LAPORAN PRAKTIKUM PENGEMBANGAN APLIKASI BERGERAK

Android Fundamental 5 - WEEK 11

Disusun oleh:

Nama : Afif Imam Rahadi

Nim : L0122006

Kelas : A

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA

UNIVERSITAS SEBELAS MARET

2024

1. **Screenshot Source Code**

   Source code ini merupakan source code yang digunakan untuk praktikum ke-11. Terdapat beberapa SC yang nantinya akan diperlihatkan dan dijelaskan dalam laporan ini.

Main Activity,kt

```
package com.l0122006.afifimam.week11

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import androidx.core.view.ViewCompat
import androidx.core.view.WindowCompat
import androidx.core.view.WindowInsetsCompat

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets ^setOnApplyWindowInsetsListener
        }
        window.statusBarColor = getColor(android.R.color.transparent)
        val windowInsetsController = WindowCompat.getInsetsController(window, window.decorView)
        windowInsetsController.isAppearanceLightStatusBars = true
    }

    @Suppress( ...names: "UNUSED_PARAMETER")
    fun toCalculator(view: View) {
        val intent = Intent( packageContext: this, CalculatorActivity::class.java)
        startActivity(intent)
    }

    @Suppress( ...names: "UNUSED_PARAMETER")
    fun toTimer(view: View) {
```

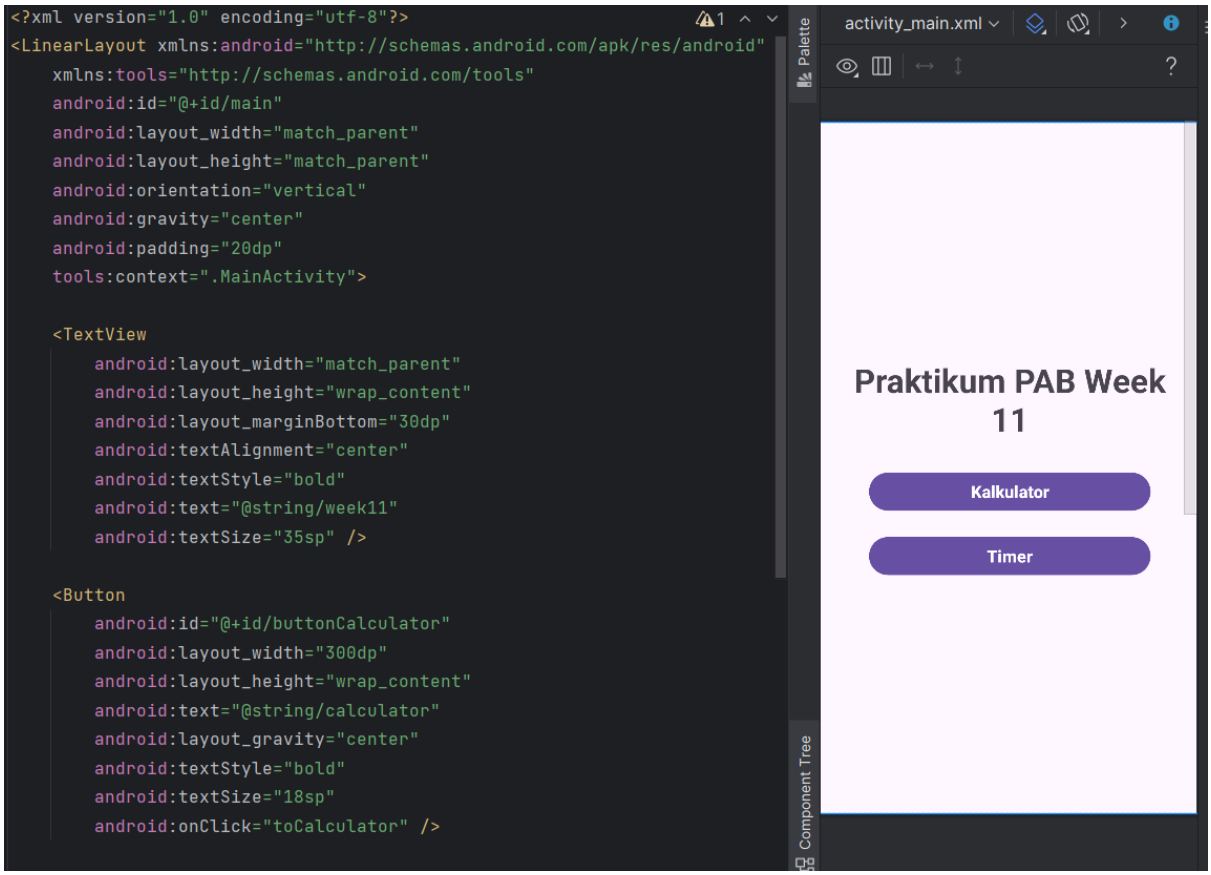
Code di atas merupakan kelas MainActivity, Kelas ini mewarisi dari AppCompatActivity, memberikannya sifat dan perilaku dari sebuah aktivitas standar Android. Aktivitas ini merupakan bagian dari paket com.l0122006.afifimam.week11 dan menggunakan berbagai komponen kerangka kerja Android dan pustaka kompatibilitas, yang diimpor pada awal file.

Metode onCreate dioverride untuk menyiapkan aktivitas saat pertama kali dibuat. Metode ini memanggil super.onCreate untuk memastikan inisialisasi kelas dasar terjadi, dan kemudian mengatur tampilan konten ke activity_main.xml menggunakan setContentView. Untuk memastikan elemen UI ditempatkan dengan benar terhadap bilah sistem (seperti bilah status dan bilah navigasi), sebuah OnApplyWindowInsetsListener diatur pada tampilan utama (R.id.main). Pemantau ini menyesuaikan padding dari tampilan berdasarkan pada masukan jendela sistem.

Warna bilah status diatur menjadi transparan dengan window.statusBarColor = getColor(android.R.color.transparent), memungkinkan tata letak aktivitas untuk meluas ke area bilah status untuk tampilan yang lebih imersif. WindowInsetsController kemudian diperoleh untuk mengatur tampilan ikon dan teks bilah status menjadi terang, memastikan agar terlihat pada latar belakang yang terang.

Dua metode, toCalculator dan toTimer, didefinisikan untuk menangani klik tombol. Dianotasi dengan @Suppress("UNUSED_PARAMETER") untuk mengabaikan peringatan tentang parameter yang tidak digunakan, metode ini membuat intents untuk memulai CalculatorActivity dan TimerActivity secara berturut-turut. Metode-metode ini memfasilitasi navigasi dari MainActivity ke aktivitas lain dalam aplikasi saat tombol-tombol yang sesuai diklik.

Activity_main.xml



Tata letak dimulai dengan sebuah LinearLayout, yang merupakan wadah untuk menata elemen-elemen tampilan. Pengaturan android:orientation="vertical" menandakan bahwa elemen-elemen akan ditata secara vertikal di dalamnya.

Elemen pertama dalam tata letak adalah sebuah TextView yang menampilkan teks "week11" di tengah layar. Properti android:textAlignment="center" mengatur teks agar terpusat secara horizontal, sementara android:textSize="35sp" menentukan ukuran teksnya.

Kemudian, terdapat dua tombol Button. Tombol pertama memiliki teks "Calculator", yang bertujuan untuk membuka fitur kalkulator aplikasi, sedangkan tombol kedua memiliki teks

"Timer" untuk membuka fitur timer aplikasi. Properti android:layout_gravity="center" pada kedua tombol tersebut mengatur agar tombol-tombol tersebut terletak di tengah secara horizontal.

Setiap tombol memiliki properti android:onClick, yang menentukan metode yang akan dipanggil ketika tombol tersebut ditekan. Untuk tombol "Calculator", metode yang dipanggil adalah toCalculator(), sementara untuk tombol "Timer", metode yang dipanggil adalah toTimer(). Ini memungkinkan aplikasi untuk menanggapi aksi pengguna dan melakukan navigasi ke fitur-fitur yang sesuai.

Calculator Activity.kt

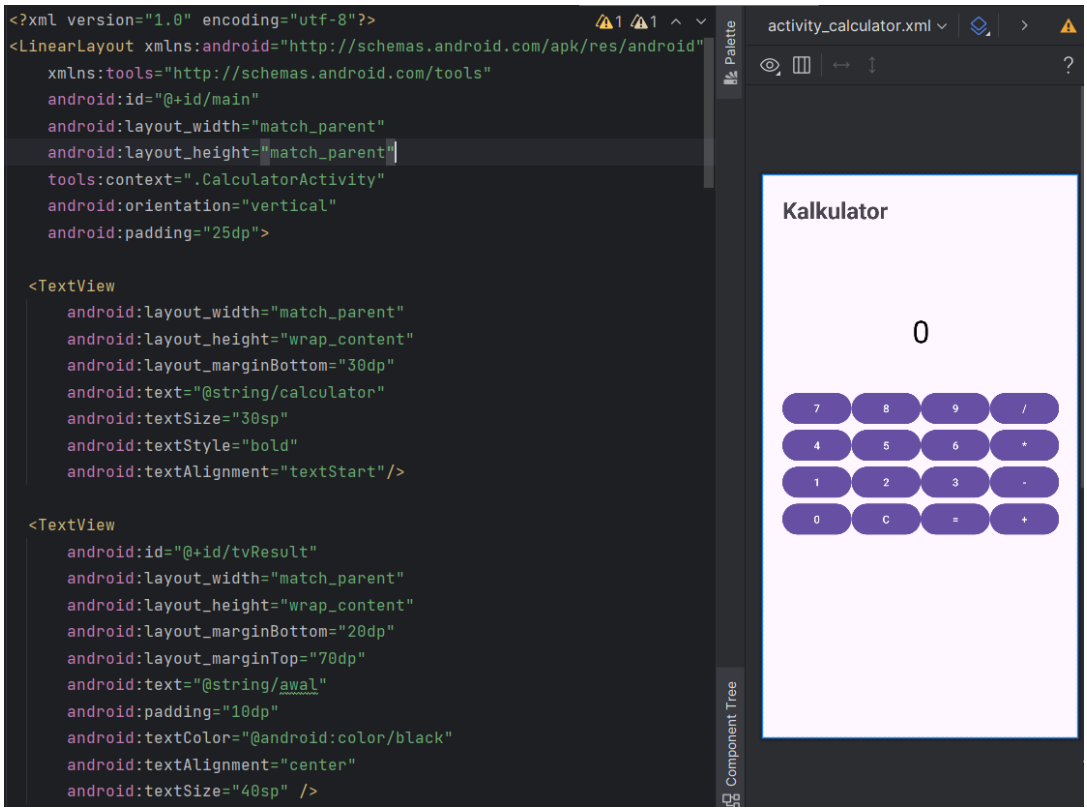
```
package com.l0122006.afifimam.week11

import ...

class CalculatorActivity : AppCompatActivity() {

    private lateinit var binding: ActivityCalculatorBinding
    private val viewModel: MainViewModel by viewModels()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityCalculatorBinding.inflate(layoutInflater)
        setContentView(binding.root)

        viewModel.result.observe( owner: this, Observer { result ->
            binding.tvResult.text = result
        })

        setButtonListeners()
    }

    private fun setButtonListeners() {
        binding.btn0.setOnClickListener { viewModel.onDigit( digit: '0') }
        binding.btn1.setOnClickListener { viewModel.onDigit( digit: '1') }
        binding.btn2.setOnClickListener { viewModel.onDigit( digit: '2') }
        binding.btn3.setOnClickListener { viewModel.onDigit( digit: '3') }
        binding.btn4.setOnClickListener { viewModel.onDigit( digit: '4') }
        binding.btn5.setOnClickListener { viewModel.onDigit( digit: '5') }
        binding.btn6.setOnClickListener { viewModel.onDigit( digit: '6') }
        binding.btn7.setOnClickListener { viewModel.onDigit( digit: '7') }
        binding.btn8.setOnClickListener { viewModel.onDigit( digit: '8') }
        binding.btn9.setOnClickListener { viewModel.onDigit( digit: '9') }
```

Kode di atas mengimplementasikan kelas CalculatorActivity dalam sebuah aplikasi Android. Kelas ini bertanggung jawab untuk menampilkan antarmuka pengguna dan mengelola logika kalkulator. Ini adalah kelas turunan dari AppCompatActivity, yang berarti kelas ini merupakan sebuah aktivitas dalam aplikasi Android.

Variabel binding digunakan untuk mengakses elemen-elemen antarmuka pengguna dalam layout XML. Variabel viewModel digunakan untuk mengelola logika dan data kalkulator menggunakan ViewModel. Dalam metode onCreate(), layout XML activity_calculator di-inflate menggunakan binding dan di-set sebagai konten tampilan aktivitas.

LiveData result dari ViewModel dipantau menggunakan Observer. Ketika ada perubahan pada LiveData, TextView tvResult dalam layout diperbarui dengan hasil yang baru. Metode setButtonListeners() digunakan untuk menetapkan listener untuk setiap tombol dalam layout kalkulator. Ini memungkinkan penanganan interaksi pengguna seperti mengklik tombol angka atau operator matematika. Melalui kelas CalculatorActivity, pengguna dapat memanfaatkan fitur-fitur kalkulator yang disediakan oleh aplikasi, dengan logika perhitungan yang dikelola oleh MainViewModel.

Activity_calculator.xml



Elemen root adalah LinearLayout, yang digunakan sebagai wadah untuk menata elemen-elemen tampilan secara vertikal. Atribut xmlns:android dan xmlns:tools digunakan untuk mengidentifikasi namespace untuk elemen-elemen dan atribut khusus Android dan Android Studio. TextView pertama menampilkan judul "calculator". Atribut android:textAlignment="textStart" mengatur teksnya untuk dimulai dari kiri, android:textSize="30sp" mengatur ukuran teks, dan android:textStyle="bold" mengatur gaya teksnya menjadi tebal.

TextView kedua adalah tempat untuk menampilkan hasil perhitungan kalkulator. android:id="@+id/tvResult" memberikan id untuk referensi di dalam kode Kotlin nanti.

Properti android:textSize="40sp" mengatur ukuran teksnya lebih besar dari judul, dan android:textColor="@android:color/black" mengatur warna teksnya menjadi hitam.

GridLayout berisi tombol-tombol kalkulator yang ditempatkan dalam bentuk grid. Atribut android:columnCount="4" dan android:rowCount="5" menentukan jumlah kolom dan baris dalam grid. Setiap tombol memiliki atribut android:layout_columnWeight="1" untuk memastikan bahwa setiap kolom memiliki lebar yang sama.

Setiap Button memiliki atribut android:id untuk memberikan id unik. Label tombol didefinisikan menggunakan atribut android:text. Tombol-tombol ini akan menampilkan angka 0-9, operator matematika, tombol "Clear" untuk menghapus input, dan tombol "Equals" untuk menampilkan hasil perhitungan.

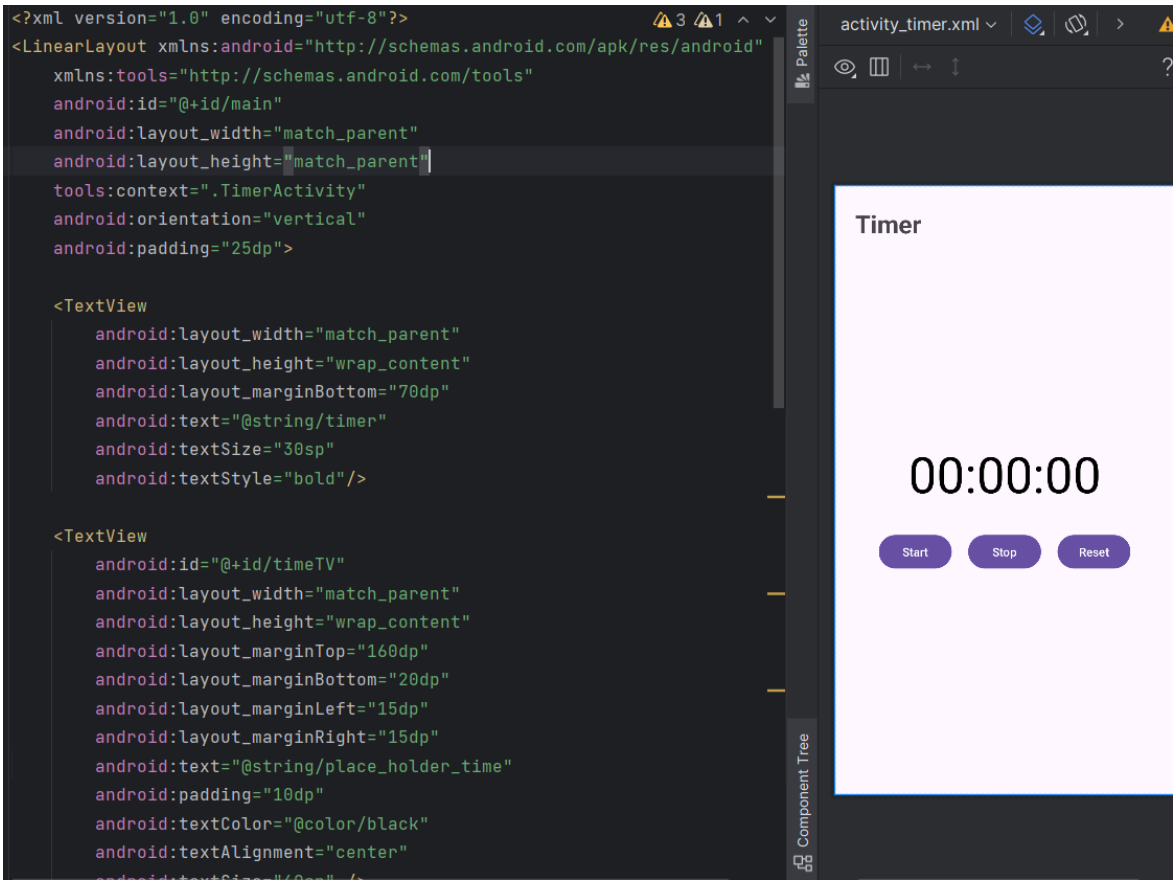
Timer Activity.kt

```
package com.l0122006.afifimam.week11

import ...

class TimerActivity : AppCompatActivity() {

    private lateinit var liveDataTimerViewModel: MainViewModel
    private lateinit var activityTimerBinding: ActivityTimerBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        activityTimerBinding = ActivityTimerBinding.inflate(layoutInflater)
        setContentView(activityTimerBinding.root)

        liveDataTimerViewModel = ViewModelProvider( owner: this)[MainViewModel::class.java]
        timesGoing()

        activityTimerBinding.startButton.setOnClickListener { it: View!
            liveDataTimerViewModel.startTimer()
        }

        activityTimerBinding.stopButton.setOnClickListener { it: View!
            liveDataTimerViewModel.stopTimer()
        }

        activityTimerBinding.resetButton.setOnClickListener { it: View!
            liveDataTimerViewModel.resetTimer()
        }

        window.statusBarColor = getColor(android.R.color.transparent)
        val windowInsetsController = WindowCompat.getInsetsController(window, window.decorView)
```

Kelas TimerActivity adalah turunan dari AppCompatActivity, yang merupakan aktivitas dalam aplikasi Android. Pada metode onCreate(), layout XML activity_timer di-inflate menggunakan ActivityTimerBinding untuk mengakses elemen-elemen antarmuka pengguna. ViewModel liveDataTimerViewModel diinisialisasi menggunakan ViewModelProvider untuk mengelola logika timer menggunakan kelas MainViewModel.

Metode timesGoing() digunakan untuk menampilkan waktu yang berlalu dalam format jam:menit. Observer elapsedTimeObserver dipasang untuk memperbarui TextView timeTV dalam layout dengan waktu yang berubah.Tombol-tombol dalam layout (startButton, stopButton, resetButton) diberi fungsi yang sesuai menggunakan setOnClickListener(), yang akan memanggil metode yang relevan dalam liveDataTimerViewModel saat tombol diklik. Pada bagian akhir onCreate(), status bar diatur menjadi transparan untuk estetika visual yang lebih baik menggunakan window.statusBarColor dan WindowCompat.getInsetsController().

Activity_timer.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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".TimerActivity"
    android:orientation="vertical"
    android:padding="25dp">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="70dp"
        android:text="@string/timer"
        android:textSize="30sp"
        android:textStyle="bold"/>

    <TextView
        android:id="@+id/timeTV"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="160dp"
        android:layout_marginBottom="20dp"
        android:layout_marginLeft="15dp"
        android:layout_marginRight="15dp"
        android:text="@string/place_holder_time"
        android:padding="10dp"
        android:textColor="@color/black"
        android:textAlignment="center"
```

Elemen root adalah LinearLayout, yang digunakan sebagai wadah untuk menata elemen-elemen tampilan secara vertikal. Atribut xmlns:android dan xmlns:tools digunakan untuk mengidentifikasi namespace untuk elemen-elemen dan atribut khusus Android dan Android Studio. TextView pertama menampilkan judul "timer". Atribut android:textSize="30sp" mengatur ukuran teksnya, dan android:textStyle="bold" mengatur gaya teksnya menjadi tebal.

TextView kedua adalah tempat untuk menampilkan waktu timer. android:id="@+id/timeTV" memberikan id untuk referensi di dalam kode Kotlin nanti. Properti android:textSize="60sp" mengatur ukuran teksnya lebih besar dari judul, dan android:textColor="@color/black" mengatur warna teksnya menjadi hitam.

LinearLayout berisi tombol-tombol untuk mengontrol timer. Tombol-tombol ini adalah “Start”, “Stop”, dan “Reset”. Setiap tombol memiliki atribut android:id untuk memberikan id unik, dan label tombol didefinisikan menggunakan atribut android:text.

Main View Model.kt

```
package com.l0122006.afifimam.week11

import android.os.Handler
import android.os.Looper
import androidx.lifecycle.LiveData
import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel

class MainViewModel : ViewModel() {

    // Calculator
    private val _result = MutableLiveData<String>()
    val result: LiveData<String> get() = _result

    private var currentInput = ""
    private var operator: Char? = null
    private var operand1: Double? = null

    fun onDigit(digit: Char) {
        currentInput += digit
        _result.value = currentInput
    }

    fun onOperator(op: Char) {
        operator = op
        operand1 = currentInput.toDoubleOrNull()
        currentInput = ""
    }

    fun onEquals() {
        val operand2 = currentInput.toDoubleOrNull()
        if (operand1 != null && operand2 != null && operator != null) {
```

ViewModel ini merupakan bagian dari arsitektur MVVM (Model-View-ViewModel) dalam pengembangan aplikasi Android. ViewModel bertanggung jawab untuk menyediakan data yang diperlukan oleh antarmuka pengguna (UI) dan menjaga data tetap persisten saat terjadi perubahan konfigurasi atau siklus hidup activity.

Dalam ViewModel ini, terdapat dua bagian utama: kalkulator sederhana dan timer. Bagian kalkulator sederhana memiliki fungsi-fungsi seperti onDigit, onOperator, onEquals, dan onClear yang digunakan untuk melakukan operasi perhitungan dasar. Setiap fungsi ini memanipulasi nilai-nilai seperti currentInput, operator, dan operand1 untuk melakukan perhitungan yang tepat, kemudian hasilnya disimpan dalam LiveData _result.

Bagian timer menggunakan fungsi-fungsi seperti startTimer, stopTimer, dan resetTimer untuk mengelola timer. Timer diimplementasikan menggunakan Handler dan Runnable, yang berjalan di thread utama. Waktu yang telah berlalu disimpan dalam LiveData _elapsedTime, yang akan diperbarui setiap detik menggunakan handler.

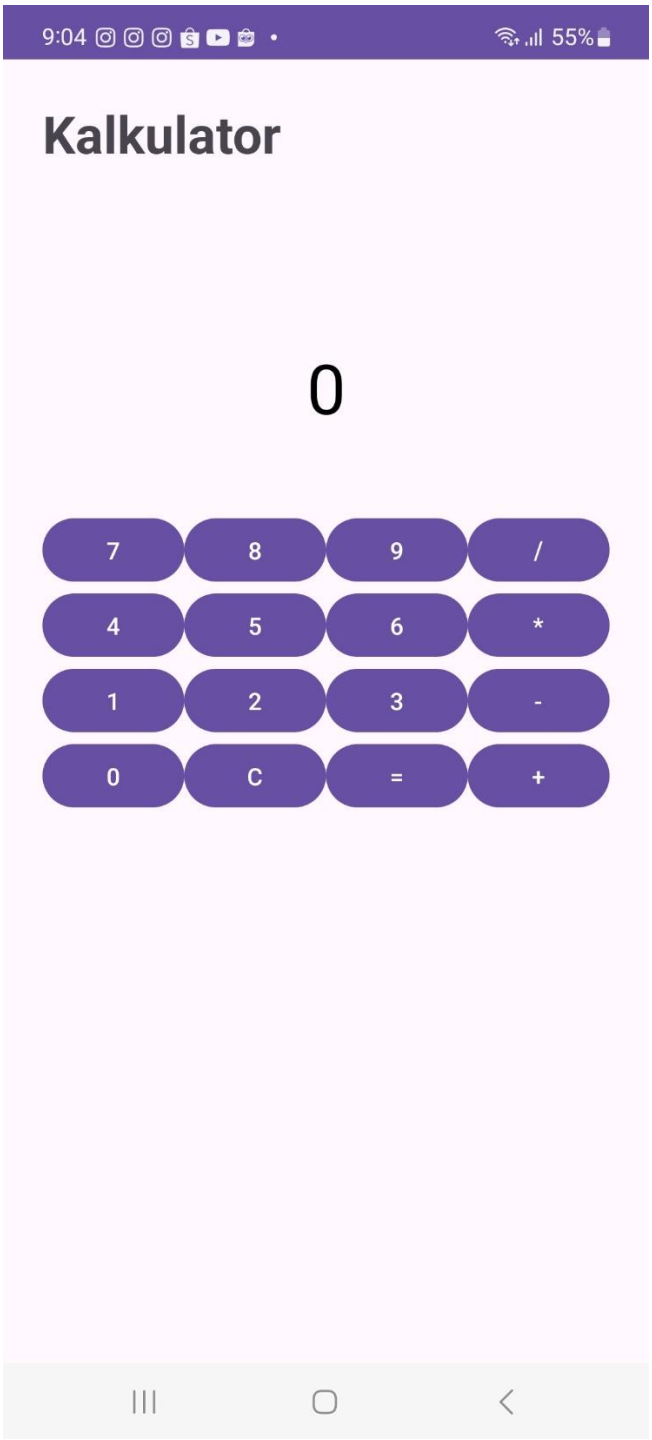
2. **Screenshot Terminal**

   Berikut adalah hasil pada emulator apabila source code dijalankan. Saya menggukan hp karena membuat program lebih ringan untuk dijalankan.

Halaman Utama

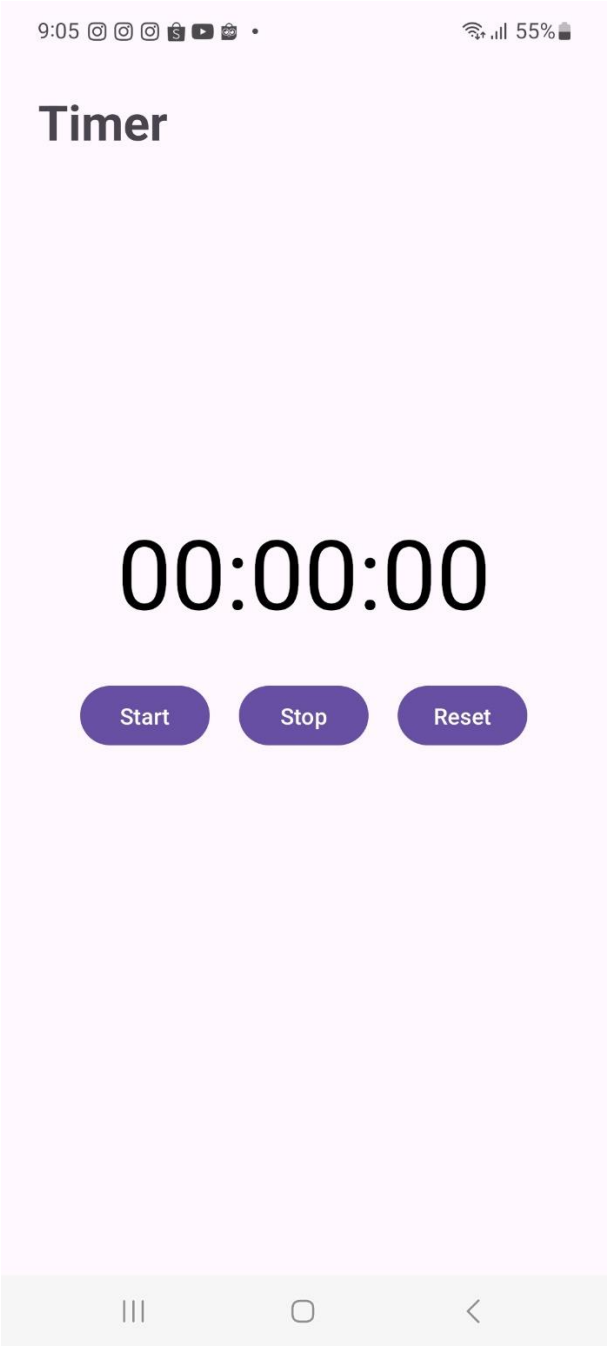
Ini adalah hasil dari halaman utama dari praktikum ke-11. Pada halaman ini terdapat judul berupa teks praktikum dan di bawahnya terdapat 2 button yang masing-masing mengarah ke halaman yang berbeda. Button kalkulator jika dipencet akan mengarah ke activity kalkulator sedangkan button timer akan masuk ke activity timer.

Halaman Kalkulator



Ini adalah halaman kalkulator. Pada halaman ini terdapat judul halaman pada bagian atas kiri. Lalu terdapat angka 0 sebagai default angka sebelum melakukan perhitungan. Lalu di bawahnya terdapat angka-angka dan operasi yang dapat dilakukan dalam kalkulator ini. Hasil dari operasi kalkulator akan berbentuk decimal.

Halaman Timer



Ini adalah halaman timer. Sama seperti kalkulator, pada bagian atas kiri terdapat judul halaman. Di bawahnya terdapat format timer berupa HH:MM:SS sehinnga timer akan diperlihatkan perdetik. Di bawahnya, terdapat button untuk melakukan start, stop, dan reset pada timer yang dapat digunakan oleh user.