

# **LAPORAN TUGAS UJIAN TENGAH SEMESTER**

## **Pemrograman Mobile 1**

Dosen Pengajar : Ibu Nova Agustina, S.T., M.Kom.



Di susun Oleh :

Nama : HarisNurpazri

NPM : 23552011143

Kelas/Semester : TIF RP 23 CNS B | Semester 4

Departemen : Teknik Informatika

**DEPARTEMEN TEKNIK INFORMATIKA  
UNIVERSITAS TEKNOLOGI BANDUNG TAHUN  
2025**

## A. SOAL

### 1. Essay

- Apa fungsi `setOnClickListener`?
- Apa syarat pemanggilan method `setOnClickListener`? Buat contohnya dan screenshot source code nya!
- Error apa yang terjadi jika file kotlin salah menginisialisasi `findViewById` atau objek pada xml belum diinisialisasi? Screenshot logcat-nya!
- Buat sebuah contoh program untuk menampilkan pesan error `NullPointerException`!
- Kumpulkan dalam bentuk pdf di Elearning (Soal essay digabung dengan soal studi kasus cek point 7 Studi Kasus)

### 2. Studi Kasus

- Buatlah sebuah program sederhana yang terdiri dari 4 Activity menggunakan Android Native (Java + XML) yang terdiri dari:
  - SplashScreen Activity
  - Login Activity
  - Register Activity
  - List Chating
- Ketentuan: Silahkan membuat splashscreen dengan baik.
- Pada Register Activity, minimal terdapat objek: `TextView`, `EditText`, `Button`, `ImageView`!
- Tampilkan event Log, Toast dan Toast pada saat Button Register di klik.
- Pada List Chating terdapat data yang ditampilkan dalam listview
- Upload project di Github.
- Jelaskan fungsi setiap baris source code pada file kotlin dan submit dalam bentuk pdf pada Elearning
- Link Github harus tercantum pada pdf (point 7)

## B. JAWABAN

### 1. Essay

- `setOnClickListener` adalah method yang digunakan untuk menangani event klik pada sebuah View (misalnya tombol). Dengan method ini, kita bisa menentukan aksi apa yang akan dilakukan ketika View tersebut diklik oleh pengguna.
- Syarat pemanggilan method `setOnClickListener` dan Contohnya

#### 1) Syarat

- View yang ingin diberi listener harus sudah diinisialisasi, biasanya dengan `findViewById`
- Biasanya dipanggil setelah layout di-set dengan `setOnClickListener`

#### 2) Contohnya:

```
import android.os.Bundle
import android.widget.Button
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
```

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

```

val button = findViewById<Button>(R.id.myButton)
button.setOnClickListener {
    Toast.makeText(this, "Tombol diklik!", Toast.LENGTH_SHORT).show()
}
}
}

```

Pada contoh di atas, `setOnClickListener` dipanggil pada objek `button` yang sudah diinisialisasi dengan `findViewById`.

- c. Jika kamu memanggil `findViewById` dengan ID yang salah atau objek belum ada di layout, biasanya akan terjadi error `NullPointerException` saat kamu mencoba mengakses objek tersebut.

```

java.lang.NullPointerException: Attempt to invoke virtual method 'void android.widget.Button.setOnClickListener()' on a null object reference

```

- d. Contoh program menampilkan pesan error `NullPointerException`

```

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

        val button: Button? = null
        // Ini akan menyebabkan NullPointerException karena button null
        button!!.setOnClickListener {
            Toast.makeText(this, "Ini tidak akan jalan", Toast.LENGTH_SHORT).show()
        }
    }
}

```

Pada contoh di atas, `button` di-set `null`, tapi dipaksa untuk dipanggil method `setOnClickListener` dengan operator `!!` sehingga menyebabkan `NullPointerException`.

## 2. Studi Kasus

### a. Pengertian

Aplikasi ini merupakan sebuah aplikasi mobile bernama Class ES ( Class Ellectornic Schedule ) yaitu sebuah aplikasi untuk membuat sebuah jadwal dengan berbasis electronic yang dirancang menggunakan editor android studio dan menggunakan bahasa kotlin.

### b. Penjelasan Kode

#### 1) SpalshScreenActivity.kt

- package com.example.esapp  
( Mendefinisikan paket tempat file ini berada, yaitu com.example.esapp. Ini membantu mengorganisasi kode )
- import android.content.Intent  
import android.os.Bundle  
import android.os.Handler

```
import android.os.Looper
import androidx.appcompat.app.AppCompatActivity
( Intent untuk berpindah antar activity, Bundle untuk menyimpan data state activity,
Handler dan Looper untuk menjalankan kode dengan penundaan,
AppCompatActivity sebagai superclass activity yang mendukung fitur modern )
```

- class SplashScreenActivity : AppCompatActivity() {  
( Mendefinisikan kelas SplashScreenActivity yang merupakan turunan dari AppCompatActivity, artinya ini adalah sebuah activity di Android )
- private val splashTimeout: Long = 2500 // 2.5 detik  
( Mendeklarasikan variabel splashTimeout bertipe Long dengan nilai 2500 milidetik (2,5 detik), yang akan digunakan sebagai durasi tampilan splash screen )
- override fun onCreate(savedInstanceState: Bundle?) {  
( Fungsi onCreate adalah metode lifecycle yang dipanggil saat activity pertama kali dibuat )
- super.onCreate(savedInstanceState)  
( Memanggil implementasi onCreate dari superclass (AppCompatActivity) untuk memastikan setup dasar activity berjalan dengan benar )
- setContentView(R.layout.activity\_splash\_screen)  
( Mengatur layout yang akan ditampilkan pada activity ini, yaitu layout dengan nama activity\_splash\_screen.xml )
- Handler(Looper.getMainLooper()).postDelayed({  
( Membuat objek Handler yang terhubung ke thread utama (UI thread) dan menjalankan kode di dalam blok { ... } setelah penundaan tertentu )
- startActivity(Intent(this, LoginActivity::class.java))  
( Setelah penundaan, menjalankan intent untuk berpindah ke LoginActivity )
- finish()  
( Menutup SplashScreenActivity agar ketika pengguna menekan tombol kembali, tidak kembali ke splash screen )
- splashTimeout)  
( Menentukan durasi penundaan, yaitu sesuai nilai splashTimeout (2,5 detik) )

## 2) LoginActivity.kt

- import android.content.Context
- import android.content.Intent
- import android.os.Bundle
- import android.widget.Toast
- import androidx.appcompat.app.AppCompatActivity
- import com.google.android.material.button.MaterialButton
- import com.google.android.material.textfield.TextInputEditText

```
import android.widget.TextView
```

( Context untuk akses konteks aplikasi, Intent untuk berpidah antar activity, Bundle untuk menyimpan untuk menyimpan state activity, Toast untuk menampilkan pesan singkat, AppCompatActivity sebagai kelas dasar activity, MaterialButton dan TextInputEditText dari Material Design untuk UI, TextView untuk menampilkan teks )

- class LoginActivity : AppCompatActivity() {  
( Mendefinisikan kelas LoginActivity yang merupakan turunan dari AppCompatActivity )
- private lateinit var etUsername: TextInputEditText  
private lateinit var etPassword: TextInputEditText  
private lateinit var btnLogin: MaterialButton  
private lateinit var tvRegister: TextView  
( etUsername untuk input username, etPassword untuk input password, btnLogin tombol login, tvRegister teks yang bisa diklik untuk menuju halaman registrasi )
- override fun onCreate(savedInstanceState: Bundle?) {  
( Fungsi lifecycle yang dipanggil saat activity dibuat )
- super.onCreate(savedInstanceState)  
( Memanggil fungsi onCreate dari superclass untuk setup dasar )
- setContentView(R.layout.activity\_login)  
( Mengatur layout activity menggunakan file activity\_login.xml )
- etUsername = findViewById(R.id.etUsername)  
etPassword = findViewById(R.id.etPassword)  
btnLogin = findViewById(R.id.btnLogin)  
tvRegister = findViewById(R.id.tvRegister)  
( Menghubungkan variabel dengan komponen UI di layout berdasarkan ID masing-masing )
- btnLogin.setOnClickListener {  
( Menambahkan event listener saat tombol login ditekan )
- val usernameInput = etUsername.text.toString().trim()  
val passwordInput = etPassword.text.toString().trim()  
( Mengambil teks dari input username dan password, lalu menghapus spasi di awal dan akhir )
- if (usernameInput.isEmpty() || passwordInput.isEmpty()) {  
Toast.makeText(this, "Username dan password harus diisi",  
Toast.LENGTH\_SHORT).show()  
return@setOnClickListener}

( Mengecek apakah username atau password kosong. Jika iya, tampilkan pesan dan hentikan proses login )

- ```
val sharedPref = getSharedPreferences("UserPrefs", Context.MODE_PRIVATE)
val savedUsername = sharedPref.getString("username", null)
val savedPassword = sharedPref.getString("password", null)
```

( Mengakses data yang tersimpan di SharedPreferences dengan nama "UserPrefs" untuk mendapatkan username dan password yang sudah terdaftar )
- ```
if (savedUsername == null || savedPassword == null) {
    Toast.makeText(this, "Belum ada akun terdaftar, silakan daftar dulu",
        Toast.LENGTH_SHORT).show()
} else if (usernameInput == savedUsername && passwordInput ==
    savedPassword) {
```

( Jika belum ada data akun tersimpan, tampilkan pesan agar pengguna mendaftar dulu, Jika username dan password yang dimasukkan cocok dengan data tersimpan, lanjutkan login )
- ```
val intent = Intent(this, MainActivity::class.java)
startActivity(intent)
finish()
```

( Membuat intent untuk pindah ke MainActivity, menjalankan activity tersebut, dan menutup LoginActivity )
- ```
} else {
    Toast.makeText(this, "Username atau password salah",
        Toast.LENGTH_SHORT).show()
}
```

( Jika username atau password salah, tampilkan pesan kesalahan )
- ```
tvRegister.setOnClickListener {
    startActivity(Intent(this, RegisterActivity::class.java))
}
```

( Menambahkan event listener pada teks register, ketika diklik akan membuka RegisterActivity )

### 3) RegisterActivity.kt

- ```
import android.content.Context
import android.content.Intent
import android.os.Bundle
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.google.android.material.button.MaterialButton
import com.google.android.material.textfield.TextInputEditText
import android.widget.TextView
```

( Context untuk akses konteks aplikasi, Intent untuk berpindah antar activity, Bundle untuk menyimpan state activity, Toast untuk menampilkan pesan singkat, AppCompatActivity sebagai kelas dasar activity, MaterialButton dan

TextInputEditText dari Material Design untuk UI, TextView untuk menampilkan teks )

- class RegisterActivity : AppCompatActivity() {  
( Mendefinisikan kelas RegisterActivity yang merupakan turunan dari AppCompatActivity )
- private lateinit var etUsername: TextInputEditText  
private lateinit var etPassword: TextInputEditText  
private lateinit var etConfirmPassword: TextInputEditText  
private lateinit var btnRegister: MaterialButton  
private lateinit var tvLogin: TextView  
( etUsername untuk input username, etPassword untuk input password, etConfirmPassword untuk input konfirmasi password, btnRegister tombol register, tvLogin teks yang bisa diklik untuk menuju halaman )
- override fun onCreate(savedInstanceState: Bundle?) {  
( Fungsi lifecycle yang dipanggil saat activity dibuat )
- super.onCreate(savedInstanceState)  
( Memanggil fungsi onCreate dari superclass untuk setup dasar )
- setContentView(R.layout.activity\_register)  
( Mengatur layout activity menggunakan file activity\_register.xml )
- etUsername = findViewById(R.id.etUsername)  
etPassword = findViewById(R.id.etPassword)  
etConfirmPassword = findViewById(R.id.etConfirmPassword)  
btnRegister = findViewById(R.id.btnRegister)  
tvLogin = findViewById(R.id.tvLogin)  
( Menghubungkan variabel dengan komponen UI di layout berdasarkan ID masing-masing )
- btnRegister.setOnClickListener {  
( Menambahkan event listener saat tombol register ditekan )
- val username = etUsername.text.toString().trim()  
val password = etPassword.text.toString().trim()  
val confirmPassword = etConfirmPassword.text.toString().trim()  
( Mengambil teks dari input username, password, dan konfirmasi password, lalu menghapus spasi di awal dan akhir )
- if (username.isEmpty() || password.isEmpty() || confirmPassword.isEmpty()) {  
Toast.makeText(this, "Mohon isi semua data", Toast.LENGTH\_SHORT).show()  
} else if (password != confirmPassword) {  
Toast.makeText(this, "Password dan konfirmasi tidak sama", Toast.LENGTH\_SHORT).show()

```
} else {
```

( Mengecek apakah ada input yang kosong, jika iya tampilkan pesan, Jika password dan konfirmasi password tidak sama, tampilkan pesan kesalahan, Jika semua valid, lanjut ke proses penyimpanan data )

- ```
val sharedPref = getSharedPreferences("UserPrefs", Context.MODE_PRIVATE)
with(sharedPref.edit()) {
    putString("username", username)
    putString("password", password)
    apply()}
( Mengakses SharedPreferences dengan nama "UserPrefs" dalam mode privat,
Menyimpan username dan password ke SharedPreferences secara permanen )
```
- ```
Toast.makeText(this, "Pendaftaran berhasil, silakan login",
Toast.LENGTH_SHORT).show()
startActivity(Intent(this, LoginActivity::class.java))
finish()
( Menampilkan pesan bahwa pendaftaran berhasil, Membuka LoginActivity agar
pengguna bisa login, Menutup RegisterActivity agar tidak bisa kembali ke halaman
registrasi dengan tombol kembali )
```
- ```
tvLogin.setOnClickListener {
startActivity(Intent(this, LoginActivity::class.java))
finish() }
( Menambahkan event listener pada teks login, ketika diklik akan
membuka LoginActivity dan menutup RegisterActivity )
```

#### 4) MainActivity.kt

- ```
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
( Bundle untuk menyimpan state activity, AppCompatActivity sebagai kelas dasar
activity, LinearLayoutManager dan RecyclerView untuk menampilkan daftar
dengan tampilan scroll )
```
- ```
class MainActivity : AppCompatActivity() {
( Mendefinisikan kelas MainActivity yang merupakan turunan dari
AppCompatActivity )
```
- ```
override fun onCreate(savedInstanceState: Bundle?) {
( Fungsi lifecycle yang dipanggil saat activity dibuat )
```
- ```
super.onCreate(savedInstanceState)
( Memanggil fungsi onCreate dari superclass untuk setup dasar )
```



- setContentView(R.layout.activity\_main)  
( Mengatur layout activity menggunakan file activity\_main.xml )
- val recyclerView = findViewById<RecyclerView>(R.id.recycler\_view)  
( Menghubungkan variabel recyclerView dengan komponen RecyclerView di layout berdasarkan ID recycler\_view )
- recyclerView.layoutManager = LinearLayoutManager(this)  
( Mengatur layout manager RecyclerView menjadi LinearLayoutManager, yang menampilkan item secara vertikal berurutan )
- recyclerView.setHasFixedSize(true)  
( Memberi tahu RecyclerView bahwa ukuran item tidak akan berubah, sehingga meningkatkan performa )
- val itemList = listOf(  
Item("Selasa", "Pemrograman Berorientasi Objek 2", R.drawable.img1),  
Item("Selasa", "Jaringan Komputer 2", R.drawable.img2),  
Item("Rabu", "Pemrograman Mobile 1", R.drawable.img3),  
Item("Kamis", "Digital Preneurship", R.drawable.img4),  
Item("Kamis", "Teknik Kompilasi", R.drawable.img5),  
Item("Jumat", "Objek Oriented Analisis Design", R.drawable.img6),  
Item("Jumat", "Keamanan Komputer", R.drawable.img7) )  
( Membuat daftar itemList yang berisi objek ItemList. Setiap objek berisi hari, nama mata kuliah, dan gambar (resource drawable) )
- val adapter = AdapterList(itemList)  
( Membuat instance adapter AdapterList dengan data itemList. Adapter ini bertugas menghubungkan data dengan tampilan RecyclerView )
- recyclerView.adapter = adapter  
( Mengatur adapter RecyclerView dengan adapter yang sudah dibuat agar data bisa ditampilkan )

##### 5) AdapterList.kt

- import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.ImageView  
import android.widget.TextView  
import androidx.recyclerview.widget.RecyclerView  
import com.bumptech.glide.Glide  
( LayoutInflater untuk mengubah file XML layout menjadi objek View, View dan ViewGroup sebagai elemen dasar UI, ImageView dan TextView untuk menampilkan gambar dan teks RecyclerView untuk membuat daftar yang bisa di-scroll, Glide adalah library pihak ketiga untuk memuat dan menampilkan gambar secara efisien )

- class AdapterList(private val itemList: List<ItemList>) : RecyclerView.Adapter<AdapterList.ViewHolder>() {  
 ( Mendefinisikan kelas AdapterList yang merupakan adapter untuk RecyclerView, Adapter ini menerima daftar data itemList bertipe List<ItemList>, Adapter ini mengatur bagaimana data ditampilkan di dalam RecyclerView, AdapterList.ViewHolder adalah kelas internal yang akan mengelola tampilan tiap item )
  - class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView){  
 val imageVi  
 ( Mendefinisikan kelas ViewHolder yang menyimpan referensi ke komponen UI dalam satu item daftar, ViewHolder menerima itemView yang merupakan layout item, Baris ini belum lengkap, tapi biasanya di sini akan ada deklarasi variabel untuk ImageView, TextView, dan lain-lain yang ada di layout item )
- 6) ItemList.kt
- import android.adservices.adid.AdId  
 ( Baris ini mengimpor kelas AdId dari paket android.adservices.adid. Namun, dalam kode yang kamu berikan, kelas ini tidak digunakan, jadi kemungkinan ini impor yang tidak diperlukan atau sisa dari kode lain )
  - data class ItemList(  
 val judul: String,  
 val subjudul: String,  
 val imageResId: Int)
- ( Mendefinisikan sebuah **data class** bernama ItemList. Data class ini berfungsi sebagai model data yang menyimpan tiga properti: judul bertipe String — biasanya untuk menyimpan judul item, subjudul bertipe String — biasanya untuk menyimpan deskripsi atau subjudul item, imageResId bertipe Int — menyimpan ID resource gambar (biasanya dari folder drawable) )

### C. LINK

- GitHub  
 ([https://github.com/harisnurpazri/UTS\\_PM1-Haris\\_Nurpazri-23552011143](https://github.com/harisnurpazri/UTS_PM1-Haris_Nurpazri-23552011143))
- Drive Run  
 (<https://drive.google.com/drive/folders/1Gg6gAP0JPz2Ua1LwC6YNdYjCoF0modnd?usp=sharing>)

## D. OUTPUN APLIKASI

