

# MODUL PRAKTIKUM PEMROGRAMAN UNTUK PERANGKAT BERGERAK 1

Indra Azimi, S.T., M.T.

Reza Budiawan, S.T., M.T., OCA



D3 Rekayasa Perangkat Lunak Aplikasi  
Telkom University

# Daftar Isi

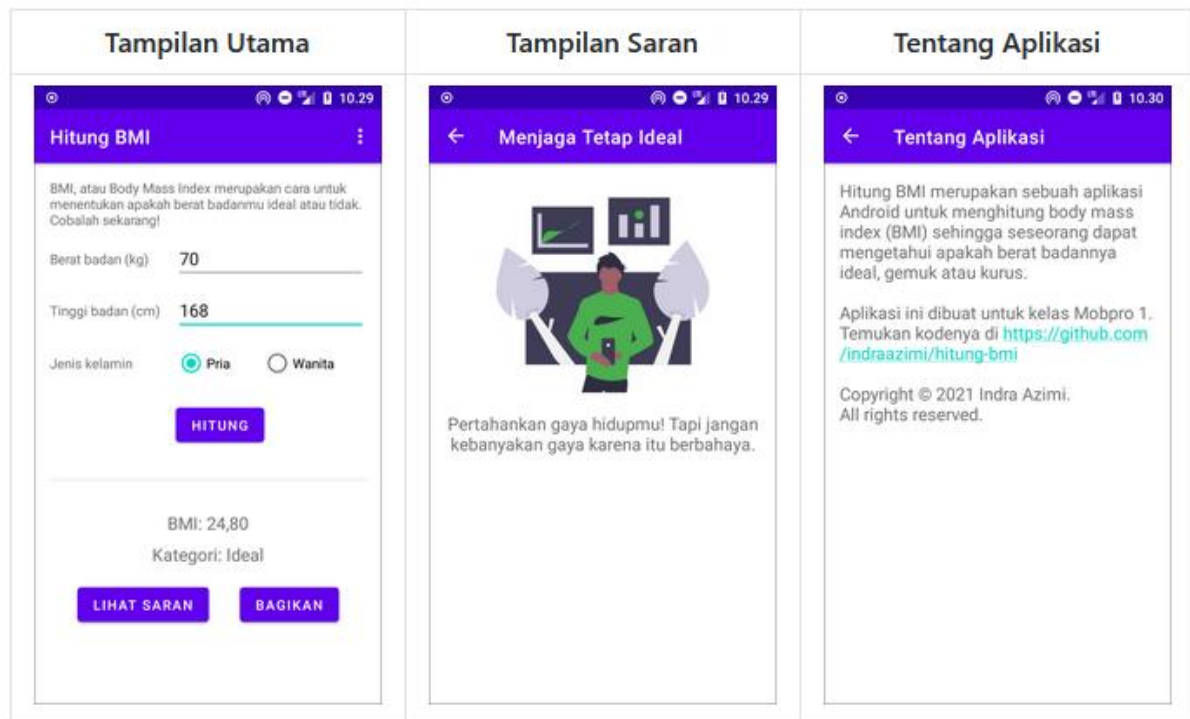
Modul 08: App Navigation .....	1
1. Overview .....	1
2. Getting Started.....	2
3. Task .....	2
3.1. Refactor fitur hitung BMI ke fragment .....	3
3.2. Membuat fragment lihat saran.....	6
3.3. Menambahkan navigation component .....	10
3.4. Membuat navigasi sederhana.....	12
3.5. Mengirim data saat bernavigasi .....	15
3.6. Membuat navigasi menggunakan menu .....	18
3.7. Membuat navigasi ke app lain .....	23
4. Summary .....	27
5. Challenge.....	27

# Modul 08: App Navigation

## 1. Overview

Pada modul kali ini, kita akan membuat sebuah aplikasi yang menggunakan beberapa layar untuk ditampilkan. Jadi tidak hanya menggunakan satu layar seperti yang sudah dilakukan sebelumnya. Kemudian untuk berpindah dari satu layar ke layar lainnya, kita akan menggunakan komponen navigasi yang merupakan bagian dari Android Jetpack.

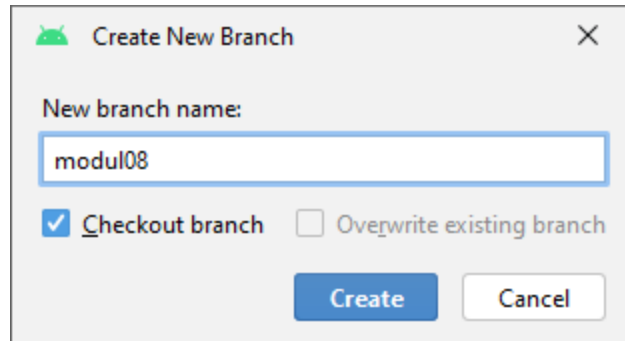
Pembuatan aplikasi ini meneruskan proses pengembangan aplikasi perhitungan BMI di modul 07, yaitu App Architecture. Aplikasi yang akan kita buat dapat dilihat versi lengkapnya di repository Github <https://github.com/indraazimi/hitung-bmi> pada branch navigation. Berikut ini merupakan tampilan akhir dari aplikasi yang akan memiliki tiga buah layar.



Ketika aplikasi dijalankan, akan muncul tampilan utama. Di sini pengguna dapat memasukkan datanya lalu menghitung BMI dan kategorinya. Kemudian pengguna dapat melihat saran. Tampilan saran ini bersifat dinamis, sesuai dengan kategori BMI yang dimiliki pengguna. Selain itu pengguna juga dapat membagikan hasil perhitungan BMI ini ke aplikasi lain.

## 2. Getting Started

Karena modul 7 sudah ada Git-nya, di modul ini kita tidak membuat repository baru, namun cukup membuat branch baru saja. Buka project Hitung BMI dari modul 7 di Android Studio. Pastikan saat ini kita berada di branch **modul07** dan aplikasi di branch tersebut berjalan dengan baik. Selanjutnya klik menu Git > New Branch, lalu masukkan “modul08” sebagai nama branch yang baru, lalu klik Create sesuai gambar. Kerjakan task-task pada modul ini di branch yang baru dibuat.



Task pada modul ini juga akan membuat beberapa class baru. Pada pembuatannya, perhatikan penempatan package class tersebut, dan pastikan tidak menghapus kode package yang telah terbentuk pada class yang sudah ada. Karena kesalahan package dapat menyebabkan kode aplikasi menjadi error dan aplikasi menjadi tidak bisa dijalankan.

**Dilarang keras untuk copy – paste kode dari modul/sumber lain!**

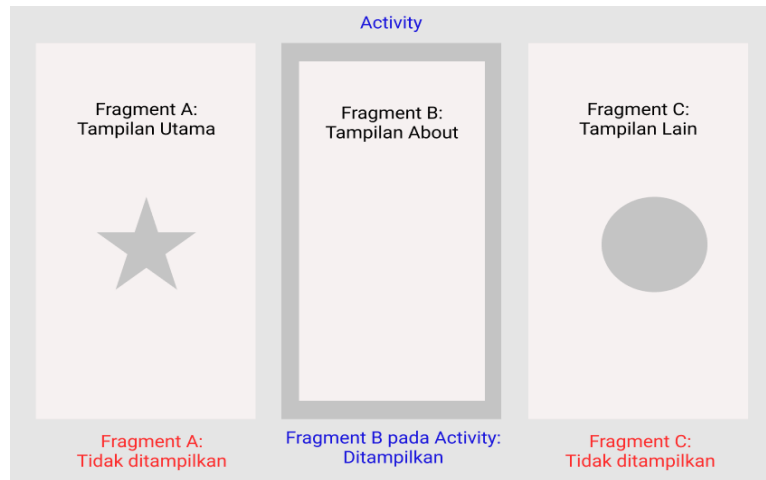
**Ngoding pelan-pelan akan membuat kamu lebih jago di masa depan.**

**Lakukan commit setiap selesai 1 sub-task. Selamat ngoding!**

## 3. Task

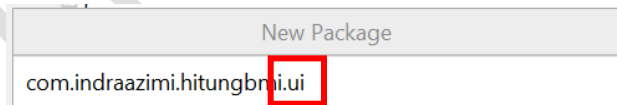
Ketika ingin membuat aplikasi dengan beberapa layar, kita dihadapkan pada dua pilihan. Kita bisa membuat satu activity untuk setiap layar, sehingga pada akhirnya kita akan memiliki banyak activity. Atau kita bisa menggunakan pendekatan satu fragment untuk setiap layar, sehingga pada akhirnya kita cukup menggunakan satu activity saja. Pendekatan kedua ini disebut sebagai “single activity – multiple fragment”, dan cara inilah yang kita pakai di praktikum sekarang.

Menurut [dokumentasi Android](#), fragment adalah bagian UI aplikasi yang dapat digunakan kembali. Fragment menghadirkan modularitas dan memungkinkan kita membagi UI menjadi bagian-bagian terpisah. Dengan kata lain, fragment akan berisi tampilan kepada pengguna, sedangkan activity akan berfungsi sebagai jendela yang menampilkan fragment tersebut. Ilustrasinya:

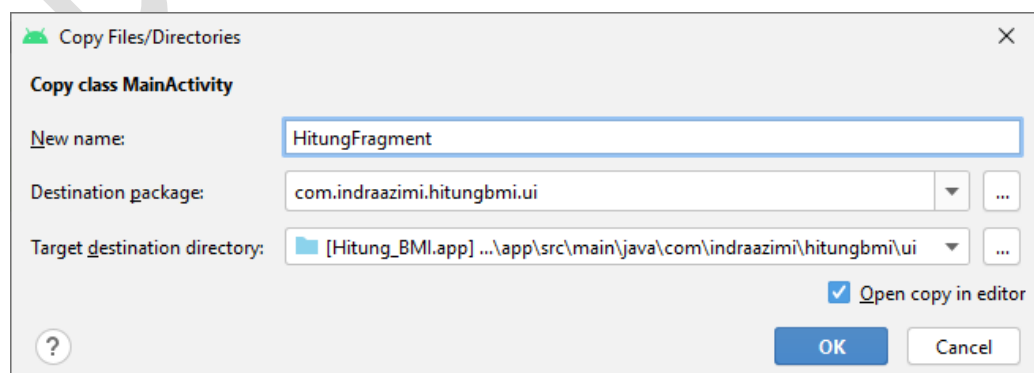


### 3.1. Refactor fitur hitung BMI ke fragment

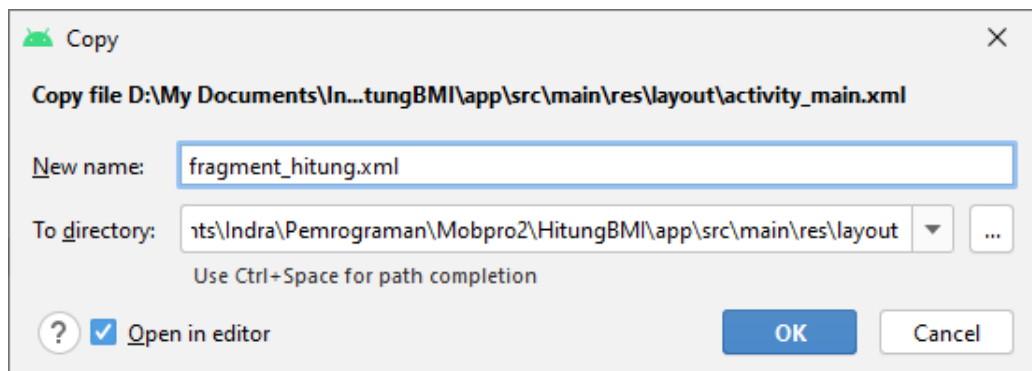
Oleh karena kita akan menggunakan pendekatan “single activity – multiple fragment”, langkah awal yang perlu dilakukan adalah memindahkan kode aplikasi sebelumnya yang ada di activity ke fragment. Agar project kita lebih rapi dan teratur, kita akan mengelompokkan fragment ini ke package baru bernama ui. Klik kanan pada package, pilih New → package, dan tambahkan “ui” setelah hitungbmi..



Selanjutnya, copy file MainActivity.kt, kemudian paste ke package ui dengan nama HitungFragment..



Fragment baru ini akan memerlukan tampilan berupa file XML. Untuk itu, copy activity\_main.xml dan paste sebagai fragment\_hitung.xml di folder layout.



Sesuaikan context dari fragment\_hitung.xml berikut.

```
<ScrollView 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">
    tools:context=".ui.HitungFragment">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        ...
    </androidx.constraintlayout.widget.ConstraintLayout>

</ScrollView>
```

Kembalilah ke HitungFragment dan lakukan penyesuaian AppCompatActivity menjadi Fragment. Lalu sesuaikan juga binding dan viewModel seperti ini.

```
class MainActivity : AppCompatActivity() {
class HitungFragment : Fragment() {

    private lateinit var binding: ActivityMainBinding
    private lateinit var binding: FragmentHitungBinding

    private val viewModel: MainViewModel by lazy {
        ViewModelProvider(this)[MainViewModel::class.java]
        ViewModelProvider(requireActivity())[MainViewModel::class.java]
    }
}
```

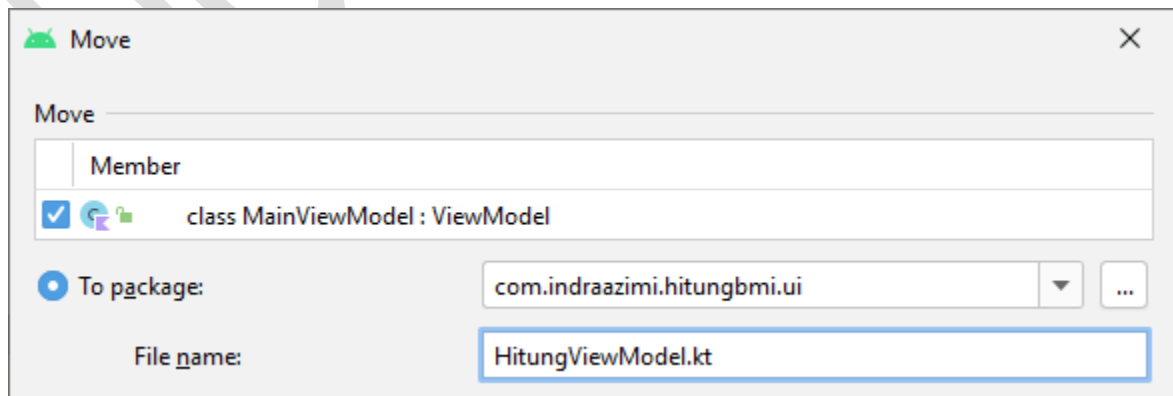
Selanjutnya kita akan membagi method onCreate menjadi 2, yaitu onCreateView dan onViewCreated.

```
override fun onCreate(savedInstanceState: Bundle?) {  
    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,  
        savedInstanceState: Bundle?): View {  
        super.onCreate(savedInstanceState)  
        binding = ActivityMainBinding.inflate(LayoutInflater)  
        binding = FragmentHitungBinding.inflate(LayoutInflater, container, false)  
        setContentView(binding.root)  
        return binding.root  
    }  
  
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  
        binding.button.setOnClickListener { hitungBmi() }  
        viewModel.getHasilBmi().observe(this, { showResult(it) })  
        viewModel.getHasilBmi().observe(requireActivity(), { showResult(it) })  
    }  
}
```

Terakhir, sesuaikan kode pada perintah Toast agar tidak terjadi error. Karena Fragment bukan kelas turunan dari Context, maka ubah semua kode this pada Toast menjadi context seperti contoh berikut.

```
Toast.makeText(this, R.string.berat_invalid, Toast.LENGTH_LONG).show()  
Toast.makeText(context, R.string.berat_invalid, Toast.LENGTH_LONG).show()  
  
Toast.makeText(this, R.string.tinggi_invalid, Toast.LENGTH_LONG).show()  
Toast.makeText(context, R.string.tinggi_invalid, Toast.LENGTH_LONG).show()  
  
Toast.makeText(this, R.string.gender_invalid, Toast.LENGTH_LONG).show()  
Toast.makeText(context, R.string.gender_invalid, Toast.LENGTH_LONG).show()
```

HitungFragment sudah selesai di-update.. Sekarang pindahkan MainViewModel ke package ui agar menjadi satu dengan HitungFragment. Ubah juga namanya menjadi HitungViewModel agar konsisten.



Sekarang kita update file MainActivity. Di sini kita akan mengembalikan kode ke bentuk awalnya, yaitu kode default MainActivity.kt saat membuat project pertama kali.

```
class MainActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
}
```

Selanjutnya ubah kode activity\_main.xml sehingga menampilkan fragment hitung yang telah dibuat. Pastikan android:name sesuai dengan nama package masing-masing.

```
<?xml version="1.0" encoding="utf-8"?>  
<fragment xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:id="@+id/container"  
    android:name="com.indraazimi.hitungbmi.ui.HitungFragment"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context=".MainActivity"  
    tools:viewBindingIgnore="true" />
```

Setelah melakukan semua hal di atas, kita telah memindahkan (refactor) fitur hitung BMI dari activity ke fragment. Run project dan pastikan aplikasi berjalan seperti sebelumnya. Aplikasi menampilkan layar untuk menerima input pengguna. Ketika tombol “Hitung” ditekan, maka akan keluar nilai BMI dan kategorinya. Pastikan aplikasi sudah berjalan dengan baik, dan lakukan commit dengan commit message sesuai judul task 3.1 ini.

### 3.2. Membuat fragment lihat saran

Pada task ini, kita akan membuat sebuah tampilan yang menampilkan saran kepada pengguna tergantung kategori yang dihasilkan dari perhitungan. Seperti yang telah dibahas di Overview, tampilan saran ini nantinya akan memunculkan gambar dan teks. Untuk itu, tambahkan gambar yang akan digunakan (file Zip terlampir) ke dalam folder drawable terlebih dahulu. Selanjutnya tambahkan juga string berikut di strings.xml.

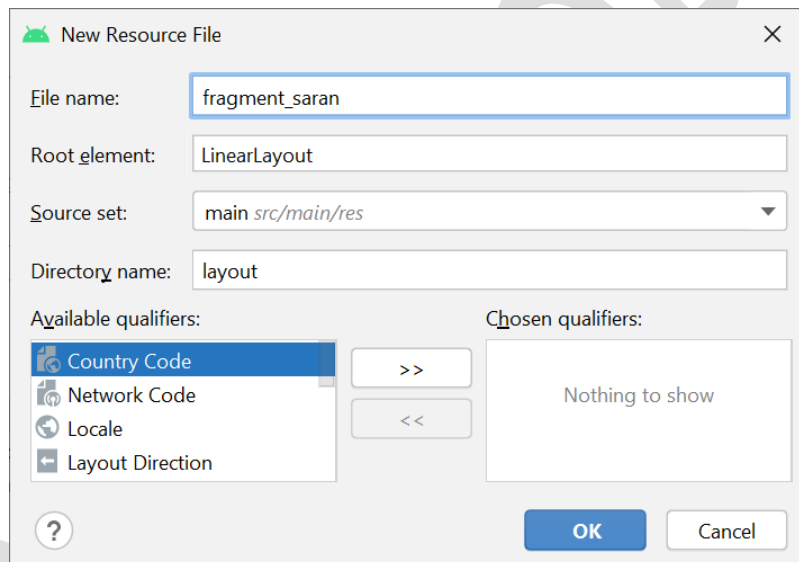


```

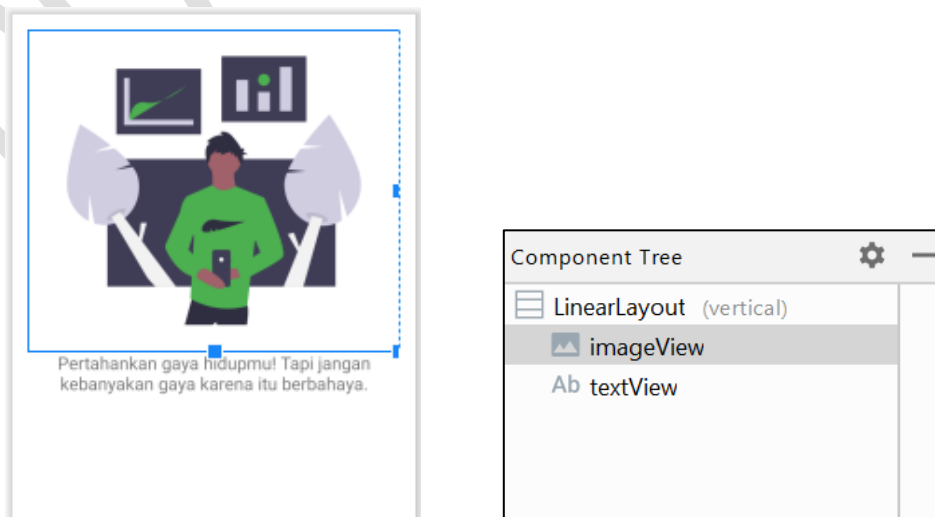
<resources>
    ...
    <string name="judul_kurus">Mengatasi Kekurusan</string>
    <string name="judul_ideal">Menjaga Tetap Ideal</string>
    <string name="judul_gemuk">Mengatasi Kegemukan</string>
    <string name="saran_kurus">Cobalah untuk makan porsi besar, seperti burger
    berukuran raksasa ini.</string>
    <string name="saran_ideal">Pertahankan gaya hidupmu! Tapi jangan kebanyakan
    gaya karena itu berbahaya.</string>
    <string name="saran_gemuk">Cobalah untuk olahraga lari. Kemanapun boleh,
    asal tidak lari dari kehidupan.</string>
</resources>

```

Berikutnya, buat XML baru dengan klik kanan pada /res/layout pilih New → Layout Resource File.



Atur tampilan sebagai berikut. Gunakan LinearLayout dengan 1 ImageView dan 1 TextView.



Kode XML dari tampilan di atas adalah seperti ini:

```
<?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="16dp"
    android:orientation="vertical"
    android:weightSum="2">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:contentDescription="@null"
        tools:src="@drawable/ideal" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:textSize="18sp"
        tools:text="@string/saran_ideal" />

</LinearLayout>
```

Langkah selanjutnya, tambahkan sebuah class `SaranFragment.kt` pada package `ui`. Klik kanan pada package `ui`, lalu pilih `New` → `Kotlin Class/File`, tuliskan `SaranFragment` dengan bagian “Class” terpilih. Tambahkan sifat fragment pada class ini, dan hubungkan tampilan `fragment_saran.xml` dengan `SaranFragment.kt` dengan meng-override `onCreateView`.

```
class SaranFragment : Fragment() {

    private lateinit var binding: FragmentSaranBinding

    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?): View {
        binding = FragmentSaranBinding.inflate(layoutInflater, container, false)
        return binding.root
    }
}
```

Berikutnya, tambahkan kode untuk mengubah user interface tergantung kategori yang dihasilkan dari perhitungan, sehingga tampilan saran akan bersifat dinamis.

```
private fun updateUI(kategori: KategoriBmi) {
    val actionBar = (requireActivity() as AppCompatActivity).supportActionBar
    when (kategori) {
        KategoriBmi.KURUS -> {
            actionBar?.title = getString(R.string.judul_kurus)
            binding.imageView.setImageResource(R.drawable.kurus)
            binding.textView.text = getString(R.string.saran_kurus)
        }
        KategoriBmi.IDEAL -> {
            actionBar?.title = getString(R.string.judul_ideal)
            binding.imageView.setImageResource(R.drawable.ideal)
            binding.textView.text = getString(R.string.saran_ideal)
        }
        KategoriBmi.GEMUK -> {
            actionBar?.title = getString(R.string.judul_gemuk)
            binding.imageView.setImageResource(R.drawable.gemuk)
            binding.textView.text = getString(R.string.saran_gemuk)
        }
    }
}
}
```

Selanjutnya buat method onCreateView, lalu panggil method updateUI.

```
override fun onCreateView(view: View, savedInstanceState: Bundle?) {
    updateUI(KategoriBmi.KURUS)
}
```

Sampai di sini, fragment saran sudah terbentuk layout dan juga class-nya. Berikutnya, sesuaikan tampilan pada activity\_main.xml untuk menampilkan SaranFragment ini dengan mengubah bagian atribut android:name seperti berikut.

```
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/container"
    android:name="com.indraazimi.hitungbmi.ui.HitungFragment"
    android:name="com.indraazimi.hitungbmi.ui.SaranFragment"
    ... />
```

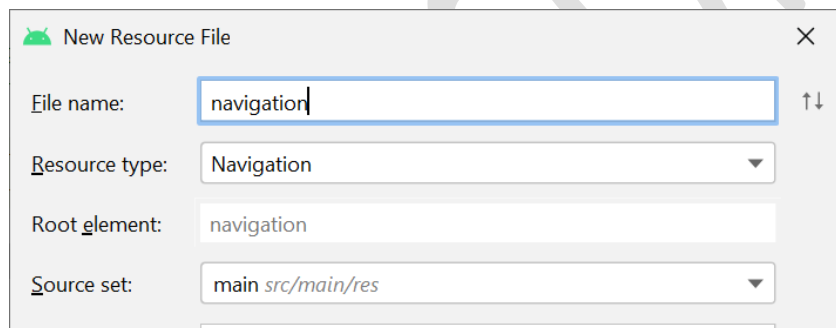
Setelah melakukan perubahan di atas, lakukan run project. Aplikasi sekarang memunculkan tampilan yang memperlihatkan saran untuk kategori KURUS. Setelah memastikan aplikasi berjalan dengan baik, lakukan commit dengan commit message sesuai judul task 3.2 ini.

### 3.3. Menambahkan navigation component

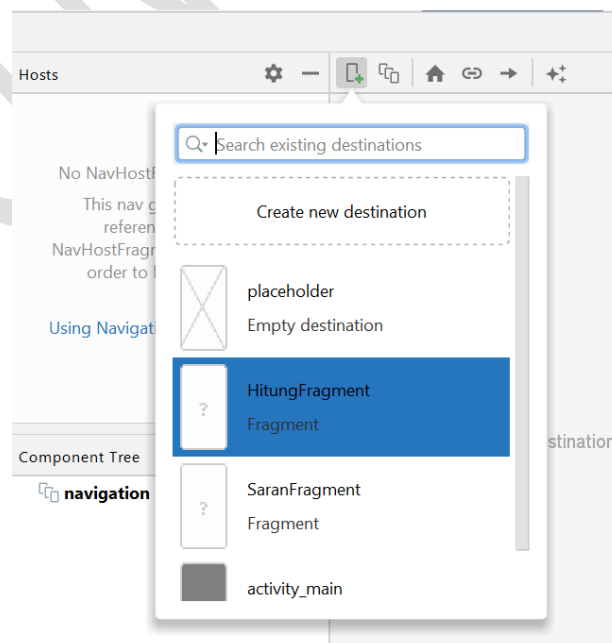
Pada task ini, kita akan menambahkan navigation component yang memungkinkan adanya navigasi (perpindahan) antara tampilan menghitung BMI dengan tampilan saran. Langkah awal task ini adalah menambahkan dependensi pada app/build.gradle. Jangan lupa lakukan Gradle sync.

```
dependencies {  
    ...  
    implementation 'androidx.navigation:navigation-fragment-ktx:2.4.2'  
    implementation 'androidx.navigation:navigation-ui-ktx:2.4.2'  
}
```

Setelah menambahkan dependensi, buat sebuah file navigation.xml dengan klik kanan folder /res, pilih New → Android Resource File. Pilih Resource type “Navigation” dengan nama “navigation”.



Pada file yang terbentuk, klik ikon tambah fragment, lalu pilih “HitungFragment”.



Sesuaikan labelnya sehingga kode yang terbentuk adalah sebagai berikut:

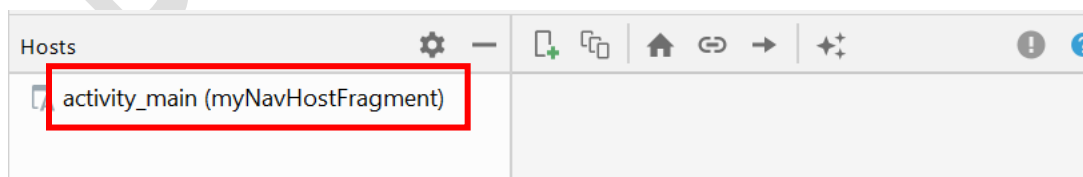
```
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/navigation"
    app:startDestination="@id/hitungFragment">

    <fragment
        android:id="@+id/hitungFragment"
        android:name="com.indraazimi.hitungbmi.ui.HitungFragment"
        android:label="@string/app_name"
        tools:layout="@layout/fragment_hitung"/>
</navigation>
```

Gunakan navigation ini pada activity\_main.xml. Sehingga, tampilan ini menjadi host pada navigation, dan menjadi “jendela” bagi tampilan yang dituliskan pada komponen fragment.

```
<?xml version="1.0" encoding="utf-8"?>
<fragment 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/container"
    android:name="com.indraazimi.hitungbmi.ui.SaranFragment"
    android:id="@+id/myNavHostFragment"
    android:name="androidx.navigation.fragment.NavHostFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:defaultNavHost="true"
    app:navGraph="@navigation/navigation"
    tools:context=".MainActivity"
    tools:viewBindingIgnore="true" />
```

Setelah menambahkan atribut pada activity\_main.xml, buka kembali navigation.xml, maka akan terdefinisi host-nya seperti pada gambar berikut.



Run project, aplikasi akan menampilkan tampilan input untuk menghitung BMI. Setelah memastikan aplikasi berjalan dengan baik, lakukan commit dengan commit message sesuai judul task 3.3 ini.

### 3.4. Membuat navigasi sederhana

Setelah mendefinisikan komponen navigasi pada task sebelumnya, berikutnya kita akan membuat navigasi sederhana dengan menambahkan tombol pada tampilan utama ke tampilan saran. Untuk itu, kita tambahkan tombol pada tampilan `fragment_hitung.xml`.

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

    <androidx.constraintlayout.widget.ConstraintLayout
        ...

        <TextView
            android:id="@+id/kategoriTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:layout_marginBottom="16dp"
            android:textSize="18sp"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/bmiTextView"
            tools:text="Kategori: Ideal" />

        <Button
            android:id="@+id/saranButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="16dp"
            android:layout_marginTop="16dp"
            android:layout_marginEnd="16dp"
            android:layout_marginBottom="16dp"
            android:text="@string/lihat_saran"
            android:visibility="invisible"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/kategoriTextView"
            tools:visibility="visible" />

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

<resources>
    ...
    <string name="lihat_saran">Lihat Saran</string>
    ...
</resources>
```

Hasil dari penambahan kode tombol di atas dapat dilihat pada gambar berikut.

BMI, atau Body Mass Index merupakan cara untuk menentukan apakah berat badanmu ideal atau tidak. Cobalah sekarang!

Berat badan (kg) \_\_\_\_\_

Tinggi badan (cm) \_\_\_\_\_

Jenis kelamin ☐ Pria ☐ Wanita

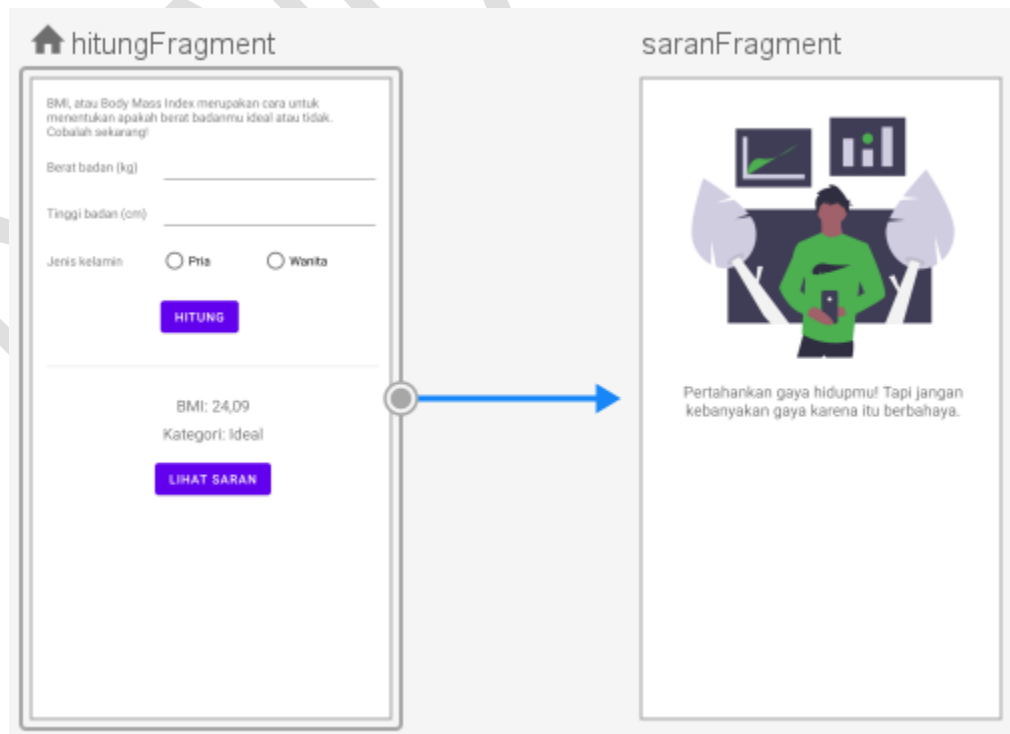
**HITUNG**

---

BMI: 24,09  
Kategori: Ideal

**LIHAT SARAN**

Berikutnya, buka navigation.xml lalu tambahkan fragment lihat saran dengan cara klik pada ikon tambah fragment. Kemudian tambahkan jalur navigasi dari hitungFragment ke saranFragment dengan cara drag bulatan di sisi kanan hitungFragment ke saranFragment. Maka akan muncul garis biru seperti gambar di bawah ini.



Setelah melakukan perubahan pada design navigation.xml, kode yang terbentuk adalah:

```
<navigation ...>

    <fragment
        android:id="@+id/hitungFragment"
        android:name="com.indraazimi.hitungbmi.ui.HitungFragment"
        android:label="@string/app_name"
        tools:layout="@layout/fragment_hitung">
        <action
            android:id="@+id/action_hitungFragment_to_saranFragment"
            app:destination="@id/saranFragment" />
        </fragment>
    <fragment
        android:id="@+id/saranFragment"
        android:name="com.indraazimi.hitungbmi.ui.SaranFragment"
        tools:layout="@layout/fragment_saran"/>
</navigation>
```

Jalur yang dibentuk merupakan aksi "<action ...>" yang terdapat pada navigation.xml. Aksi ini kita panggil di action listener saat menekan tombol saran. Untuk itu, ubah HitungFragment.kt menjadi:

```
override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    binding.button.setOnClickListener { hitungBmi() }
    binding.saranButton.setOnClickListener {
        it.findNavController().navigate(
            R.id.action_hitungFragment_to_saranFragment
        )
    }
    viewModel.getHasilBmi().observe(requireActivity(), { showResult(it) })
}
```

Jangan lupa untuk menampilkan tombol saran ketika menampilkan hasil perhitungan BMI.

```
private fun showResult(result: HasilBmi?) {
    if (result == null) return

    binding.bmiTextView.text = getString(R.string.bmi_x, result.bmi)
    binding.kategoriTextView.text = getString(R.string.kategori_x,
        getKategoriLabel(result.kategori))
    binding.saranButton.visibility = View.VISIBLE
}
```



Pada tahap ini, project dapat di-run dan tombol saran dapat di-klik yang mengarahkan ke tampilan saran. Berikutnya, kita akan membuat tombol up (tanda panah yang ada di pojok kiri atas aplikasi). Caranya, ubah MainActivity.kt sebagai berikut:

```
class MainActivity : AppCompatActivity() {  
  
    private lateinit var navController: NavController  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        navController = findNavController(R.id.myNavHostFragment)  
        NavigationUI.setupActionBarWithNavController(this, navController)  
    }  
  
    override fun onSupportNavigateUp(): Boolean {  
        return navController.navigateUp()  
    }  
}
```

Run project, dan pada saat memunculkan tampilan saran, maka tombol up akan dimunculkan di pojok kiri atas aplikasi. Setelah memastikan aplikasi berjalan dengan baik, lakukan commit dengan commit message sesuai judul task 3.4 ini.

### 3.5. Mengirim data saat bernavigasi

Sampai tahap ini, aplikasi kita sudah menampilkan 2 layar kepada pengguna. Akan tetapi tampilan saran masih statis, hanya menampilkan untuk pilihan kurus. Sedangkan 2 kategori lain (gemuk dan ideal) belum ditampilkan. Untuk itu, diperlukan pengiriman data dari layar utama (HitungFragment) ke layar saran (SaranFragment).

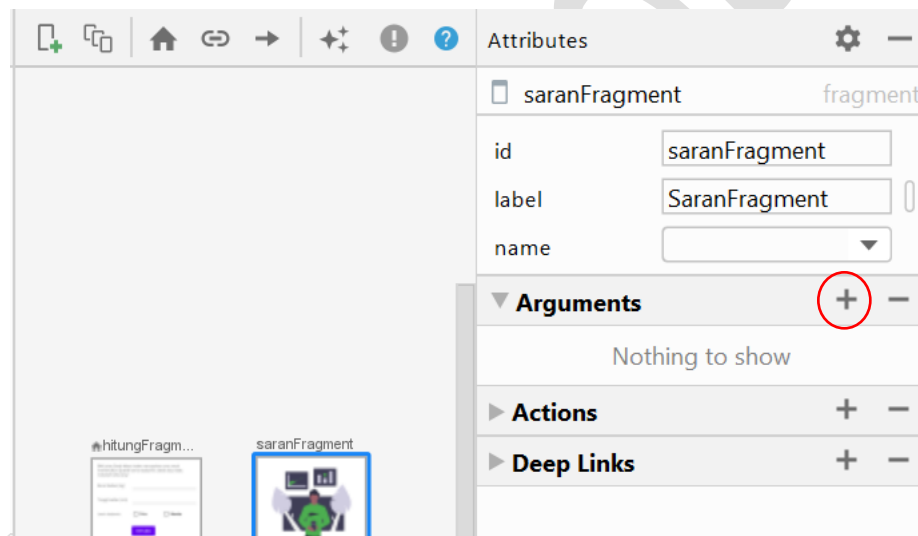
Langkah awal, tambahkan classpath SafeArgs pada dependencies project/build.gradle:

```
buildscript {  
    ...  
    dependencies {  
        classpath "com.android.tools.build:gradle:7.0.4"  
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:1.6.10"  
        classpath "androidx.navigation:navigation-safe-args-gradle-plugin:2.4.2"  
    }  
}
```

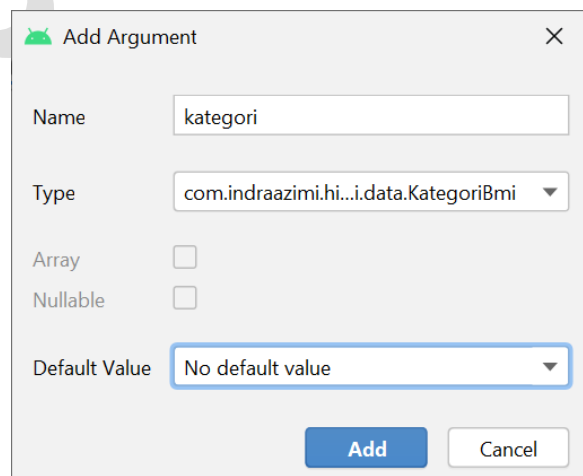
Berikutnya, tambahkan plugin SafeArgs pada app/build.gradle:

```
plugins {  
    id 'com.android.application'  
    id 'kotlin-android'  
    id 'androidx.navigation.safeargs.kotlin'  
}  
...
```

Lakukan sinkronisasi Gradle (klik sync now di pojok kanan atas). Selanjutnya pada navigation.xml, tambahkan argument yang akan diterima oleh saranFragment. Caranya, dengan saranFragment dalam kondisi terpilih, klik simbol “+” pada Arguments Attributes.



Pada bagian Name, berikan nilai “kategori”. Pada bagian Type pilih “Custom Enum”, dan pilih enumerasi yang telah dibuat sebelumnya, yaitu KategoriBmi.



Pada bagian editor, akan tertulis kode berikut:

```
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    ...
    <fragment
        android:id="@+id/saranFragment"
        android:name="com.indraazimi.hitungbmi.ui.SaranFragment"
        android:label="SaranFragment"
        tools:layout="@layout/fragment_saran"/>
        tools:layout="@layout/fragment_saran">
        <argument
            android:name="kategori"
            app:argType="com.indraazimi.hitungbmi.data.KategoriBmi" />
    </fragment>
</navigation>
```

Kita akan menggunakan data argument yang sudah dibuat sebelumnya untuk mengirimkan nilai kategori BMI dari HitungFragment.kt ke SaranFragment.kt. Oleh karena data tersebut ada di view model, maka ubah HitungViewModel menjadi:

```
class HitungViewModel : ViewModel() {

    private val hasilBmi = MutableLiveData<HasilBmi?>()
    private val navigasi = MutableLiveData<KategoriBmi?>()

    ...
    fun getHasilBmi(): LiveData<HasilBmi?> = hasilBmi

    fun mulaiNavigasi() {
        navigasi.value = hasilBmi.value?.kategori
    }

    fun selesaiNavigasi() {
        navigasi.value = null
    }

    fun getNavigasi() : LiveData<KategoriBmi?> = navigasi
}
```

Selanjutnya, ubah HitungFragment sebagai berikut:

```
override fun onCreateView(view: View, savedInstanceState: Bundle?) {
    binding.button.setOnClickListener { hitungBmi() }
    binding.saranButton.setOnClickListener {
        it.findNavController().navigate(
            R.id.action_hitungFragment_to_saranFragment
        )
    }
    binding.saranButton.setOnClickListener { viewModel.mulaiNavigasi() }
```

```

viewModel.getHasilBmi().observe(requireActivity(), { showResult(it) })
viewModel.getNavigasi().observe(viewLifecycleOwner, {
    if (it == null) return@observe
    findNavController().navigate(HitungFragmentDirections
        .actionHitungFragmentToSaranFragment(it))
    viewModel.selesaiNavigasi()
})
}

```

Jika pada bagian HitungFragmentDirections dinyatakan error, lakukan rebuild project dengan memilih Build → Rebuild Project. Saat ini, HitungFragment sudah memberikan navigasi beserta argument yang diperlukan (yaitu variable kategoriBmi) ke SaranFragment. Berikutnya, lakukan perubahan di SaranFragment agar dapat menampilkan data sesuai argument yang diterima.

```

class SaranFragment : Fragment() {

    private lateinit var binding: FragmentSaranBinding
    private val args: SaranFragmentArgs by navArgs()

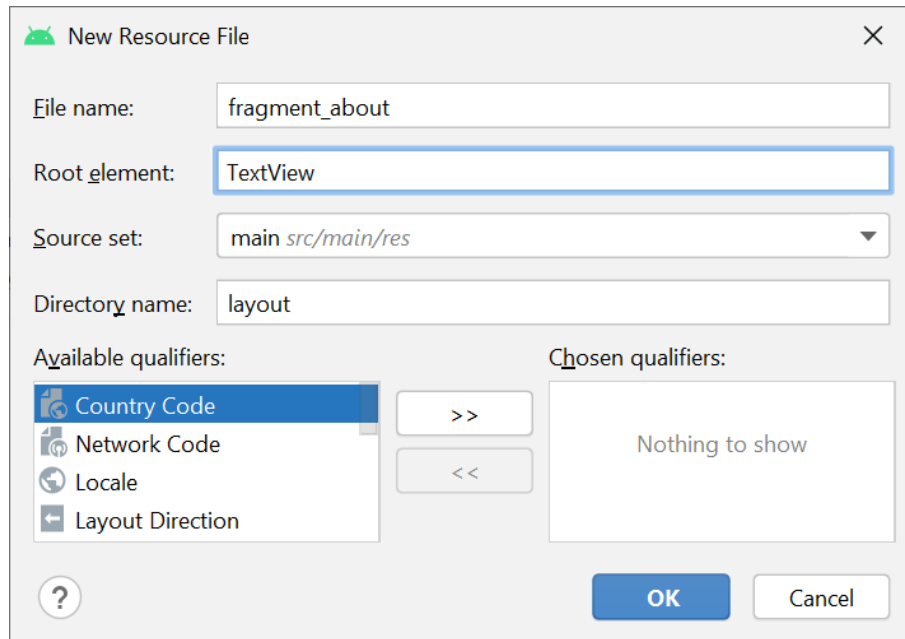
    ...
    override fun onCreateView(view: View, savedInstanceState: Bundle?) {
        updateUI(KategoriBmi.KURUS)
        updateUI(args.kategori)
    }
    ...
}

```

Jika terdapat error, lakukan import pada kode yang bersesuaian. Jalankan kembali aplikasi, dan perhatikan bahwa saran sudah ditampilkan sesuai dengan hasil perhitungan BMI yang dilakukan. Setelah memastikan aplikasi berjalan dengan baik, lakukan commit dengan commit message sesuai judul task 3.5 ini.

### 3.6. Membuat navigasi menggunakan menu

Pada task ini, kita akan membuat overflow menu (muncul dalam bentuk 3 titik di pojok kanan atas) yang menampilkan layar “Tentang Aplikasi” dengan informasi berupa copyright. Langkah pertama, buat sebuah tampilan fragment\_about pada /res/layout. Klik kanan pada /res/layout, pilih New → LayoutResource File. Berikan nama fragment\_about dengan root element TextView.



Atur konten pada fragment\_about.xml dan strings.xml sebagai berikut:

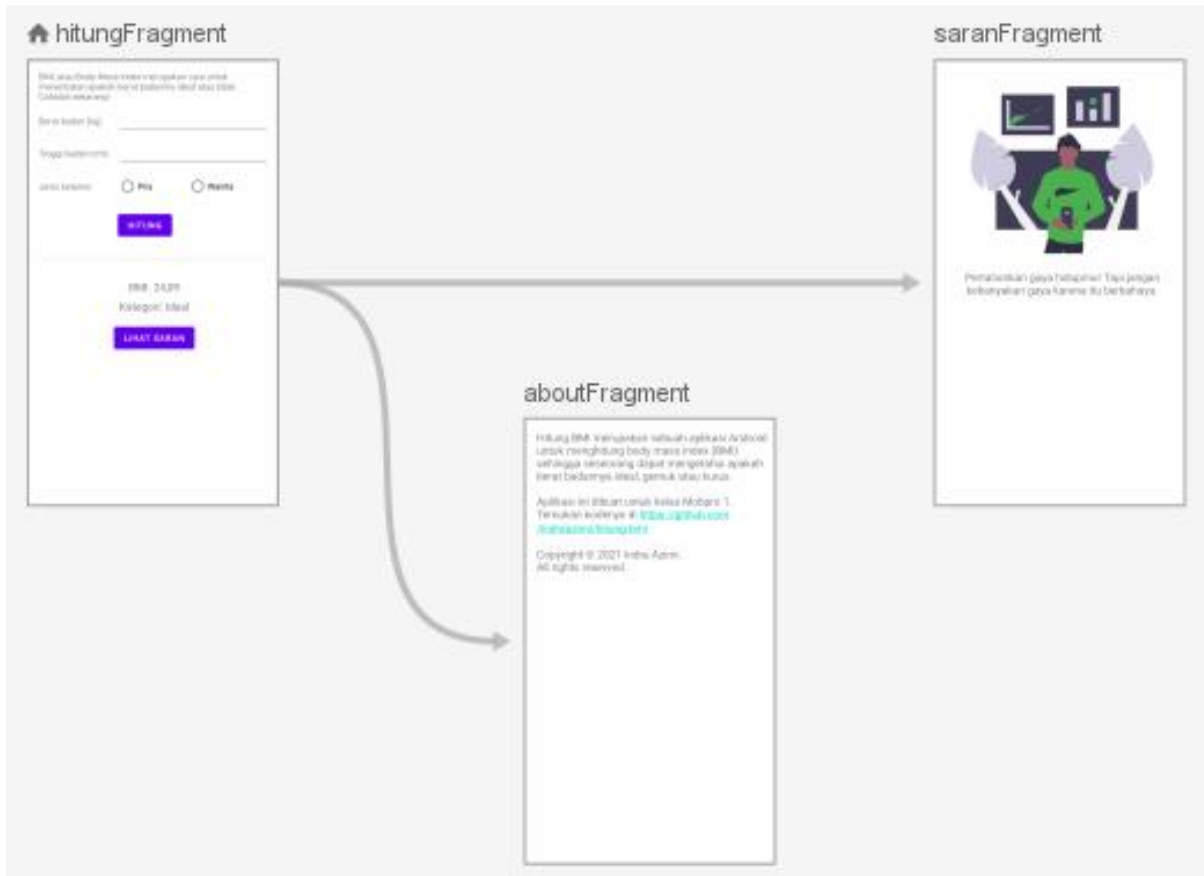
```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="16dp"
    android:autoLink="web"
    android:text="@string/copyright"
    android:textSize="18sp" />

<resources>
    ...
    <string name="tentang_aplikasi">Tentang Aplikasi</string>
    <string name="copyright">Hitung BMI merupakan sebuah aplikasi Android untuk
menghitung body mass index (BMI) sehingga seseorang dapat mengetahui apakah
berat badannya ideal, gemuk atau kurus.\n\nAplikasi ini dibuat untuk kelas
Mobpro 1. Temukan kodenya di https://github.com/indraazimi/hitung-
bmi\n\nCopyright © 2021 Indra Azimi.\nAll rights reserved.</string>
</resources>
```

Berikutnya, pada package ui, buat sebuah class AboutFragment.kt (klik kanan pada package pilih New → Kotlin Class/File). Berikan sifat fragment pada class ini, dan sekaligus hubungkan dengan layout XML fragment\_about. Sesuaikan import dengan nama package masing-masing.

```
class AboutFragment : Fragment(R.layout.fragment_about)
```

Layar fragment untuk menampilkan about sudah terbentuk, dan hal lain yang perlu dilakukan adalah mendaftarkan fragment ini pada navigasi. Buka navigation.xml, tambahkan aboutFragment dengan menekan tanda “+” seperti sebelumnya. Lalu, tambahkan aksi dari hitungFragment ke aboutFragment dengan menarik garis dari kedua fragment tersebut. Perhatikan arah panah, pastikan mengarah ke aboutFragment dari hitungFragment.



Berikut adalah kode dan tampilan yang terbentuk:

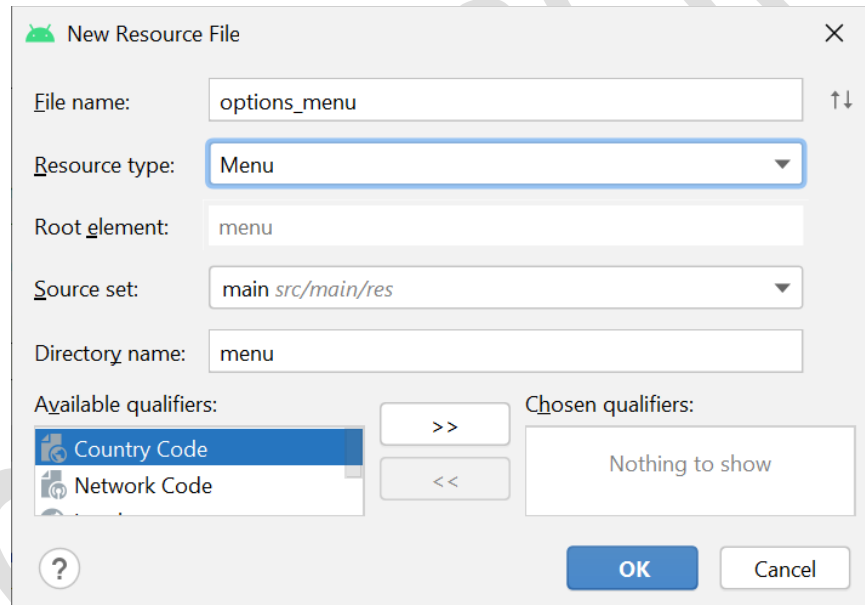
```
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
...
    <fragment
        android:id="@+id/hitungFragment"
        ...>
        <action
            android:id="@+id/action_hitungFragment_to_saranFragment"
            app:destination="@id/saranFragment" />
        <action
            android:id="@+id/action_hitungFragment_to_aboutFragment"
            app:destination="@id/aboutFragment" />
    </fragment>
```

```

<fragment
    android:id="@+id/saranFragment"
    ...
</fragment>
<fragment
    android:id="@+id/aboutFragment"
    android:name="com.indraazimi.hitungbmi.ui.AboutFragment"
    android:label="@string/tentang_aplikasi"
    tools:layout="@layout/fragment_about"/>
</navigation>

```

Untuk menampilkan layar fragment “Tentang Aplikasi” di atas, kita memerlukan option menu. Buat sebuah option menu dengan cara klik kanan pada folder /res, pilih New → Android Resource File. Berikan nama options\_menu dan pilih Resource type “Menu”.



Pada bagian design, dari palette ke canvas tarik komponen “Menu Item”. Atur attribute id bernama menu\_about dan title diambil dari strings.xml (name: tentang\_aplikasi) dengan tulisan “Tentang Aplikasi”. Kode yang terbentuk adalah sebagai berikut:

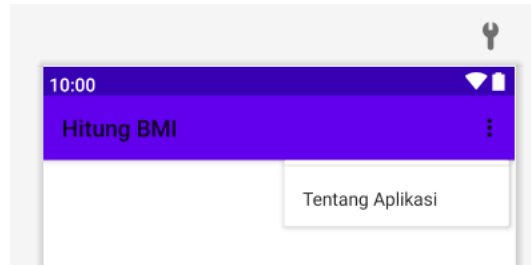
```

<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/menu_about"
        android:title="@string/tentang_aplikasi"
        app:showAsAction="never" />
</menu>

```

Design tampilan menu ditampilkan sebagai berikut:



Selanjutnya, gunakan options menu ini pada HitungFragment.kt. Lakukan override terhadap 2 method onCreateOptionsMenu dan onOptionsItemSelected. Pada onCreateOptionsMenu, inflate menu item R.menu.options\_menu yang telah dibentuk sebelumnya. Sedangkan pada onOptionsItemSelected, kita akan menambahkan aksi sehingga layar about ditampilkan. Aksi ini diambil dari <action> navigation sebelumnya. Lakukan import jika kode berwarna merah/error.

```
class HitungFragment : Fragment() {
    ...
    override fun onCreateOptionsMenu(menu: Menu, inflater: MenuInflater) {
        super.onCreateOptionsMenu(menu, inflater)
        inflater.inflate(R.menu.options_menu, menu)
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        if (item.itemId == R.id.menu_about) {
            findNavController().navigate(
                R.id.action_hitungFragment_to_aboutFragment
            )
            return true
        }
        return super.onOptionsItemSelected(item)
    }
    ...
}
```

Dikarenakan overflow menu/options menu ini ditampilkan pada fragment, diperlukan penambahan kode "setHasOptionsMenu(true)" pada onCreateView.

```
override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?): View {
    binding = FragmentHitungBinding.inflate(inflater, container, false)
    setHasOptionsMenu(true)
    return binding.root
}
```



Jalankan aplikasi, pastikan sekarang di pojok kanan atas muncul option menu yang menampilkan teks “Tentang Aplikasi”. Ketika menu ini di-klik, maka layar yang menampilkan informasi tentang aplikasi (AboutFragment) akan muncul. Setelah memastikan aplikasi berjalan dengan baik, lakukan commit dengan commit message sesuai judul task 3.6 ini.

### 3.7. Membuat navigasi ke app lain

Terkadang kita memerlukan navigasi yang tidak hanya mengarah ke layar yang sudah dibentuk sebelumnya, tapi juga ke aplikasi lain. Pada task kali ini, kita akan menambahkan share button yang akan mengarahkan kita ke aplikasi lain (messaging app).

Langkah pertama pada task ini adalah mengubah tampilan layar pada saat menampilkan hasil BMI. Pada layar ini, tambahkan sebuah share button. Agar mudah mengatur visibility state dari saranButton dan shareButton, gabungkan keduanya dalam satu grup. Pada fragment\_hitung.xml dan strings.xml lakukan perubahan sebagai berikut:

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

    <androidx.constraintlayout.widget.ConstraintLayout
        ...

        <Button
            android:id="@+id/saranButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="16dp"
            android:layout_marginTop="16dp"
            android:layout_marginEnd="16dp"
            android:layout_marginBottom="16dp"
            android:text="@string/lihat_saran"
            android:visibility="invisible"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintEnd_toStartOf="@+id/shareButton"
            app:layout_constraintHorizontal_chainStyle="packed"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/kategoriTextView"
            tools:visibility="visible" />
```

```

<Button
    android:id="@+id/shareButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginEnd="16dp"
    android:text="@string/bagikan"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/saranButton"
    app:layout_constraintTop_toTopOf="@+id/saranButton" />

<androidx.constraintlayout.widget.Group
    android:id="@+id/buttonGroup"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:visibility="invisible"
    app:constraint_referenced_ids="saranButton,shareButton"
    tools:visibility="visible" />

</androidx.constraintlayout.widget.ConstraintLayout>

</ScrollView>

<resources>
    ...
    <string name="bagikan">Bagikan</string>
</resources>

```

Dengan hasil akhir tampilan sebagai berikut:

BMI, atau Body Mass Index merupakan cara untuk menentukan apakah berat badanmu ideal atau tidak. Cobalah sekarang!

Berat badan (kg)

Tinggi badan (cm)

Jenis kelamin ☐ Pria ☐ Wanita

**HITUNG**

BMI: 24,09  
Kategori: Ideal

**LIHAT SARAN** **BAGIKAN** 16

Berikutnya, tambahkan string template untuk share pada strings.xml. Template ini akan kita gunakan pada kode Kotlin saat memberikan aksi klik pada button Share.

```
<resources>
    ...
    <string name="agikan_template">Berat badan: %1$s kg\nTinggi badan: %2$s
cm\nJenis kelamin: %3$s\n\n%4$s\n%5$s</string>
</resources>
```

Selanjutnya, kita berikan aksi button pada HitungFragment.kt.

```
override fun onCreateView(view: View, savedInstanceState: Bundle?) {
    binding.button.setOnClickListener { hitungBmi() }
    binding.saranButton.setOnClickListener { viewModel.mulaiNavigasi() }
    binding.shareButton.setOnClickListener { shareData() }

    viewModel.getHasilBmi().observe(requireActivity(), { showResult(it) })
    ...
}

private fun shareData() {
    val selectedId = binding.radioGroup.checkedRadioButtonId
    val gender = if (selectedId == R.id.priaRadioButton)
        getString(R.string.pria)
    else
        getString(R.string.wanita)

    val message = getString(R.string.agikan_template,
        binding.beratEditText.text,
        binding.tinggiEditText.text,
        gender,
        binding.bmiTextView.text,
        binding.kategoriTextView.text
    )

    val shareIntent = Intent(Intent.ACTION_SEND)
    shareIntent.setType("text/plain").putExtra(Intent.EXTRA_TEXT, message)
    if (shareIntent.resolveActivity(
        requireActivity().packageManager) != null) {
        startActivity(shareIntent)
    }
}
```

Terakhir, ubah method `showResult` pada `HitungFragment.kt` agar tombol Bagikan menjadi terlihat.

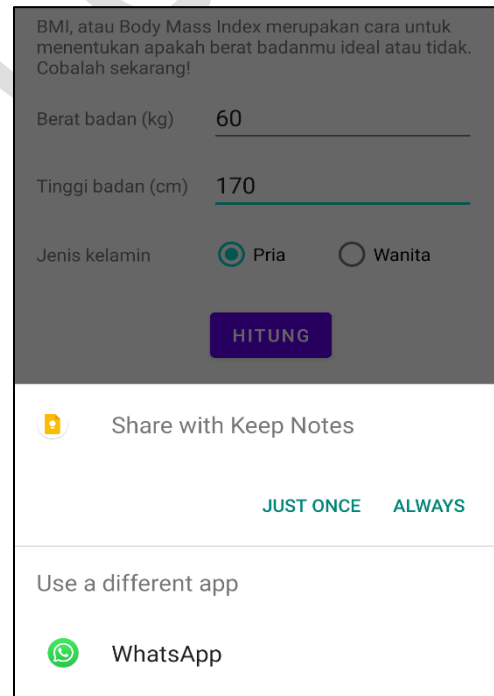
```
private fun showResult(result: HasilBmi?) {
    if (result == null) return

    binding.bmiTextView.text = getString(R.string.bmi_x, result.bmi)
    binding.kategoriTextView.text = getString(R.string.kategori_x,
        getKategoriLabel(result.kategori))
    binding.saranButton.visibility = View.VISIBLE
    binding.buttonGroup.visibility = View.VISIBLE
}
```

Aksi untuk melakukan navigasi ke app lain terlihat pada method `shareData`. Pada method ini, informasi dari masukan pengguna akan diambil dan dimasukkan ke dalam Intent. Intent yang digunakan merupakan intent implisit dengan action `ACTION_SEND`. Intent implisit akan memberikan pilihan pada user untuk memilih aplikasi apa yang dijalankan untuk intent tersebut.

Jalankan aplikasi, masukkan tinggi, berat, jenis kelamin, dan tekan tombol hitung. Fokus kita pada task ini adalah tombol bagikan. Jika tombol ini diklik, maka akan muncul pilihan untuk share melalui aplikasi lain seperti gambar di samping. Jika aplikasi telah berjalan sesuai dengan harapan, lakukan commit dengan isi commit message berupa judul dari task 3.7 ini.

Praktikum telah usai. Selanjutnya pastikan commit yang dilakukan sudah sesuai jumlah task. Berikutnya, lakukan push kode project ke repository Github menggunakan menu `Git > Push...` Refresh repository di browser dan pastikan kodenya sudah sesuai dengan kode versi lokal (komputer/laptop masing-masing). Semangat!!



## 4. Summary

Pada modul kali ini kita telah membuat sebuah aplikasi Android yang menampilkan beberapa tampilan kepada pengguna menggunakan pattern single activity – multiple fragment. Selain itu, kita juga telah melakukan pengiriman data argument antar layar fragment pada aplikasi dan juga melakukan navigasi ke app lain dengan menggunakan intent implisit.

## 5. Challenge

Pada saran fragment, hal yang ditampilkan adalah gambar dan juga saran untuk masing-masing kategori. Terkadang, user perlu diingatkan input yang telah mereka berikan. Untuk itu, tampilkan masukan dari pengguna yang diberikan melalui HitungFragment untuk dimunculkan juga pada SaranFragment. Tambahkan juga menu share pada fragment yang memperlihatkan saran.