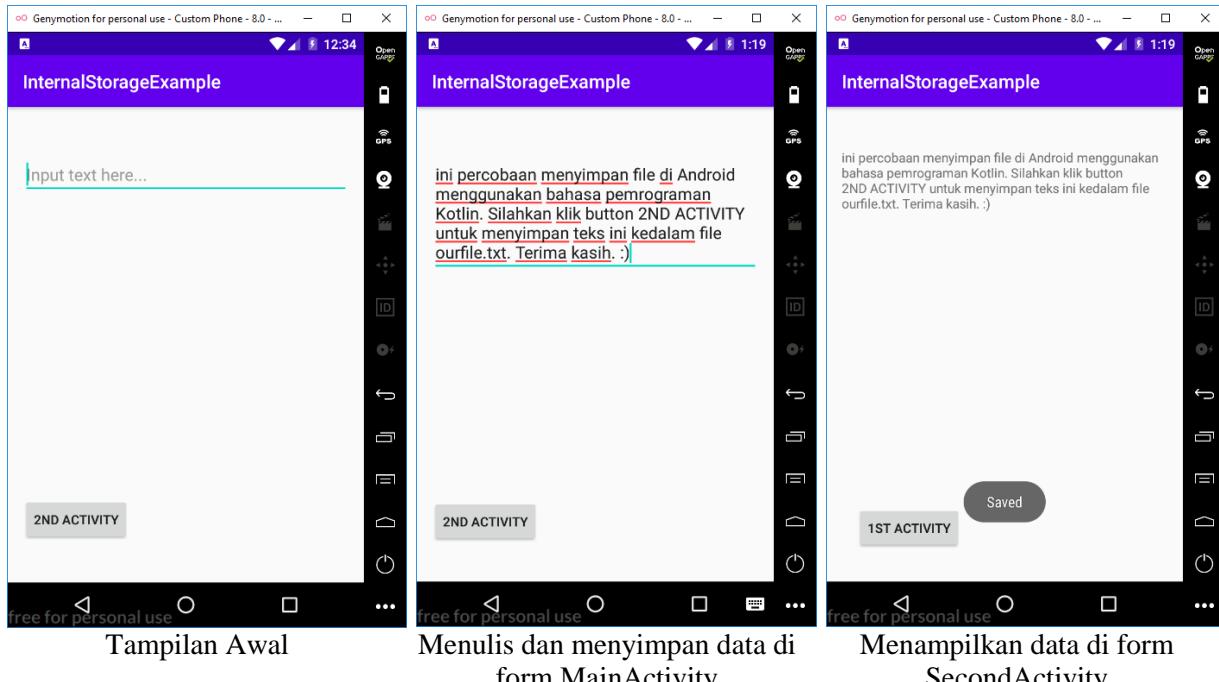
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_second)
        btnmainactivity.setOnClickListener {
            startActivity(Intent(this, MainActivity::class.java))
        }
    }

    override fun onResume() {
        super.onResume()
        loadData()
    }

    private fun loadData() {
        val filename = "ourfile.txt"
        Thread(Runnable {
            val input = openFileInput(filename)
            input.use {
                var buffer = StringBuilder()
                var bytes_read = input.read()
                while (bytes_read != -1) {
                    buffer.append(bytes_read.toChar())
                    bytes_read = input.read()
                }
                runOnUiThread(Runnable {
                    txtoutput.setText(buffer.toString())
                })
            }
        }).start()
    }
}

```

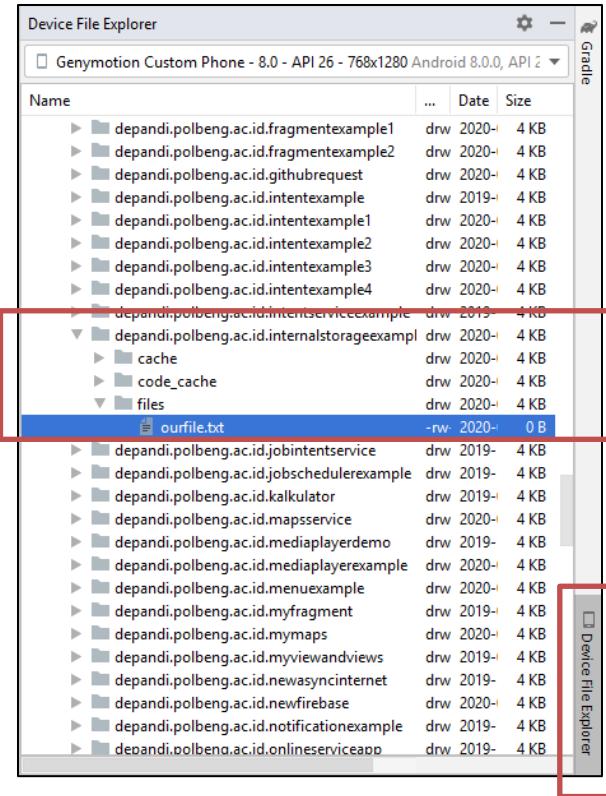
8. Jalankan program di Emulator Android. Perhatikan hasilnya dan lakukan analisa terhadap kode sumber *activity_main.xml*, *activity_second.xml* dan *ActivityMain.kt*, *ActivitySecond.kt* dilaporan praktikum.



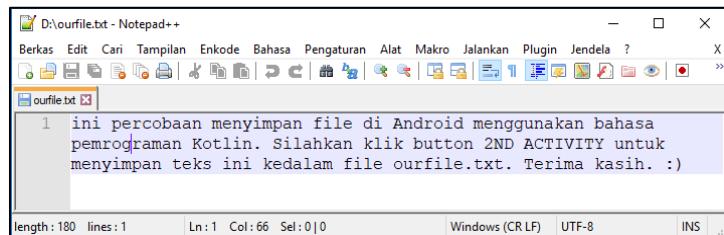
Carilah data *MainActivity.xml* pada Device File Explorer dalam struktur folder *data* → *data* → *nama_lengkap_project_anda* → *files* → *ourfile.txt*.

Contoh nama lengkap project adalah seperti berikut :

depandi.polbeng.ac.id. internalstorageexample

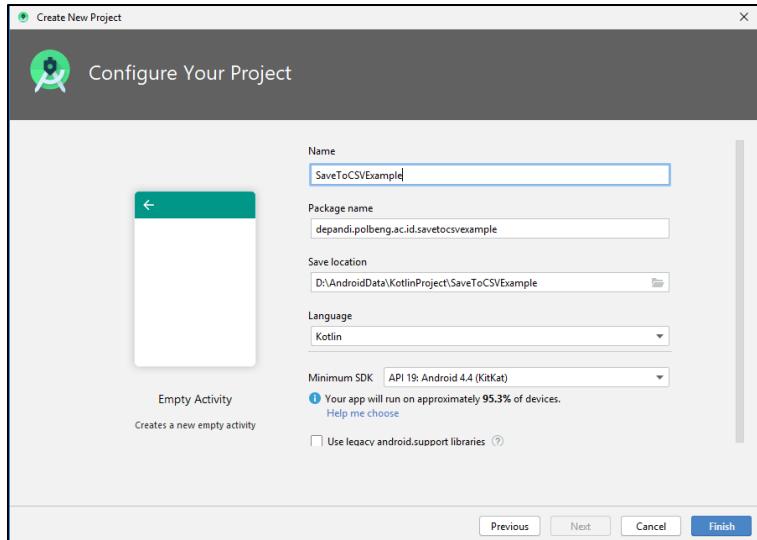


Selanjutnya klik kanan pada file `ourfile.txt` pilih `Save as...` untuk menyimpan file preferensi bersama yang dibuat di folder lokal disk anda. Kemudian ditampilkan file yang disimpan tersebut kedalam *editor* teks anda seperti notepad++. Isi filenya adalah seperti berikut:



Menyimpan data dalam format csv

1. Buatlah *project* baru dengan konfigurasi seperti berikut :



2. Setelah selesai membuat project baru, selanjutnya bukalah file konfigurasi module build.gradle(Module: app), tambahkan library tambahan kotlin-csv yang diberi tanda kotak merah pada gambar berikut. Klik **Sync Now** untuk menerapkan penambahan *library* dalam project.

```

21     minifyEnabled false
22     proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
23 }
24 }
25
26 }
27
28 dependencies {
29     implementation fileTree(dir: 'libs', include: ['*.jar'])
30     implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
31     implementation 'androidx.appcompat:appcompat:1.1.0'
32     implementation 'androidx.core:core-ktx:1.2.0'
33     implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
34     implementation 'com.github.doyaaaaaken:kotlin-csv-jvm:0.7.3' █
35     testImplementation 'junit:junit:4.12'
36     androidTestImplementation 'androidx.test.ext:junit:1.1.1'
37     androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
38 }

```

3. Setelah selesai membuat project baru, selanjutnya editlah file strings.xml pada folder values, tambahkan beberapa variabel dan nilai string sebagai berikut:

```

<resources>
    <string name="app_name">SaveToCSVExample</string>
    <string name="read_data">Read Data</string>
    <string name="save_data">Save Data</string>
    <string name="x_value">X Value</string>
    <string name="y_value">Y Value</string>
    <string name="z_value">Z Value</string>
    <string name="data_csv">Data CSV</string>
</resources>

```

4. Editlah file *activity_main.xml* sehingga kode *layout* menjadi seperti berikut :

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

    <Button
        android:id="@+id/btnRead"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:text="@string/read_data"
        android:onClick="readData"/>

    <Button
        android:id="@+id/btnSave"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/save_data"
        android:layout_above="@+id/btnRead"
        android:layout_centerHorizontal="true"
        android:onClick="saveData"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_alignParentTop="true"
        android:id="@+id/linearLayout">

        <EditText
            android:id="@+id/etX"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:hint="@string/x_value"
            android:inputType="numberDecimal"
            android:textAlignment="center" />

        <EditText
            android:id="@+id/etY"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:hint="@string/y_value"
            android:inputType="numberDecimal"
            android:textAlignment="center" />

        <EditText
            android:id="@+id/etZ"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
```

```

        android:layout_weight="1"
        android:hint="@string/z_value"
        android:inputType="numberDecimal"
        android:textAlignment="center" />
    </LinearLayout>

    <TextView
        android:padding="10dp"
        android:id="@+id/tvData"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/linearLayout"
        android:text="@string/data_csv" />

</RelativeLayout>

```

5. Buka file sumber *MainActivity.kt* editlah kode program hingga menjadi seperti berikut :

```

package depandi.polbeng.ac.id.savetocsvexample

import android.Manifest
import android.app.Activity
import android.content.pm.PackageManager
import android.os.Build
import android.os.Bundle
import android.os.Environment
import android.view.View
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import com.github.doyaaaaaken.kotlincsv.dsl.csvReader
import com.github.doyaaaaaken.kotlincsv.dsl.csvWriter
import kotlinx.android.synthetic.main.activity_main.*
import java.io.File
import java.io.IOException

class MainActivity : AppCompatActivity() {

    var baseDir = Environment.getExternalStorageDirectory().absolutePath
    var fileName = "test.csv"
    var filePath = baseDir + File.separator.toString() + fileName
    var myFile: File? = null

    // Storage Permissions
    private val REQUEST_EXTERNAL_STORAGE = 1

    private val PERMISSIONS_STORAGE = arrayOf(
        Manifest.permission.READ_EXTERNAL_STORAGE,
        Manifest.permission.WRITE_EXTERNAL_STORAGE
    )

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        if (shouldAskPermissions()) {
            verifyStoragePermissions(this);
        }
    }
}

```

```

        }

    }

    private fun shouldAskPermissions(): Boolean {
        return Build.VERSION.SDK_INT > Build.VERSION_CODES.LOLLIPOP_MR1
    }

    private fun verifyStoragePermissions(activity: Activity?) {
        // Check if we have write permission
        val permission = ActivityCompat.checkSelfPermission(activity!!, Manifest.permission.WRITE_EXTERNAL_STORAGE)
        if (permission != PackageManager.PERMISSION_GRANTED) {
            // We don't have permission so prompt the user
            ActivityCompat.requestPermissions(
                activity,
                PERMISSIONS_STORAGE,
                REQUEST_EXTERNAL_STORAGE
            )
        }
    }

    fun saveData(view: View) {
        val xVal: String = etX.text.toString()
        val yVal: String = etY.text.toString()
        val zVal: String = etZ.text.toString()
        val rows = listOf(xVal, yVal, zVal)
        try {
            csvWriter().open(filePath, append = true) {
                writeRow(rows)
            }
            Toast.makeText(applicationContext, "Data berhasil di simpan!", Toast.LENGTH_SHORT).show()
        } catch (e: IOException) {
            Toast.makeText(applicationContext, "Data tidak berhasil di simpan! $e", Toast.LENGTH_SHORT).show()
        }
    }

    fun readData(view: View) {
        myFile = File(filePath)
        // File exist
        if (myFile!!.exists() && !myFile!!.isDirectory) {
            val rows: List<List<String>> = csvReader().readAll(myFile!!)
            var baris = ""
            var koma = ","
            for (row in rows) {
                baris += row[0] + koma + row[1] + koma + row[2] + "\n"
            }
            tvData.text = baris
        } else {
            Toast.makeText(applicationContext, "File csv belum dibuat!", Toast.LENGTH_SHORT).show()
        }
    }
}

```

6. Bukalah file AndroidManifest.xml tambahkan baris code berwarna kuning seperti berikut ini.

```

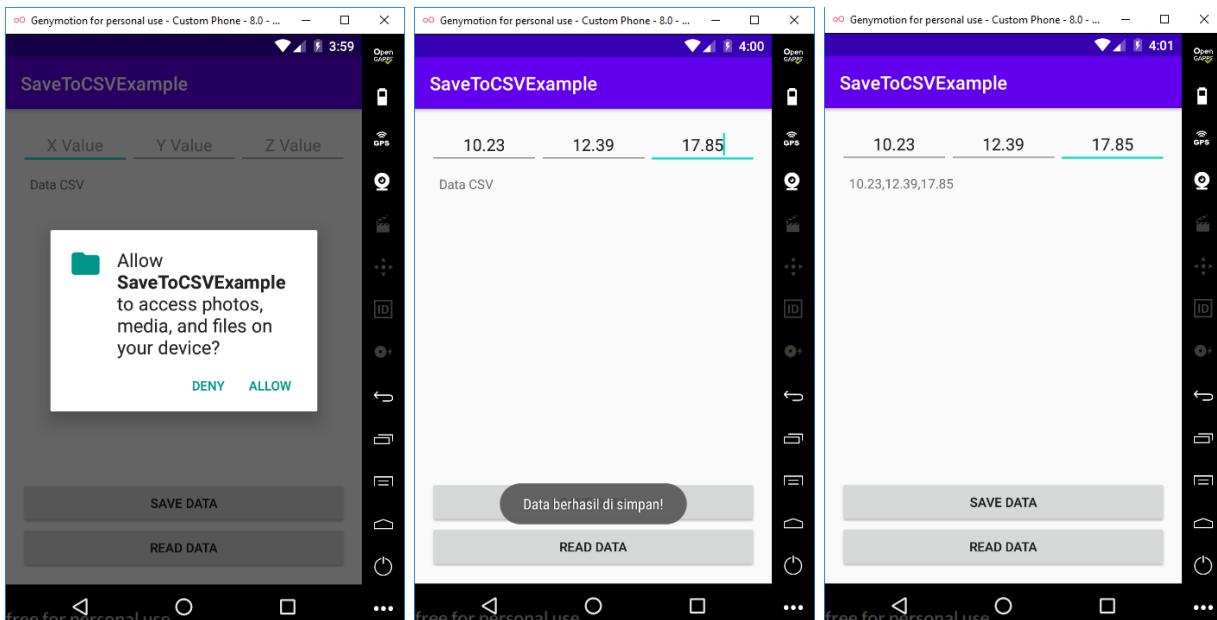
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="depandi.porbeng.ac.id.savetocsvexample">

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

7. Jalankan program di Emulator Android. Perhatikan hasilnya dan lakukan analisa terhadap kode layout *activity_main.xml* dan kode logika program *ActivityMain.kt* dilaporan praktikum.

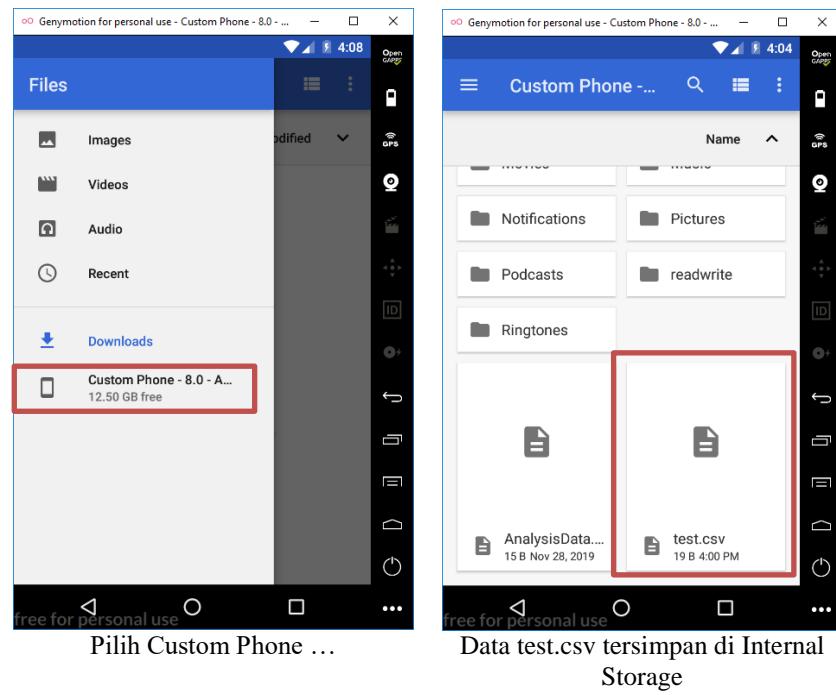


Klik Allow untuk mengizinkan akses Internal Storage

Untuk melihat file csv yang telah disimpan di internal storage bukalah file explorer emulator Android dan pilih Internal Storage (Custom Phone – 8.0...).

Menyimpan Data di Internal Storage dalam format CSV

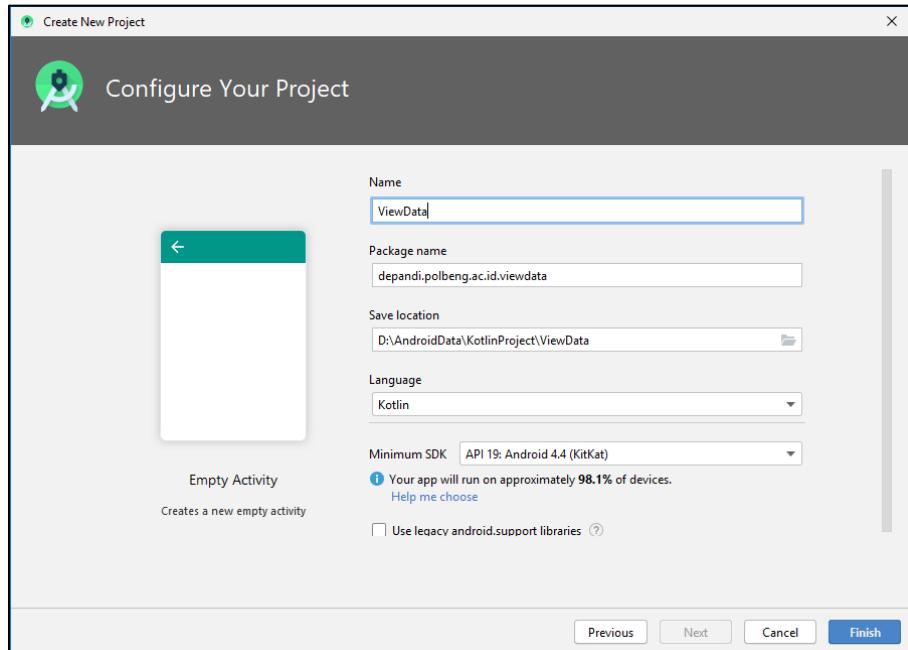
Menampilkan Data di Internal Storage dalam format CSV



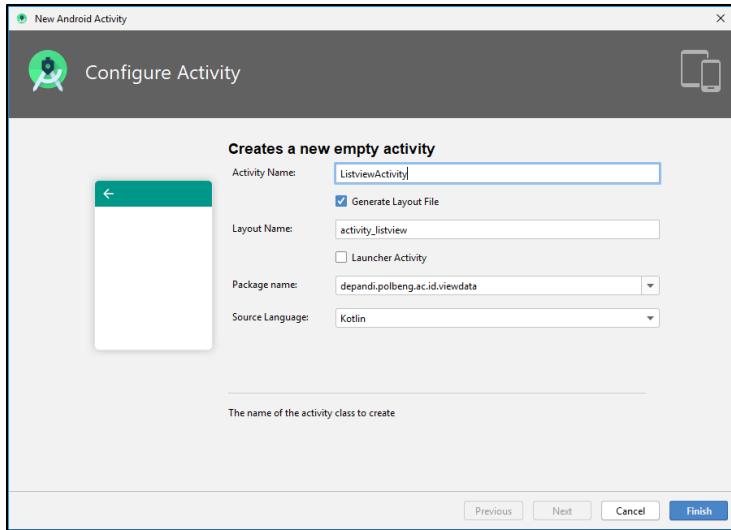
Menampilkan Data Menggunakan Listview dan Recyclerview

Menampilkan data kedalam Listview

1. Buatlah *project* baru dengan konfigurasi seperti berikut :



2. Setelah selesai membuat project baru, buatlah *activity* baru dengan nama **ListviewActivity** dan tipe **Empty Activity**.



3. Selanjutnya editlah file *strings.xml* pada folder *values*, tambahkan beberapa variabel dan nilai *string* sebagai berikut:

```
<resources>
    <string name="app_name">ViewData</string>
    <string name="btn_list_view">Tampilkan Listview</string>
    <string-array name="jurusan">
        <item>Teknik Informatika</item>
        <item>Bahasa</item>
        <item>Teknik Perkapalan</item>
        <item>Teknik Mesin</item>
        <item>Teknik Elektro</item>
        <item>Administrasi Niaga</item>
        <item>Teknik Sipil</item>
        <item>Kemaritiman</item>
    </string-array>
</resources>
```

4. Editlah file *activity_main.xml* sehingga kode *layout* menjadi seperti berikut :

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

    <Button
        android:id="@+id/btnlistview"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/btn_list_view" />

</LinearLayout>
```

5. Buka file sumber *MainActivity.kt* editlah kode program hingga menjadi seperti berikut :

```
package depandi.polbeng.ac.id.viewdata
```

```

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        btnlistview.setOnClickListener{
            actionPindah(ListviewActivity())
        }
    }

    private fun actionPindah(simpleActivity: Any) {
        startActivity(Intent(applicationContext, simpleActivity::class.java))
    }
}

```

9. Editlah file *activity_listview.xml* sehingga kode *layout* menjadi seperti berikut :

```

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

    <ListView
        android:id="@+id/listview"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
</LinearLayout>

```

10. Bukalah file sumber *ListviewActivity.kt* editlah kode program hingga menjadi seperti berikut :

```

package depandi.polbeng.ac.id.viewdata

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.ArrayAdapter
import kotlinx.android.synthetic.main.activity_listview.*
import java.util.*

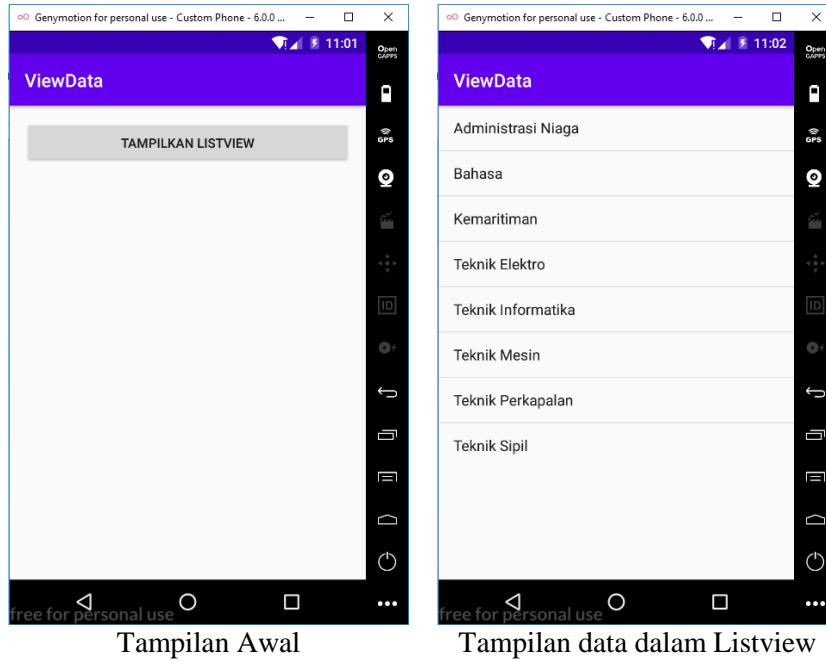
class ListviewActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_listview)
        var dataArray: Array<String> =
resources.getStringArray(R.array.jurusan)
        Arrays.sort(dataArray)
        var adapter = ArrayAdapter(this,
        android.R.layout.simple_list_item_1, dataArray)
        listview.adapter = adapter
    }
}

```

```
}
```

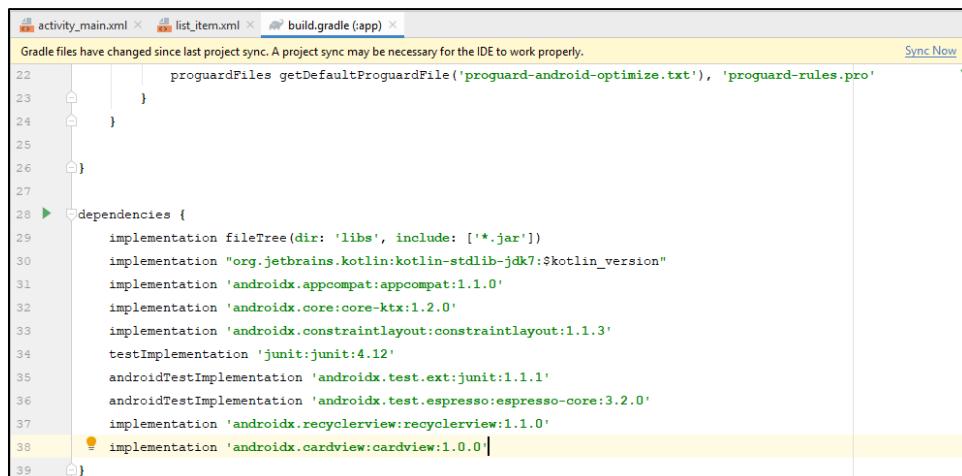
11. Jalankan program di Emulator Android. Perhatikan hasilnya dan lakukan analisa terhadap kode sumber *activity_main.xml*, *activity_listview.xml* dan *ActivityMain.kt*, *ListviewActivity.kt* dilaporan praktikum.



Menampilkan data kedalam Recycleview

12. Lakukan modifikasi pada proyek ViewData sebelumnya. Tambahkan pustaka untuk mendukung tampilan *recyclerview* dan *cardview* pada proyek ViewData. Klik **Sync Now** untuk menambahkan pustaka.

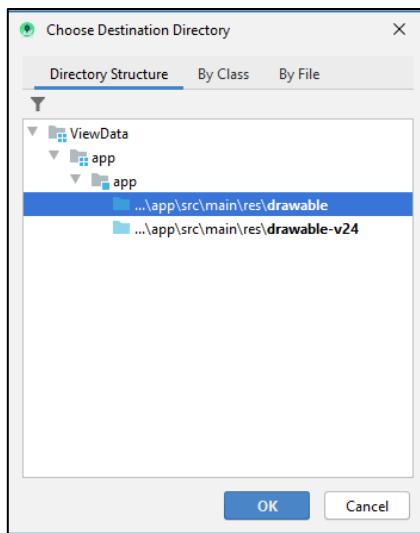
- implementation 'androidx.recyclerview:recyclerview:1.1.0'
- implementation 'androidx.cardview:cardview:1.0.0'



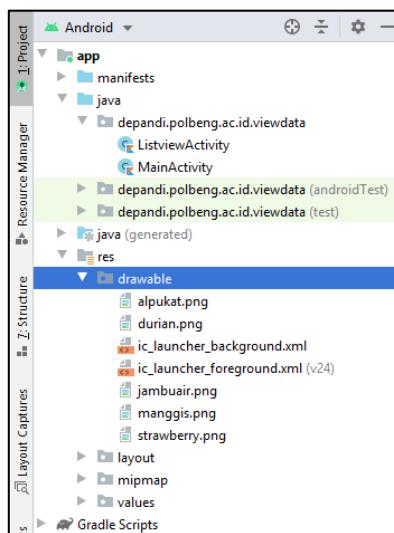
13. Tambahkan gambar kedalam folder drawable dengan cara copy-paste. Adapun file gambar yang dimasukkan kedalam folder drawable adalah :

- R.drawable.alpukat,
- R.drawable.durian,
- R.drawable.jambuair,
- R.drawable.manggis,
- R.drawable.strawberry

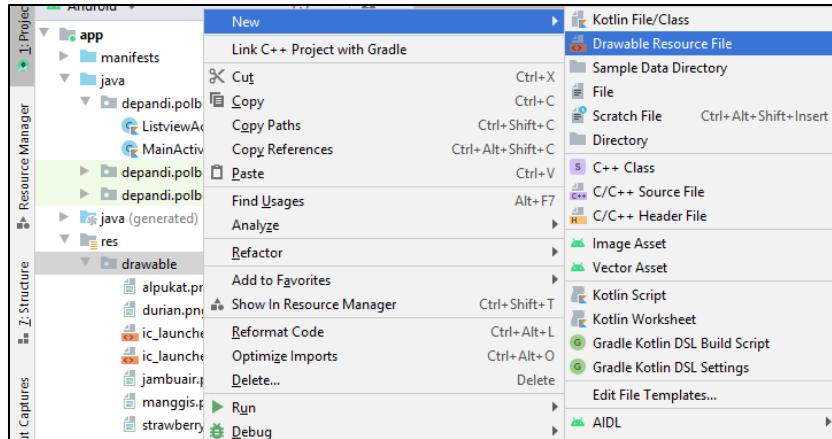
Pilih direktori tujuan drawable dan klik OK untuk melanjutkan → OK untuk menyimpan.



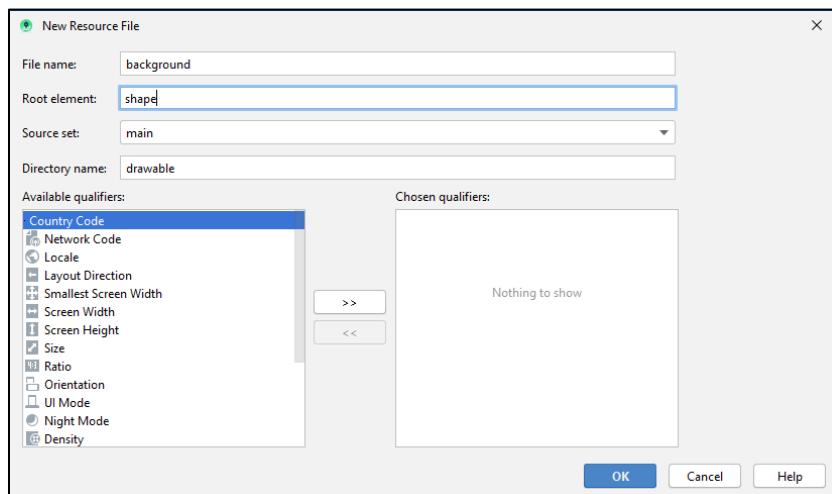
Hasil penyalinan gambar kedalam folder drawable.



14. Buatlah shape baru dengan cara klik kanan pada folder drawable → New → Drawable Resource File



Masukkan File name: background dan Root element: shape pada form isian:



Masukkan kode berikut pada file background.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >
    <gradient
        android:type="linear"
        android:startColor="#01c701"
        android:endColor="#FF00ffff"
        android:angle="45"/>
</shape>
```

15. Editlah file *strings.xml* pada folder *values*, tambahkan beberapa variabel dan nilai *string* sebagai berikut:

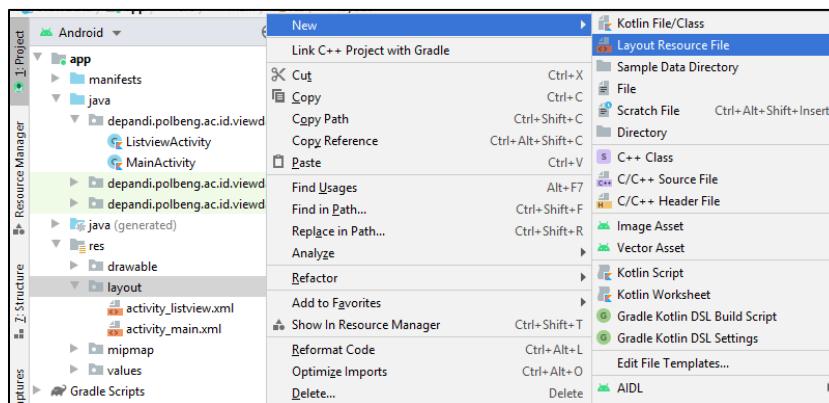
```
<resources>
    <string name="app_name">ViewData</string>
    <string name="btn_list_view">Tampilkan Listview</string>
    <string-array name="jurusan">
        <item>Teknik Informatika</item>
        <item>Bahasa</item>
        <item>Teknik Perkapalan</item>
        <item>Teknik Mesin</item>
```

```

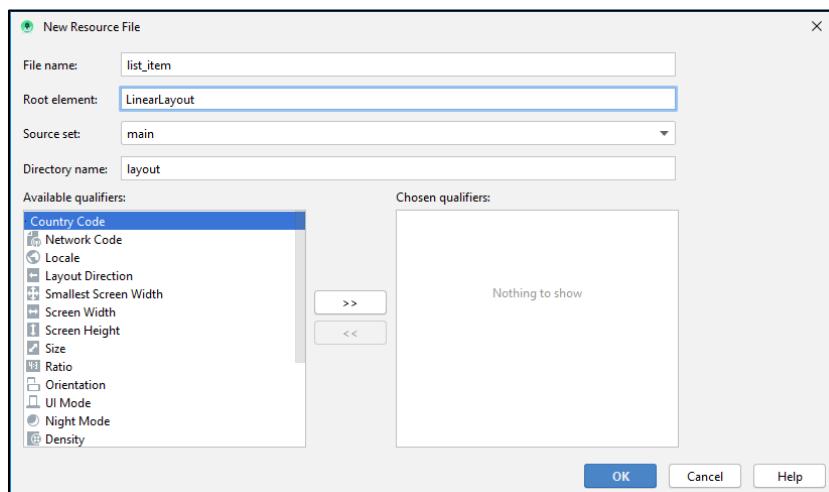
<item>Teknik Elektro</item>
<item>Administrasi Niaga</item>
<item>Teknik Sipil</item>
<item>Kemaritiman</item>
</string-array>
<string name="gambar_buah">Gambar Buah</string>
<string name="nama_buah">Nama Buah</string>
<string name="btn_recyler_view">Tampilkan Recyclerview</string>
</resources>

```

16. Buatlah layout baru dengan nama list_item.xml di folder layout. Caranya klik kanan pada folder layout → New → Layout Resource File



Masukkan File name: list_item dan Root element: LinearLayout pada form isian:



Masukkan kode berikut pada file list_item.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:orientation="vertical">

    <androidx.cardview.widget.CardView>

```

```

    android:id="@+id/cardview"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="3dp"
    app:cardCornerRadius="5dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="5dp"
        android:gravity="center"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/imglsitiitm"
            android:layout_width="match_parent"
            android:layout_height="150dp"
            android:src="@drawable/durian"
            android:contentDescription="@string/gambar_buah" />

        <TextView
            android:id="@+id/namabuah"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="@string/nama_buah" />
    </LinearLayout>
</androidx.cardview.widget.CardView>
</LinearLayout>

```

17. Editlah file *activity_main.xml* sehingga kode *layout* menjadi seperti berikut :

```

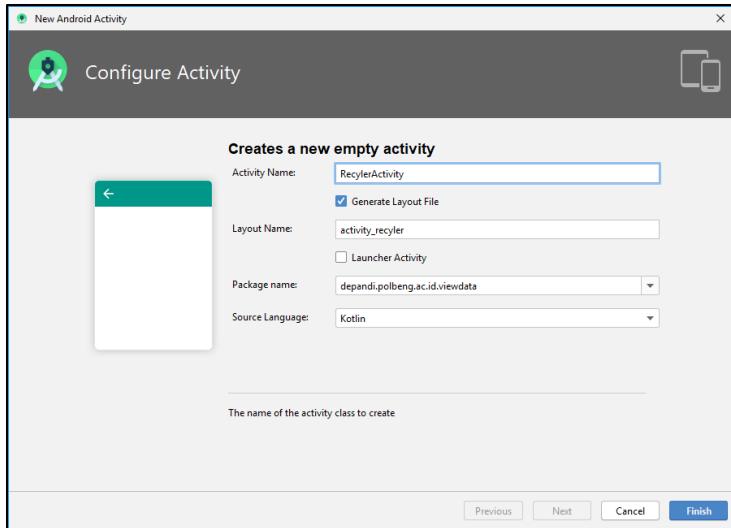
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"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

    <Button
        android:id="@+id/btnlistview"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/btn_list_view"/>

    <Button
        android:id="@+id/btnrecyclerview"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/btn_recycler_view"/>

```

18. Buatlah activity baru dengan nama RecylerActivity dan tipe Empty Activity.



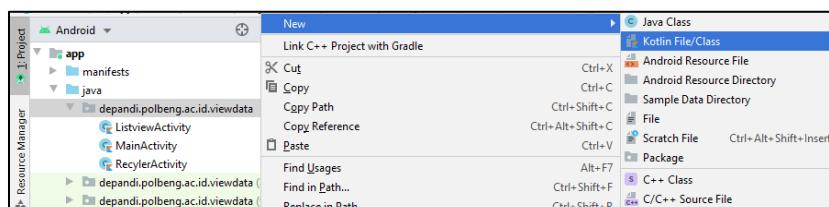
19. Editlah file *activity_recycler.xml* sehingga kode *layout* menjadi seperti berikut :

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

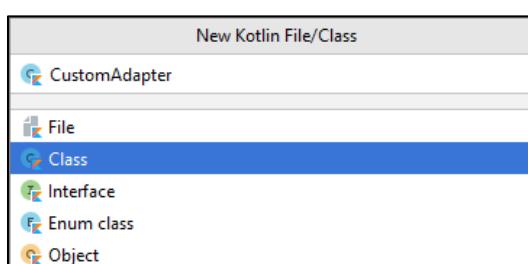
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerview"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

    </androidx.recyclerview.widget.RecyclerView>
</LinearLayout>
```

20. Selanjutnya buatlah kelas baru dengan nama CustomAdapter. Caranya klik kanan pada folder base package proyek anda → New → Kotlin File/Class



Pada kotak dialog masukkan nama CustomAdapter dan jenis Class pada kotak dialog isian.



Masukkan kode berikut pada file CustomAdapter.kt:

```
package depandi.polbeng.ac.id.viewdata

import android.app.Activity
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView

class CustomAdapter : RecyclerView.Adapter<CustomAdapter.Myholder> {
    private var c: Activity? = null
    private var databuah: Array<String>? = null
    private var datagambar: Array<Int>? = null

    constructor(recylerActivity: RecylerActivity, dataBuah: Array<String>, dataGambar: Array<Int>) {
        c = recylerActivity
        databuah = dataBuah
        datagambar = dataGambar
    }

    //set data ke recyclerview
    override fun onBindViewHolder(holder: Myholder, position: Int) {
        //manggil dan set data
        holder.txt?.text = databuah?.get(position)
        holder.img?.setImageResource(datagambar?.get(position)!!)
    }

    //mencreate pada layout list item
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): Myholder {
        var view = LayoutInflater.from(c).inflate(R.layout.list_item,
        parent, false)
        return Myholder(view)
    }

    //tampilkan data
    override fun getItemCount(): Int {
        //return data gmbr
        return databuah?.size!!
    }

    //deklarasi id pada customAdapter, id yang ada pada list item
    class Myholder(itemView: View?) : RecyclerView.ViewHolder(itemView!!)
    {
        var txt: TextView? = itemView?.findViewById(R.id.namabuah)
        var img: ImageView? = itemView?.findViewById(R.id.imglsitietm)
    }
}
```

21. Buka file sumber RecylerActivity.kt editlah kode program hingga menjadi seperti berikut ini:

```
package depandi.polbeng.ac.id.viewdata
```

```

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.recyclerview.widget.LinearLayoutManager
import kotlinx.android.synthetic.main.activity_recyler.*

class RecylerActivity : AppCompatActivity() {

    //deklarasi
    var dataBuah =
        arrayOf("alpukat", "durian", "jambuair", "manggis", "strawberry")
    var dataGambar= arrayOf(
        R.drawable.alpukat,
        R.drawable.durian,
        R.drawable.jambuair,
        R.drawable.manggis,
        R.drawable.strawberry)

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_recyler)

        //custom adapter, nnt dibuatkan constructor nanti akan di buat
        contrut di custom adpater
        var adapter = CustomAdapter(this, dataBuah, dataGambar)
        //buat layout manaager
        var liner = LinearLayoutManager(applicationContext)
        //manggil listnya
        recylerview.adapter = adapter
        recylerview.layoutManager = liner
    }
}

```

22. Bukalah file sumber *MainActivity.kt* editlah kode program hingga menjadi seperti berikut:

```

package depandi.polbeng.ac.id.viewdata

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

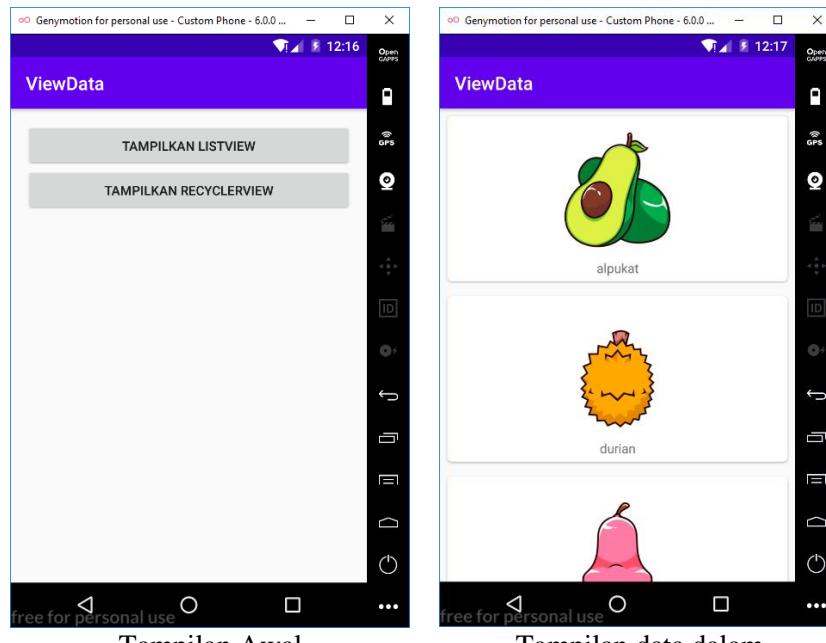
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        btnlistview.setOnClickListener{
            actionPindah(ListviewActivity())
        }
        btnrecylerview.setOnClickListener{
            actionPindah(RecylerActivity())
        }
    }

    private fun actionPindah(simpleActivity: Any) {
        startActivity(Intent(applicationContext,simpleActivity::class.java))
    }
}

```

```
}
```

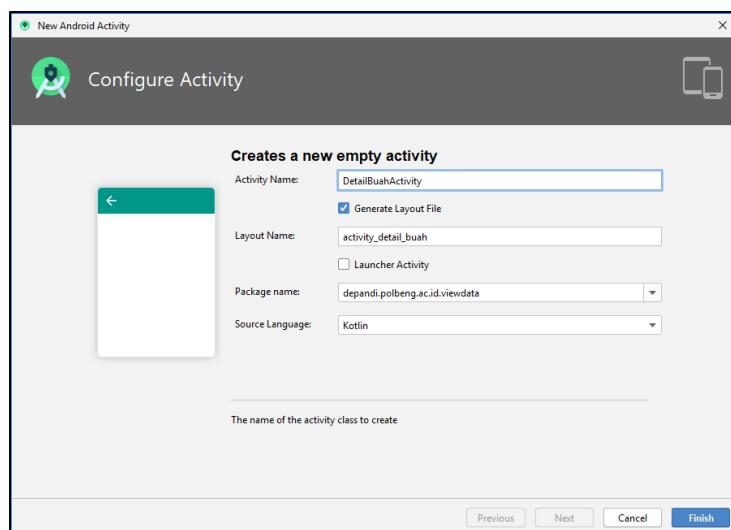
23. Jalankan program di Emulator Android. Perhatikan hasilnya dan lakukan analisa terhadap kode sumber *activity_main.xml*, *list_item.xml*, *activity_recycler.xml* dan *ActivityMain.kt*, *CustomAdapter.kt*, *RecyclerActivity.kt* dilaporan praktikum.



Tampilan Awal

Tampilan data dalam
Recyclerview

24. Selanjutnya kita akan menambahkan aksi/*event* membuka *activity* baru ketika salah satu item pada Recyclerview di klik. Buatlah activity baru dengan nama *DetailBuahActivity* dan jenis empty activity.



25. Editlah file *activity_detail_buah.xml* sehingga kode *layout* menjadi seperti berikut :

```
<?xml version="1.0" encoding="utf-8" ?>
```

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="5dp"
    android:orientation="vertical"
    tools:context=".DetailBuahActivity">

    <TextView
        android:id="@+id/detailtxt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="@string/nama_buah"
        android:textSize="30sp" />

    <ImageView
        android:id="@+id/imgdetail"
        android:layout_width="match_parent"
        android:layout_height="240dp"
        android:src="@drawable/alpukat" />
</LinearLayout>

```

26. Bukalah file sumber *DetailBuahActivity.kt* editlah kode program hingga menjadi seperti berikut:

```

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_detail_buah.*

class DetailBuahActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_detail_buah)

        //membuat variabel dan di tangkap data dri intent
        var tangkap = intent.getStringExtra("txt")
        var tangkap2 = intent.getIntExtra("img", 0)

        //menangkap data intent dan di set ke layout
        detailtxt.text = tangkap
        imgdetail.setImageResource(tangkap2)
    }
}

```

27. Bukalah file sumber *CustomAdapter.kt* editlah kode program tambahkan kode yang berwarna kuning seperti berikut ini:

```

package depandi.polbeng.ac.id.viewdata

import android.app.Activity
import android.content.Intent
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView

```

```

import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView

class CustomAdapter : RecyclerView.Adapter<CustomAdapter.Myholder> {
    private var c: Activity? = null
    private var databuah: Array<String>? = null
    private var datagambar: Array<Int>? = null

    constructor(recylerActivity: RecylerActivity, dataBuah: Array<String>, dataGambar: Array<Int>) {
        c = recylerActivity
        databuah = dataBuah
        datagambar = dataGambar
    }

    //set data ke recyclerview
    override fun onBindViewHolder(holder: Myholder, position: Int) {
        //manggil dan set data
        holder.txt?.text = databuah?.get(position)
        holder.img?.setImageResource(datagambar?.get(position)!!)
        holder.img?.setOnClickListener {
            var intent = Intent(c, DetailBuahActivity::class.java)
            intent.putExtra("txt", databuah?.get(position))
            intent.putExtra("img", datagambar?.get(position))
            c?.startActivity(intent)
        }
    }

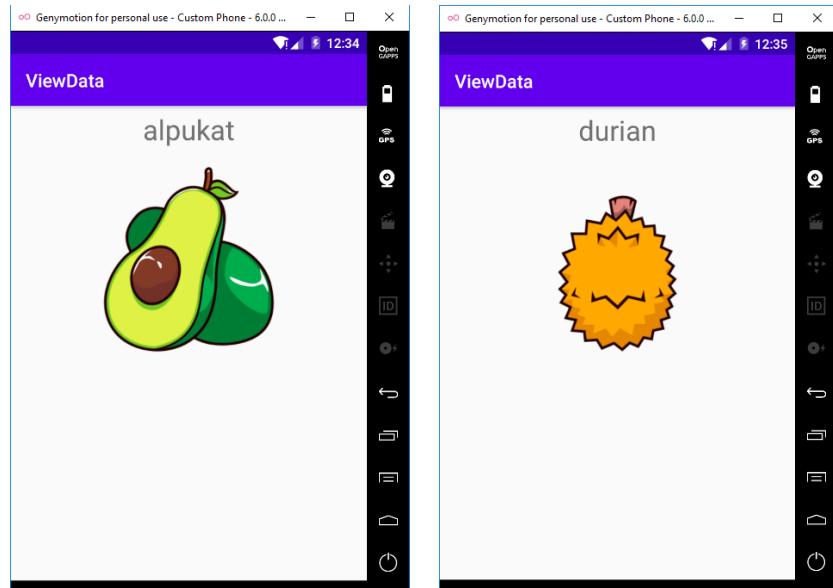
    //mencreate pada layout list item
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): Myholder {
        var view = LayoutInflater.from(c).inflate(R.layout.list_item,
        parent, false)
        return Myholder(view)
    }

    //tampilkan data
    override fun getItemCount(): Int {
        //return data gmbr
        return databuah?.size!!
    }

    //deklarasi id pada customAdapter, id yang ada pada list_item
    class Myholder(itemView: View?) : RecyclerView.ViewHolder(itemView!!)
}

```

28. Jalankan program di Emulator Android. Perhatikan hasilnya dan lakukan analisa terhadap kode sumber activity_detail_buah.xml dan CustomAdapter.kt, DetailBuahActivity.kt dilaporan praktikum.



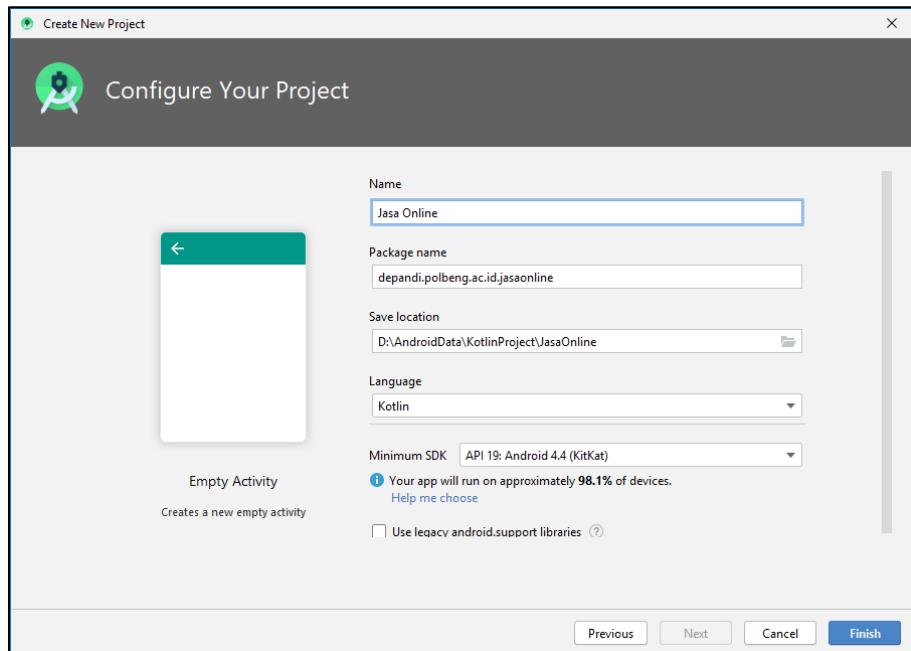
Tampilan item alpukat di klik

Tampilan item durian di klik

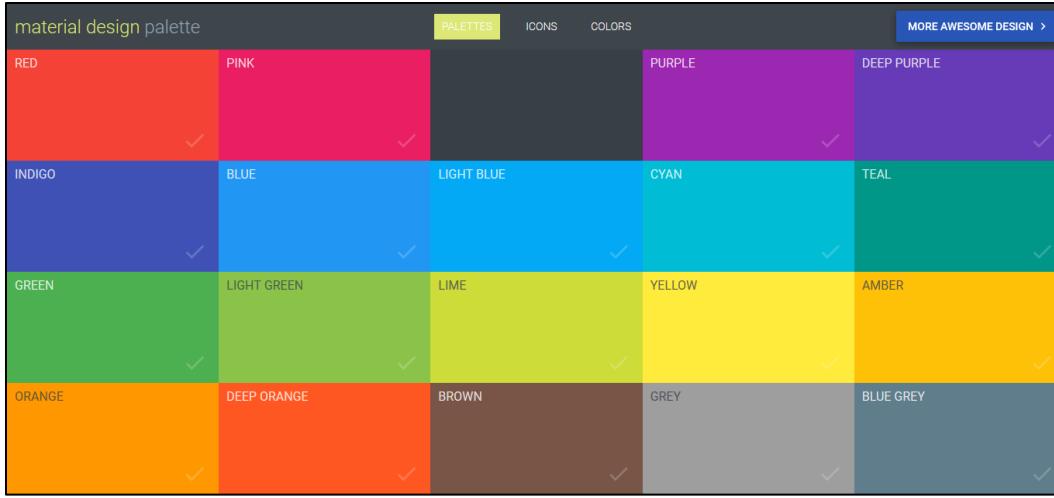
Membuat Navigation Drawer dengan menggunakan Fragment

Menambahkan theme color

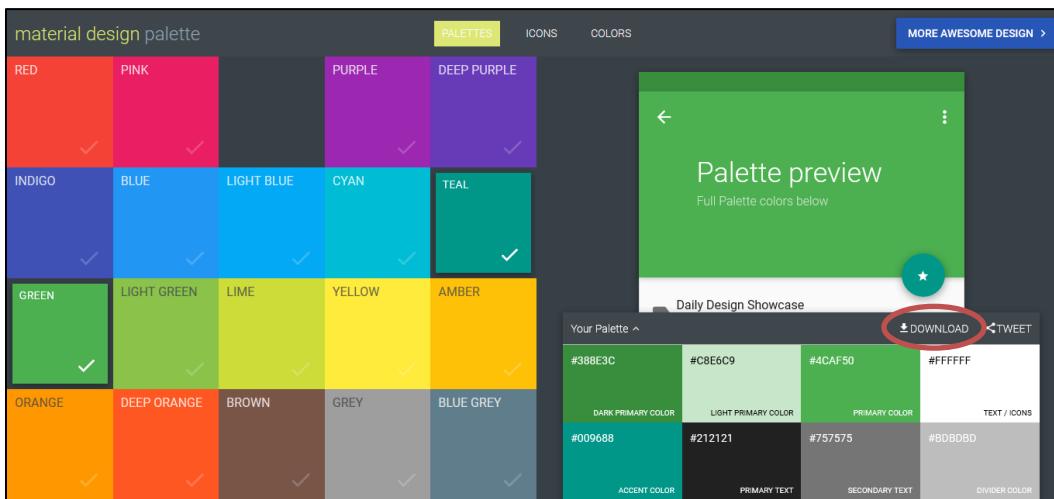
1. Buatlah *project* baru dengan konfigurasi seperti berikut :



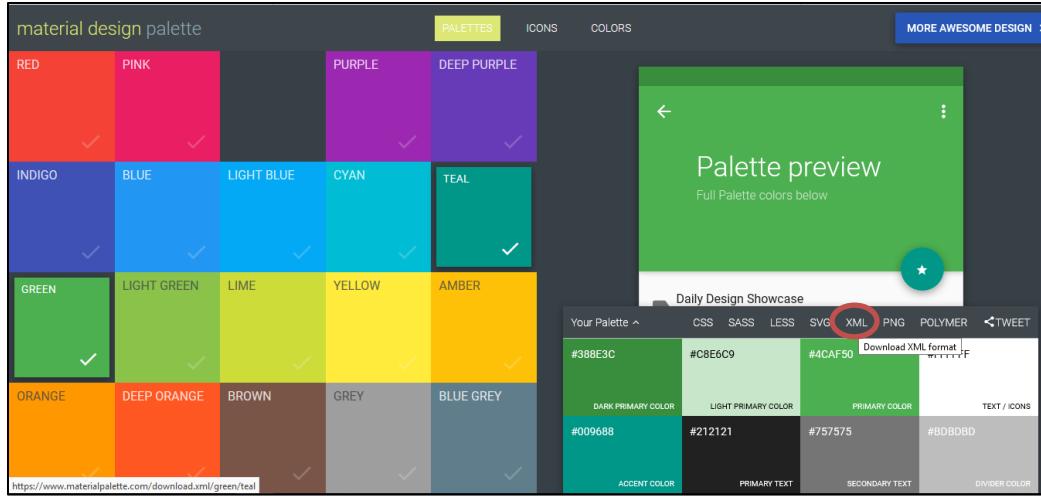
2. Tunggu proses pembuatan *project* baru selesai. Buka halaman web berikut untuk memilih material theme color <https://www.materialpalette.com/>.



Selanjutnya pilihlah dua warna yang akan dijadikan theme color, base color (warna utama) dengan mengklik salah satu warna pada material design palette misalnya warna GREEN, dan klik lagi warna berikutnya yaitu secondary color untuk button misalnya TEAL. Ulangi langkah diatas jika menginginkan warna yang lain dengan memilih warna base kemudian dilanjutkan warna secondary. Sehingga tampilannya menjadi seperti berikut.



Setelah memilih selanjutnya downloadlah kode warna dalam format xml. Pada pilihan Download di kanan bawah tampilan pallete.



3. Setelah mendownload file kode warna dalam format xml (colors_green_teal.xml). Buka file tersebut kedalam editor teks seperti notepad++.

```

1  <!-- Palette generated by Material Palette --
materialpalette.com/green/teal -->
2  <?xml version="1.0" encoding="utf-8"?>
3  <resources>
4      <color name="primary">#4CAF50</color>
5      <color name="primary_dark">#388E3C</color>
6      <color name="primary_light">#C8E6C9</color>
7      <color name="accent">#009688</color>
8      <color name="primary_text">#212121</color>
9      <color name="secondary_text">#757575</color>
10     <color name="icons">#FFFFFF</color>
11     <color name="divider">#BDBDBD</color>
12 </resources>

```

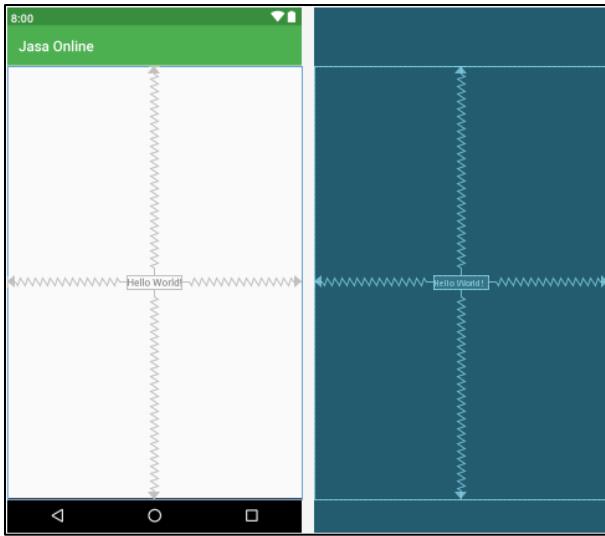
4. Buka kembali proyek Jasa Online pada Android Studio IDE. Buka file colors.xml di folder app/res/values dan tambahkan beberapa komponen warna pada file kode warna yang telah di download (colors_green_teal.xml) dengan cara copy-paste dan ubah name color hingga menjadi seperti berikut:

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#4CAF50</color>
    <color name="colorPrimaryDark">#388E3C</color>
    <color name="colorPrimaryLight">#C8E6C9</color>
    <color name="colorAccent">#009688</color>
    <color name="colorPrimaryText">#212121</color>
    <color name="colorSecondaryText">#757575</color>
    <color name="colorIcons">#FFFFFF</color>
    <color name="colorDivider">#BDBDBD</color>
</resources>

```

5. Buka file activity_main.xml dan lihat dalam mode preview design. Perhatikan, sekarang warna theme color sudah berubah menjadi warna yang kita inginkan.



Membuat Navigation Drawer dengan menggunakan Fragment

6. Masih dalam proyek Jasa Online. Tambahkan library material design (implementation 'com.google.android.material:material:1.1.0') yang diberi kotak berwarna merah berikut pada file build.gradle(Module: app). Klik Sync Now untuk menerapkan penambahan library dalam project.

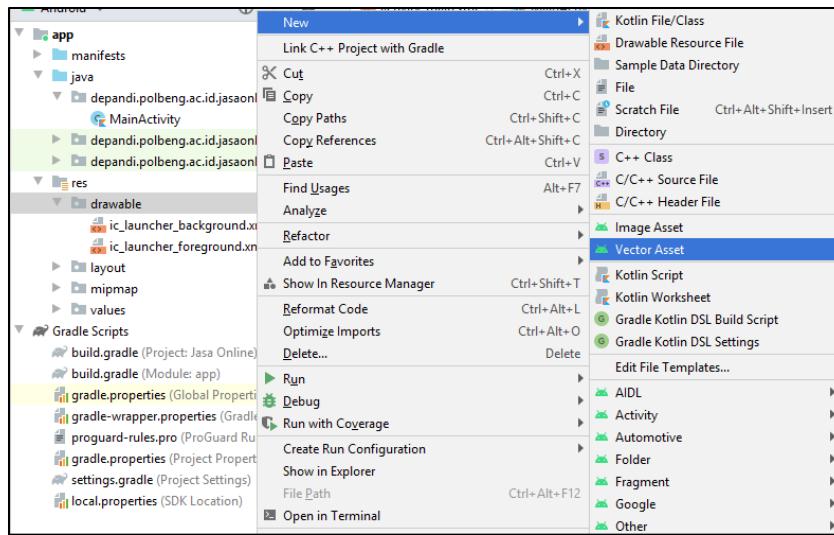
```

activity_main.xml ✘ build.gradle (app) ✘ MainActivity.kt ✘ colors.xml ✘
Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.
Sync Now

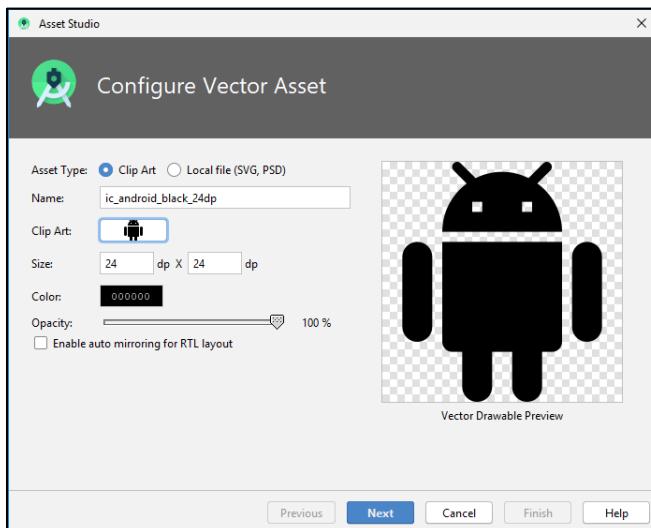
19    buildTypes {
20        release {
21            minifyEnabled false
22            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
23        }
24    }
25
26
27
28    dependencies {
29        implementation fileTree(dir: 'libs', include: ['*.jar'])
30        implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
31        implementation 'androidx.appcompat:appcompat:1.1.0'
32        implementation 'androidx.core:core-ktx:1.2.0'
33        implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
34        testImplementation 'junit:junit:4.12'
35        androidTestImplementation 'androidx.test.ext:junit:1.1.1'
36        androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
37        implementation 'com.google.android.material:material:1.0.0' // Red box highlights this line
38    }

```

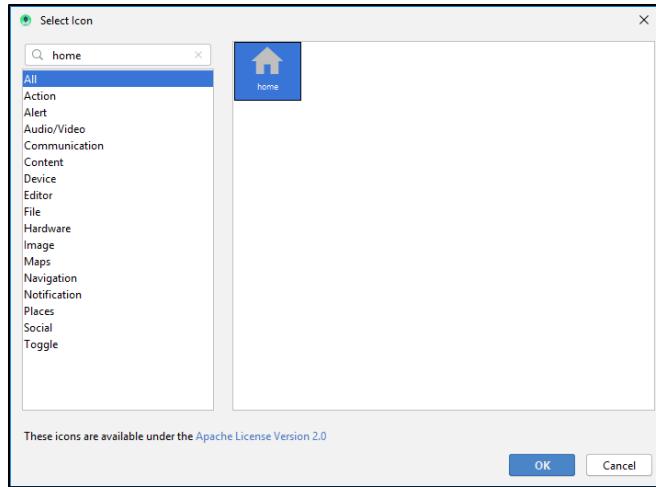
7. Langkah awal ialah menambahkan beberapa file sumber daya icon kedalam folder drawable. Caranya adalah klik kanan pada folder drawable → New → Vector Asset



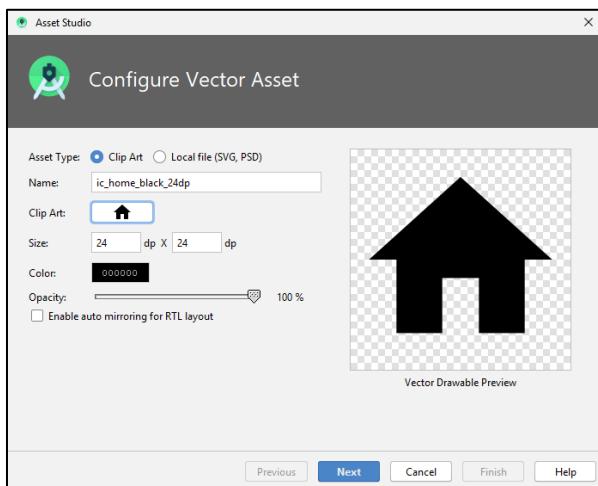
Pilih icon yang tersedia pada Clip Art, caranya klik icon Android yang ada pada clip art.



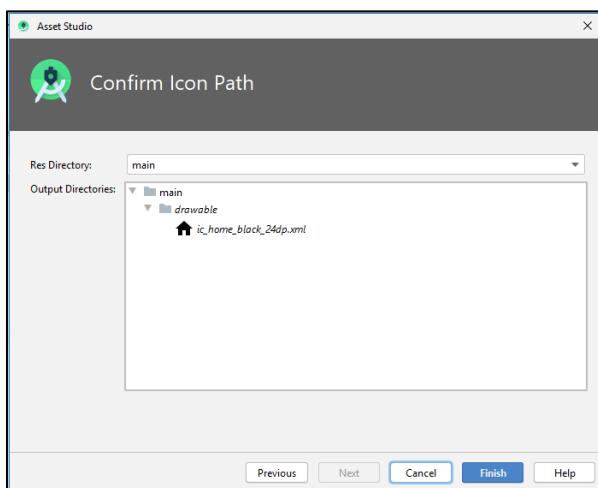
Maka akan muncul form pemilihan icon yang akan ditambahkan kedalam folder drawable. Carilah icon home dengan memasukkan kata kunci home pada textfield search icon dan pilihlah icon home yang tersedia seperti berikut.



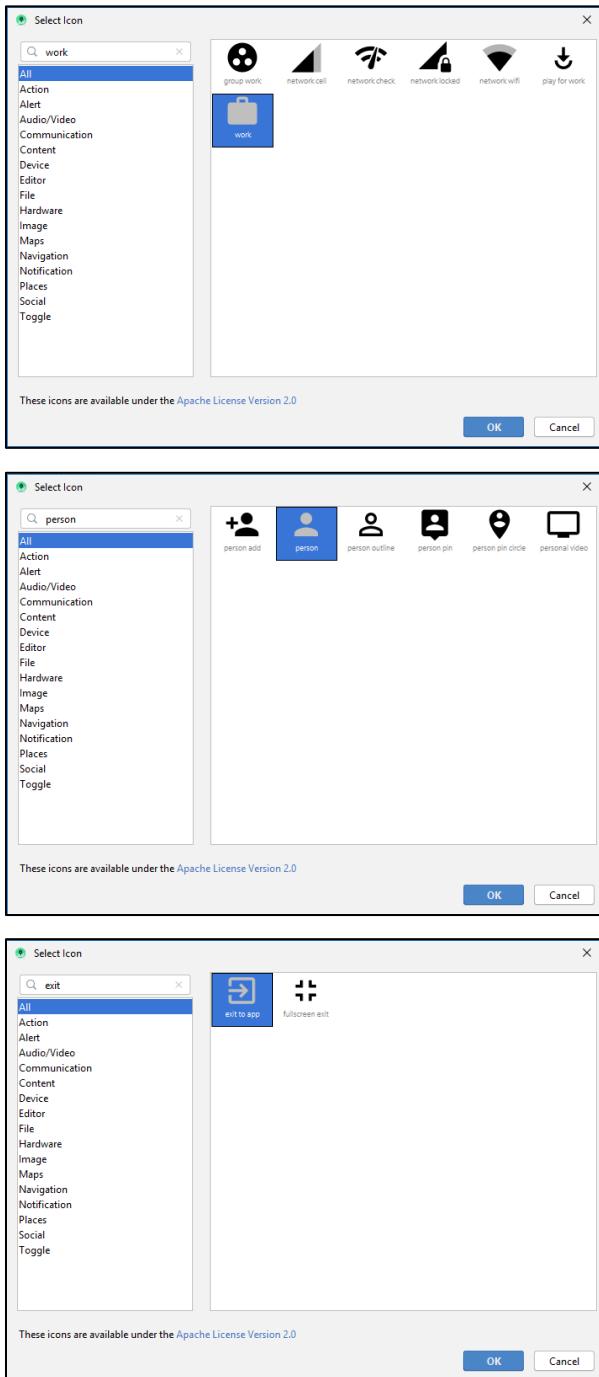
Selanjutnya ubahlah nama icon jika diperlukan, pada contoh ini biarkan saja nama defaultnya, dan klik next.



Klik finish untuk menyelesaikan penambahan icon didalam folder drawable.



Lakukan langkah yang sama untuk menambahkan icon ic_work_black_24dp, ic_person_black_24dp dan ic_exit_to_app_black_24dp.



8. Selanjutnya editlah file *strings.xml* pada folder *values*, tambahkan beberapa variabel dan nilai *string* sebagai berikut:

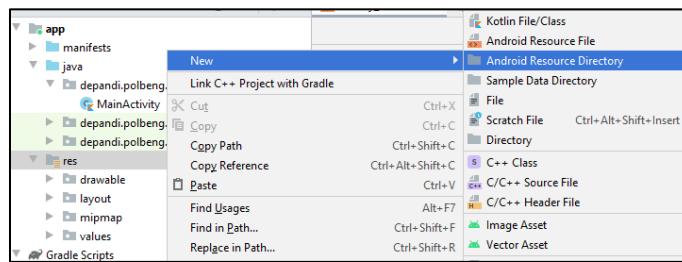
```
<resources>
    <string name="app_name">Jasa Online</string>
    <string name="beranda">Beranda</string>
```

```

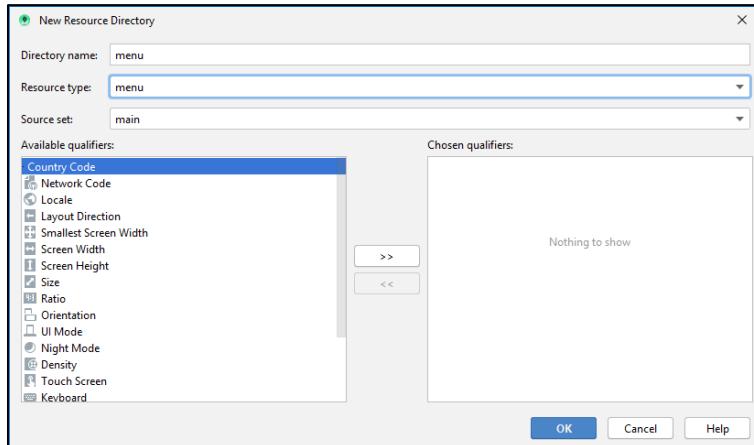
<string name="jasa_pengguna">Jasa Pengguna</string>
<string name="pengaturan_akun">Pengaturan Akun</string>
<string name="profil">Profil</string>
<string name="logout">Logout</string>
<string name="nama_header">Nama di Header</string>
<string name="email_header">Email di Header</string>
<string name="image_header">image header</string>
<string name="fragment_beranda">Ini fragment Beranda Pengguna</string>
<string name="fragment_jasa">Ini fragment Jasa Pengguna</string>
<string name="fragment_profil">Ini fragment Profil Pengguna</string>
</resources>

```

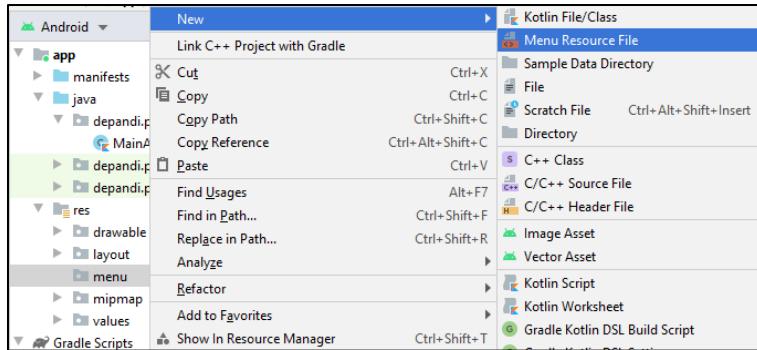
9. Buatlah folder baru dengan nama menu didalam folder res. Caranya klik kanan pada folder res → New → Android Resource Directory.



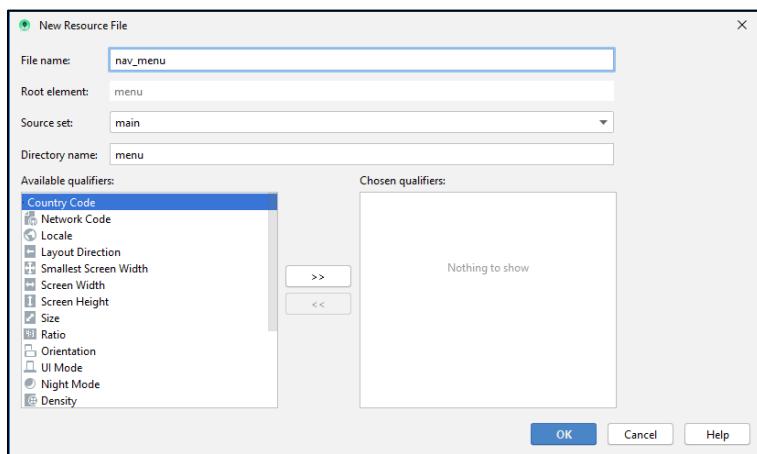
Masukkan Directory name: menu dan Resource type: menu pada form isian. Maka folder menu akan dibuat pada folder res.



10. Pada folder menu yang telah dibuat, buatlah menu baru dengan nama nav_menu.xml. Caranya klik kanan pada folder menu → New → Menu Resource File.



Masukkan nama menu dengan nama nav_menu dan klik OK.



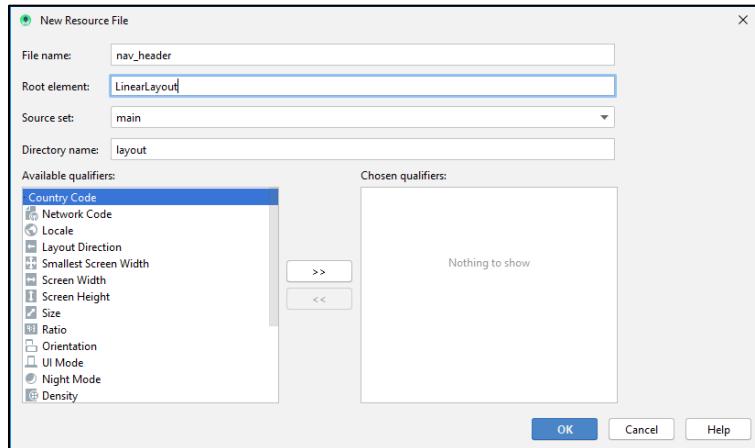
Bukalah file nav_menu.xml. Ubahlah kode menu menjadi seperti berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <group>
        <item
            android:id="@+id/nav_beranda"
            android:icon="@drawable/ic_home_black_24dp"
            android:title="@string/beranda"/>
        <item
            android:id="@+id/nav_jasa_pengguna"
            android:icon="@drawable/ic_work_black_24dp"
            android:title="@string/jasa_pengguna"/>
    </group>

    <item android:title="@string/pengaturan_akun">
        <menu>
            <item
                android:id="@+id/nav_profile"
                android:icon="@drawable/ic_person_black_24dp"
                android:title="@string/profil"/>
            <item
                android:id="@+id/nav_logout"
                android:icon="@drawable/ic_exit_to_app_black_24dp"
                android:title="@string/logout"/>
        </menu>
    </item>
</item>
```

```
</menu>
```

11. Selanjutnya kita akan membuat header layout dengan nama file nav_header.xml didalam folder layout. Caranya klik kanan pada folder layout → New → Layout Resource File. Masukkan File name: nav_header dan Root element: LinearLayout, klik OK untuk membuat file layout.



Bukalah file nav_header.xml. Ubahlah kode layout menjadi seperti berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:background="#4CAF50"
    android:gravity="bottom"
    android:paddingStart="15dp"
    android:paddingEnd="15dp"
    android:paddingBottom="15dp">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="75dp"
        android:layout_height="75dp"
        app:srcCompat="@mipmap/ic_launcher_round"
        android:layout_marginBottom="10dp"
        android:contentDescription="@string/image_header" />

    <TextView
        android:id="@+id/tvNamaHeader"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/nama_header"
        android:textColor="@android:color/white"
        android:textStyle="bold"
        android:textSize="18sp"/>

    <TextView
        android:id="@+id/tvEmailHeader"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/email_header"
        android:textColor="@android:color/white"
        android:textSize="16sp" />

    </LinearLayout>

```

12. Tambahkan style theme pada file style.xml di folder res/values/. Buka file style.xml dan tambahkan beberapa style theme berikut:

```

<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

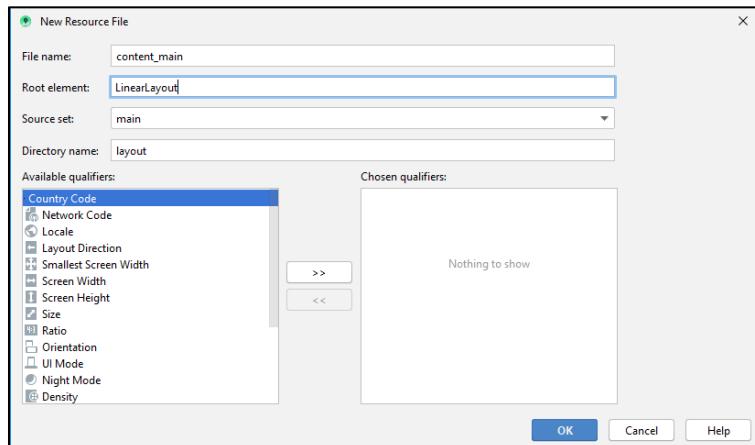
    <style name="AppTheme.NoActionBar">
        <item name="windowActionBar">false</item>
        <item name="windowNoTitle">true</item>
    </style>

    <style name="AppTheme.AppBarOverlay"
        parent="ThemeOverlay.AppCompat.Dark.ActionBar"/>

    <style name="AppTheme.PopupOverlay"
        parent="ThemeOverlay.AppCompat.Light"/>

```

13. Buatlah layout content dengan nama file content_main.xml didalam folder layout. Caranya klik kanan pada folder layout → New → Layout Resource File. Masukkan File name: content_main dan Root element: LinearLayout, klik OK untuk membuat file layout.



Bukalah file content_main.xml. Ubahlah kode layout menjadi seperti berikut:

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_height="wrap_content"
        android:layout_width="match_parent"
        android:theme="@style/AppTheme.AppBarOverlay">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay">
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:orientation="horizontal"
                android:gravity="center_vertical">
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="@string/app_name"
                    android:textColor="@android:color/white"
                    style="@style/TextAppearance.AppCompat.Widget.ActionBar.Title"/>
            </LinearLayout>
        </androidx.appcompat.widget.Toolbar>

    </com.google.android.material.appbar.AppBarLayout>

    <FrameLayout
        android:id="@+id/frame_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</LinearLayout>

```

14. Tambahkan android:theme pada file AndroidManifest.xml proyek anda pada folder manifests.

```

<?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="depandi.polbeng.ac.id.jasaonline">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity">

```

```

        android:theme="@style/AppTheme.NoActionBar">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER"
/>
        </intent-filter>
    </activity>
</application>

</manifest>

```

15. Berikutnya kita akan mengubah layout main_activity sehingga menjadi tampilan drawer.

Bukalah file main_activity.xml dalam folder layout, ubahlah kode layout hingga menjadi seperti berikut:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start"
    tools:context=".MainActivity">

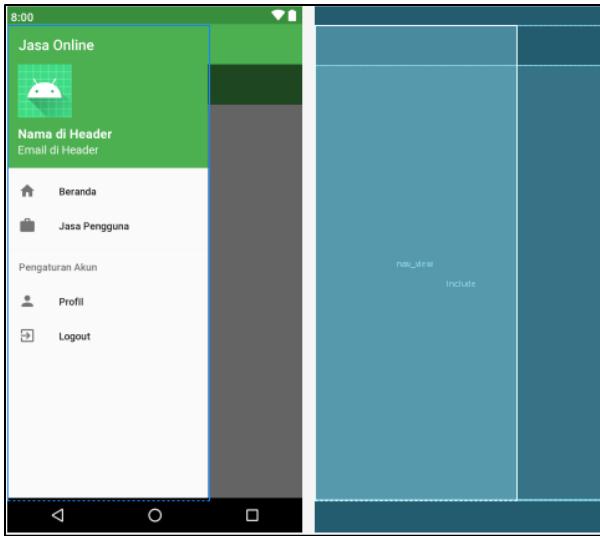
    <include
        layout="@layout/content_main"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

    <com.google.android.material.navigation.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header"
        app:menu="@menu/nav_menu"/>

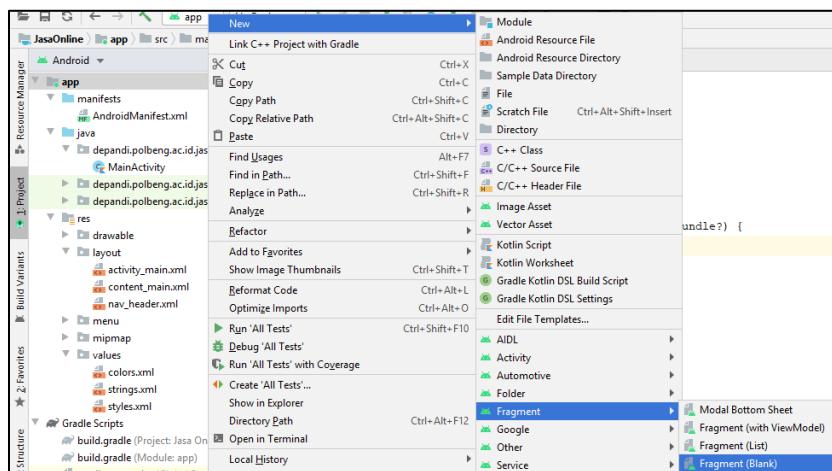
</androidx.drawerlayout.widget.DrawerLayout>

```

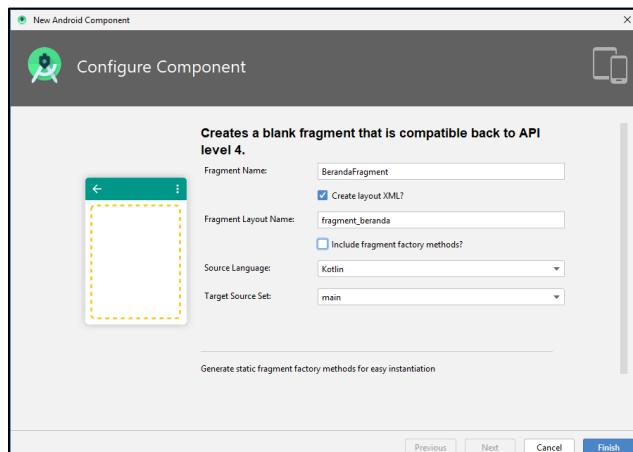
Jika tidak ada kesalahan dalam membuat design maka akan tampil navigation drawer seperti berikut ini dalam mode preview design.



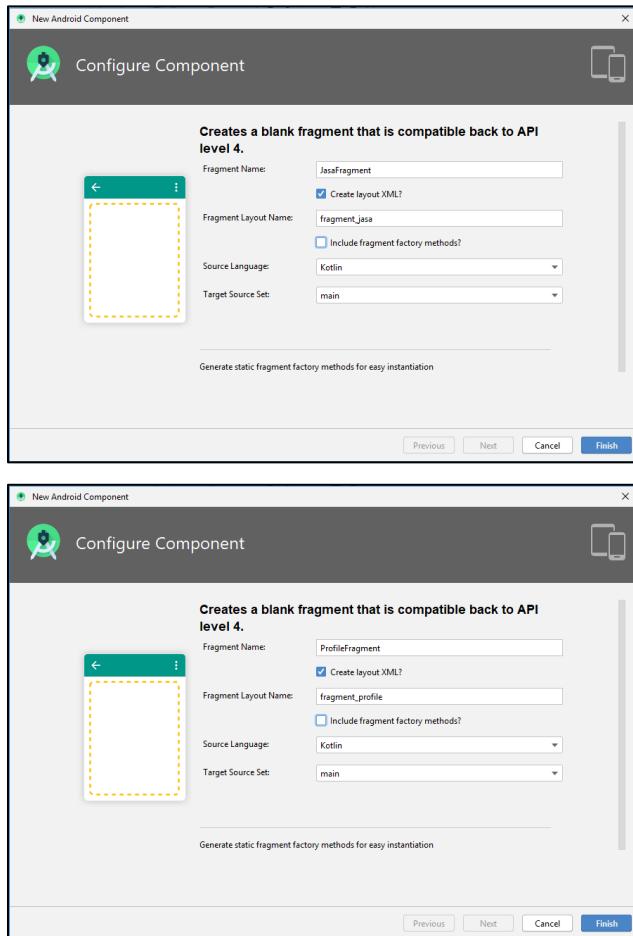
16. Tambahkan tiga buah fragment pada proyek. Caranya adalah klik kanan pada app → New → Fragment → Pilih Jenis Fragment (Blank)



Masukkan Fragment Name: **BerandaFragment** dan jangan lupa untuk menghilangkan centang pada *Include fragment factory method*?



Lakukan langkah yang sama untuk menambahkan JasaFragment dan ProfileFragment seperti berikut:



17. Ubahlah kode layout fragment_beranda menjadi seperti berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".BerandaFragment">

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/fragment_beranda"
        android:textAlignment="center"
        android:textSize="20sp"
        android:layout_centerInParent="true"/>

</RelativeLayout>
```

18. Ubahlah kode layout fragment_jasa menjadi seperti berikut:

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".JasaFragment">

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/fragment_jasa"
        android:textAlignment="center"
        android:textSize="20sp"
        android:layout_centerInParent="true" />

</RelativeLayout>

```

19. Ubahlah kode layout fragment_profile menjadi seperti berikut:

```

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

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/fragment_profil"
        android:textAlignment="center"
        android:textSize="20sp"
        android:layout_centerInParent="true" />

</RelativeLayout>

```

20. Kemudian bukalah file ActivityMain.kt, tambahkan beberapa kode untuk menggunakan drawer sehingga menjadi seperti berikut:

```

package depandi.polbeng.ac.id.jasaonline

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.MenuItem
import androidx.appcompat.app.ActionBarDrawerToggle
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.widget.Toolbar
import android.core.view.GravityCompat
import androidx.drawerlayout.widget.DrawerLayout
import androidx.fragment.app.FragmentTransaction
import com.google.android.material.navigation.NavigationView
import com.google.android.material.snackbar.Snackbar
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity(),
    NavigationView.OnNavigationItemSelectedListener {

```

```

lateinit var toolbar: Toolbar
lateinit var drawerLayout: DrawerLayout
lateinit var navView: NavigationView
lateinit var berandaFragment: BerandaFragment
lateinit var jasaFragment: JasaFragment
lateinit var profileFragment: ProfileFragment

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    toolbar = findViewById(R.id.toolbar)
    setSupportActionBar(toolbar)

    drawerLayout = findViewById(R.id.drawer_layout)
    navView = findViewById(R.id.nav_view)

    val toggle = ActionBarDrawerToggle(
        this, drawerLayout, toolbar, 0, 0
    )

    drawerLayout.addDrawerListener(toggle)
    toggle.syncState()
    navView.setNavigationItemSelectedListener(this)

    openFragment(R.id.nav_beranda)
}

override fun onNavigationItemSelected(item: MenuItem): Boolean {
    openFragment(item.itemId)
    return true
}

private fun openFragment(fragment_id: Int) {
    when (fragment_id) {
        R.id.nav_beranda -> {
            berandaFragment = BerandaFragment()
            supportFragmentManager
                .beginTransaction()
                .replace(R.id.frame_layout, berandaFragment)

            .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_OPEN)
            .commit()
        }
        R.id.nav_jasa_pengguna -> {
            jasaFragment = JasaFragment()
            supportFragmentManager
                .beginTransaction()
                .replace(R.id.frame_layout, jasaFragment)

            .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_OPEN)
            .commit()
        }
        R.id.nav_profile -> {
            profileFragment = ProfileFragment()
            supportFragmentManager
                .beginTransaction()

```

```

        .replace(R.id.frame_layout, profileFragment)

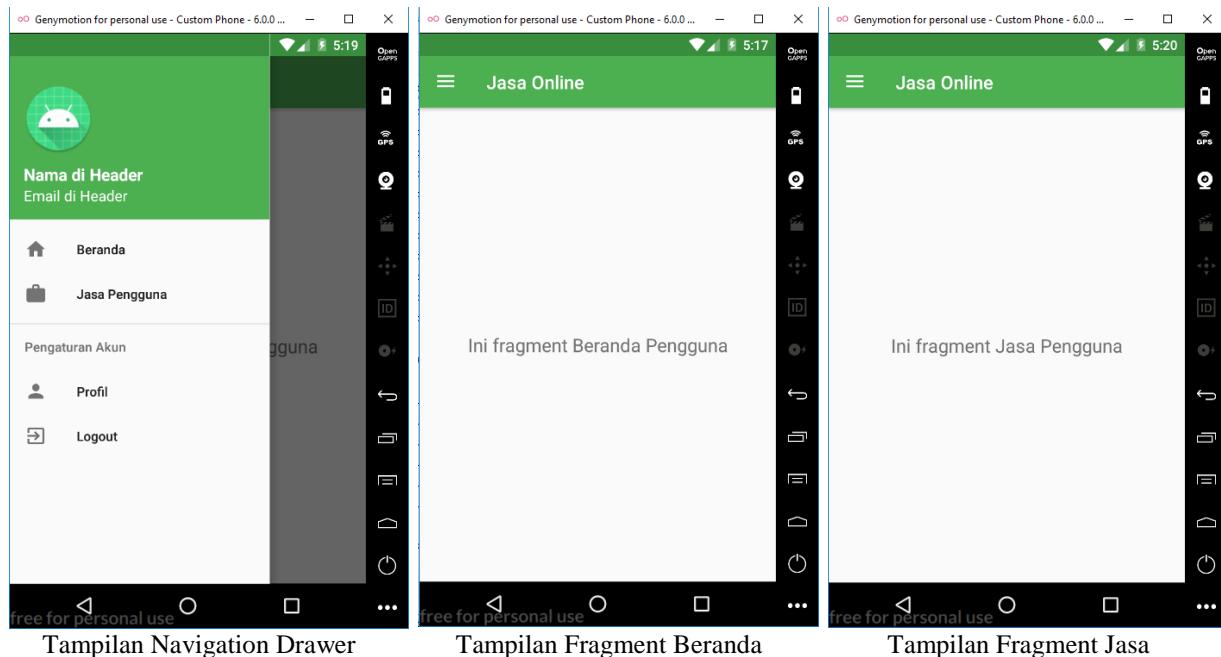
.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_OPEN)
        .commit()
    }

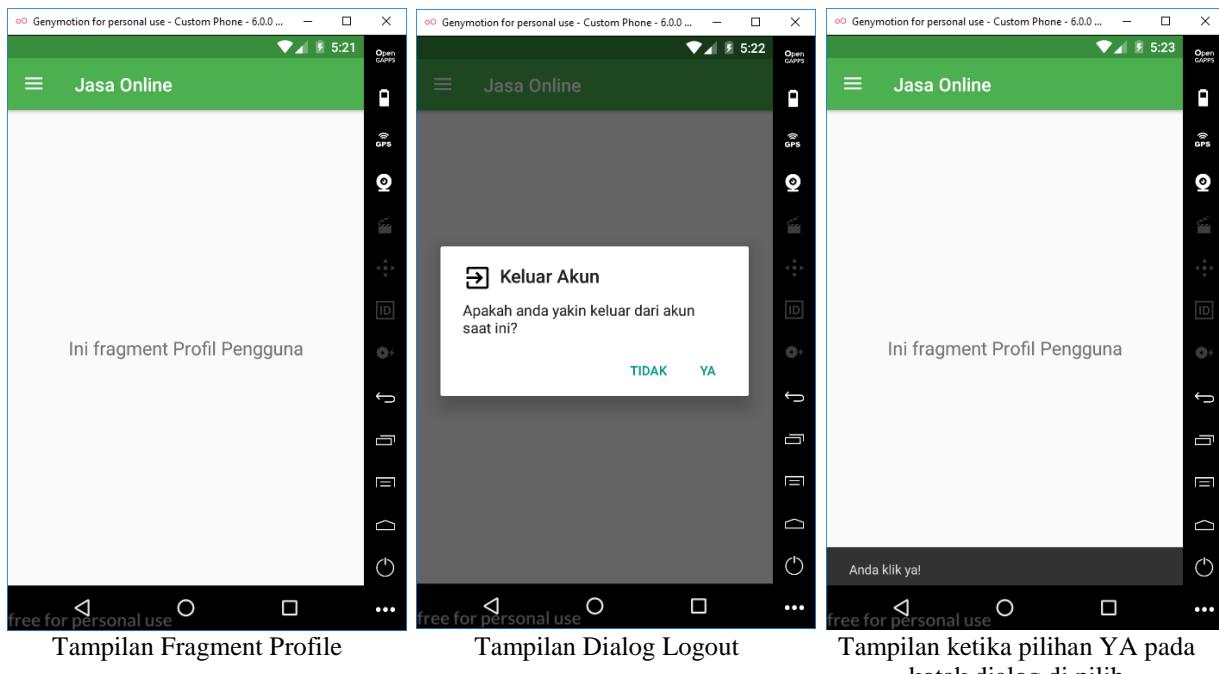
R.id.nav_logout -> {
    val builder = AlertDialog.Builder(this)
    builder.setTitle("Keluar Akun")
    builder.setMessage("Apakah anda yakin keluar dari akun
saat ini?")
    builder.setIcon(R.drawable.ic_exit_to_app_black_24dp)
    builder.setPositiveButton("Ya") { dialog, _ ->
        dialog.dismiss()
        Snackbar.make(drawer_layout, "Anda klik ya!", Snackbar.LENGTH_LONG).show()
    }
    builder.setNegativeButton("Tidak") { dialog, _ ->
        dialog.dismiss()
        Snackbar.make(drawer_layout, "Anda klik tidak!", Snackbar.LENGTH_LONG).show()
    }
    val alertDialog: AlertDialog = builder.create()
    alertDialog.show()
}

drawerLayout.closeDrawer(GravityCompat.START)
}
}

```

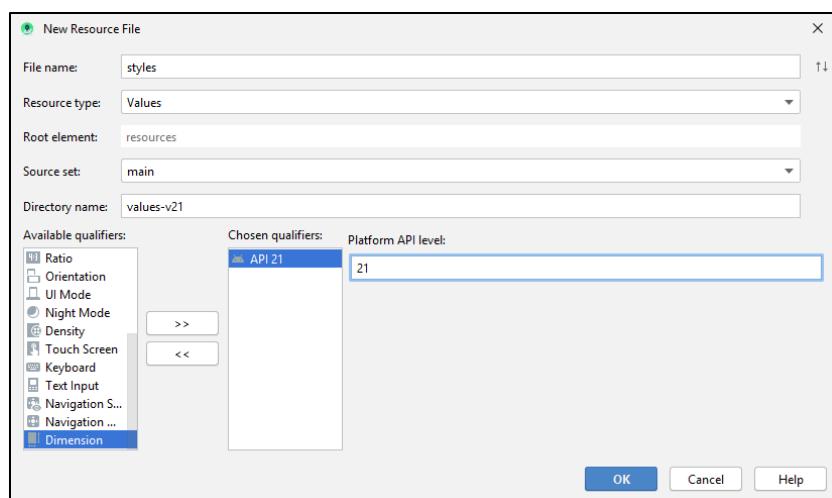
21. Jalankan program di Emulator Android. Perhatikan hasilnya dan lakukan analisa terhadap kode sumber yang ada dan dokumentasikan dilaporan praktikum.





22. Buatlah style baru untuk perangkat yang memiliki versi API 21 ke atas, tujuannya ialah membuat status bar aplikasi menjadi transparan. Caranya klik kanan pada app → New → Android Resource File. Masukkan pengaturan field seperti berikut:

- File Name: styles
- Resource type: Values
- Pilih version dari Available qualifiers dan klik button >>
- Platform API Lever: 21



Klik OK untuk membuat style.

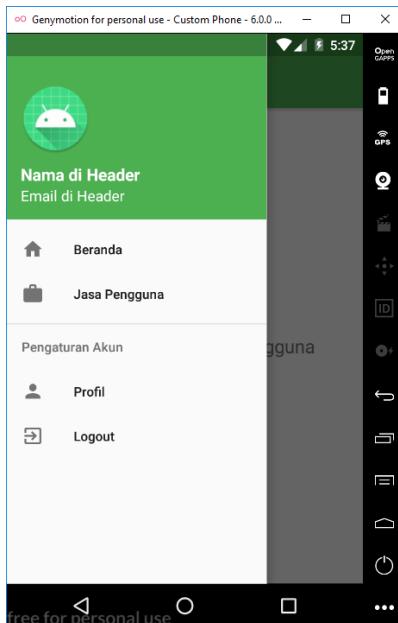
23. Ubahlah isi kode pada file styles.xml (v21) yang telah dibuat, sehingga menjadi seperti berikut:

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="AppTheme.NoActionBar">
        <item name="windowActionBar">false</item>
        <item name="windowNoTitle">true</item>
        <item
            name="android:statusBarColor">@android:color/transparent</item>
    </style>
</resources>

```

24. Jalankan program di Emulator Android (API 21). Perhatikan hasilnya dan lakukan analisa terhadap hasil tampilan drawer sekarang dan sebelumnya.



Tampilan Navigation Drawer

Membuat RESTful API pada Local Web Server

Sekilas tentang RESTful API

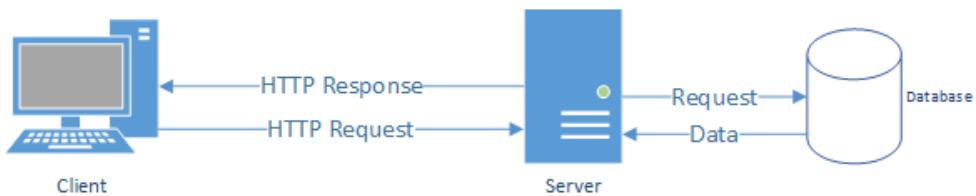
Sebelum kita membuat RESTful API ada baiknya kita memahami terlebih dahulu apa itu RESTful API. RESTful API adalah salah satu jenis implementasi dari API (*Application Programming Interface*). API itu sendiri adalah antarmuka komputasi untuk sebuah komponen atau sistem perangkat lunak yang mendefenisikan bagaimana cara komponen atau system lain dapat menggunakannya. Ia mendefenisikan pemanggilan (*calls*) atau permintaan (*request*) yang dibuat, bagaimana membuatnya, format data apa yang dapat digunakan, penamaan yang harus diikuti dan lain-lain (Wikipedia, 2020). Adapun tujuan dari API itu dibuat antara lain ialah untuk membuat aplikasi dengan performa yang baik, memudahkan pengembangan aplikasi pada

platform yang berbeda dan memudahkan integrasi sebuah aplikasi yang berbeda bahasa pemrograman.

REST (REpresentational State Transfer) adalah sebuah metode arsitektur standard berbasis web untuk komunikasi yang menggunakan protocol HTTP dalam melakukan pertukaran data, dan metode ini sering digunakan pengembangan aplikasi. Tujuannya ialah untuk membuat sistem dengan performa yang baik, cepat, skalabilitas khususnya dalam pertukaran data dan komunikasi.

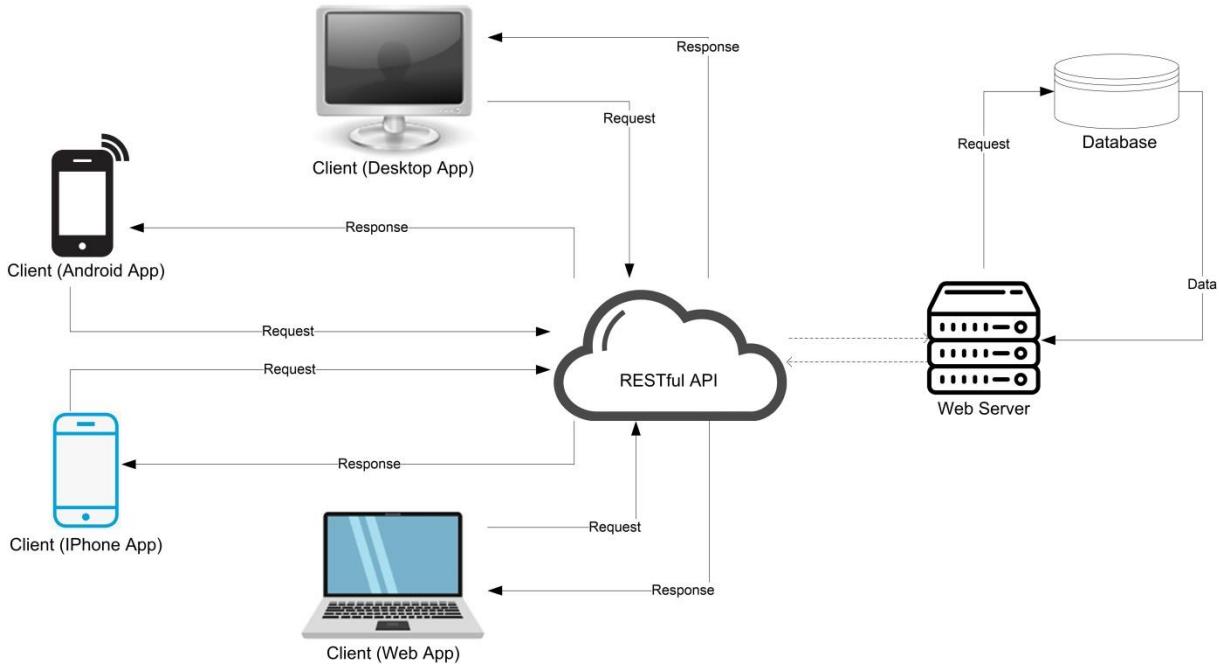
Kenapa kita membutuhkan RESTful API?

Untuk menjawab pertanyaan diatas perhatikan arsitektur perangkat lunak berikut:



Sumber : <http://mfikri.com/en/blog/nodejs-restful-api-mysql>

Gambar diatas adalah salah satu jenis arsitektur perangkat lunak *Client–Server*. Cara kerjanya adalah *client* mengirim permintaan ke *server* melalui **HTTP Request** dan *server* menyediakan sebuah respon melalui **HTTP Response**. Respon yang diberikan oleh server biasanya dalam format **HTML**. Bayangkan jika kita mengembangkan aplikasi berbasis *mobile* seperti Android dan IOS, apakah Android dan IOS mengharapkan respon berupa format HTML dari *server*? Tentu tidak, karena Android dan IOS menggunakan bahasa pemrograman yang berbeda dan tidak mengenali respon HTML pada umumnya. Itulah alasannya kenapa kita membutuhkan RESTful API, RESTful API menjadi jembatan komunikasi antara *client* dan *server* dalam menangani permintaan dan respon. Jadi, server tidak perlu mengirim respon dalam bentuk HTML yang belum tentu mendukung semua platform aplikasi yang dikirim hanyalah data yang biasanya berformat JSON atau XML.



Gambar: Arsitektur RESTful API.

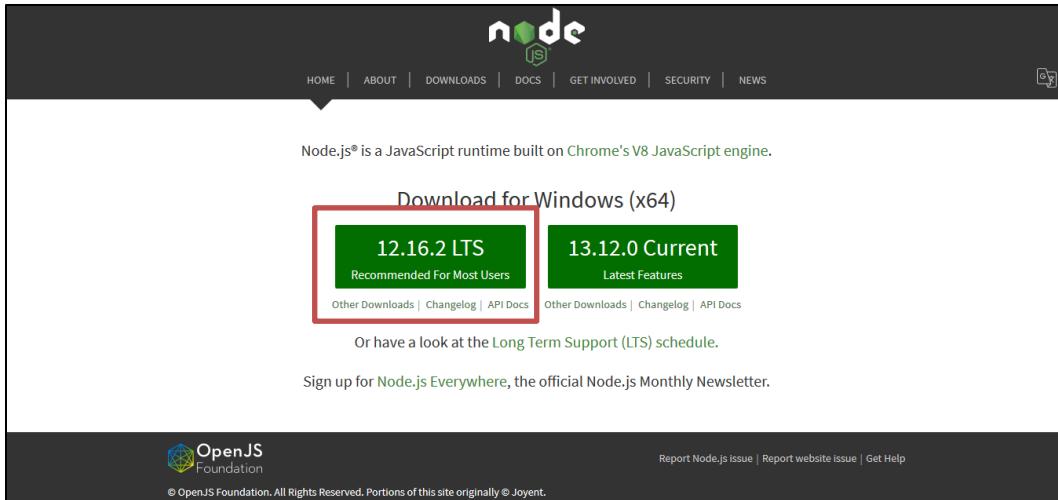
Berdasarkan arsitektur RESTful API pada gambar diatas memungkinkan beberapa platform yang berbeda untuk mengakses RESTful API yang kita buat seperti platform Desktop, Web dan Mobile. Ada banyak RESTful API yang bisa digunakan, pada praktikum ini kita menggunakan NodeJS dengan beberapa dependency yang dibutuhkan dan MySQL sebagai database.

Mempersiapkan perangkat lunak yang dibutuhkan

Adapun perangkat lunak yang perlu dipersiapkan adalah sebagai berikut:

- NodeJS (RESTful API)
- XAMPP (Apache Web Server dan database MySQL)
- Postman (Menguji request dan response layanan API)
- Visual Studio Code (Editor Teks JavaScript)

1. Download dan install software NodeJS. Bukalah halaman web : <https://nodejs.org/en/> silahkan download software NodeJS yang terakhir dirilis dan direkomendasikan untuk kebanyakan user (misal versi 12.16.2 LTS).



- Setelah berhasil diinstal, langkah berikutnya ialah melakukan pemeriksaan versi npm. Bukalah command prompt di folder project anda (jika belum ada buat terlebih dahulu misal jasa_online_api) ketik cmd di search direktori.



Masukkan perintah ‘npm –v’ di command prompt.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.17134.1365]
(c) 2018 Microsoft Corporation. All rights reserved.

D:\Project_NodeJS\jasa_online_api>npm -v
6.14.4

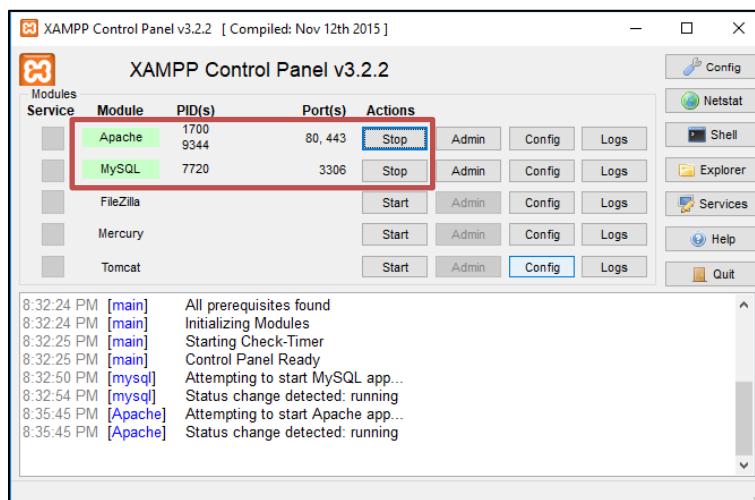
D:\Project_NodeJS\jasa_online_api>
```

Adapun versi npm yang digunakan oleh penulis adalah versi 6.14.4

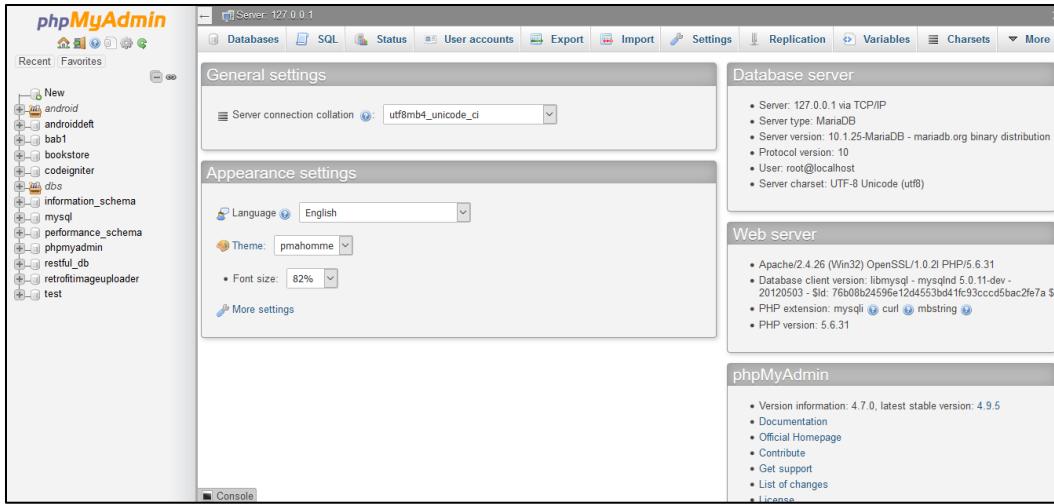
- Download dan install software XAMPP untuk keperluan webserver dan database mysql. Bukalah halaman web : <https://www.apachefriends.org/index.html> silahkan download software XAMPP yang terakhir dirilis sesuai dengan platform yang anda inginkan.



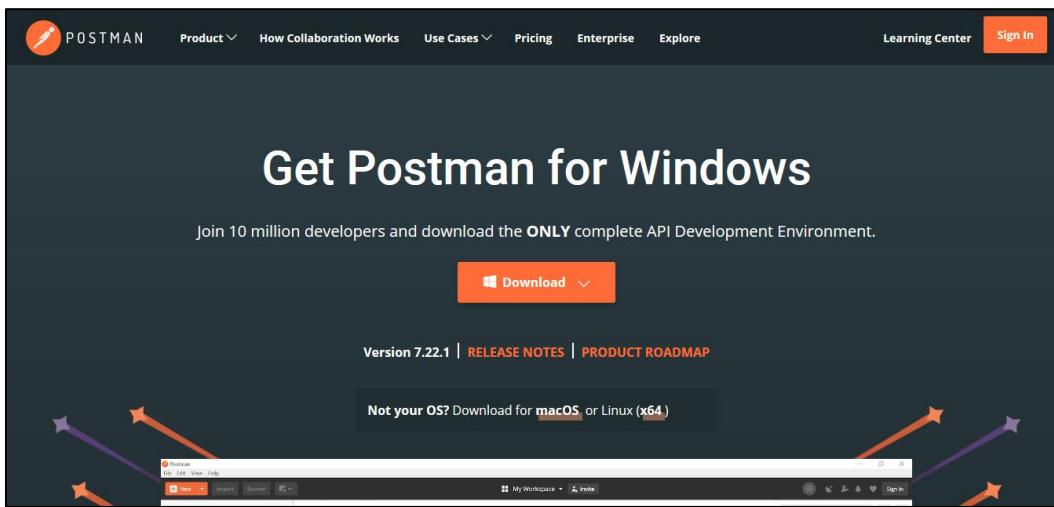
4. Setelah selesai instalasi. Jalankan layanan webserver Apache dan database MySQL pada aplikasi XAMPP Control Panel.



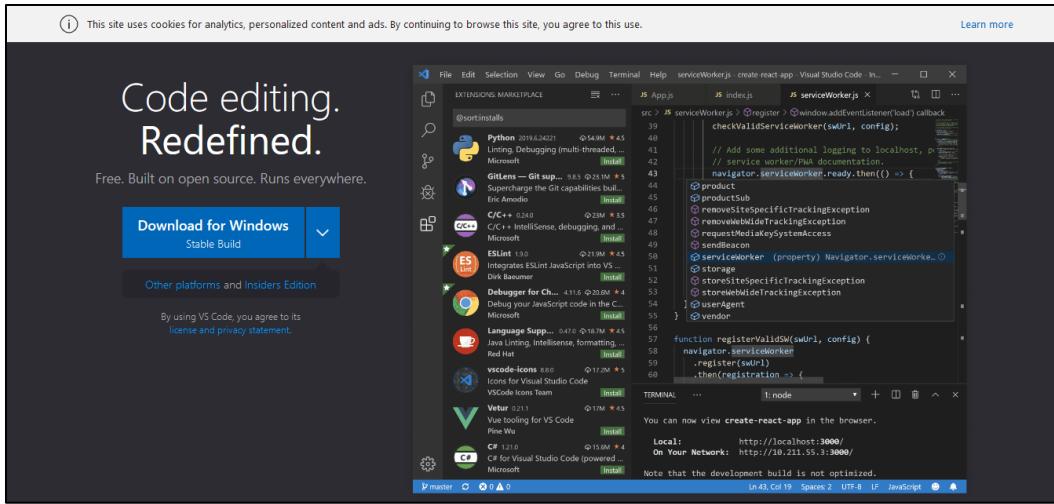
5. Buka halaman phpmyadmin untuk memeriksa apakah software XAMPP telah bekerja dengan baik dengan mengetikkan alamat url : <http://localhost/phpmyadmin/> di web browser.



6. Download dan install software Postman untuk keperluan pengujian request dan response layanan API yang kita buat. Buka halaman web : <https://www.postman.com/downloads/> silahkan download software Postman yang terakhir dirilis sesuai dengan platform yang anda inginkan.



7. Download dan install software Visual Studio Code sebagai editor teks bahasa pemrograman JavaScript (NodeJS). Buka halaman web : <https://code.visualstudio.com/> silahkan download software Visual Studio Code yang terakhir dirilis sesuai dengan platform yang anda inginkan.



Membuat database aplikasi di phpMyAdmin

1. Bukalah halaman phpmyadmin di web browser. Masukkan query “create database dbs_jasa_online;” pada tab SQL.

Hasil query:

2. Masukkan query “show database;” pada tab SQL.

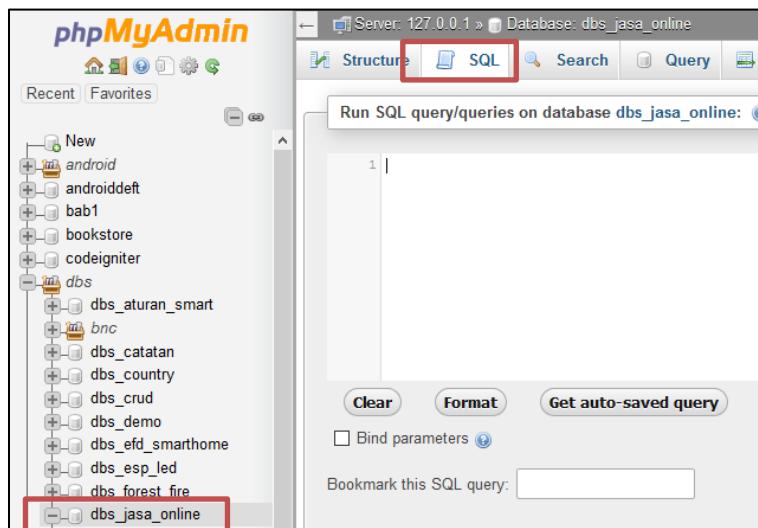
Your SQL query has been executed successfully

```
show databases
```

+ Options Database

- android
- android_api
- androiddef
- bab1**
- bookstore
- codeigniter
- dbs_aturan_smart
- dbs_bnc_baby
- dbs_bnc_full
- dbs_catatan
- dbs_country
- dbs_crud
- dbs_demo
- dbs_efd_smarthome
- dbs_esp_led
- dbs_forest_fire**
- dbs_jasa_online**

- Pilih database “dbs_jasa_online” pada side tab database dan pilih tab SQL.



- Masukkan query berikut pada field query:

```
CREATE TABLE IF NOT EXISTS tbl_user(
    id int(11) NOT NULL AUTO_INCREMENT,
    nama varchar(50) NOT NULL,
    tanggal_lahir varchar(50),
    jenis_kelamin varchar(50),
    nomor_hp varchar(20),
    alamat varchar(255),
```

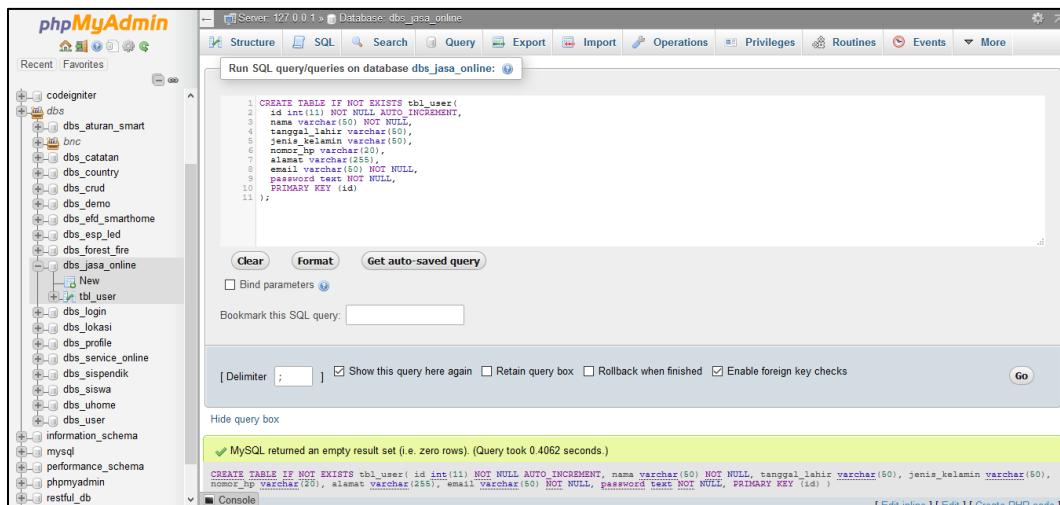
```

email varchar(50) NOT NULL,
password text NOT NULL,
PRIMARY KEY (id)
);

```

Pada tabel tbl_user terdapat satu primary key yaitu field id yang juga bersifat AUTO INCREAMENT.

Hasil query:



5. Masukkan query berikut pada field query:

```

CREATE TABLE IF NOT EXISTS `tbl_jasa`(
  `id_jasa` int NOT NULL AUTO_INCREMENT PRIMARY KEY,
  `id_user` int NOT NULL,
  `nama_jasa` varchar(50) NOT NULL,
  `deskripsi_singkat` varchar(255) NOT NULL,
  `uraian_deskripsi` text,
  `rating` int,
  `gambar` varchar(100) NOT NULL,
  CONSTRAINT `PK_ID_USER` FOREIGN KEY (`id_user`)
    REFERENCES `tbl_user`(`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

```

Pada tabel tbl_jasa terdapat satu primary key yaitu field id_jasa yang bersifat AUTO INCREAMENT. Satu foreign key yaitu field id_user dengan constraint PK_ID_USER yang menerapkan aturan ON DELETE CASCADE artinya jika suatu id atau user yang berasal dari tabel tbl_user terhapus maka field yang berelasi dengannya di tbl_jasa juga turut terhapus dan ON UPDATE CASCADE artinya jika suatu data pada tbl_user misalnya data pada field nama berubah maka data yang berelasi dengan tabel jasa juga akan turut terupdate.

Hasil query:

phpMyAdmin

Server: 127.0.0.1 | Database: dbs_jasa_online

Structure SQL Search Query Export Import Operations Privileges Routines Events More

Recent Favorites

Run SQL query/queries on database dbs_jasa_online: [?](#)

```
1 CREATE TABLE IF NOT EXISTS `tbl_jasa`(
2     `id_jasa` int NOT NULL AUTO_INCREMENT PRIMARY KEY,
3     `id_user` int NOT NULL,
4     `nama_jasa` varchar(50) NOT NULL,
5     `deskripsi_singkat` varchar(256) NOT NULL,
6     `urlan_deskripsi` text,
7     `rating` int,
8     `gambar` varchar(100) NOT NULL,
9     CONSTRAINT `PK_ID_JASA` FOREIGN KEY (`id_user`)
10    REFERENCES `tbl_user`(id)
11    ON DELETE CASCADE
12    ON UPDATE CASCADE
13 )
```

[Clear](#) [Format](#) [Get auto-saved query](#)

Bind parameters [?](#)

Bookmark this SQL query:

[Delimiter: ;] Show this query here again Retain query box Rollback when finished Enable foreign key checks [Go](#)

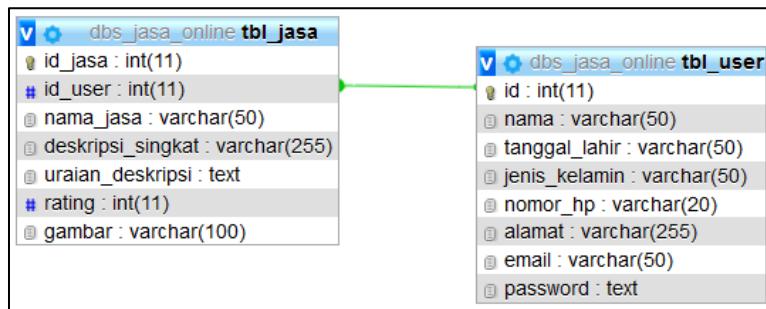
Hide query box

MySQL returned an empty result set (i.e. zero rows). (Query took 0.3120 seconds)

```
CREATE TABLE IF NOT EXISTS `tbl_jasa`(`id_jasa` int NOT NULL AUTO_INCREMENT PRIMARY KEY, `id_user` int NOT NULL, `nama_jasa` varchar(50) NOT NULL, `deskripsi_singkat` varchar(256) NOT NULL, `urlan_deskripsi` text, `rating` int, `gambar` varchar(100) NOT NULL, CONSTRAINT `PK_ID_JASA` FOREIGN KEY (`id_user`) REFERENCES `tbl_user`(id) ON DELETE CASCADE ON UPDATE CASCADE)
```

[Edit inline](#) [\[Edit \]](#) [\[Create PHP code \]](#)

6. Berikut tampilan relasi antar tabel `tbl_jasa` dan `tbl_user` pada database `dbs_jasa_online` dari menu designer.



7. Cobalah untuk menambahkan data pada tbl_user dengan sintaks sql berikut:

```
INSERT INTO tbl_user(nama, tanggal_lahir, jenis_kelamin, nomor_hp, alamat, email, password) VALUES ('Anton','1 Januari 2020', 'Pria', '081211111111', 'Jl. Damai', 'anton@gmail.com', 'tes12345'), ('Budi','2 Januari 2020', 'Pria', '081222222222', 'Jl. Sentosa', 'budi@gmail.com', 'tes12345'), ('Cut Monika','3 Januari 2020', 'Wanita', '081233333333', 'Jl. Makmur', 'cut@gmail.com', 'tes12345');
```

Hasil query select data:

Showing rows 0 - 2 (3 total, Query took 0.0034 seconds.)

```
SELECT * FROM tbl_user
```

Server: 127.0.0.1 > Database: dbs_jasa_online > Table: tbl_user

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#) [Privileges](#) [Operations](#) [Tracking](#) [Triggers](#)

	Edit	Copy	Delete	id	nama	tanggal_lahir	jenis_kelamin	nomor_hp	alamat	email	password
<input type="checkbox"/>	Edit	Copy	Delete	1	Anton	1 Januari 2020	Pria	081211111111	Jl. Damai	anton@gmail.com	tes12345
<input type="checkbox"/>	Edit	Copy	Delete	2	Budi	2 Januari 2020	Pria	081222222222	Jl. Sentosa	budi@gmail.com	tes12345
<input type="checkbox"/>	Edit	Copy	Delete	3	Cut Monika	3 Januari 2020	Wanita	081233333333	Jl. Makmur	cut@gmail.com	tes12345

[Check all](#) [With selected:](#) [Edit](#) [Copy](#) [Delete](#) [Export](#)

Show all | Number of rows: Filter rows: Sort by key:

8. Cobalah untuk menambahkan data pada `tbl_jasa` dengan sintaks sql berikut:

```
INSERT INTO tbl_jasa(id_user, nama_jasa, deskripsi_singkat, uraian_deskripsi, rating, gambar) VALUES
(1,'Service AC Barokah', 'Service segala merk AC', 'Rumah atau kantor hajar', 0,'test3.jpg'),
(3,'Tukang Pipa Hidayah', 'Memperbaiki segala pipa yang bocor', 'Rumah atau kantor siap', 0, 'test2.jpg'),
(3,'Tukang Rumah', 'Buat rumah ukuran apa saja', 'Rumah kayu dan batu lewat', 0, 'test3.jpg');
```

Hasil query select data:

	<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	id_jasa	id_user	nama_jasa	deskripsi_singkat	uraian_deskripsi	rating	gambar
	<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	Service AC Barokah	Service segala merk AC	Rumah atau kantor hajar	0	test3.jpg
	<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	3	Tukang Pipa Hidayah	Memperbaiki segala pipa yang bocor	Rumah atau kantor siap	0	test2.jpg
	<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	3	Tukang Rumah	Buat rumah ukuran apa saja	Rumah kayu dan batu lewat	0	test3.jpg

- Untuk menguji relasi antar tabel tbl_user dan tbl_jasa cobalah untuk menghapus data user dengan id user 3 dan select data pada tbl_jasa.

```
DELETE FROM tbl_user WHERE id = 3;
SELECT * FROM tbl_jasa;
```

Hasil query:

phpMyAdmin								
Server: 127.0.0.1 > Database: dbs_jasa_online > Table: tbl_jasa								
	<input type="checkbox"/> Browse <input type="checkbox"/> Structure <input type="checkbox"/> SQL <input type="checkbox"/> Search <input type="checkbox"/> Insert <input type="checkbox"/> Export <input type="checkbox"/> Import <input type="checkbox"/> Privileges <input type="checkbox"/> Operations <input type="checkbox"/> Tracking <input type="checkbox"/> Triggers							
	Show query box	<p>✓ 1 row affected. (Query took 0.1628 seconds.)</p> <pre>DELETE FROM tbl_user WHERE id = 3</pre>						
		<p>✓ Showing rows 0 - 0 (1 total, Query took 0.0020 seconds.)</p> <pre>SELECT * FROM tbl_jasa</pre>						
	+ Options	<p><input type="checkbox"/> Show all <input type="checkbox"/> Number of rows: 25 <input type="checkbox"/> Filter rows: Search this table</p>						
		<p><input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete</p>	1	1	Service AC Barokah	Service segala merk AC	Rumah atau kantor hajar	0 test3.jpg
		<p><input type="checkbox"/> Check all With selected: <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete <input type="checkbox"/> Export</p>						
		<p><input type="checkbox"/> Show all <input type="checkbox"/> Number of rows: 25 <input type="checkbox"/> Filter rows: Search this table</p>						
	Query results operations	<p><input type="checkbox"/> Print <input type="checkbox"/> Copy to clipboard <input type="checkbox"/> Export <input type="checkbox"/> Display chart <input type="checkbox"/> Create view</p>						
		<p><input type="checkbox"/> Console</p>						

Coba perhatikan data pada tbl_jasa dengan id_user 3 telah terhapus. Hal ini dikarenakan penerapan aturan ON DELETE CASCADE yang telah dijelaskan pada pembahasan sebelumnya.

Membuat RESTful API dengan NodeJS

Adapun deskripsi RESTful API yang akan dibuat akan menangani perintah tambah, hapus, ubah, dan menampilkan data pada tabel tbl_user dan tbl_jasa. Berikut adalah jenis-jenis *request* yang tersedia pada rancangan RESTful API yang dibuat:

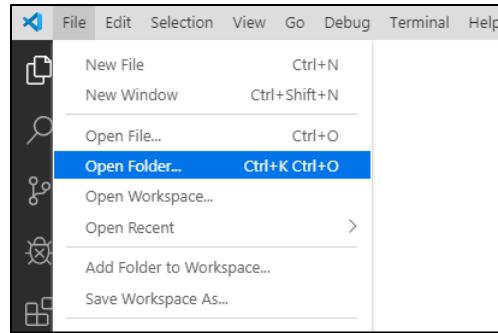
Metode	Url	Aksi
GET	/	Menampilkan pesan dari server
GET	/users	Menampilkan seluruh data user
GET	/users/:user_id	Menampilkan data user dengan id tertentu
GET	/login	Melakukan login user dengan email dan password tertentu
POST	/users	Mendaftarkan akun user
PUT	/users	Mengubah data user
DELETE	/users/:id	Menghapus user dengan id tertentu
POST	/services	Menambahkan jasa
GET	/services	Menampilkan seluruh jasa user
GET	/services/:id	Menampilkan data jasa dengan id user tertentu
DELETE	/services/:id_jasa	Menghapus jasa dengan id jasa tertentu
PUT	/services	Mengubah data jasa (dengan seluruh field termasuk image sebagai query)
PUT	/services/:id_jasa	Mengubah data jasa (dengan beberapa field tertentu tidak termasuk image sebagai query)

Secara singkat kita akan membuat 7 file berikut untuk membuat RESTful API:

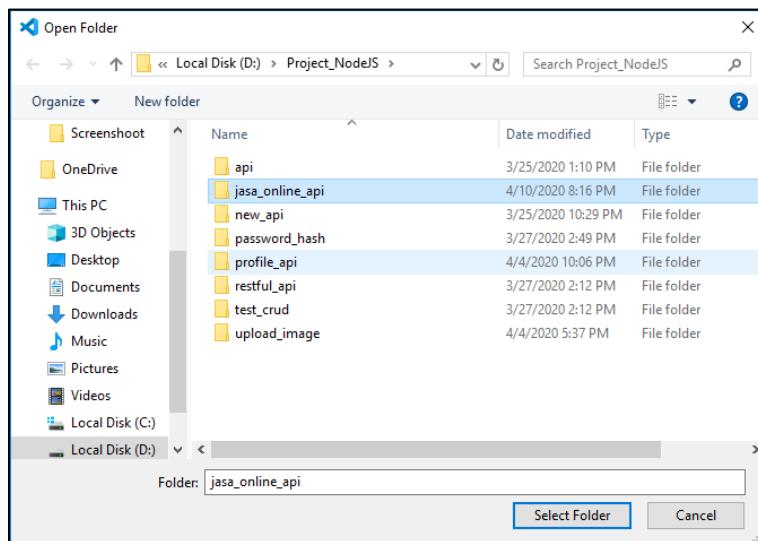
1. Membuat express web server pada file ***index.js***
2. Membuat file yang menangani koneksi ke database dengan nama ***conn.js***
3. Membuat file kontrol aplikasi dengan nama ***controller.js***
4. Membuat file routing aplikasi dengan nama ***routes.js***
5. Membuat file yang menangani enkripsi password user dengan nama ***encrypt_password.js***
6. Membuat file yang menangani respon API dalam format JSON dengan nama ***res.js***
7. Membuat file yang menangani request yang disertai image dari client dengan nama ***helpers.js***

Berikut adalah langkah-langkah untuk membuat RESTful API:

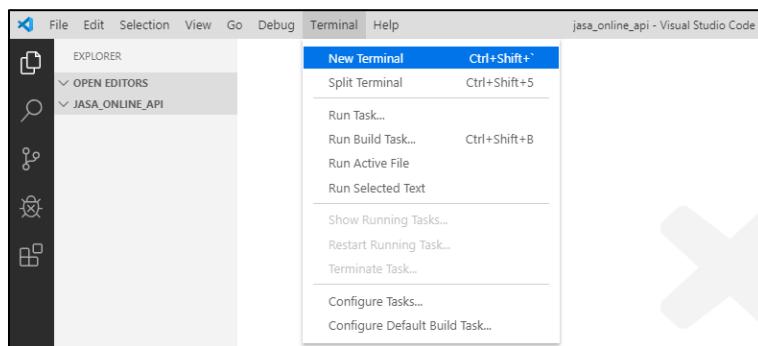
1. Bukalah aplikasi Visual Studio Code. Pilih menu File → Open Folder...



Arahkan ke folder project yang telah dibuat “jasa_online_api”



2. Bukalah terminal baru di Visual Studio Code caranya klik Menu Terminal → New Terminal.



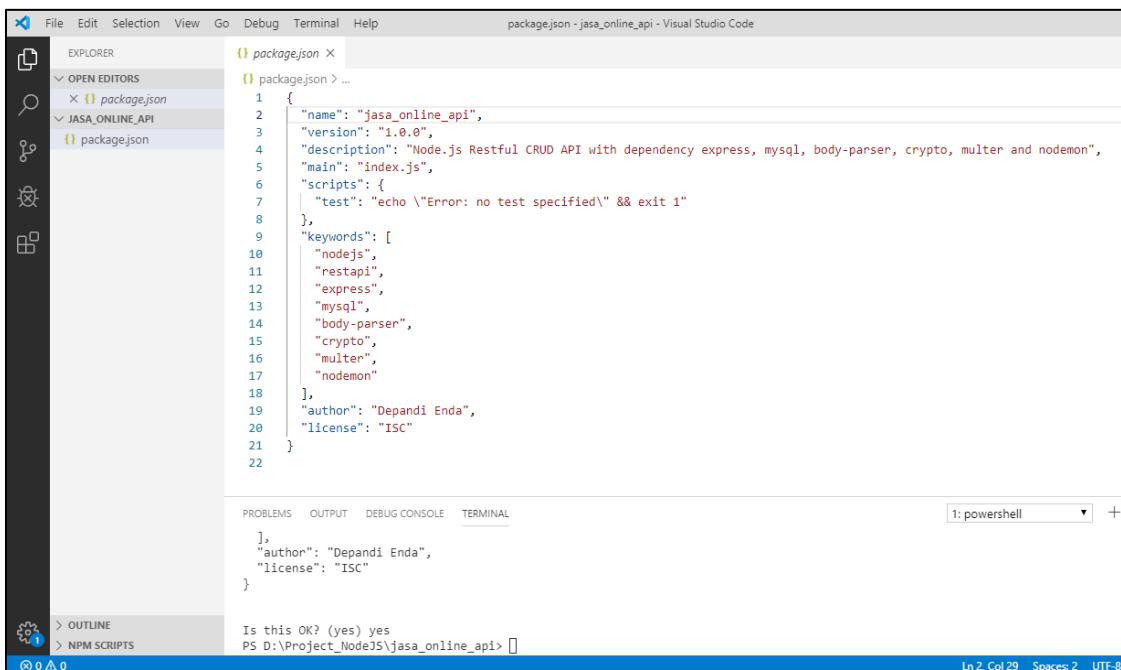
Tampilan terminal di Visual Studio Code:



3. Masukkan sintaks : ‘npm init’ pada terminal dan masukkan konfigurasi package seperti berikut:

```
package name: (jasa_online_api)
version: (1.0.0)
description: Node.js Restful CRUD API with dependency express, mysql, body-parser, crypto, multer and nodemon
entry point: (index.js)
test command:
git repository:
keywords: nodejs, restapi, express, mysql, body-parser, crypto, multer, nodemon
author: Depandi Enda
license: (ISC)
```

Hasilnya akan membuat sebuah file package.json pada folder proyek seperti berikut:



```
File Edit Selection View Go Debug Terminal Help
package.json - jasa_online_api - Visual Studio Code

EXPLORER OPEN EDITORS JASA_ONLINE_API package.json
1 {
  "name": "jasa_online_api",
  "version": "1.0.0",
  "description": "Node.js Restful CRUD API with dependency express, mysql, body-parser, crypto, multer and nodemon",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [
    "nodejs",
    "restapi",
    "express",
    "mysql",
    "body-parser",
    "crypto",
    "multer",
    "nodemon"
  ],
  "author": "Depandi Enda",
  "license": "ISC"
}

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Is this OK? (yes) yes
PS D:\Project_NodeJS\jasa_online_api>
```

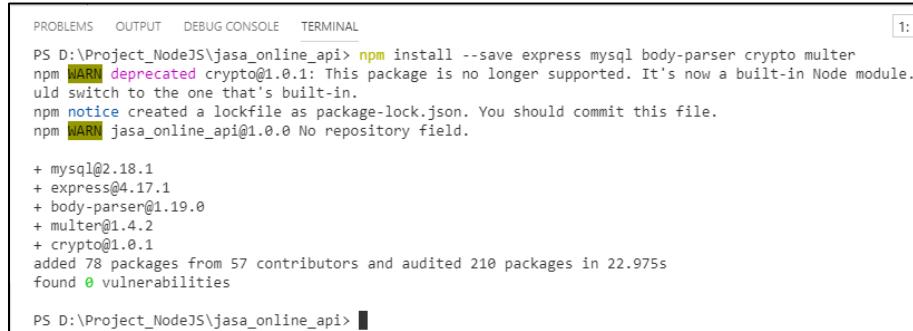
4. Menginstal beberapa dependency yang dibutuhkan.

- Express (node.js framework untuk menangani routing, middleware dan view system)
- MySQL (MySQL driver untuk node.js)
- Body-parser (middleware untuk menangani request post body)
- Crypto (untuk mengenkripsi password user)
- Multer (menangani multipart data seperti image)
- Nodemon (menangkap jejak perubahan pada file dalam project dan secara otomatis merestart server)

Cara menginstall dependency jalankan sintaks berikut di terminal:

```
npm install --save express mysql body-parser crypto multer
```

Hasil instalasi:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Project_NodeJS\jasa_online_api> npm install --save express mysql body-parser crypto multer
npm [WARN] deprecated crypto@1.0.1: This package is no longer supported. It's now a built-in Node module.
  old switch to the one that's built-in.
npm [notice] created a lockfile as package-lock.json. You should commit this file.
npm [WARN] jasa_online_api@1.0.0 No repository field.

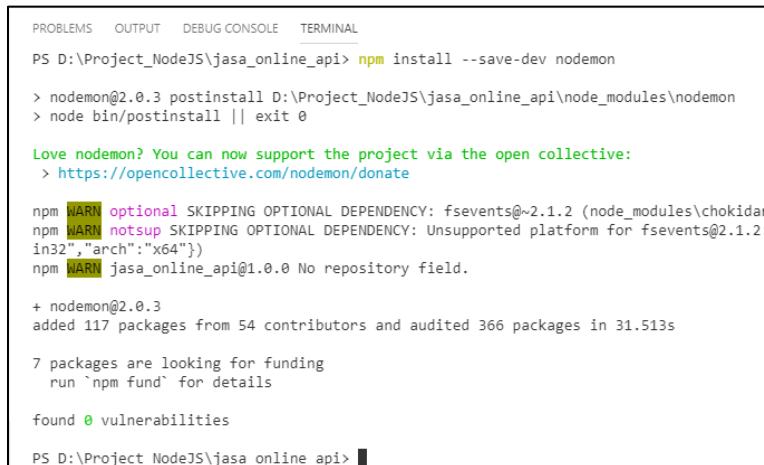
+ mysql@2.18.1
+ express@4.17.1
+ body-parser@1.19.0
+ multer@1.4.2
+ crypto@1.0.1
added 78 packages from 57 contributors and audited 210 packages in 22.975s
found 0 vulnerabilities

PS D:\Project_NodeJS\jasa_online_api>
```

Jalankan sintaks berikut untuk menginstal dependency nodemon:

```
npm install --save-dev nodemon
```

Hasil instalasi:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Project_NodeJS\jasa_online_api> npm install --save-dev nodemon
> nodemon@2.0.3 postinstall D:\Project_NodeJS\jasa_online_api\node_modules\nodemon
> node bin/postinstall || exit 0

Love nodemon? You can now support the project via the open collective:
> https://opencollective.com/nodemon/donate

npm [WARN] optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.1.2 (node_modules\chokidar\node_modules\fsevents)
npm [WARN] notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.2
  os: win32 "arch":"x64"}
npm [WARN] jasa_online_api@1.0.0 No repository field.

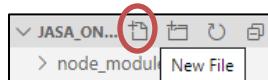
+ nodemon@2.0.3
added 117 packages from 54 contributors and audited 366 packages in 31.513s

7 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

PS D:\Project_NodeJS\jasa_online_api>
```

- Untuk membuat file baru didalam project anda caranya klik pada icon +document seperti pada gambar dibawah ini:



Masukkan nama file baru yang akan dibuat misalnya : index.js

- Buatlah file conn.js dan masukkan kode berikut:

```
var mysql = require('mysql');

var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "",
  database: "dbs_jasa_online"
});

con.connect(function (err){
```

```

    if(err){
        console.log("error: " + err.message);
        process.exit(1)
    }
    console.log('Mysql terkoneksi...');

});

module.exports = con;

```

7. Buatlah file res.js dan masukkan kode berikut:

```

'use strict';

//jika query sukses maka format json yang dikembalikan yaitu message, error = false dan data
exports.ok_with_data = function(message, values, res) {
    var data = {
        'message': message,
        'error': false,
        'data': values
    };
    res.end(JSON.stringify(data));
};

//jika query sukses maka format json yang dikembalikan yaitu message, error = false
exports.ok = function(message, res) {
    var data = {
        'message': message,
        'error': false
    };
    res.end(JSON.stringify(data));
};

//jika query gagal maka format json yang dikembalikan yaitu message, error = true
exports.gagal = function(message, res){
    var data = {
        'message': message,
        'error': true
    };
    res.end(JSON.stringify(data));
};

```

8. Buatlah file helpers.js dan masukkan kode berikut:

```

//menangani request yang disertai image
var multer = require('multer');
var path = require('path');

exports.imageFilter = function(req, file, cb) {
    // Accept images only
    if (!file.originalname.match(/\.(jpg|JPG|jpeg|JPEG|png|PNG|gif|GIF)$/)) {
        req.fileValidationError = 'Only image files are allowed!';
        return cb(new Error('Only image files are allowed!'), false);
    }
    cb(null, true);
};

```

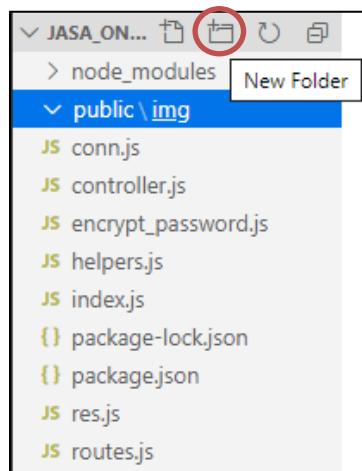
```

exports.storage = multer.diskStorage({
    destination: function(req, file, cb) {
        cb(null, './public/img/');
    },

    // By default, multer removes file extensions so let's add them back
    filename: function(req, file, cb) {
        cb(null, file.fieldname + '-'
        + Date.now() + path.extname(file.originalname));
    }
});

```

9. Kita akan menyimpan file gambar yang diupload pada folder public/img, oleh karena itu buatlah folder tersebut di dalam folder proyek anda.



10. Buatlah file encrypt_password.js dan masukkan kode berikut:

```

'use strict';
var crypto = require('crypto');

/**
 * generates random string of characters i.e salt
 * @function
 * @param {number} length - Length of the random string.
 */
var genRandomString = function(length){
    return crypto.randomBytes(Math.ceil(length/2))
        .toString('hex') /* convert to hexadecimal format */
        .slice(0,length); /* return required number of characters */
};

/**
 * hash password with sha512.
 * @function
 * @param {string} password - List of required fields.
 * @param {string} salt - Data to be validated.
 */
var sha512 = function(password, salt){
    var hash = crypto.createHmac('sha512', salt); /* Hashing algorithm sha512 */
}

```

```

        hash.update(password);
        var value = hash.digest('base64');
        return {
            salt:salt,
            passwordHash:value
        };
    };

exports.saltHashPassword = function(userpassword) {
    try {
        var salt = genRandomString(16); /* Gives us salt of length 16 */
        var passwordData = sha512(userpassword, salt);
        //console.log('UserPassword = '+userpassword);
        //console.log('Passwordhash = '+passwordData.passwordHash);
        //console.log('nSalt = '+passwordData.salt);
        passwordData = passwordData.passwordHash + "$" + passwordData.salt
        return passwordData;
    }catch (e) {
        return "";
    }
}

exports.compareHashPassword = function(userpassword, hashsaltpassword){
    try {
        var pass = hashsaltpassword.split("$");
        let salt = pass[1];
        let hash = crypto.createHmac('sha512', salt).update(userpassword).digest("base64");
        if (hash === pass[0]){
            //console.log('Password cocok');
            return true;
        }else{
            //console.log('Password tidak cocok!');
            return false;
        }
    }catch (e) {
        return false;
    }
}
}

```

11. Buatlah file controller.js dan masukkan kode berikut:

```

'use strict';

var response = require('./res');
var connection = require('./conn');
var encrypt = require('./encrypt_password');

exports.index = function(req, res) {
    response.ok("Hello, pesan ini dari sisi server NodeJS Restful API!", res)
};

exports.users = function(req, res) {
    connection.query('SELECT * FROM tbl_user', function (error, rows, fields){
        var message;
        if(error){

```

```

        message = "Error ketika menampilkan semua data user!";
        response.gagal(message, res);
    } else{
        message = "Sukses menampilkan semua data user";
        response.ok_with_data(message, rows, res);
    }
});

exports.findUsers = function(req, res) {

    var user_id = req.params.user_id;
    var message;

    if (!user_id) {
        message = "Kehilangan beberapa parameter yang dibutuhkan!";
        return response.gagal(message, res);
    }

    connection.query('SELECT * FROM tbl_user where id = ?',
    [ user_id ],
    function (error, rows, fields){
        if(error){
            message = "Error ketika mencari user yang dimaksud!";
            response.gagal(message, res);
        } else{
            if(rows.length > 0){
                message = "User ditemukan";
                response.ok_with_data(message, rows[0], res);
            }else{
                message = "User tidak ditemukan!";
                response.gagal(message, res);
            }
        }
    });
};

exports.loginUser = async function(req, res) {

    var email = req.query.email;
    var myPassword = req.query.password;
    var message;

    if (!email || !myPassword) {
        message = "Kehilangan beberapa parameter yang dibutuhkan!";
        return response.gagal(message, res);
    }

    await connection.query('SELECT * FROM tbl_user where email = ?',
    [ email ],
    function(error, rows, fields){
        if(rows.length > 0){
            if(encrypt.compareHashPassword(myPassword, rows[0].password)){
                message = "Login berhasil";
                response.ok_with_data(message, rows[0], res);
            }
        }
    });
}

```

```

        }else{
            message = "Password user tidak cocok!";
            response.gagal(message, res);
        }
    }else{
        message = "User tidak terdaftar!";
        response.gagal(message, res);
    }
});

exports.registerUser = async function(req, res) {

    var nama = req.body.nama;
    var tanggal_lahir = req.body.tanggal_lahir;
    var jenis_kelamin = req.body.jenis_kelamin;
    var nomor_hp = req.body.nomor_hp;
    var alamat = req.body.alamat;
    var email = req.body.email;
    var plain_password = req.body.password;
    var message;

    if (!nama || !tanggal_lahir || !jenis_kelamin || !nomor_hp || !alamat || !email || !plain_password) {
        message = "Kehilangan beberapa parameter yang dibutuhkan!";
        return response.gagal(message, res);
    }

    await connection.query('SELECT * FROM tbl_user where email = ?',
    [ email ],
    function(error, rows, fields){
        if(rows.length > 0){
            message = "Email user telah terdaftar, silahkan pilih yang lain!";
            response.gagal(message, res);
        }else{
            var encrypt_password = encrypt.saltHashPassword(plain_password);
            connection.query('INSERT INTO tbl_user (nama, tanggal_lahir, jenis_kelamin, nomor_hp, alamat, email, password) values (?,?,?,?,?,?,?)',
            [ nama, tanggal_lahir, jenis_kelamin, nomor_hp, alamat, email, encrypt_password ],
            function (error, rows, fields){
                if(error){
                    message = "Error ketika menambahkan user baru!";
                    response.gagal(message, res);
                } else{
                    message = "Berhasil menambahkan user baru";
                    response.ok(message, res);
                }
            });
        }
    });
};

exports.updateUser = async function(req, res) {

```

```

var user_id = req.body.id;
var nama = req.body.nama;
var tanggal_lahir = req.body.tanggal_lahir;
var jenis_kelamin = req.body.jenis_kelamin;
var nomor_hp = req.body.nomor_hp;
var alamat = req.body.alamat;
var email = req.body.email;
var plain_password = req.body.password;
var message;

if (!nama || !tanggal_lahir || !jenis_kelamin || !nomor_hp || !alamat || !email) {
    message = "Kehilangan beberapa parameter yang dibutuhkan!";
    return response.gagal(message, res);
}

var sql_query;
var params;
if(plain_password == ""){
    sql_query = 'UPDATE tbl_user SET nama = ?, tanggal_lahir = ?, jenis_kelamin = ?, nomor_hp = ?, alamat = ?, email = ? WHERE id = ?';
    params = [ nama, tanggal_lahir, jenis_kelamin, nomor_hp, alamat, email, user_id ]
} else{
    sql_query = 'UPDATE tbl_user SET nama = ?, tanggal_lahir = ?, jenis_kelamin = ?, nomor_hp = ?, alamat = ?, email = ?, password = ? WHERE id = ?';
    var encrypt_password = encrypt.saltHashPassword(plain_password);
    params = [ nama, tanggal_lahir, jenis_kelamin, nomor_hp, alamat, email, encrypt_password, user_id ]
}

await connection.query('SELECT * FROM tbl_user where email = ? and id <> ?', [email, user_id ],
function(error, rows, fields){
    if(rows.length > 0){
        message = "Email user telah terdaftar, silahkan pilih yang lain!";
        response.gagal(message, res);
    } else{
        connection.query(sql_query, params, function (error, rows, fields){
            if(error){
                message = "Error ketika mengubah data user!";
                response.gagal(message, res);
            } else{
                message = "Berhasil mengubah data user";
                response.ok(message, res);
            }
        });
    }
});
};

exports.deleteUser = async function(req, res) {

var user_id = req.params.id;
var message;

```

```

if (!user_id) {
    message = "Kehilangan beberapa parameter yang dibutuhkan!";
    return response.gagal(message, res);
}

await connection.query('SELECT * FROM tbl_user where id = ?',
[ user_id ],
function(error, rows, fields){
    if(rows.length > 0){
        connection.query('DELETE FROM tbl_user WHERE id = ?',
        [ user_id ],
        function (error, rows, fields){
            if(error){
                message = "Error ketika menghapus user!";
                response.gagal(message, res);
            } else{
                message = "Berhasil menghapus user";
                response.ok(message, res);
            }
        });
    } else{
        message = "User tidak ditemukan!";
        response.gagal(message, res);
    }
});
};

//limit file size
const MAX_SIZE = 1 * 1000 * 1000; // 1 MB
//for request with image multipart
var helpers = require('./helpers');
var multer = require('multer');
var fs = require('fs');

exports.addService = function(req, res){

    let upload = multer({storage: helpers.storage, limits: { fileSize: MAX_SIZE },
    fileFilter: helpers.imageFilter}).single('file');

    upload(req, res, function(err) {
        var message;
        // req.file contains information of uploaded file

        if (req.fileValidationError) {
            message = req.fileValidationError;
            return response.gagal(message, res);
        }
        else if (!req.file) {
            message = "Please select an image to upload or choose image under 1MB";
            return response.gagal(message, res);
        }
        else if (err instanceof multer.MulterError) {
            message = "MulterError upload image";
            return response.gagal(message, res);
        }
    });
}

```

```

        return response.gagal(message, res);
    }
    else if (err) {
        message = "Error upload image";
        return response.gagal(message, res);
    }

    // req.body contains information of text fields, if there were any
    var id_user = req.body.id_user;
    var nama_jasa = req.body.nama_jasa;
    var deskripsi_singkat = req.body.deskripsi_singkat;
    var uraian_deskripsi = req.body.uraian_deskripsi;
    var rating = req.body.rating;
    var gambar = req.body.gambar;

    if (!id_user || !nama_jasa || !deskripsi_singkat || !uraian_deskripsi || !rating || !gambar) {
        message = "Kehilangan beberapa parameter yang dibutuhkan!";
        return response.gagal(message, res);
    }

    gambar = req.file.filename;

    connection.query('INSERT INTO tbl_jasa (id_user, nama_jasa, deskripsi_singkat, uraian_deskripsi, rating, gambar) values (?, ?, ?, ?, ?, ?)', [ id_user, nama_jasa, deskripsi_singkat, uraian_deskripsi, rating, gambar ],
        function (error, rows, fields){
            if(error){
                //removeImage;
                fs.unlink('./public/img/' + req.file.filename, function (err) {
                    if (err) throw err;
                    // if no error, file has been deleted successfully
                    //console.log('File deleted!');
                });
                message = "Error ketika menambahkan jasa!";
                response.gagal(message, res);
            } else{
                message = "Berhasil menambahkan jasa";
                response.ok(message, res);
            }
        });
    });
};

exports.getServices = function(req, res){
    connection.query('SELECT tbl_jasa.id_jasa, tbl_user.nama as nama_penyedia, tbl_user.nomor_hp, tbl_jasa.nama_jasa, tbl_jasa.deskripsi_singkat, tbl_jasa.uraian_deskripsi, tbl_jasa.rating, tbl_jasa.gambar FROM tbl_jasa, tbl_user WHERE tbl_user.id = tbl_jasa.id_user ORDER BY tbl_jasa.id_jasa DESC', function (error, rows, fields){
        var message;
        if(error){
            message = "Error ketika menampilkan semua data jasa!";
            response.gagal(message, res);
        }
    });
};

```

```

    } else{
        message = "Sukses menampilkan semua data jasa";
        response.ok_with_data(message, rows, res);
    }
});

};

exports.getUserServices = function(req, res){
    var user_id = req.params.id;
    var message;

    if (!user_id) {
        message = "Kehilangan beberapa parameter yang dibutuhkan!";
        return response.gagal(message, res);
    }

    connection.query('SELECT tbl_jasa.id_jasa, tbl_user.nama as nama_penyedia, tb
l_user.nomor_hp, tbl_jasa.nama_jasa, tbl_jasa.deskripsi_singkat, tbl_jasa.uraian_
deskripsi, tbl_jasa.rating, tbl_jasa.gambar FROM tbl_jasa, tbl_user WHERE tbl_us
er.id = tbl_jasa.id_user AND tbl_user.id = ? ORDER BY tbl_jasa.id_jasa DESC',
    [ user_id ],
    function (error, rows, fields){
        if(error){
            message = "Error ketika menampilkan semua data jasa pengguna!";
            response.gagal(message, res);
        } else{
            message = "Sukses menampilkan semua data jasa pengguna";
            response.ok_with_data(message, rows, res);
        }
    });
};

exports.deleteService = async function(req, res){
    var id_jasa = req.params.id_jasa;
    var message;

    if (!id_jasa) {
        message = "Kehilangan beberapa parameter yang dibutuhkan!";
        return response.gagal(message, res);
    }

    await connection.query('SELECT * FROM tbl_jasa where id_jasa = ?',
    [ id_jasa ],
    function(error, rows, fields){
        if(rows.length > 0){
            var filename = rows[0].gambar
            connection.query('DELETE FROM tbl_jasa WHERE id_jasa = ?',
            [ id_jasa ],
            function (error, rows, fields){
                if(error){
                    message = "Error ketika menghapus jasa!";
                    response.gagal(message, res);
                } else{
                    fs.unlink('./public/img/' + filename, function (err) {
                        if (err) throw err;
                    });
                }
            });
        }
    });
};

```

```

        // if no error, file has been deleted successfully
        //console.log('File deleted!');
    });
    message = "Berhasil menghapus jasa";
    response.ok(message, res);
}
});
}};

} else{
    message = "Jasa tidak ditemukan!";
    response.gagal(message, res);
}
});

exports.editServiceReplaceImage = function(req, res){

    let upload = multer({storage: helpers.storage, limits: { fileSize: MAX_SIZE }
, fileFilter: helpers.imageFilter}).single('file');

    upload(req, res, async function(err) {
        var message;
        // req.file contains information of uploaded file

        if (req.fileValidationError) {
            message = req.fileValidationError;
            return response.gagal(message, res);
        }
        else if (!req.file) {
            message = "Please select an image to upload or choose image under 1MB ";
            return response.gagal(message, res);
        }
        else if (err instanceof multer.MulterError) {
            message = "MulterError upload image";
            return response.gagal(message, res);
        }
        else if (err) {
            message = "Error upload image";
            return response.gagal(message, res);
        }

        // req.body contains information of text fields, if there were any
        var id_jasa = req.body.id_jasa;
        var nama_jasa = req.body.nama_jasa;
        var deskripsi_singkat = req.body.deskripsi_singkat;
        var uraian_deskripsi = req.body.uraian_deskripsi;
        var gambar = req.body.gambar;

        if (!id_jasa || !nama_jasa || !deskripsi_singkat || !uraian_deskripsi ||
!gambar) {
            message = "Kehilangan beberapa parameter yang dibutuhkan!";
            return response.gagal(message, res);
        }

        gambar = req.file.filename;
    });
}
});
```

```

        await connection.query('SELECT * FROM tbl_jasa where id_jasa = ?',
        [ id_jasa ],
        function(error, rows, fields){
            if(rows.length > 0){
                var sql_query = 'UPDATE tbl_jasa SET nama_jasa = ?, deskripsi_singkat = ?, uraian_deskripsi = ?, gambar = ? WHERE id_jasa = ?';
                var params = [ nama_jasa, deskripsi_singkat, uraian_deskripsi, gambar, id_jasa ]
                var filename = rows[0].gambar
                connection.query(sql_query, params, function (error, rows, fields){
                    if(error){
                        message = "Error ketika mengubah data jasa!";
                        response.gagal(message, res);
                    } else{
                        fs.unlink('./public/img/' + filename, function (err) {
                            if (err) throw err;
                        });
                        message = "Berhasil mengubah data jasa";
                        response.ok(message, res);
                    }
                });
            } else{
                message = "Jasa yang mau diubah telah terhapus!";
                response.gagal(message, res);
            }
        });
    });

exports.editService = async function(req, res){

    // req.body contains information of text fields, if there were any
    var id_jasa = req.params.id_jasa;
    var nama_jasa = req.body.nama_jasa;
    var deskripsi_singkat = req.body.deskripsi_singkat;
    var uraian_deskripsi = req.body.uraian_deskripsi;
    var message;

    if (!id_jasa || !nama_jasa || !deskripsi_singkat || !uraian_deskripsi) {
        message = "Kehilangan beberapa parameter yang dibutuhkan!";
        return response.gagal(message, res);
    }

    await connection.query('SELECT * FROM tbl_jasa where id_jasa = ?',
    [ id_jasa ],
    function(error, rows, fields){
        if(rows.length > 0){
            var sql_query = 'UPDATE tbl_jasa SET nama_jasa = ?, deskripsi_singkat = ?, uraian_deskripsi = ? WHERE id_jasa = ?';
            var params = [ nama_jasa, deskripsi_singkat, uraian_deskripsi, id_jasa ]
            connection.query(sql_query, params, function (error, rows, fields){
                if(error){

```

```

        message = "Error ketika mengubah data jasa!";
        response.gagal(message, res);
    } else{
        message = "Berhasil mengubah data jasa";
        response.ok(message, res);
    }
});
}else{
    message = "Jasa yang mau diubah telah terhapus!";
    response.gagal(message, res);
}
});
};

```

12. Buatlah file routes.js dan masukkan kode berikut:

```

'use strict';

module.exports = function(app) {
    var todoList = require('./controller');

    //routing untuk index
    app.route('/')
        .get(todoList.index);

    //routing untuk user
    app.route('/users')
        .get(todoList.users);

    app.route('/users/:user_id')
        .get(todoList.findUsers);

    app.route('/login')
        .get(todoList.loginUser);

    app.route('/users')
        .post(todoList.registerUser);

    app.route('/users')
        .put(todoList.updateUser);

    app.route('/users/:id')
        .delete(todoList.deleteUser);

    //routing untuk jasa
    app.route('/services')
        .post(todoList.addService);

    app.route('/services')
        .get(todoList.getServices);

    app.route('/services/:id')
        .get(todoList.getUserServices);

    app.route('/services/:id_jasa')
        .delete(todoList.deleteService);
}

```

```

    app.route('/services')
        .put(todoList.editServiceReplaceImage);

    app.route('/services/:id_jasa')
        .put(todoList.editService);
}

```

13. Buatlah file index.js dan masukkan kode berikut:

```

var express = require('express'),
app = express(),
port = process.env.PORT || 3000,
bodyParser = require('body-parser');

app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());
app.use(express.static('public'));

var routes = require('./routes');
routes(app);

app.listen(port);
console.log('RESTful API Server dimulai pada port: ' + port);

```

14. Langkah berikutnya ialah menjalankan server. Untuk menjalankan server bisa melalui dua cara yaitu dengan sintaks:

PS D:\Project_NodeJS\jasa_online_api> **nodemon index.js**

Jika mau menjalankan live server, setiap membuat perubahan pada file otomatis akan merestart server gunakanlah sintaks ini untuk mendebug sintaks javascript.

Ketika menjalankan sintaks diatas terdapat error permasalahan izin akses seperti berikut:

```
+ CategoryInfo: PermissionDenied: (:) [Set-ExecutionPolicy], UnauthorizedAccessException
+ FullyQualifiedErrorId: System.UnauthorizedAccessException,Microsoft.PowerShell.Commands.SetExecutionPolicyCommand
```

Masukkan sintaks berikut di power shell:

PS D:\Project_NodeJS\jasa_online_api> **Set-ExecutionPolicy -Scope CurrentUser**

cmdlet Set-ExecutionPolicy at command pipeline position 1

Supply values for the following parameters:

ExecutionPolicy: **remotesigned**

Selain itu untuk menjalankan server bisa juga dilakukan dengan sintaks:

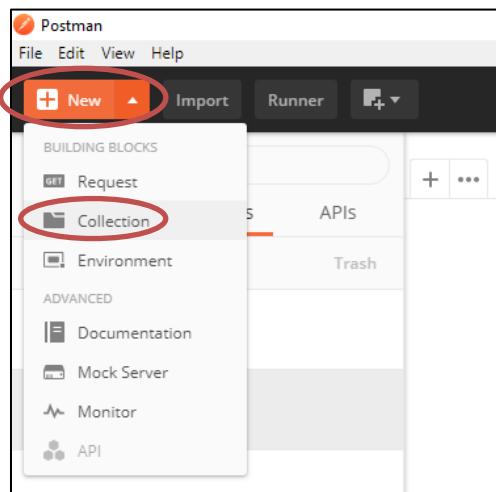
PS D:\Project_NodeJS\jasa_online_api> **node index.js**

Sintaks ini menjalankan file tetapi tidak merekam setiap perubahan pada file. Kita bisa menggunakan sintaks ini untuk menguji API pada aplikasi postman.

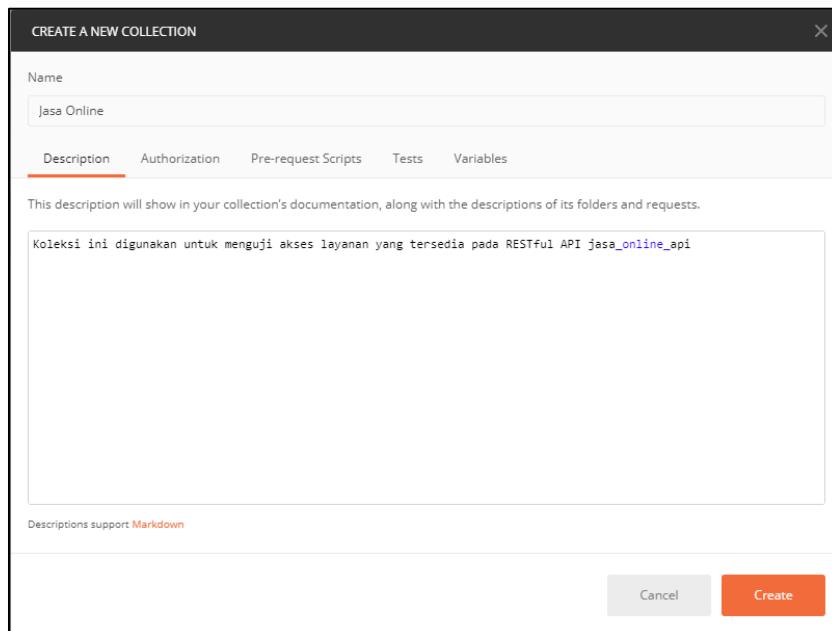
Menguji Akses Layanan API Pada Local Web Server

Menguji akses layanan user

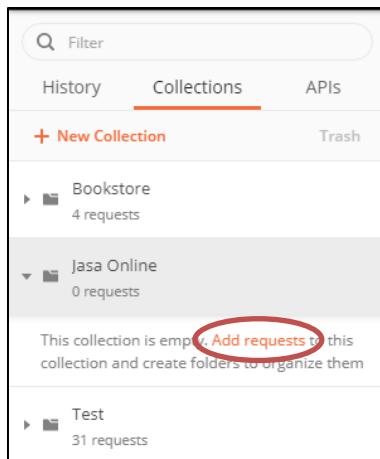
1. Pastikan server node js telah dijalankan dengan sintaks : *node index.js*
2. Buka aplikasi Postman. Buatlah koleksi baru pada dropdown + New dan pilih Collection.



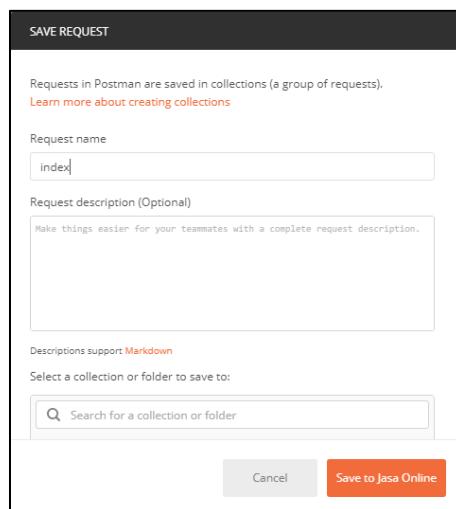
3. Masukkan nama koleksi 'Jasa Online' dan description seperti pada form isian berikut, klik Create untuk membuat koleksi.



4. Pada koleksi jasa online tambahkan request baru untuk mengakses index aplikasi. Klik Add requests untuk menambahkan request.



5. Masukkan nama request 'index' dan klik button Save to Jasa Online.



6. Pilih metode GET pada pilihan request di postman. Masukkan spesifikasi akses layanan seperti berikut:

Metode : GET

Url : <http://localhost:3000/>

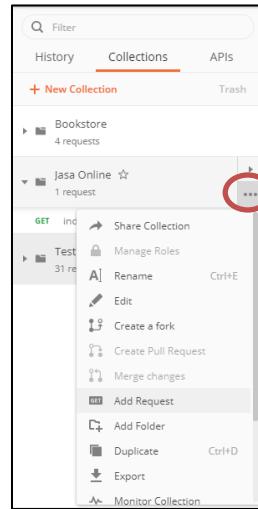
A screenshot of the Postman Request interface. On the left, a sidebar shows a list of methods: GET, POST, PUT, PATCH, DELETE, COPY, HEAD, OPTIONS, LINK, UNLINK, PURGE, LOCK, UNLOCK, PROPFIND, and VIEW. The 'GET' method is selected, indicated by a red arrow labeled 'Pilihan metode'. In the center, the 'Headers' tab is selected under the 'Body' tab. The URL field contains 'http://localhost:3000/'. Red arrows point from the 'Alamat Url' and 'Request Body' labels to their respective sections in the interface. At the bottom, there is a note: 'Hit Send to get a response'.

Hasil respon:

The screenshot shows the Postman interface with a collection named 'index'. A GET request is made to 'http://localhost:3000/index'. The response status is 200 OK, time is 132 ms, and size is 204 B. The response body is a JSON object:

```
1 "message": "Hello, pesan ini dari sisi server NodeJS Restful API!",  
2 "error": false
```

7. Tambahkan request baru didalam koleksi Jasa Online dengan nama getAllUser. Caranya adalah klik menu ... disebelah kanan bawah koleksi Jasa Online pilih Add Request.



Masukkan spesifikasi akses layanan seperti berikut:

Metode : GET

Url : <http://localhost/3000/users/>

Hasil respon:

```

1  {
2      "message": "Sukses menampilkan semua data user",
3      "error": false,
4      "data": [
5          {
6              "id": 1,
7              "nama": "Anton",
8              "tanggal_lahir": "1 Januari 2020",
9              "jenis_kelamin": "Pria",
10             "nomor_hp": "081211111111",
11             "alamat": "Jl. Damai",
12             "email": "anton@gmail.com",
13             "password": "tes12345"
14         },
15         {
16             "id": 2,
17             "nama": "Budi",
18             "tanggal_lahir": "2 Januari 2020",
19             "jenis_kelamin": "Pria",
20             "nomor_hp": "081222222222",
21             "alamat": "Jl. Sentosa",
22             "email": "budi@gmail.com",
23             "password": "tes12345"
24         }
25     ]
26 }

```

8. Tambahkan request baru didalam koleksi Jasa Online dengan nama getUserId. Masukkan spesifikasi akses layanan seperti berikut:

Metode : GET

Url : <http://localhost:3000/users/1>

Hasil respon:

```

1  {
2      "message": "User ditemukan",
3      "error": false,
4      "data": {
5          "id": 1,
6          "nama": "Anton",
7          "tanggal_lahir": "1 Januari 2020",
8          "jenis_kelamin": "Pria",
9          "nomor_hp": "081211111111",
10         "alamat": "Jl. Damai",
11         "email": "anton@gmail.com",
12         "password": "tes12345"
13     }
14 }

```

9. Tambahkan request baru didalam koleksi Jasa Online dengan nama userLoginEmailError.

Masukkan spesifikasi akses layanan seperti berikut:

Metode : GET

Url : <http://localhost:3000/login>

Params:

Key	Value
email	cut@gmail.com
password	tes12345

Hasil respon:

The screenshot shows a Postman collection named "userLoginEmailError". A GET request is made to `http://localhost:3000/login?email=cut@gmail.com&password=tes12345`. The "Params" tab is selected, showing two query parameters: "email" with value "cut@gmail.com" and "password" with value "tes12345". The "Body" tab shows a JSON response with the following content:

```

1  {
2   "message": "User tidak terdaftar!",
3   "error": true
4 }

```

10. Tambahkan request baru didalam koleksi Jasa Online dengan nama userLoginPasswordError.

Masukkan spesifikasi akses layanan seperti berikut:

Metode : GET

Url : <http://localhost/3000/login>

Params:

Key	Value
email	anton@gmail.com
password	tes123456

Hasil respon:

The screenshot shows a Postman collection named "userLoginPasswordError". A GET request is made to `http://localhost:3000/login?email=anton@gmail.com&password=tes123456`. The "Params" tab is selected, showing two query parameters: "email" with value "anton@gmail.com" and "password" with value "tes123456". The "Body" tab shows a JSON response with the following content:

```

1  {
2   "message": "Password user tidak cocok!",
3   "error": true
4 }

```

Lakukan sekali lagi, masukkan spesifikasi akses layanan seperti berikut:

Metode : GET

Url : <http://localhost:3000/login>

Params:

Key	Value
email	anton@gmail.com
password	tes12345

Hasil respon:

The screenshot shows a Postman collection named 'userLoginPasswordError'. A GET request is made to 'http://localhost:3000/login?email=anton@gmail.com&password=tes12345'. The 'Params' tab shows 'email' and 'password' as query parameters. The response status is 200 OK, and the JSON body is: { "message": "Password user tidak cocok!", "error": true }.

Kenapa hasilnya tidak seperti yang diharapkan (sukses login)? Berilah analisa dan tanggapan anda tentang request ini.

11. Tambahkan request baru didalam koleksi Jasa Online dengan nama addUser. Masukkan spesifikasi akses layanan seperti berikut:

Metode : POST

Url : <http://localhost:3000/users>

Body:

Key	Value
nama	Danu
tanggal_lahir	1 Januari 2000
jenis_kelamin	Pria
nomor_hp	085233333333
alamat	Jl. Bantan
email	danu@gmail.com
password	tes12345

Harus diperhatikan bahwa **key body** yang diinput dalam aplikasi postman **harus sama** dengan argumen pada **request body** yang terdapat dalam **fungsi registerUser** di controller.js.

Perhatikan potongan kode program berikut:

```
exports.registerUser = async function(req, res) {
```

```
    var nama = req.body.nama;
```

```

var tanggal_lahir = req.body.tanggal_lahir;
var jenis_kelamin = req.body.jenis_kelamin;
var nomor_hp = req.body.nomor_hp;
var alamat = req.body.alamat;
var email = req.body.email;
var plain_password = req.body.password;
var message;

if (!nama || !tanggal_lahir || !jenis_kelamin || !nomor_hp || !alamat || !email || !plain_password) {
    message = "Kehilangan beberapa parameter yang dibutuhkan!";
    return response.gagal(message, res);
}

await connection.query('SELECT * FROM tbl_user where email = ?',
[ email ],
function(error, rows, fields){
    if(rows.length > 0){
        message = "Email user telah terdaftar, silahkan pilih yang lain!";
        response.gagal(message, res);
    }else{
        var encrypt_password = encrypt.saltHashPassword(plain_password);
        connection.query('INSERT INTO tbl_user (nama, tanggal_lahir, jenis_kelamin, nomor_hp, alamat, email, password) values (?,?,?,?,?,?,?)',
        [ nama, tanggal_lahir, jenis_kelamin, nomor_hp, alamat, email, encrypt_password ],
        function (error, rows, fields){
            if(error){
                message = "Error ketika menambahkan user baru!";
                response.gagal(message, res);
            } else{
                message = "Berhasil menambahkan user baru";
                response.ok(message, res);
            }
        });
    }
});

```

Hasil respon:

The screenshot shows the Postman interface with a POST request to `http://localhost:3000/users`. The request body is a JSON object with fields: `nama` (Danu), `tanggal_lahir` (1 Januari 2000), `jenis_kelamin` (Pria), `nomor_hp` (085233333333), `alamat` (Jl. Bantan), `email` (danu@gmail.com), and `password` (tes12345). The response status is 200 OK, and the message is "Berhasil menambahkan user baru", with error set to false.

Key	Value
<code>nama</code>	Danu
<code>tanggal_lahir</code>	1 Januari 2000
<code>jenis_kelamin</code>	Pria
<code>nomor_hp</code>	085233333333
<code>alamat</code>	Jl. Bantan
<code>email</code>	danu@gmail.com
<code>password</code>	tes12345

```

1
2   "message": "Berhasil menambahkan user baru",
3   "error": false

```

Masukkan spesifikasi akses layanan seperti berikut:

Metode : POST

Url : <http://localhost:3000/users>

Body:

Key	Value
<code>nama</code>	Danu Gagal
<code>tanggal_lahir</code>	1 Januari 2000
<code>jenis_kelamin</code>	Pria
<code>nomor_hp</code>	085233333333
<code>alamat</code>	Jl. Bantan
<code>email</code>	danu@gmail.com
<code>password</code>	tes12345

Hasil respon:

12. Tambahkan request baru didalam koleksi Jasa Online dengan nama userLogin. Cobalah untuk melakukan login dengan user yang baru dibuat yaitu danu. Masukkan spesifikasi akses layanan seperti berikut:

Metode : GET

Url : <http://localhost:3000/login>

Params:

Key	Value
email	danu@gmail.com
password	tes12345

Hasil respon:

13. Tambahkan request baru didalam koleksi Jasa Online dengan nama updateUser. Masukkan spesifikasi akses layanan seperti berikut:

Metode : PUT

Url : <http://localhost:3000/users>

Body:

Key	Value
nama	Antoni Saputra
tanggal_lahir	1 Januari 2002
jenis_kelamin	Pria
nomor_hp	085211111111
alamat	Jl. Durian
email	antoni@gmail.com
password	tes12345
id	1

Harus diperhatikan bahwa **key body** yang diinput dalam aplikasi postman **harus sama** dengan argumen pada **request body** yang terdapat dalam **fungsi updateUser** di controller.js.

Perhatikan potongan kode program berikut:

```
exports.updateUser = async function(req, res) {

    var user_id = req.body.id;
    var nama = req.body.nama;
    var tanggal_lahir = req.body.tanggal_lahir;
    var jenis_kelamin = req.body.jenis_kelamin;
    var nomor_hp = req.body.nomor_hp;
    var alamat = req.body.alamat;
    var email = req.body.email;
    var plain_password = req.body.password;
    var message;

    if (!nama || !tanggal_lahir || !jenis_kelamin || !nomor_hp || !alamat || !email) {
        message = "Kehilangan beberapa parameter yang dibutuhkan!";
        return response.gagal(message, res);
    }

    var sql_query;
    var params;
    if(plain_password == ""){
        sql_query = 'UPDATE tbl_user SET nama = ?, tanggal_lahir = ?, jenis_kelamin = ?, nomor_hp = ?, alamat = ?, email = ? WHERE id = ?';
        params = [ nama, tanggal_lahir, jenis_kelamin, nomor_hp, alamat, email, user_id ]
    }else{
        sql_query = 'UPDATE tbl_user SET nama = ?, tanggal_lahir = ?, jenis_kelamin = ?, nomor_hp = ?, alamat = ?, email = ?, password = ? WHERE id = ?';
        var encrypt_password = encrypt.saltHashPassword(plain_password);
        params = [ nama, tanggal_lahir, jenis_kelamin, nomor_hp, alamat, email, encrypt_password, user_id ]
    }
}
```

```

await connection.query('SELECT * FROM tbl_user where email = ? and id <> ?', [email, user_id], function(error, rows, fields){
    if(rows.length > 0){
        message = "Email user telah terdaftar, silahkan pilih yang lain!";
        response.gagal(message, res);
    }else{
        connection.query(sql_query, params, function (error, rows, fields){
            if(error){
                message = "Error ketika mengubah data user!";
                response.gagal(message, res);
            } else{
                message = "Berhasil mengubah data user";
                response.ok(message, res);
            }
        });
    }
});

```

Hasil respon:

The screenshot shows the Postman interface with the following details:

- Method:** PUT
- URL:** http://localhost:3000/users
- Body (x-www-form-urlencoded):**

KEY	VALUE	DESCRIPTION
nama	Antoni Saputra	
tanggal_lahir	1 Januari 2002	
jenis_kelamin	Pria	
nomor_hp	085211111111	
alamat	Jl. Durian	
email	antonii@gmail.com	
password	tes12345	
id	1	
- Status:** 200 OK
- Time:** 167 ms
- Size:** 178 B

14. Jalankan kembali layanan getUserId, maka hasilnya data akan berubah perhatikan pada field password, setelah diubah data password menjadi password terenkripsi hash seperti berikut:

```

1 {
2   "message": "User ditemukan",
3   "error": false,
4   "data": {
5     "id": 1,
6     "name": "Antoni Saputra",
7     "tanggal_lahir": "1 Januari 2002",
8     "jenis_kelamin": "Pria",
9     "nomor_hp": "085211111111",
10    "alamat": "Jl. Durian",
11    "email": "anton@gmail.com",
12    "password": "huXO@o4PhErhYDqUvp5XL0jetJCLXMrMmqc7Fc8dVtkzdhpgy6m5RlkBFzsPVYJ821xkFYbuJxiRLxu0b1XRg==$9ea42afb3816ed87"
13  }
14 }

```

15. Cobalah untuk menjalankan layanan userLogin lagi dengan user yang baru diubah yaitu anton. Masukkan spesifikasi akses layanan seperti berikut:

Metode : GET

Url : <http://localhost:3000/login>

Params:

Key	Value
email	anton@gmail.com
password	tes12345

Hasil respon:

```

1 {
2   "message": "Login berhasil",
3   "error": false,
4   "data": {
5     "id": 1,
6     "name": "Antoni Saputra",
7     "tanggal_lahir": "1 Januari 2002",
8     "jenis_kelamin": "Pria",
9     "nomor_hp": "085211111111",
10    "alamat": "Jl. Durian",
11    "email": "anton@gmail.com",
12    "password": "huXO@o4PhErhYDqUvp5XL0jetJCLXMrMmqc7Fc8dVtkzdhpgy6m5RlkBFzsPVYJ821xkFYbuJxiRLxu0b1XRg==$9ea42afb3816ed87"
13  }
14 }

```

16. Tambahkan request baru didalam koleksi Jasa Online dengan nama updateUserEmailError.

Masukkan spesifikasi akses layanan seperti berikut:

Metode : PUT

Url : <http://localhost:3000/users>

Body:

Key	Value
nama	Antoni Saputra
tanggal_lahir	1 Januari 2002
jenis_kelamin	Pria
nomor_hp	085211111111
alamat	Jl. Durian
email	budi@gmail.com
password	tes12345
id	1

Hasil respon:

The screenshot shows the Postman application interface. A PUT request is being made to the URL `http://localhost:3000/users`. The request body contains the following key-value pairs, all of which have checkboxes checked:

KEY	VALUE
nama	Antoni Saputra
tanggal_lahir	1 Januari 2002
jenis_kelamin	Pria
nomor_hp	085211111111
alamat	Jl. Durian
email	budi@gmail.com
password	tes12345
id	1

The response status is 200 OK, the time taken is 18 ms, and the size is 203 B. The response body is displayed in JSON format:

```
1 "message": "Email user telah terdaftar, silahkan pilih yang lain!",  
2 "error": true
```

17. Tambahkan request baru didalam koleksi Jasa Online dengan nama deleteUser. Masukkan spesifikasi akses layanan seperti berikut untuk menghapus user dengan id = 2:

Metode : DELETE

Url : <http://localhost:3000/users/2>

Hasil respon:

The screenshot shows the Postman interface with a successful DELETE request to `http://localhost:3000/users/2`. The response status is 200 OK, and the response body is:

```

1 [
2   "message": "Berhasil menghapus user",
3   "error": false
4 ]

```

Jalankan sekali lagi layanan menghapus user dengan id = 2. Hasilnya adalah seperti berikut:

The screenshot shows the Postman interface with an unsuccessful DELETE request to `http://localhost:3000/users/2`. The response status is 200 OK, and the response body is:

```

1 [
2   "message": "User tidak ditemukan!",
3   "error": true
4 ]

```

Menguji akses layanan jasa

- Pastikan server node.js telah dijalankan dengan sintaks : `node index.js`
- Tambahkan request baru didalam koleksi Jasa Online dengan nama addService. Masukkan spesifikasi akses layanan seperti berikut:

Metode : POST

Url : <http://localhost:3000/services>

Body:

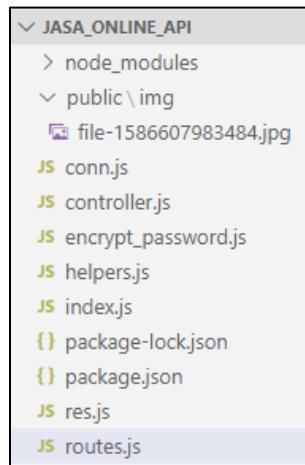
Key	Value
id_user	1
nama_jasa	Service Diesel Antoni
deskripsi_singkat	Service Diesel Berkualitas
uraian_deskripsi	Contoh Uraian Deskripsi
rating	0

gambar	test.jpg
file	diesel.jpg

Hasil respon:

The screenshot shows a Postman interface with a POST request to 'addService' at 'http://localhost:3000/services'. The 'Body' tab is selected, showing form-data fields: id_user (1), nama_jasa (Service Diesel Antoni), deskripsi_singkat (Service Diesel Berkualitas), uraian_deskripsi (Contoh Uraian Deskripsi), rating (0), gambar (test.jpg), and file (diesel.jpg). The response status is 200 OK.

Hasil upload image dalam folder public/img dapat dilihat pada struktur file project di Visual Studio Code, adapun nama file yang degenerate otomatis adalah file-1586607983484.jpg:



Cobalah untuk mengupload image yang berukuran lebih besar dari 1 Mb. Masukkan spesifikasi akses layanan seperti berikut:

Metode : POST

Url : <http://localhost:3000/services>

Body:

Key	Value
id_user	1
nama_jasa	Service AC Antoni

deskripsi_singkat	Service AC Berkualitas
uraian_deskripsi	Contoh Uraian Deskripsi
rating	0
gambar	test.jpg
file	diesel.jpg

Pada bagian key file uploadlah gambar yang berukuran lebih besar dari 1 Mb. Contoh file IMG_4774cp.jpg dengan ukuran 3.62Mb.

Hasil respon:

KEY	VALUE	DESCRIPTION
id_user	1	
nama_jasa	Service AC Antoni	
deskripsi_singkat	Service AC Berkualitas	
uraian_deskripsi	Contoh Uraian Deskripsi	
rating	0	
gambar	test.jpg	
file	IMG_4774cp.jpg	

Status: 200 OK Time: 149 ms Size: 208 B Save Response

Selanjutnya cobalah mengupload file yang bukan berekstensi image pada key file. Contoh file colors_green_teal.xml dengan ukuran file 1 Kb.

Hasil respon:

KEY	VALUE	DESCRIPTION
id_user	1	
nama_jasa	Service AC Antoni	
deskripsi_singkat	Service AC Berkualitas	
uraian_deskripsi	Contoh Uraian Deskripsi	
rating	0	
gambar	test.jpg	
file	colors_green_teal.xml	

Status: 200 OK Time: 44 ms Size: 179 B Save Response

Masukkan spesifikasi akses layanan seperti berikut, dimana id_user yang diinput tidak terdapat di tabel tbl_user:

Metode : POST

Url : <http://localhost:3000/services>

Body:

Key	Value
id_user	2
nama_jasa	Service AC Antoni
deskripsi_singkat	Service AC Berkualitas
uraian_deskripsi	Contoh Uraian Deskripsi
rating	0
gambar	test.jpg
file	ac.jpg

Hasil respon:

The screenshot shows the Postman application interface. At the top, there are two tabs: 'POST addService' and 'GET getAllUser'. Below the tabs, the URL 'http://localhost:3000/services' is entered. On the right side, there are 'Send' and 'Save' buttons. The main area has tabs for 'Params', 'Authorization', 'Headers (9)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Body' tab is selected and set to 'form-data'. It contains seven key-value pairs: 'id_user' (value: 2), 'nama_jasa' (value: Service AC Antoni), 'deskripsi_singkat' (value: Service AC Berkualitas), 'uraian_deskripsi' (value: Contoh Uraian Deskripsi), 'rating' (value: 0), 'gambar' (value: test.jpg), and 'file' (value: ac.jpg). Below the body table, there are 'Pretty', 'Raw', 'Preview', 'Visualize', and 'Text' buttons. The response section shows a status of '200 OK', a time of '125 ms', and a size of '180 B'. The response body is a JSON object: { "message": "Error ketika menambahkan jasa!", "error": true }.

Ubahlah id_user menjadi 3 dan nama_jasa Service AC Danu dan jalankan kembali request diatas. Hasilnya akan menjadi seperti berikut:

The screenshot shows the Postman interface with a POST request to 'addService' at 'http://localhost:3000/services'. The 'Body' tab is selected, showing a form-data structure with the following fields:

Key	Value	Description
id_user	3	
nama_jasa	Service AC Danu	
deskripsi_singkat	Service AC Berkualitas	
uraian_deskripsi	Contoh Uraian Deskripsi	
rating	0	
gambar	test.jpg	
file	ac.jpg	

The response status is 200 OK with a message: "{'message': 'Berhasil menambahkan jasa', 'error': false}".

3. Tambahkan request baru didalam koleksi Jasa Online dengan nama getServices. Masukkan spesifikasi akses layanan seperti berikut:

Metode : GET

Url : <http://localhost:3000/services>

Hasil respon:

The screenshot shows the Postman interface with a GET request to 'getServices' at 'http://localhost:3000/services'. The 'Body' tab is selected, showing a JSON response with the following data:

```

1  {
2   "message": "Sukses menampilkan semua data jasa",
3   "error": false,
4   "data": [
5     {
6       "id_jasa": 3,
7       "nama_penyedia": "Danu",
8       "nomor_hp": "085233333333",
9       "nama_jasa": "Service AC Danu",
10      "deskripsi_singkat": "Service AC Berkualitas",
11      "uraian_deskripsi": "Contoh Uraian Deskripsi",
12      "rating": 0,
13      "gambar": "file-1586607983484.jpg"
14    },
15    {
16      "id_jasa": 1,
17      "nama_penyedia": "Antoni Saputra",
18      "nomor_hp": "085211111111",
19      "nama_jasa": "Service Diesel Antoni",
20      "deskripsi_singkat": "Service Diesel Berkualitas",
21      "uraian_deskripsi": "Contoh Uraian Deskripsi",
22      "rating": 0,
23      "gambar": "file-1586607983484.jpg"
24    }
  ]
}

```

The response status is 200 OK with a message: "{'message': 'Sukses menampilkan semua data jasa', 'error': false}".

4. Tambahkan request baru didalam koleksi Jasa Online dengan nama getUserServices. Masukkan spesifikasi akses layanan seperti berikut:

Metode : GET

Url : <http://localhost:3000/services/1>

Hasil respon:

```

1  {
2      "message": "Sukses menampilkan semua data jasa pengguna",
3      "error": false,
4      "data": [
5          {
6              "id_jasa": 1,
7              "nama_penyedia": "Antoni Saputra",
8              "nomor_hp": "085211111111",
9              "nama_jasa": "Service Diesel Antoni",
10             "deskripsi_singkat": "Service Diesel Berkualitas",
11             "uraian_deskripsi": "Contoh Uraian Deskripsi",
12             "rating": 0,
13             "gambar": "file-1586607983484.jpg"
14         }
15     ]
16 }

```

5. Tambahkan request baru didalam koleksi Jasa Online dengan nama editService. Masukkan spesifikasi akses layanan seperti berikut:

Metode : PUT

Url : <http://localhost:3000/services>

Body:

Key	Value
id_user	1
nama_jasa	Service Diesel Antoni
deskripsi_singkat	Service Diesel Berkualitas
uraian_deskripsi	Contoh Uraian Deskripsi
gambar	motor.jpg
file	motor.jpg

Hasil respon:

```

1  {"message": "Berhasil mengubah data jasa", "error": false}

```

Untuk melihat perubahan jalankan request getServices. Hasilnya seperti berikut:

```

1   "message": "Sukses menampilkan semua data jasa",
2   "error": false,
3   "data": [
4     {
5       "id_jasa": 3,
6       "nama_penyedia": "Danu",
7       "nomor_hp": "085233333333",
8       "nama_jasa": "Service AC Danu",
9       "deskripsi_singkat": "Service AC Berkualitas",
10      "uraian_deskripsi": "Contoh Uraian Deskripsi",
11      "rating": 0,
12      "gambar": "file-1586609394527.jpg"
13    },
14    {
15      "id_jasa": 1,
16      "nama_penyedia": "Antoni Saputra",
17      "nomor_hp": "085211111111",
18      "nama_jasa": "Service Motor Antoni",
19      "deskripsi_singkat": "Service Motor Berkualitas",
20      "uraian_deskripsi": "Contoh Uraian Deskripsi",
21      "rating": 0,
22      "gambar": "file-1586610467603.jpg"
23    }
24  ]
25 ]
26

```

6. Tambahkan request baru didalam koleksi Jasa Online dengan nama editServiceId. Masukkan spesifikasi akses layanan seperti berikut:

Metode : PUT

Url : <http://localhost:3000/services/1>

Body:

Key	Value
nama_jasa	Antoni Service Motor Bergaransi
deskripsi_singkat	Terpercaya, handal dan awet
uraian_deskripsi	Service motor bergaransi 1 bulan

Hasil respon:

```

1   [{"message": "Berhasil mengubah data jasa", "error": false}]

```

Untuk melihat perubahan jalankan request getServices. Hasilnya seperti berikut:

```

1   {
2     "message": "Sukses menampilkan semua data jasa",
3     "error": false,
4     "data": [
5       {
6         "id_jasa": 3,
7         "nama_penyedia": "Danu",
8         "nomor_hp": "085233333333",
9         "nama_jasa": "Service AC Danu",
10        "deskripsi_singkat": "Service AC Berkualitas",
11        "uraian_deskripsi": "Contoh Uraian Deskripsi",
12        "rating": 0,
13        "gambar": "file-1586609394527.jpg"
14      },
15      {
16        "id_jasa": 1,
17        "nama_penyedia": "Antoni Saputra",
18        "nomor_hp": "085211111111",
19        "nama_jasa": "Antoni Service Motor Bergaransi",
20        "deskripsi_singkat": "Terpercaya, handal dan awet",
21        "uraian_deskripsi": "Service motor bergaransi 1 bulan",
22        "rating": 0,
23        "gambar": "file-1586610467603.jpg"
24      }
25    ]
26  }

```

7. Tambahkan request baru didalam koleksi Jasa Online dengan nama deleteService. Masukkan spesifikasi akses layanan seperti berikut:

Metode : DELETE

Url : <http://localhost/3000/services/3>

Hasil respon:

```

1   {"message": "Berhasil menghapus jasa", "error": false}

```

Coba jalankan sekali lagi request diatas, hasilnya seperti berikut:

```

1   {"message": "Jasa tidak ditemukan!", "error": true}

```

Tugas Praktikum

- Tuliskan pemahaman anda mengenai contoh koding program praktikum 5 dan screen shoot hasilnya.
- Buatlah laporan resmi praktikum 5.

Daftar Referensi

<https://pratamawijaya.com/programming/android-recyclerview-kotlin/>

<https://www.androdocs.com/tutorials/implementing-navigation-drawer-in-android-app-using-kotlin.html>

https://en.wikipedia.org/wiki/Application_programming_interface

<https://medium.com/@kiddy.xyz/tutorial-restful-api-node-js-express-mysql-part-1-527868bfb9d1>

<https://medium.com/@kiddy.xyz/tutorial-restful-api-node-js-express-mysql-part-2-eff96ebb4b7f>

<http://mfikri.com/en/blog/nodejs-restful-api-mysql>

<https://stackabuse.com/handling-file-uploads-in-node-js-with-expres-and-multer/>

** ☺ Selamat Mengerjakan ☺ ***