

MODUL PRAKTIKUM PEMROGRAMAN UNTUK PERANGKAT BERGERAK 1

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

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

Cahyana S.T., M.Kom.



D3 Rekayasa Perangkat Lunak Aplikasi
Telkom University

Daftar Isi

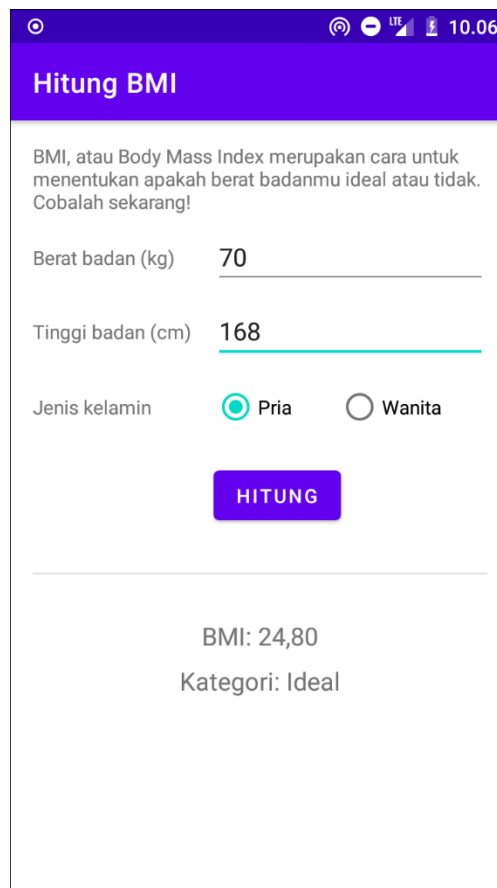
| | |
|--|----|
| Modul 03: Layouts | 2 |
| 1. Overview | 2 |
| 2. Getting Started..... | 3 |
| 3. Task | 3 |
| 3.1. Membuat form perhitungan BMI | 3 |
| 3.2. Menambahkan view binding..... | 10 |
| 3.3. Membuat perhitungan BMI | 11 |
| 3.4. Debugging error dan perbaikannya | 14 |
| 3.5. Menangani tampilan di layar kecil | 16 |
| 4. Summary | 18 |
| 5. Challenge..... | 18 |

Modul 03: Layouts

1. Overview

Pada modul kali ini, kita akan membuat sebuah aplikasi sederhana bernama “Hitung BMI”. Aplikasi ini menghitung BMI atau body mass index seseorang berdasarkan berat, tinggi dan jenis kelaminnya. Selain itu, ditampilkan juga kategori dari berat tubuh berdasarkan nilai BMI tersebut, apakah ideal, gemuk atau kurus. Aplikasi yang kita buat dapat dilihat versi lengkapnya di repository Github ini: <https://github.com/indraazimi/hitung-bmi>

Setelah mempraktekkan modul ini, diharapkan kita mengetahui bagaimana membuat tampilan menggunakan ConstraintLayout, meminta input dari pengguna dan memprosesnya. Berikut adalah tampilan akhir dari aplikasi yang akan dibuat.



The screenshot shows a mobile application titled "Hitung BMI" with a purple header. Below the header, there is a descriptive text: "BMI, atau Body Mass Index merupakan cara untuk menentukan apakah berat badanmu ideal atau tidak. Cobalah sekarang!". The form contains three input fields: "Berat badan (kg)" with the value "70", "Tinggi badan (cm)" with the value "168", and "Jenis kelamin" with radio buttons for "Pria" (selected) and "Wanita". A purple "HITUNG" button is positioned below the inputs. A horizontal line separates the input section from the result section, which displays "BMI: 24,80" and "Kategori: Ideal". The status bar at the top shows the time as 10.06.

2. Getting Started

Pertama, jalankan Android Studio, lalu buat project baru bernama “Hitung BMI”. Tuliskan “org.d3ifxxx.hitungbmi” sebagai nama package. Pilih minimum SDK 19, dengan Empty Activity sebagai template awal. Silahkan baca kembali modul pertama jika diperlukan. Berikutnya, kita akan menambahkan beberapa komponen pada project Android Studio ini. Silahkan ikuti task di modul ini langkah per langkah.

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

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

3. Task

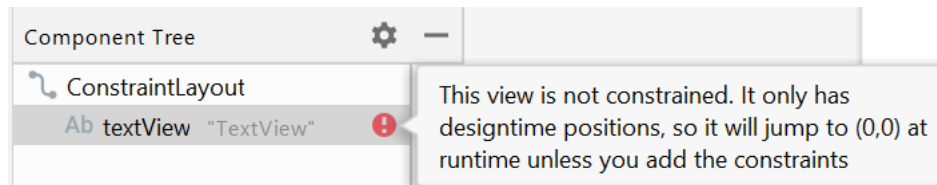
3.1. Membuat form perhitungan BMI

Task pertama yang kita lakukan adalah membuat tampilan form untuk menghitung BMI dengan mengatur layout activity_main.xml. Kita mulai dari canvas yang kosong untuk memahami bagaimana komponen ditempatkan pada ConstraintLayout. Jadi, hapus TextView yang terbentuk secara default di bagian canvas. Dan, pastikan ukuran lebar juga tinggi dari ConstraintLayout sudah bernilai “match_parent”.

Berikutnya, tambahkan TextView baru dari Pallette ke Canvas. Komponen yang baru ditambahkan ini akan memiliki 4 buah titik di sekitarnya yang menyatakan “constraint” atau batasan penempatan bagi komponen tersebut. Pengaturan constraint ini perlu diperhatikan karena jika tidak diberikan, ketika aplikasi dijalankan, semua komponen akan berkumpul di pojok kiri atas dan tidak menempati tempat yang seharusnya.



Jika constraint tidak diberikan dengan baik seperti pada gambar di atas, di bagian Component Tree akan terdapat simbol error dengan warna merah yang menyatakan “This view is not constrained...”. Namun jika komponen telah diberikan constraint dengan benar, maka simbol ini tidak muncul di bagian Component Tree. Sebuah komponen minimal harus diberi satu constraint horizontal dan juga satu constraint vertikal.



Dan secara kode, hal yang tertulis adalah:

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="TextView"
    tools:layout_editor_absoluteX="94dp"
    tools:layout_editor_absoluteY="94dp" />
```

Perhatikan 2 baris terakhir pada kode XML di atas. Terdapat “tools:layout_editor_absoluteX” dan “tools:layout_editor_absoluteY”. Hal ini berarti, komponen text ini diatur penempatannya pada titik 94dp ke kanan dan bawah dari titik pojok kiri atas. Akan tetapi, di sini terdapat namespace “tools”. Properties yang dikenai namespace “tools” ini, hanya berefek pada saat mode preview, sedangkan pada saat run, tidak berpengaruh.

Sehingga, walaupun pada saat di-preview di Android Studio, posisi sudah ditempatkan dengan cara digeser peletakannya di bagian design, pada saat dijalankan di smartphone/emulator, komponen akan muncul di pojok kiri atas. Untuk mencegah agar ini tidak terjadi, terhadap TextView yang baru saja ditambahkan, atur constraint dengan cara menghubungkan bulatan pada komponen ke tepi layar seperti pada gambar berikut.



Terdapat tiga constraint yang harus ditambahkan (atas, kiri dan kanan). Setelah menambahkan constraint, simbol error di Component Tree akan berubah menjadi simbol warning. Warning muncul karena diperlukan nilai text yang diambil dari strings.xml, bukan diketik langsung (lihat modul sebelumnya terkait string yang ditampilkan ke layar). Berikut kode yang terbentuk untuk komponen TextView yang telah diatur constraint-nya.

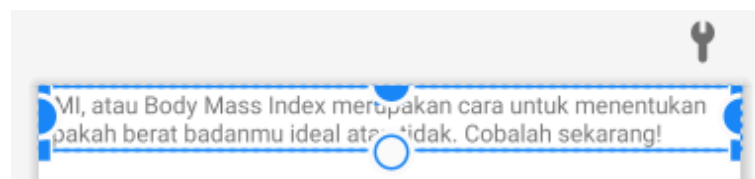
```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="TextView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

layout_constraintTop_toTopOf merupakan constraint panah atas, layout_constraintStart_toStartOf sebagai constraint yang diperlihatkan dalam panah kiri dan layout_constraintEnd_toEndOf merupakan constraint panah kanan,. Ketiga panah ini menuju tepi layar yang diacu menggunakan "parent" pada bagian layout. Jika dihubungkan dengan komponen lain, maka value dari constraint ini akan bernilai "id" dari komponen lain tersebut.

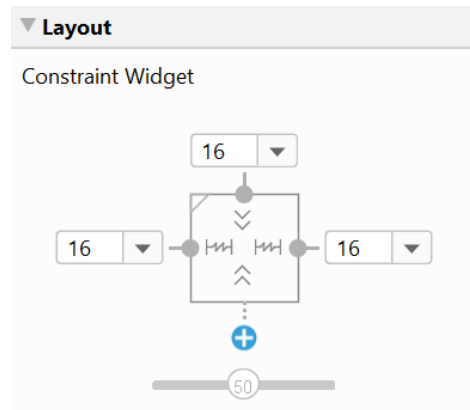
Lanjutkan penyesuaian untuk komponen TextView ini dengan mengubah nilai properties berikut:

| Property | Nilai | Penjelasan |
|---------------|--------------------------|--|
| layout_width | Odp | Lebar komponen menyesuaikan ukuran layar |
| layout_height | wrap_content | Panjang komponen mengikuti ukuran text yang tertulis di dalamnya |
| text | Diambil dari strings.xml | Menampilkan teks <i>"BMI, atau Body Mass Index merupakan cara untuk menentukan apakah berat badanmu ideal atau tidak. Cobalah sekarang!"</i> |

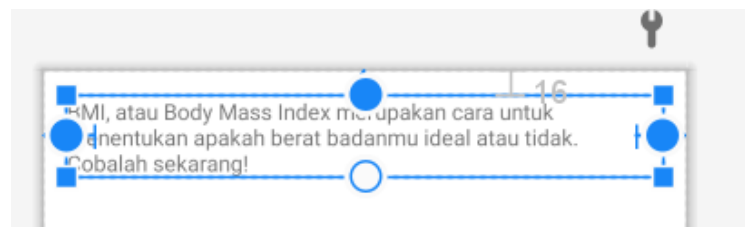
Berikut adalah tampilan yang didapatkan.



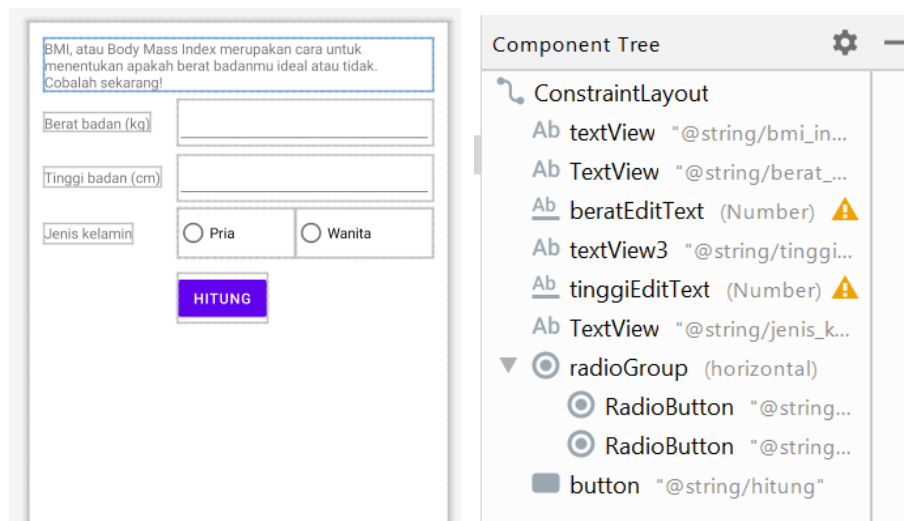
Terlihat bahwa komponen tidak memiliki jarak terhadap tepi layar, baik atas, kiri, ataupun kanan. Hal ini bukan tampilan yang baik, sehingga kita harus memberikan jaraknya. Untuk tampilan handphone, gunakan jarak yang seragam, sebanyak 16dp. Pengaturan ini, dapat dilakukan dengan mengatur di bagian constraint widget.



Berikut tampilan yang terbentuk:



Kita telah menampilkan satu komponen yang terdapat pada ConstraintLayout. Berikutnya, aturlah komponen-komponen dengan melakukan drag-n-drop, dan pengaturan constraint dengan jarak antar komponen sebesar 8dp sehingga didapat tampilan akhir sebagai berikut:



Agar tampilan TextView (komponen text) dan EditText (komponen input) ditampilkan bersisian dengan align center secara horizontal, atur constraint sebagai berikut:



Hal lain yang perlu diperhatikan juga adalah property “inputType” dari EditText. Berikan tipe number. Hal ini membatasi hanya karakter numerik (number) yang dapat diberikan sebagai input. Selain itu, perhatikan bahwa komponen RadioButton berada pada hirarki setelah Radio Group. Hal ini perlu dilakukan agar radio button dapat dipilih salah satu saja. Fungsi Radio Group di sini sama dengan komponen Button Group pada Java Swing.

Hasil akhir dari kode XML pada bagian layout dan string adalah sebagai 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">

    <TextView
        android:id="@+id/textView"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:text="@string/bmi_intro"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:labelFor="@id/beratEditText"
        android:text="@string/berat_badan"
        app:layout_constraintBottom_toBottomOf="@+id/beratEditText"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/beratEditText" />
```



```

<EditText
    android:id="@+id/beratEditText"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="16dp"
    android:ems="10"
    android:inputType="number"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/textView3"
    app:layout_constraintTop_toBottomOf="@+id/textView" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:labelFor="@id/tinggiEditText"
    android:text="@string/tinggi_badan"
    app:layout_constraintBottom_toBottomOf="@+id/tinggiEditText"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/tinggiEditText" />

<EditText
    android:id="@+id/tinggiEditText"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="16dp"
    android:ems="10"
    android:inputType="number"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/beratEditText"
    app:layout_constraintTop_toBottomOf="@+id/beratEditText" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:text="@string/jenis_kelamin"
    app:layout_constraintBottom_toBottomOf="@+id/radioGroup"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/radioGroup" />

```

```

<RadioGroup
    android:id="@+id/radioGroup"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="16dp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/tinggiEditText"
    app:layout_constraintTop_toBottomOf="@+id/tinggiEditText">

    <RadioButton
        android:id="@+id/priaRadioButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/pria" />

    <RadioButton
        android:id="@+id/wanitaRadioButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/wanita" />

</RadioGroup>

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="@string/hitung"
    app:layout_constraintStart_toStartOf="@+id/radioGroup"
    app:layout_constraintTop_toBottomOf="@+id/radioGroup" />

</androidx.constraintlayout.widget.ConstraintLayout>

<resources>
    <string name="app_name">Hitung BMI</string>

    <string name="bmi_intro">BMI, atau Body Mass Index merupakan cara untuk
    menentukan apakah berat badanmu ideal atau tidak. Cobalah sekarang!</string>
    <string name="berat_badan">Berat badan (kg)</string>
    <string name="tinggi_badan">Tinggi badan (cm)</string>
    <string name="jenis_kelamin">Jenis kelamin</string>
    <string name="pria">Pria</string>
    <string name="wanita">Wanita</string>
    <string name="hitung">Hitung</string>
</resources>

```

Jalankan aplikasi, dan pastikan bahwa layout dapat ditampilkan dengan baik.

3.2. Menambahkan view binding

Pada task kali ini, kita akan menambahkan aksi pada komponen Button yang telah ditambahkan pada task sebelumnya. Berbeda dengan modul sebelumnya yang menggunakan `findViewById` pada komponennya, di modul kali ini kita menggunakan view binding. View binding memudahkan kita untuk berinteraksi terhadap objek komponen di layout XML pada kode Kotlin secara lebih efisien. Untuk menggunakan fitur ini, tambahkan kode berikut pada `app/build.gradle`.

```
defaultConfig {
    ...

    testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
}

buildFeatures {
    viewBinding = true
}

buildTypes {
    ...
}
```

Setelah melakukan perubahan kode, terdapat pilihan “sync now” di kanan atas. Klik pilihan tersebut. Tunggu loading selesai. Berikutnya, ubah kode `MainActivity.kt` sebagai berikut:

```
class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding

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

        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.button.setOnClickListener { hitungBmi() }
    }

    private fun hitungBmi() {
        Log.d("MainActivity", "Tombol diklik!")
    }
}
```

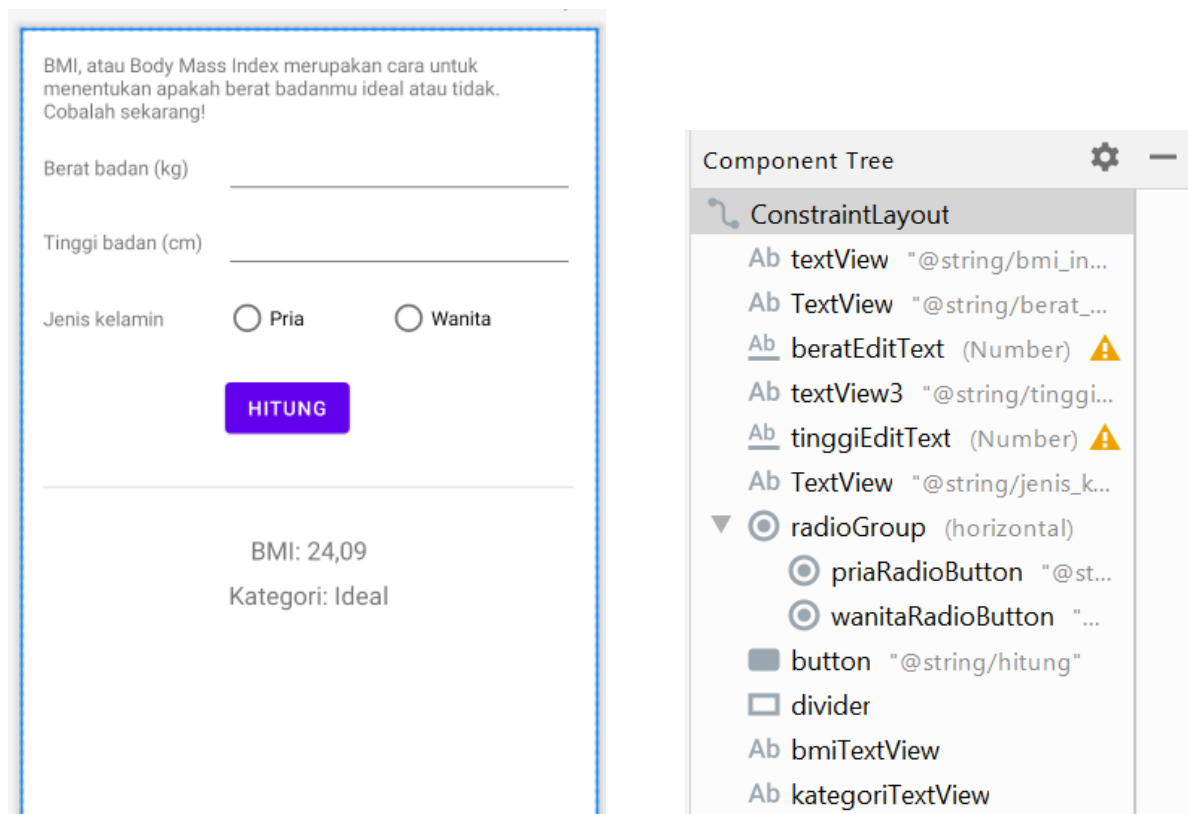
Import bagian `ActivityMainBinding` dan `Log` pada kode di atas.

Pada kode di atas, setContentView dengan layout secara langsung digantikan dengan setContentView melalui binding. Variabel binding memiliki nama sesuai nama layout yang dimiliki. Melalui variable, kita dapat mengakses objek dari komponen yang ada di XML berupa button, dan menambahkan kode setOnClickListener terhadapnya (menambahkan aksi pada button). Sehingga, tidak diperlukan lagi findViewById seperti yang dilakukan pada modul minggu sebelumnya.

Jalankan aplikasi, pastikan komponen muncul dengan baik. Saat menekan tombol, maka pada Logcat akan tampil "Tombol diklik!". Jika diperlukan, cek modul sebelumnya tentang menampilkan output di logcat.

3.3. Membuat perhitungan BMI

Pada task kali ini kita akan mengubah aksi tombol klik yang awalnya menampilkan "Tombol diklik!" di logcat menjadi melakukan perhitungan BMI. Sebelum menambahkan aksi, ubah tampilan di layout activity_main.xml dengan menambahkan sebuah divider dan dua buah komponen TextView untuk menampilkan nilai BMI dan kategori dari BMI-nya. Hasil akhir dari tampilan adalah sebagai berikut:



Kode akhir dari activity_main.xml dituliskan sebagai berikut:

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

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="@string/hitung"
        app:layout_constraintStart_toStartOf="@+id/radioGroup"
        app:layout_constraintTop_toBottomOf="@+id/radioGroup" />

    <View
        android:id="@+id/divider"
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="32dp"
        android:layout_marginEnd="16dp"
        android:background="?android:attr/listDivider"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button" />

    <TextView
        android:id="@+id/bmiTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:textSize="18sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/divider"
        tools:text="BMI: 24,09" />

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

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

Kemudian, kita juga perlu melakukan perubahan pada strings.xml dikarenakan akan digunakan di kode Kotlin sebagai aksi untuk menampilkan hasil perhitungan BMI juga penentuan kategori.

```
<resources>
    ...
    <string name="hitung">Hitung</string>
    <string name="bmi_x">BMI: %1$.2f</string>
    <string name="kurus">Kurus</string>
    <string name="ideal">Ideal</string>
    <string name="gemuk">Gemuk</string>
    <string name="kategori_x">Kategori: %1$s</string>
</resources>
```

Sebelum menuliskan kode Kotlin, kita harus tahu bahwa BMI dihitung dengan cara berat badan dibagi tinggi badan kuadrat seperti rumus di samping. Berat badan harus dalam kilogram dan tinggi badan harus dalam meter. BMI yang telah dihitung selanjutnya dikategorikan menjadi kurus, ideal atau gemuk dengan tabel berikut ini.

$$BMI = \frac{BB}{TB^2}$$

| Pria | |
|---------------|----------|
| BMI | Kategori |
| < 20,50 | Kurus |
| 20,50 – 26,99 | Ideal |
| >= 27,00 | Gemuk |

| Wanita | |
|---------------|----------|
| BMI | Kategori |
| < 18,50 | Kurus |
| 18,50 – 24,99 | Ideal |
| >= 25,00 | Gemuk |

Setelah memahami rumus dan pengkategorianya, ubah kode MainActivity.kt sebagai berikut:

```
class MainActivity : AppCompatActivity() {
    ...

    private fun hitungBmi() {
        Log.d("MainActivity", "Tombol diklik!")
        val berat = binding.beratEditText.text.toString().toFloat()
        val tinggi = binding.tinggiEditText.text.toString().toFloat() / 100
        val bmi = berat / (tinggi * tinggi)

        val selectedId = binding.radioGroup.checkedRadioButtonId
        val isMale = selectedId == R.id.priaRadioButton
        val kategori = getKategori(bmi, isMale)

        binding.bmiTextView.text = getString(R.string.bmi_x, bmi)
        binding.kategoriTextView.text = getString(R.string.kategori_x, kategori)
    }
}
```

```

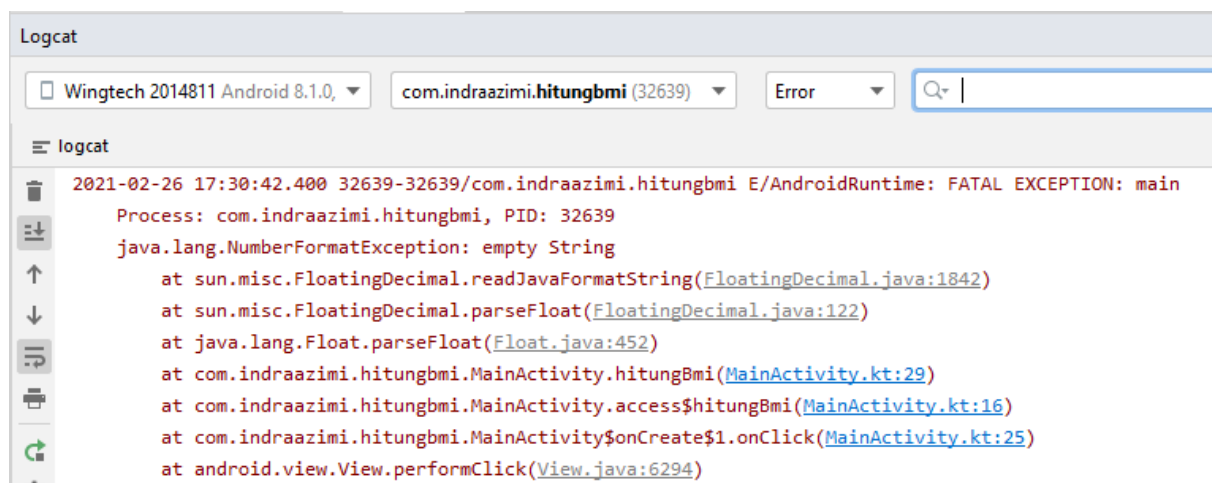
private fun getKategori(bmi: Float, isMale: Boolean): String {
    val stringRes = if (isMale) {
        when {
            bmi < 20.5 -> R.string.kurus
            bmi >= 27.0 -> R.string.gemuk
            else -> R.string.ideal
        }
    } else {
        when {
            bmi < 18.5 -> R.string.kurus
            bmi >= 25.0 -> R.string.gemuk
            else -> R.string.ideal
        }
    }
    return getString(stringRes)
}
}

```

Terdapat perubahan pada method hitungBMI, dan terdapat penambahan method getKategori yang digunakan pada method hitungBMI. Pada kode di atas, dapat dilihat bahwa terdapat pengambilan nilai string dari strings.xml menggunakan getString, dan mengakses name resource dari strings.xml. Jalankan aplikasi, maka setelah memasukkan input yang diperlukan, akan ditampilkan nilai BMI dan juga kategorinya.

3.4. Debugging error dan perbaikannya

Pada task sebelumnya, kita telah membuat aplikasi untuk menampilkan BMI dan kategori dari input yang diberikan pengguna. Akan tetapi, jika terdapat input yang kosong, aplikasi akan crash.



Karena pesan errornya “empty String”, ubah kode pada method hitungBmi di MainActivity.kt agar melakukan pengecekan apakah input kosong atau tidak, sebelum melakukan konversi input dari tipe String → Float, atau melakukan perhitungan atau penentuan berikutnya. Selain itu, lakukan juga pengecekan apakah radio button sudah terpilih salah satunya. Proses pengecekan input pengguna ini dinamakan “sanity check”.

```
private fun hitungBmi() {  
    val berat = binding.beratEditText.text.toString().toFloat()  
    val tinggi = binding.tinggiEditText.text.toString().toFloat() / 100  
    val bmi = berat / (tinggi * tinggi)  
  
    val berat = binding.beratEditText.text.toString()  
    if (TextUtils.isEmpty(berat)) {  
        Toast.makeText(this, R.string.berat_invalid, Toast.LENGTH_LONG).show()  
        return  
    }  
  
    val tinggi = binding.tinggiEditText.text.toString()  
    if (TextUtils.isEmpty(tinggi)) {  
        Toast.makeText(this, R.string.tinggi_invalid, Toast.LENGTH_LONG).show()  
        return  
    }  
    val tinggiCm = tinggi.toFloat() / 100  
  
    val selectedId = binding.radioGroup.checkedRadioButtonId  
    if (selectedId == -1) {  
        Toast.makeText(this, R.string.gender_invalid, Toast.LENGTH_LONG).show()  
        return  
    }  
    val isMale = selectedId == R.id.priaRadioButton  
    val bmi = berat.toFloat() / (tinggiCm * tinggiCm)  
    val kategori = getKategori(bmi, isMale)  
  
    binding.bmiTextView.text = getString(R.string.bmi_x, bmi)  
    binding.kategoriTextView.text = getString(R.string.kategori_x, kategori)  
}
```

Import bagian TextUtils dan Toast agar kode tidak lagi dinyatakan error oleh compiler. TextUtils digunakan untuk mengecek apakah input kosong atau tidak, sedangkan Toast digunakan untuk memberi pesan ke pengguna. Jangan lupa tambahkan juga kode strings.xml berikut:

```
<resources>  
    ...  
    <string name="berat_invalid">Berat tidak boleh kosong.</string>  
    <string name="tinggi_invalid">Tinggi tidak boleh kosong.</string>  
    <string name="gender_invalid">Pilih jenis kelamin.</string>  
</resources>
```


Jalankan aplikasi kembali, pastikan penanganan input telah dilakukan dengan baik. Selain itu, perhitungan juga tetap berfungsi seperti sebelumnya.

3.5. Menangani tampilan di layar kecil

Pada task ini kita akan menangani tampilan aplikasi di layar yang lebih kecil. Contohnya, ketika smartphone/emulator diubah orientasinya dari portrait menjadi landscape. Diperlukan penyesuaian dari segi tampilan, yaitu penambahan ScrollView dan juga pengacuan nilai `marginStart`/`marginEnd` diambil dari `dimens.xml`. Tambahkan nilai `dimens.xml` (bisa dilakukan seperti penambahan string resource) atau ketik manual sebagai berikut:

```
<resources>
    <dimen name="horizontal_margin">16dp</dimen>
</resources>
```

Berikutnya, ubah `activity_main.xml` sebagai berikut:

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

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

        <TextView
            android:id="@+id/textView"
            ...
            android:layout_marginStart="16dp"
            android:layout_marginStart="@dimen/horizontal_margin"
            android:layout_marginTop="16dp"
            android:layout_marginEnd="16dp"
            android:layout_marginEnd="@dimen/horizontal_margin"
            ... />

        <TextView
            ...
            android:layout_marginStart="16dp"
            android:layout_marginStart="@dimen/horizontal_margin"
            ... />
```

```
<EditText
    android:id="@+id/beratEditText"
    ...
android:layout_marginStart="16dp"
    android:layout_marginStart="@dimen/horizontal_margin"
    android:layout_marginTop="8dp"
android:layout_marginEnd="16dp"
    android:layout_marginEnd="@dimen/horizontal_margin"
    ... />

<TextView
    android:id="@+id/textView3"
    ...
android:layout_marginStart="16dp"
    android:layout_marginStart="@dimen/horizontal_margin"
    ... />

<EditText
    android:id="@+id/tinggiEditText"
    ...
    android:layout_marginTop="8dp"
android:layout_marginEnd="16dp"
    android:layout_marginEnd="@dimen/horizontal_margin"
    ... />

<TextView
    ...
android:layout_marginStart="16dp"
    android:layout_marginStart="@dimen/horizontal_margin"
    ... />

<RadioGroup
    android:id="@+id/radioGroup"
    ...
    android:layout_marginTop="8dp"
android:layout_marginEnd="16dp"
    android:layout_marginEnd="@dimen/horizontal_margin"
    ...>
    ...
</RadioGroup>

<Button
    android:id="@+id/button"
    ... />

<View
    android:id="@+id/divider"
    ...
android:layout_marginStart="16dp"
    android:layout_marginStart="@dimen/horizontal_margin"
    android:layout_marginTop="32dp"
android:layout_marginEnd="16dp"
    android:layout_marginEnd="@dimen/horizontal_margin"
    ... />
```

```

        <TextView
            android:id="@+id/bmiTextView"
            ... />

        <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" />

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

```

Jalankan aplikasi, dan lakukan perubahan orientasi smartphone/emulator. Pastikan tampilan bisa di-scroll untuk melihat hasil perhitungan BMI dan kategorinya.

4. Summary

Pada modul kali ini kita telah membuat sebuah aplikasi Android menggunakan beberapa komponen yaitu: ConstraintLayout, TextView, EditText, RadioGroup dan juga Button. Pada modul ini kita telah mengetahui bagaimana sebuah komponen diatur peletakannya pada sebuah ConstraintLayout. Selain itu, kita juga telah menggunakan view binding. Sehingga, objek dari komponen layout dapat diakses melalui binding-nya. Hal lain yang sudah dipelajari di modul ini adalah bagaimana memproses input dari pengguna dan memberikan tampilan yang dapat di-scroll di layar kecil.

5. Challenge

Pada kode yang telah dibangun sebelumnya, kita telah membuat button dengan aksi untuk menghitung nilai BMI dan menampilkan kategorinya. Sebagai tantangan, buatlah sebuah button tambahan untuk melakukan reset, agar aplikasi kembali ke tampilan seperti ketika baru dijalankan.