

PRAKTIKUM APLIKASI TEMAN SAYA

Pada praktikum ini akan dipelajari pembuatan aplikasi yang dapat menyimpan data teman serta menampilkannya dalam tampilan list. Maka itu, perlu dipelajari tentang penyimpanan data (database) lokal, yaitu Room. Pada praktikum ini juga akan dibahas komponen View yaitu FloatingActionButton dan RecyclerView lagi. Serta akan dibahas tentang penggunaan Fragment. Fragment adalah class yang memiliki fungsi sama seperti Activity, akan tetapi Fragment tidak bisa berdiri sendiri. Sebuah Fragment harus bergantung pada sebuah Activity. Dalam analogi sederhana, Fragment bisa dianggap sebuah child dan Activity adalah parent, sebagaimana ViewGroup dan View.

1. Membuat Layout Halaman Teman Saya

Sebelum membuat layout halaman teman saya, lakukan pembuatan project baru dengan nama project **MyFriendApp** atau yang lain silahkan. Halaman teman saya adalah halaman yang menampilkan informasi teman dalam tampilan list. Oleh karena itu, dibutuhkan komponen view RecyclerView. RecyclerView adalah komponen View yang tidak disediakan secara default. Maka, harus dilakukan penambahan library RecyclerView pada project MyFriendApp.

1.1 Menambahkan Library RecyclerView

Gradle Module App, yaitu file gradle yang salah satu kegunaannya adalah untuk konfigurasi penggunaan library pada project. Pada bagian `dependencies` pada file gradle tersebut, dapat ditambahkan library-library yang dibutuhkan dalam project. Library RecyclerView ditambahkan dengan menggunakan script berikut:

```
implementation 'androidx.recyclerview:recyclerview:1.0.0'
```

Setelah melakukan penambahan library, klik **Sync Now** yang muncul pada bagian kanan atas editor.

. A project sync may be necessary for the IDE to work properly. [Sync Now](#)

Gambar Pesan Yang Muncul Setelah Dilakukan Perubahan Pada File Gradle

1.2 Membuat Layout Halaman Teman Saya

Setiap layout pada project diletakkan pada direktori app/res/layout. Oleh karena itu, penambahan layout baru juga dilakukan pada direktori tersebut dengan:

1. Melakukan klik kanan pada direktori layout
2. Pilih New, XML, Layout XML File
3. Beri nama my_friends_fragment.xml
4. Tambahkan RecyclerView di dalam LinearLayout

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

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

</RelativeLayout>
```

1.3 Membuat Layout Halaman Tambah Teman

my_friends_add_fragment.xml

```
<?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:background="#FFFFFF">
```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tambah Teman"
    android:textSize="19sp"
    android:layout_margin="16dp"
    android:textColor="#000000"
    android:layout_gravity="center_horizontal"/>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    android:orientation="vertical">

    <EditText
        android:id="@+id/edtName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Nama"
        android:textColorHint="#CCCCCC"
        android:textColor="#000000"
        android:inputType="text"
        android:imeOptions="actionDone"/>

    <Spinner
        android:layout_marginTop="16dp"
        android:id="@+id/spinnerGender"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="#000000"
        android:entries="@array/gender_list"/>

    <EditText
        android:id="@+id/edtEmail"
        android:layout_marginTop="16dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:textColorHint="#CCCCCC"
        android:textColor="#000000"
        android:inputType="textEmailAddress"
        android:imeOptions="actionNext"/>

    <EditText
        android:id="@+id/edtTelp"
        android:layout_marginTop="16dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Telp. "
        android:textColorHint="#CCCCCC"
        android:textColor="#000000"
        android:inputType="number"
        android:imeOptions="actionNext"/>

    <EditText
        android:id="@+id/edtAddress"
        android:layout_marginTop="16dp"
        android:layout_width="match_parent"
        android:layout_height="120dp"
        android:hint="Alamat"
        android:textColorHint="#CCCCCC"
        android:textColor="#000000"

```

```

        android:inputType="textMultiLine"
        android:gravity="top"
        android:imeOptions="actionDone"/>

        <Button
            android:id="@+id/btnSave"
            android:layout_marginTop="24dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Simpan"/>

    </LinearLayout>

</LinearLayout>

```

The image shows a mobile application interface for adding a friend. It features a title 'Tambah Teman' at the top. Below the title are five input fields: 'Nama', 'Pilih kelamin' (a dropdown menu), 'Email', 'Telp.', and 'Alamat'. At the bottom of the form is a grey button labeled 'SIMPAN'.

Gambar 4.2. Layout Tambah Teman

1.4 Membuat Layout Item List Teman

Jika ingin membuat tampilan list, diharuskan membuat layout dari item informasi yang akan ditampilkan dalam bentuk list. Dalam hal ini, `my_friends_item.xml`

```

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

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:paddingLeft="16dp"
        android:paddingRight="16dp"
        android:paddingTop="8dp"
        android:paddingBottom="8dp">

        <TextView
            android:id="@+id/txtFriendName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Nama"
            android:textSize="18sp"/>

        <TextView
            android:id="@+id/txtFriendEmail"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Email"
            android:textSize="16sp"/>

        <TextView
            android:layout_marginTop="4dp"
            android:id="@+id/txtFriendTelp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Telp"
            android:textSize="20sp"/>
    </LinearLayout>

    <View
        android:layout_width="match_parent"
        android:layout_height="0.5dp"
        android:background="#000000"/>

</LinearLayout>

```

1.5 Menambahkan Floating Action Button pada my_friends_fragment.xml

Floating Action Button (FAB) adalah komponen View dari Material Design berupa Button Floating (mengambang) berbentuk lingkaran. Untuk dapat menggunakan FAB, harus menambahkan dua script pada file Gradle Module:

1. `vectorDrawables.useSupportLibrary = true` pada bagian `defaultConfig`
2. implementation `'com.google.android.material:material:1.1.0-alpha03'` pada `dependencies`

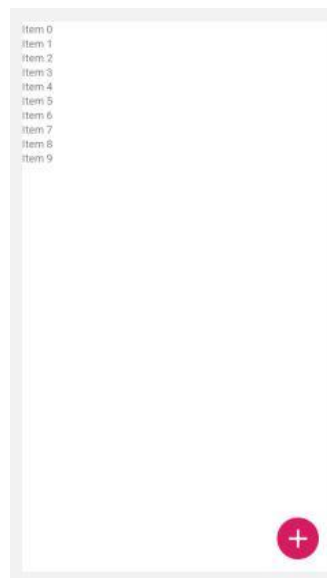
Kemudian, tambahkan FAB pada layout `my_friends_fragment.xml`

```
<?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:background="#FFFFFF"
    xmlns:app="http://schemas.android.com/apk/res-auto">

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

    <com.google.android.material.floatingactionbutton.FloatingActionButt on
        android:id="@+id/fabAddFriend"
        style="@style/Widget.Design.FloatingActionButton"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_margin="16dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/ic_add_plus_button" />

</RelativeLayout>
```



Gambar 4.3. `my_friends_fragment.xml`

1.6 Menambahkan FrameLayout Pada activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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">

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

</androidx.constraintlayout.widget.ConstraintLayout>
```

2. Membuat Fragment pada Aplikasi Teman Saya

Fragment adalah class yang pada umumnya digunakan untuk kebutuhan menampilkan lebih dari satu fitur pada satu halaman. Contoh pada aplikasi GOJEK



Gambar Halaman Aplikasi GOJEK

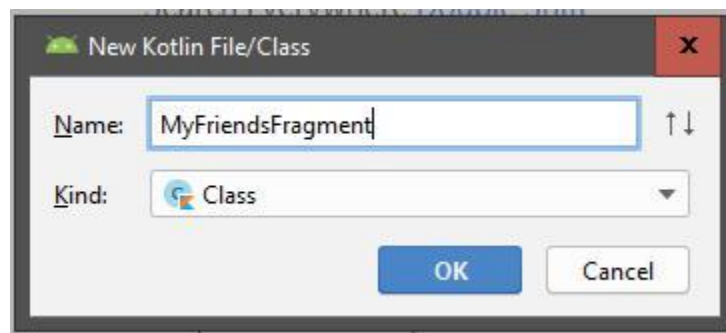
Pada aplikasi GOJEK di atas, tampilan Fragment seperti pada GoRide, GoCar, GoFood, dan GoShop yang ditampilkan dalam satu Activity. Pada project aplikasi MyFriendApp, akan

digunakan penerapan yang serupa, yaitu menggunakan dua Fragment (MyFriendsFragment.kt dan MyFriendsAddFragment.kt) pada satu Activity (MainActivity.kt)

2.1 Menambahkan Fragment MyFriendsFragment.kt

Berikut adalah langkah-langkah untuk menambahkan file .kt pada project:

1. Klik kanan pada direktori app/java/<package project>
2. Pilih New, Kotlin File/Class
3. Beri nama MyFriendsFragment dan pilih Kind: Class, OK



Gambar 4.5. Menambahkan File Kotlin Pada Project

MyFriendsFragment.kt

```
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import kotlinx.android.synthetic.*

class MyFriendsFragment : Fragment() {

    companion object {
        fun newInstance(): MyFriendsFragment {
            return MyFriendsFragment()
        }
    }

    override fun onCreateView(inflater: LayoutInflater, container:
    ViewGroup?, savedInstanceState: Bundle?): View? {
        return inflater.inflate(R.layout.my_friends_fragment,
        container, false)
    }
}
```



```

        override fun onCreateView(view: View, savedInstanceState: Bundle?) {
            super.onCreateView(view, savedInstanceState)
            initView()
        }

        private fun initView() {

        }

        override fun onDestroy() {
            super.onDestroy()
            this.clearFindViewByIdCache()
        }
    }

```

2.2 Menambahkan Fragment MyFriendsAddFragment.kt

```

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import kotlinx.android.synthetic.*

class MyFriendsAddFragment : Fragment() {

    companion object {
        fun newInstance(): MyFriendsAddFragment {
            return MyFriendsAddFragment()
        }
    }

    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View? {
        return inflater.inflate(R.layout.my_friends_add_fragment, container, false)
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        initView()
    }

    private fun initView() {

    }

    override fun onDestroy() {
        super.onDestroy()
        this.clearFindViewByIdCache()
    }
}

```

2.3 Menambahkan Fungsi gantiFragment() pada Aplikasi MyFriendApp

Fungsi gantiFragment() adalah fungsi yang digunakan untuk mengganti Fragment yang tampil pada MainActivity.kt. Berikut adalah kode MainActivity.kt setelah ditambahkan fungsi gantiFragment()

```
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.fragment.app.Fragment import
androidx.fragment.app.FragmentManager

class MainActivity : AppCompatActivity() {

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

    private fun gantiFragment(
        fragmentManager: FragmentManager,
        fragment: Fragment, frameId: Int
    ) {
        val transaction = fragmentManager.beginTransaction()
        transaction.replace(frameId, fragment)

        transaction.commit()
    }

    fun tampilMyFriendsFragment() {
        gantiFragment(supportFragmentManager,
        MyFriendsFragment.newInstance(), R.id.contentFrame)
    }

    fun tampilMyFriendsAddFragment() {
        gantiFragment(supportFragmentManager,
        MyFriendsAddFragment.newInstance(), R.id.contentFrame)
    }

}
```

2.4 Menjalankan fungsi tampilMyFriendsAddFragment pada MyFriendsFragment

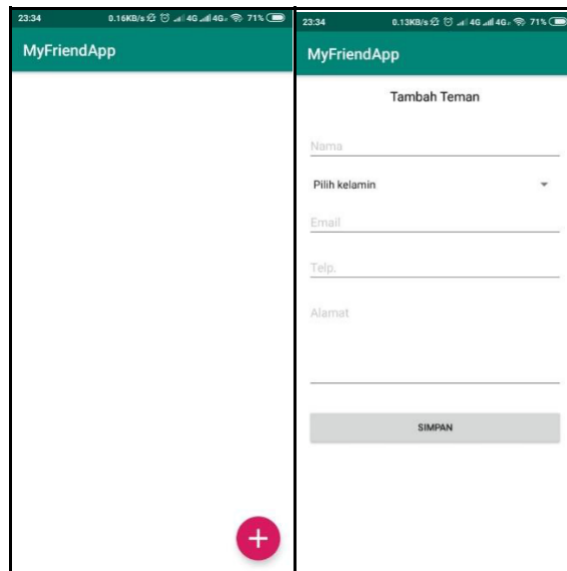
Fungsi tampilMyFriendsAddFragment pada MyFriendsFragment dijalankan ketika FAB pada MyFriendsFragment diklik. Ditambahkan pada fungsi initView():

```
fabAddFriend.setOnClickListener { (activity as
MainActivity).tampilMyFriendsAddFragment() }
```

2.5 Menjalankan fungsi tampilMyFriendsFragment pada MyFriendsAddFragment

Fungsi tampilMyFriendsFragment pada MyFriendsAddFragment dijalankan ketika Button Save pada MyFriendsAddFragment diklik. Ditambahkan pada fungsi initView():

```
btnSave.setOnClickListener { (activity as  
MainActivity).tampilMyFriendsFragment() }
```



Gambar 4.6 Aplikasi MyFriendApp

3. Menampilkan Data List Teman Pada RecyclerView

Untuk menampilkan data list pada RecyclerView, setidaknya ada 3 hal yang perlu dilakukan membuat data class untuk menampung data list item, membuat class RecyclerView Adapter untuk menampilkan data per item, dan menampilkan data list pada RecyclerView.

3.1 Membuat Data Class

Data Class adalah sebuah class yang digunakan untuk menampung data. Pada aplikasi MyFriendApp, data class digunakan untuk menampung data diri teman yang kemudian akan ditampilkan dalam tampilan list pada RecyclerView. Berikut adalah data class yang digunakan untuk menampung data diri teman tersebut:

MyFriend.kt

```
data class MyFriend (
    val nama : String,
    val kelamin : String,
    val email : String,
    val telp: String,
    val alamat: String
)
```

3.2 Membuat RecyclerViewAdapter

Untuk menampilkan data dalam tampilan list pada RecyclerView, dibutuhkan sebuah class RecyclerView Adapter. RecyclerView Adapter yang akan dibuat juga menggunakan Kotlin Android Extension sebagai pengganti findViewById, yaitu dengan mengimplementasi class LayoutContainer. Akan tetapi, fitur tersebut belum aktif secara default sehingga harus diaktifkan terlebih dahulu dengan menambahkan script di bawah ini di atas dependencies pada build.gradle (module: app)

```
androidExtensions {
    experimental = true
}
```

MyFriendAdapter.kt

```
import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView import
kotlinx.android.extensions.LayoutContainer import
kotlinx.android.synthetic.main.my_friends_item.*

class MyFriendAdapter(private val context: Context, private val
items: ArrayList<MyFriend>) :
    RecyclerView.Adapter<MyFriendAdapter.ViewHolder>() {

    override fun onCreateViewHolder(parent: ViewGroup, viewType:
    Int) =
        ViewHolder(LayoutInflater.from(context).inflate(R.layout.my_friends_
        item, parent, false))
```

```

        override fun onBindViewHolder(holder: ViewHolder, position: Int)
        {
            holder.bindItem(items.get(position))
        }

        override fun getItemCount(): Int = items.size

        class ViewHolder(override val containerView: View) :
            RecyclerView.ViewHolder(containerView), LayoutContainer {
            fun bindItem(item: MyFriend) {
                txtFriendName.text = item.nama
                txtFriendEmail.text = item.email
                txtFriendTelp.text = item.telp
            }
        }
    }
}

```

3.3 Menampilkan List Data Pada RecyclerView

Setelah membuat data class dan RecyclerView Adapter untuk menampung dan menampilkan datanya, selanjutnya harus dilakukan simulasi mengisi data ke dalam variable MutableList. Kemudian data yang ada pada variable tersebut di masukkan ke MyFriendAdapter.kt. Lalu mengatur RecyclerView untuk menggunakan MyFriendAdapter.kt sebagai adapternya. Berikut adalah langkah-langkahnya:

1. Menambahkan public variable MutableList yang menampung data dengan data class MyFriend pada MyFriendsFragment.kt

```
lateinit var listTeman : MutableList<MyFriend>
```

2. Mengisi data simulasi ke dalam listTeman

```

private fun simulasiDataTeman() {
    listTeman = ArrayList()

    listTeman.add(MyFriend("Muhammad", "Laki-laki",
        "ade@gmail.com", "085719004268", "Bandung"))
    listTeman.add(MyFriend("Al Harits", "Laki-laki",
        "rifaldi@gmail.com", "081213416171", "Bandung"))
}

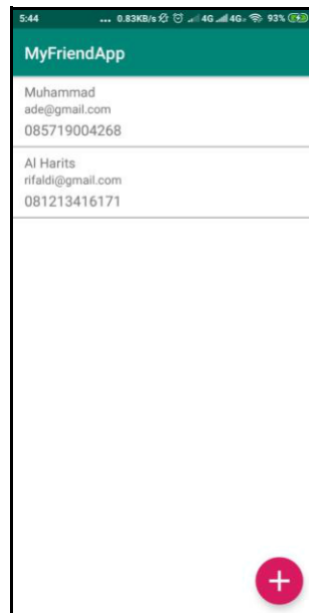
```

3. Menampilkan data listTeman ke dalam RecyclerView

```
private fun tampilTeman() {  
    listMyFriends.layoutManager = LinearLayoutManager(activity)  
    listMyFriends.adapter = MyFriendAdapter(activity!!, listTeman)  
}
```

4. Menjalankan fungsi dataTeman() dan tampilTeman()

```
private fun initView() {  
    fabAddFriend.setOnClickListener { (activity as  
MainActivity).tampilMyFriendsAddFragment() }  
  
    simulasiDataTeman()  
    tampilTeman()  
}
```



Gambar 4.7 Menampilkan Data Pada RecyclerView

4. Menambahkan Data ke Dalam Penyimpanan Data (Database) Lokal

4.1 Menambahkan Library Room

Penyimpanan data (database) lokal yang digunakan pada Aplikasi MyFriendApp adalah Room. Hal pertama yang perlu dilakukan untuk menggunakan Room pada project adalah menambahkan plugin `'kotlin-kapt'` pada `build.gradle (module: app)`. Kemudian, menambahkan library Room di dalam `dependencies`

```
apply plugin: 'kotlin-kapt'

....

implementation 'android.arch.persistence.room:runtime:1.1.1'
kapt 'android.arch.persistence.room:compiler:1.1.1'
```

4.2. Mengubah Data Class MyFriend.kt Menjadi Table (Entity)

Table pada database Room direpresentasikan dengan data class yang ditambahkan annotation `(@)` Entity. Selain itu, ditambahkan variable data sebagai primary key.

MyFriend.kt

```
@Entity
data class MyFriend (
    @PrimaryKey(autoGenerate = true)
    val temanId: Int? = null,
    val nama : String,
    val kelamin : String,
    val email : String,
    val telp: String,
    val alamat: String
)
```

4.3 Membuat Data Access Object (DAO) Untuk Menambah dan Mengambil Data Teman

Sebuah database berbasis table memerlukan sekumpulan perintah khusus untuk mengakses data yang disebut dengan SQL (Structured Query Language). Pada database Room, perlu dibuat sebuah interface berisi SQL yang disebut DAO. Sebuah DAO dapat diisi dengan perintah SQL untuk menambahkan (insert), mengubah (update), menghapus (delete), dan mengambil (get) data.

MyFriendDao.kt

```
@Dao
interface MyFriendDao {

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    fun tambahTeman(friend: MyFriend)

    @Query("SELECT * FROM MyFriend")
    fun ambilSemuaTeman(): LiveData<List<MyFriend>>

}
```

4.5 Membuat Class RoomDatabase

Hal berikutnya yang perlu dilakukan adalah membuat class RoomDatabase yang berperan sebagai class yang memberikan koneksi antara interface dengan database.

```
@Database(entities = [MyFriend::class], version = 1)
abstract class AppDatabase: RoomDatabase() {

    abstract fun myFriendDao(): MyFriendDao

    companion object {
        var INSTANCE: AppDatabase? = null

        fun getAppDataBase(context: Context): AppDatabase? {
            if (INSTANCE == null) {
                synchronized(AppDatabase::class) {
                    INSTANCE =
                        Room.databaseBuilder(context.applicationContext,
                            AppDatabase::class.java, "MyFriendAppDB").build()
                }
            }
            return INSTANCE
        }

        fun destroyDataBase() {
            INSTANCE = null
        }
    }
}
```


4.6 Membuat Fungsi tambahDataTeman(teman: MyFriend)

Perintah menambah data teman dijalankan pada Fragment MyFriendsAddFragment.kt ketika Button btnSave ditekan dan fungsi validasiInput() memasuki kondisi validasi sukses. Untuk menjalankan perintah tambah (insert) teman yang ada pada MyFriendDao.kt, dibutuhkan penggunaan coroutines. Maka itu perlu ditambahkan library coroutines pada build.gradle. Kemudian, Sync.

```
implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-core:1.1.1'
```

Sebelum menjalankan fungsi tambahTeman, perlu dibuat variable untuk menampung input pada fungsi validasiInput.

```
else -> {  
  
    val teman = MyFriend(nama = namaInput, kelamin = genderInput,  
email = emailInput, telp = telpInput, alamat = alamatInput)  
    tambahDataTeman(teman)  
  
}
```

Berikut adalah fungsi tambahTeman

```
private fun tambahDataTeman(teman: MyFriend) : Job {  
  
    return GlobalScope.launch {  
        myFriendDao?.tambahTeman(teman)  
        (activity as MainActivity).tampilMyFriendsFragment()  
    }  
  
}
```

MyFriendsAddFragment.kt

```
import android.os.Bundle  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.AdapterView  
import android.widget.Toast
```

```

import androidx.fragment.app.Fragment
import kotlinx.android.synthetic.*
import kotlinx.android.synthetic.main.my_friends_add_fragment.*
import kotlinx.coroutines.GlobalScope import
kotlinx.coroutines.Job
import kotlinx.coroutines.launch

class MyFriendsAddFragment : Fragment() {

    companion object {
        fun newInstance(): MyFriendsAddFragment {
            return MyFriendsAddFragment()
        }
    }

    private var namaInput : String = ""
    private var emailInput : String = ""
    private var telpInput : String = ""
    private var alamatInput : String = ""
    private var genderInput : String = ""

    private var db: AppDatabase? = null
    private var myFriendDao: MyFriendDao? = null

    override fun onCreateView(inflater: LayoutInflater, container:
    ViewGroup?, savedInstanceState: Bundle?): View? {
        return inflater.inflate(R.layout.my_friends_add_fragment,
        container, false)
    }

    override fun onViewCreated(view: View, savedInstanceState:
    Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        initLocalDB()
        initView()
    }

    private fun initLocalDB() {
        db = AppDatabase.getAppDataBase(activity!!)
        myFriendDao = db?.myFriendDao()
    }

    private fun initView() {
        btnSave.setOnClickListener { validasiInput() }

        setDataSpinnerGener()
    }

    private fun setDataSpinnerGener() {
        val adapter = ArrayAdapter.createFromResource(activity!!,
        R.array.gender_list,
        android.R.layout.simple_spinner_item)

        adapter.setDropDownViewResource(android.R.layout.simple_spinner_drop
        down_item)

        spinnerGender.adapter = adapter
    }

    private fun validasiInput() {
        namaInput = edtName.text.toString()
        emailInput = edtEmail.text.toString()
    }

```

```

        telpInput = edtTelp.text.toString()
        alamatInput = edtAddress.text.toString()
        genderInput = spinnerGender.selectedItem.toString()

        when{
            namaInput.isEmpty() -> edtName.error = "Nama tidak boleh
kosong"
            genderInput.equals("Pilih kelamin") -
> tampilToast("Kelamin harus dipilih")
            emailInput.isEmpty() -> edtEmail.error = "Email
tidak boleh kosong"
            telpInput.isEmpty() -> edtTelp.error = "Telp tidak boleh
kosong"
            alamatInput.isEmpty() -> edtAddress.error =
"Alamat tidak boleh kosong"

            else -> {

                val teman = MyFriend(nama = namaInput, kelamin =
genderInput, email = emailInput, telp = telpInput, alamat =
alamatInput)

                tambahDataTeman(teman)

            }

        }

        private fun tambahDataTeman(teman: MyFriend) : Job {

            return GlobalScope.launch {
                myFriendDao?.tambahTeman(teman)
                (activity as MainActivity).tampilMyFriendsFragment()
            }

        }

        private fun tampilToast(message: String) {
            Toast.makeText(activity!!, message,
Toast.LENGTH_SHORT).show()
        }

        override fun onDestroy() {
            super.onDestroy()
            this.clearFindViewByIdCache()
        }

    }

```

4.7 Mengambil Data Teman Dari Database Lokal

Perlu dilakukan beberapa perubahan pada class MyFriendsFragment.kt untuk mengambil data list teman dari database lokal, diantaranya yaitu: mengganti tipe variable listTeman dari MutableList menjadi List, menambahkan global variable AppDatabase dan MyFriendDao, inialisasi AppDatabase dan MyFriendDao, dan membuat fungsi untuk mengambil (get) data teman.

MyFriendsFragment.kt

```
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Toast
import androidx.fragment.app.Fragment
import androidx.lifecycle.Observer
import androidx.recyclerview.widget.LinearLayoutManager
import kotlinx.android.synthetic.*
import kotlinx.android.synthetic.main.my_friends_fragment.*

class MyFriendsFragment : Fragment() {

    companion object {
        fun newInstance(): MyFriendsFragment {
            return MyFriendsFragment()
        }
    }

    private var listTeman : List<MyFriend>? = null

    private var db: AppDatabase? = null
    private var myFriendDao: MyFriendDao? = null

    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View? {
        return inflater.inflate(R.layout.my_friends_fragment, container, false)
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        initLocalDB()
        initView()
    }

    private fun initLocalDB() {
        db = AppDatabase.getAppDataBase(activity!!)
        myFriendDao = db?.myFriendDao()
    }

    private fun initView() {
```

```

        fabAddFriend.setOnClickListener { (activity as
MainActivity).tampilMyFriendsAddFragment() }

        ambilDataTeman()
    }

    private fun ambilDataTeman() {

        listTeman = ArrayList()
        myFriendDao?.ambilSemuaTeman()?.observe(this, Observer { r ->

            listTeman = r

            when {
                listTeman?.size == 0 -> tampilToast("Belum ada data
teman")

                else -> {
                    tampilTeman()
                }
            }

        })

    }

    private fun tampilToast(message: String) {
        Toast.makeText(activity!!, message, Toast.LENGTH_SHORT).show()
    }

    private fun tampilTeman() {
        listMyFriends.layoutManager = LinearLayoutManager(activity)
        listMyFriends.adapter = MyFriendAdapter(activity!!, listTeman!!)
    }

    // private fun simulasiDataTeman() {
    //     listTeman = ArrayList()
    //     //
    //     listTeman.add(MyFriend("Muhammad", "Laki-laki", "ade@gmail.com",
    // "085719004268", "Bandung"))
    //     listTeman.add(MyFriend("Al Harits", "Laki-
    // laki", "rifaldi@gmail.com", "081213416171", "Bandung"))
    // }

    override fun onDestroy() {
        super.onDestroy()
        this.clearFindViewByIdCache()
    }

}

```

selesai