



LAPORAN TUGAS 1

**Disusun guna memenuhi tugas mata kuliah
Pemrograman Aplikasi Perangkat Bergerak**

Dosen Pengampu:
Fais Al Huda, S.Kom., M.Kom.

Oleh:

Pieter Christy Yan Yudhistira

235150201111068

**KELAS TIF-PAPB-D
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
TAHUN 2025**

BAB I

PENDAHULUAN

1.1 Pembahasan Tugas 1

Dalam laporan ini, saya membuat aplikasi sederhana yang mengintegrasikan tombol yang dapat menampilkan “Halo [Nama Mahasiswa]” ketika adanya interaksi pengguna (klik tombol). Instruksi yang diberikan diantaranya:

1. Membuat tombol dengan Button
2. Menampilkan teks dengan Text.
3. Menggunakan state (misalnya mutablestateof) untuk mengelola perubahan
4. Menghubungkan interaksi pengguna (klik tombol) dengan perubahan tampilan secara langsung.
5. Buat hasilkan dalam bentuk laporan sederhana yang berisi hasil run aplikasi beserta link github
6. Kumpulkan dalam bentuk pdf

Untuk link github dari hasil pengerjaan saya, saya lampirkan disini:

1. <https://github.com/Pieter414/papb/tree/master/Tugas1> (Github Tugas 1)
2. <https://github.com/Pieter414/papb/blob/master/Tugas1/app/src/main/java/com/example/tugas1/MainActivity.kt> (Github Tugas 1: File Kotlin)

1.2 Transparansi Sistem

Dalam eksperimen aplikasi sederhana, saya menggunakan Android Studio 2025 1.3.7 menggunakan Komputer Laboratorium Sistem Cerdas. Untuk infrastruktur yang saya gunakan untuk aplikasi sederhana adalah Empty Activity Project dengan minimum SDK API 24 Nougat Android 7.0 dan konfigurasi Kotlin DSL.

BAB II

SOURCE CODE DAN PEMBAHASAN

2.1 Source Code

Dalam aplikasi sederhana yang saya buat, saya memanfaatkan dari *source code* yang tersedia dari template “Empty Activity” dan melakukan modifikasi sesuai dengan instruksi Tugas 1.

Source Code 1. Import libraries dan package

```
package com.example.halostudent

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge

// import compose function
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.Surface
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment

import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.tugas1.ui.theme.Tugas1Theme
```

Dalam blok kode import tersebut, saya menggunakan libraries dan package yang sudah ada dari template seperti libraries androidx os, activity, compose.ui untuk Preview dan Modifier. Namun, saya juga menambahkan package yang digunakan seperti Button dari compose.material3, Column dan Arrangement dari compose foundation layout dan lainnya. Fungsi ini dibutuhkan mulai dari kebutuhan instruksi (Button, Text dan mutablestateof membutuhkan import) hingga estetika semata (menambahkan fungsi padding).

Source Code 2. Kelas/Class MainActivity

```
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            Tugas1Theme {
                Surface(
                    modifier = Modifier.fillMaxSize()
                )
            }
        }
    }
}
```

```

        ) {
            GreetingApp()
        }
    }
}
}
}

```

Dalam source code, saya mengikuti template Empty Activity dengan menambahkan class MainActivity sebagai titik masuk aplikasi. Dalam class ini, menggunakan onCreate function yang di-override sebagai fungsi pertama kali dan memastikan inisialisasi ComponentActivity tetap terpanggil. Fungsi enableEdgeToEdge() untuk mengatur agar tampilan aplikasi bisa muncul sampai ke tepi layar, ini juga termasuk area status bar dan navigation bar.

Lalu dalam setContent, digunakan untuk mendeklarasikan UI aplikasi. Kemudian, dalam Surface saya gunakan dalam bentuk wadah. Surface sendiri saya gunakan karena bisa mendukung tampilan lebih modern dibandingkan Scaffold, dengan konsekuensi padding secara manual. Kemudian untuk inisialisasi, saya memastikan modifier dari Surface memenuhi seluruh layar dengan fungsi fillMaxSize(). Untuk inisialisasi App, saya memanggil fungsi yang telah saya buat dibawah ini yang merupakan @Composable

Source Code 3 Fungsi GreetingApp()

```

@Preview()
@Composable
fun GreetingApp() {
    var text by remember { mutableStateOf("") }
    val namaMahasiswa = "Pieter Christy Yan Yudhistira"

    Column(
        modifier = Modifier.fillMaxSize().padding(16.dp),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Button(
            onClick = {
                text = if (text.isEmpty()) {
                    "Halo $namaMahasiswa"
                } else {
                    ""
                }
            },
            modifier = Modifier.padding(16.dp)
        ) {
            Text("Click Here")
        }

        if (text.isNotEmpty()) {
            Text(
                text = text,
            )
        }
    }
}

```

```
        style = MaterialTheme.typography.headlineSmall,  
    )  
}  
}
```

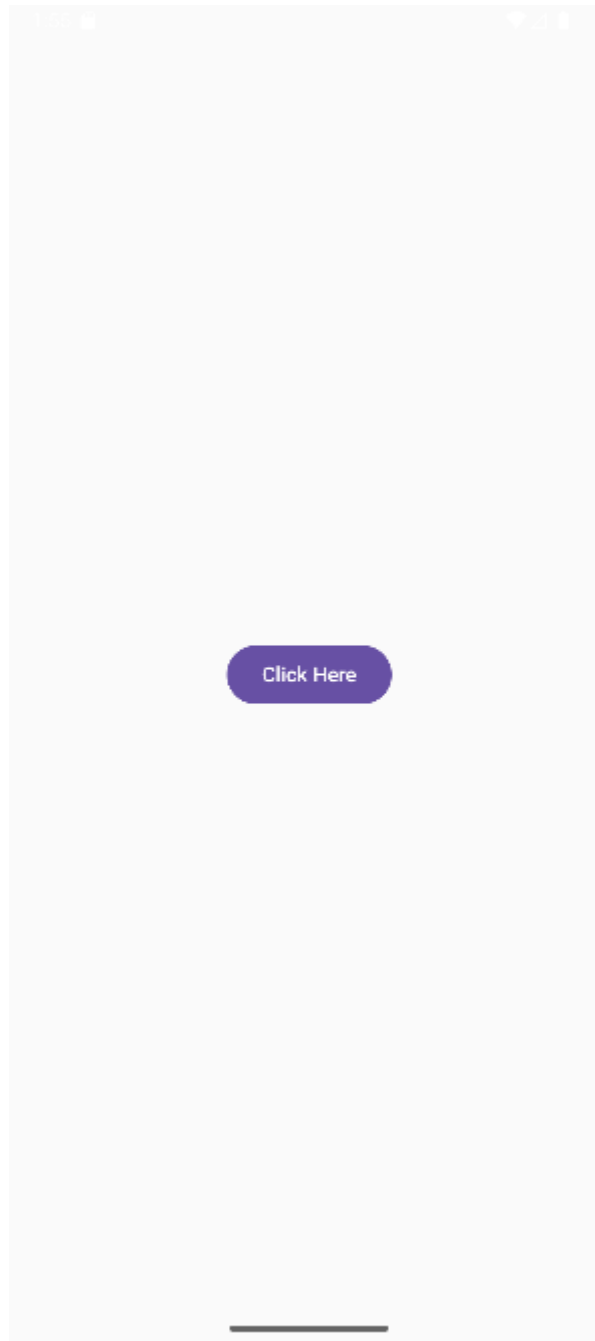
Dalam source code ini, saya mengembangkan sebagai App untuk melakukan Greeting. Pertama, saya menggunakan dua fungsi `@Composable` dan `@Preview`. `@Composable` digunakan untuk membuat UI di Jetpack Compose dan `@Preview` digunakan sebagai preview compose di Android Studio dan dapat ditampilkan. Kemudian deklarasi variabel text dan namaMahasiswa (yaitu saya), dimana text saya pastikan memiliki dapat diubah-ubah dengan `mutableStateOf` dengan default "". Kemudian saya mendeklarasikan komponen secara vertikal, dimana memastikan Column memenuhi seluruh layar dengan `fillMaxSize()` dan padding 16dp. Lalu saya juga menambahkan `verticalArrangement.Center` & `horizontalAlignment.CenterHorizontally` untuk memastikan semua button dan text di tengah layar.

Kemudian dari Column ini, dimana membangun item yang ingin ditambahkan dari instruksi. Pertama adalah tombol dengan kelas Button dengan fungsi `onClick` (memastikan interaksi dengan ditekan) dengan tampilan tombol "Click Here". Dalam deklarasi text di `onClick`, saya menggunakan fungsi `if` dimana memastikan tombol interaktif. Contohnya ketika pertama kali menekan tombol, "Halo Pieter Christy Yan Yudhistira" akan muncul. Lalu, menekan tombol berikutnya akan menghilangkan tulisan text greeting tersebut. Kemudian, memastikan style dari teks dimana menggunakan `headlineSmall` memastikan gaya teks lebih kecil.

BAB III

HASIL

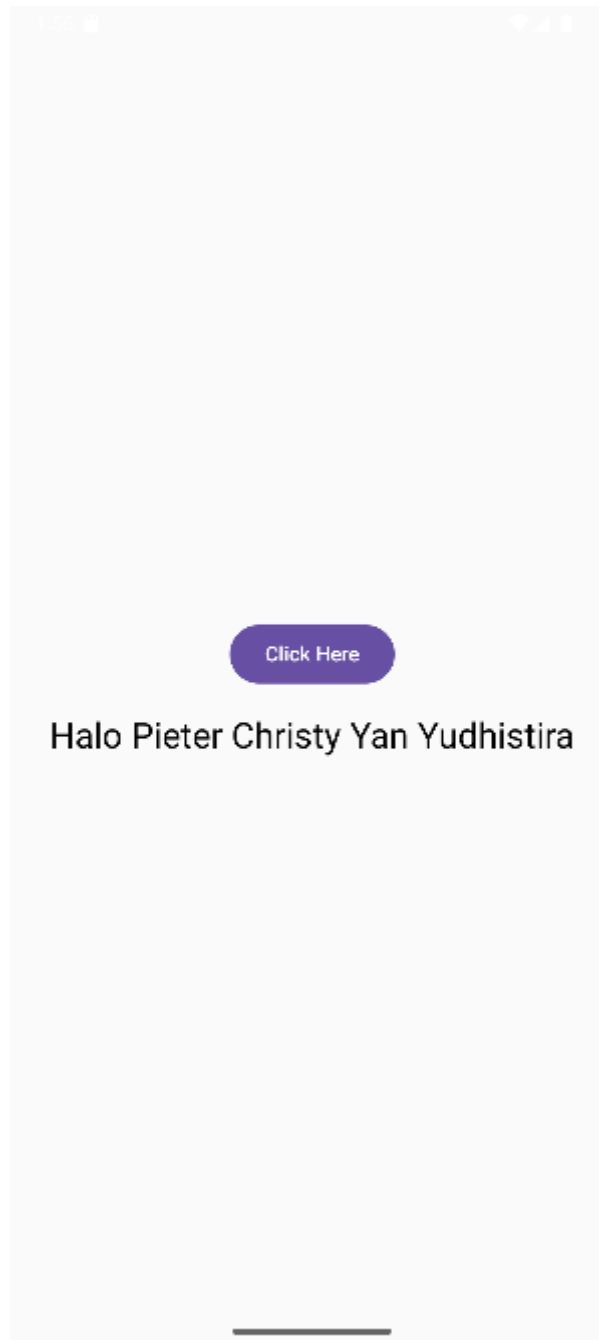
3.1 Hasil Aplikasi dari Android Studio Emulation



Gambar 1. Tampilan Awal Aplikasi

Dalam tampilan Android Studio Emulation menggunakan minimum SDK API 24, berikut tampilan aplikasinya. Pertama-tama terdapat button ungu di tengah aplikasi yang bertujuan sebagai tombol “Click Here” untuk menampilkan kata ‘Halo [Nama Mahasiswa]’.

Dalam aplikasi ini, jika ada yang menekan tombol maka tampilan aplikasi akan sesuai dengan gambaran 2.



Gambar 2. Tampilan Aplikasi setelah Menekan Tombol

Berikut adalah tampilan aplikasi yang menunjukkan sebuah teks “Halo Pieter Christy Yan Yudhistira” dimana nama tersebut adalah nama saya yang digunakan dalam aplikasi ini. Jika saya menekan tombol sekali lagi, maka tampilan aplikasi akan sesuai dengan gambar sebelumnya dan tulisan akan hilang. Sehingga tombol bisa interaktif menampilkan atau tidak menampilkan teks.

DAFTAR PUSTAKA

freeCodeCamp.org. (2021). Android App Development Tutorial for Beginners - Your First App.
<https://youtu.be/FjrKMcnKahY?feature=shared>