

LEMBAR TUGAS MAHASISWA

PEMOGRAMAN MOBILE 1

UTS

Disusun dalam rangka menyelesaikan tugas

Dosen Pengampu : Nova Agustina, ST., M.Kom.



Ditulis oleh:

Rizal Abdul Ghani 23552011086

TIF RP 23 : CID B

Semester : 4 (Empat)

PROGRAM STUDI TEKNIK INFORMATIKA

UNIVERSITAS TEKNOLOGI BANDUNG

2025

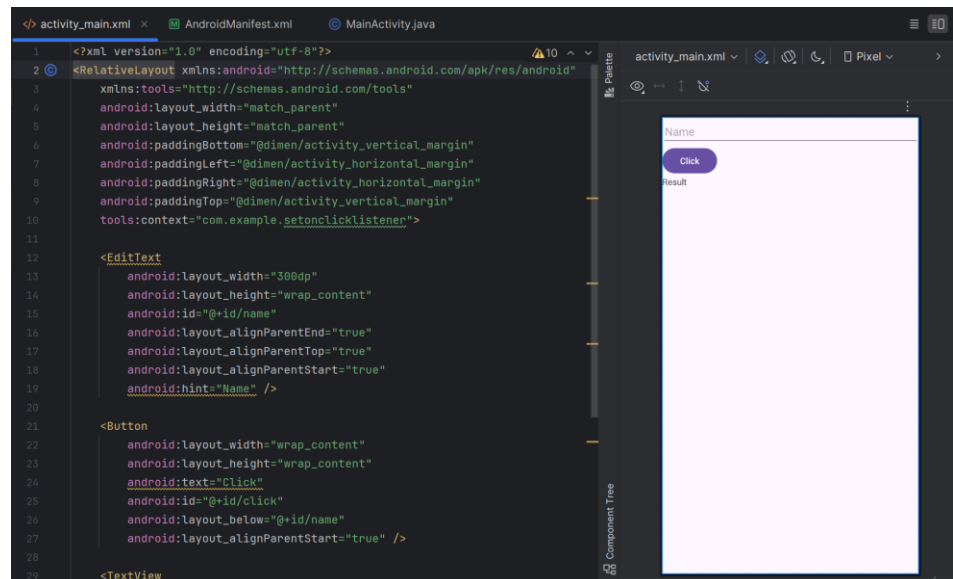
Essay

1. Apa fungsi setOnClickListener?

- setOnClickListener adalah metode di Android yang digunakan untuk memberikan perintah pada tombol, gambar atau elemen user interface yang sudah dibuat untuk diberikan aksi ketika objek tersebut di klik.

2. Apa syarat pemanggilan method setOnClickListener? Buat contohnya dan screenshot source code nya!

- Syarat agar dapat memanggil metode setOnClickListener adalah harus membuat terlebih dahulu komponen-komponen seperti tombol, text view dan layout agar dapat mengimplementasikan perintah setOnClickListener.
- Contoh:

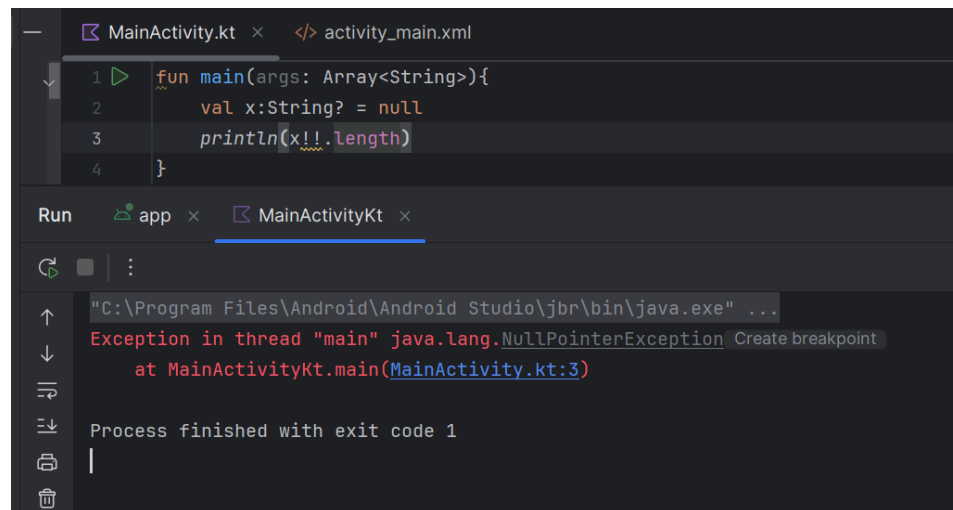


3. Error apa yang terjadi jika file kotlin salah menginisialisasi findViewById atau objek pada xml belum diinisialisasi? Screenshot logcat-nya!

- Jika Anda salah menginisialisasi findViewById atau objek pada XML belum diinisialisasi dengan benar di Kotlin, akan mengalami error NullPointerException karena mencoba mengakses properti atau metode pada objek yang null, karena pada kotlin memiliki system null safety yang mencegah kode berjalan Ketika mengacu pada objek yang null.



4. Buat sebuah contoh program untuk menampilkan pesan error `NullPointerException`!



```
1 fun main(args: Array<String>){
2     val x:String? = null
3     println(x!!.length)
4 }
```

Run app x MainActivityKt x

"C:\Program Files\Android\Android Studio\jbr\bin\java.exe" ...

Exception in thread "main" java.lang.NullPointerException Create breakpoint
at MainActivityKt.main(MainActivity.kt:3)

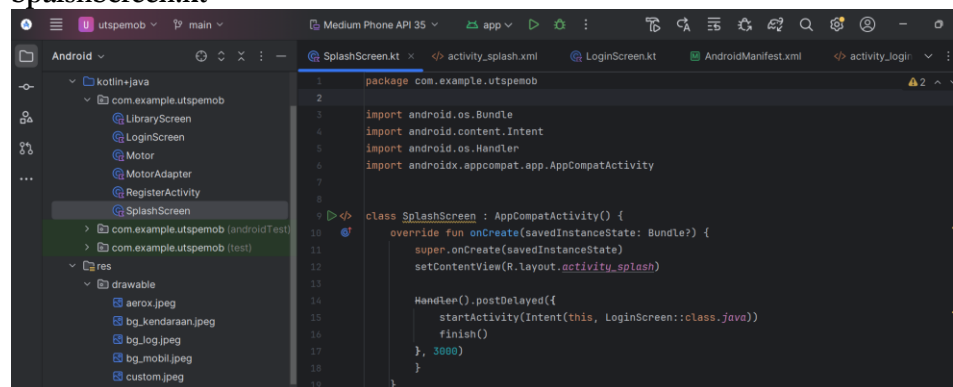
Process finished with exit code 1

Studi Kasus

1. Buatlah sebuah program sederhana yang terdiri dari 4 Activity menggunakan Android Native (Java + XML) yang terdiri dari:
 - a SplashScreen Activity
 - b Login Activity
 - c Register Activity
 - d List Chating

A. Spalsh Screen Activity beserta penjelasan:

- SpalshScreen.kt



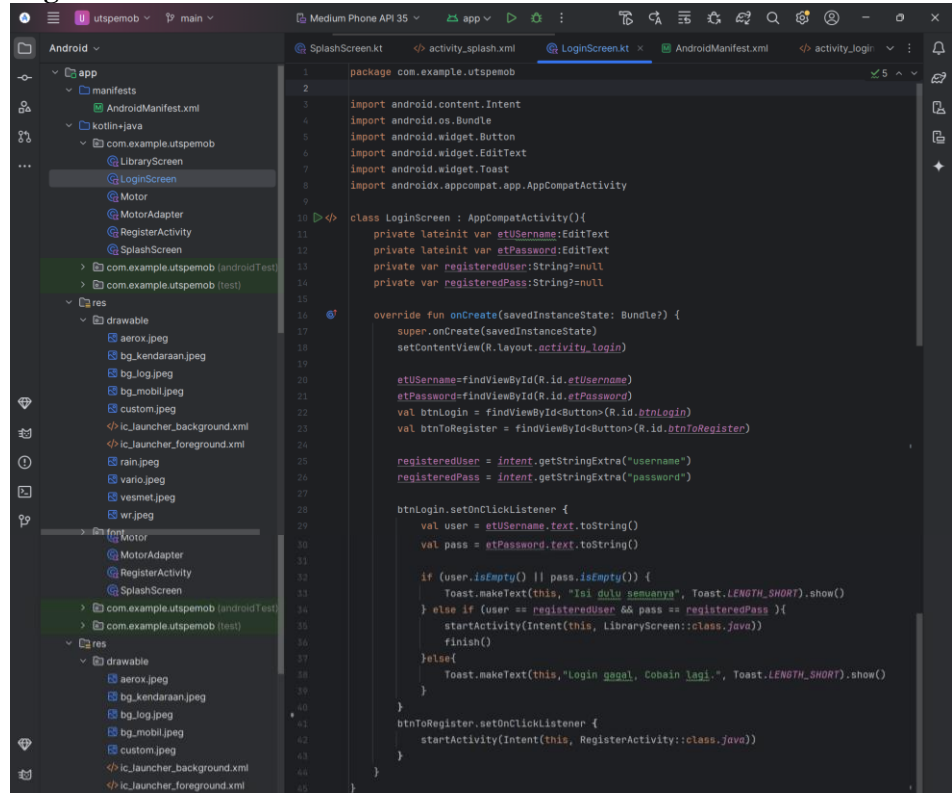
```
1 package com.example.utspemob
2
3 import android.os.Bundle
4 import android.content.Intent
5 import android.os.Handler
6 import androidx.appcompat.app.AppCompatActivity
7
8 class SplashScreen : AppCompatActivity() {
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         setContentView(R.layout.activity_splash)
12
13         Handler().postDelayed({
14             startActivity(Intent(this, LoginActivity::class.java))
15             finish()
16         }, 3000)
17     }
18 }
```

- Import berbagai class yang dibutuhkan:
- **Bundle**: untuk menyimpan state saat Activity dibuat.
 - **Intent**: untuk berpindah antar Activity.
 - **Handler**: untuk menunda eksekusi kode.
 - **AppCompatActivity**: superclass untuk activity yang mendukung fitur kompatibilitas ke belakang.

- **SplashScreen : AppCompatActivity** untuk mendefinisikan sebuah activity SplashScreen yang Dimana turunan dari AppCompatActivity
- **onCreate** dipanggil saat pertama kali activity dibuat kemudian ada setContentView (R.layout.activity_splash) untuk menampilkan layout dari activity_splash.xml.
- **Handler().postDelayed()** digunakan untuk memberikan waktu delay pada SplashScreen ke page selanjutnya selama 3 detik (3000 ms). Kemudian akan lanjut ke page berikutnya yaitu Login Screen dan memastikan agar SplashScreen tidak dapat dibuka Kembali.

B. Login Activity beserta penjelasan:

- **LoginScreen.kt**



- **LoginScreen : AppCompatActivity** untuk mendefinisikan sebuah activity LoginScreen yang dimana turunan dari AppCompatActivity.
- **private lateinit var etUsername** dan **etPassword** adalah input yang nantinya akan dilakukan oleh user ketika berada di activity_login.xml.
- **private var registeredUser** dan **registeredPass** adalah data yang sudah dimasukkan dan didaftarkan oleh user ketika akan membuat akun yang dimana nantinya akan digunakan untuk melakukan login.
- **setContentView(R.layout.activity_login)** perintah untuk memilih layout yang nantinya akan digunakan oleh LoginScreen, yang dimana layout yang digunakan adalah activity_login.xml.
- Untuk menghubungkan kotlin dengan variable yang ada di xml menggunakan perintah:
 - etUsername = findViewById(R.id.etUsername)
 - etPassword = findViewById(R.id.etPassword)
 - val btnLogin = findViewById<Button>(R.id.btnLogin)
 - val btnToRegister = findViewById<Button>(R.id.btnToRegister)

- **registeredUser** dan **registeredPass** akan mengambil data yang dikirimkan oleh RegisterActivity menggunakan Intent yang akan nantinya akan digunakan untuk validasi login.
- **btnLogin.setOnClickListener** logika pada perintah tersebut adalah seperti:
 - Ambil input user dan pass.
 - Jika kosong maka tampilkan Toast “Isi dulu semuanya”.
 - Jika cocok dengan registered maka akan pindah ke LibraryScreen.
 - Jika salah maka akan menampilkan Toast “Login gagal, Cobain lagi”.
- **btnToRegister.setOnClickListener** perintah untuk mengarahkan ke halaman RegisterActivity.

C. Register Activity beserta penjelasan:

- **RegisterActivity.kt**

```

1 package com.example.utspemob
2
3 import android.content.Intent
4 import android.os.Bundle
5 import android.widget.Button
6 import android.widget.EditText
7 import android.widget.Toast
8 import androidx.appcompat.app.AppCompatActivity
9
10 class RegisterActivity : AppCompatActivity() {
11     private lateinit var etRegUsername: EditText
12     private lateinit var etRegPassword: EditText
13     private lateinit var etConfirmPassword: EditText
14
15     override fun onCreate(savedInstanceState: Bundle?) {
16         super.onCreate(savedInstanceState)
17         setContentView(R.layout.activity_register)
18
19         etRegUsername = findViewById(R.id.etRegUsername)
20         etRegPassword = findViewById(R.id.etRegPassword)
21         etConfirmPassword = findViewById(R.id.etConfirmPassword)
22         val btnRegister = findViewById<Button>(R.id.btnRegister)
23
24         btnRegister.setOnClickListener {
25             val user = etRegUsername.text.toString()
26             val pass = etRegPassword.text.toString()
27             val confirm = etConfirmPassword.text.toString()
28
29             if (user.isEmpty() || pass.isEmpty() || confirm.isEmpty()) {
30                 Toast.makeText(this, "Isi dulu semuanya ya.", Toast.LENGTH_SHORT).show()
31             } else if (pass != confirm) {
32                 Toast.makeText(this, "Passwordnya Salah", Toast.LENGTH_SHORT).show()
33             } else {
34                 val intent = Intent(this, LoginActivity::class.java)
35                 intent.putExtra("username", user)
36                 intent.putExtra("password", pass)
37                 Toast.makeText(this, "Daftarnya Berhasil", Toast.LENGTH_SHORT).show()
38                 startActivity(intent)
39                 finish()
40             }
41         }
42     }
43 }

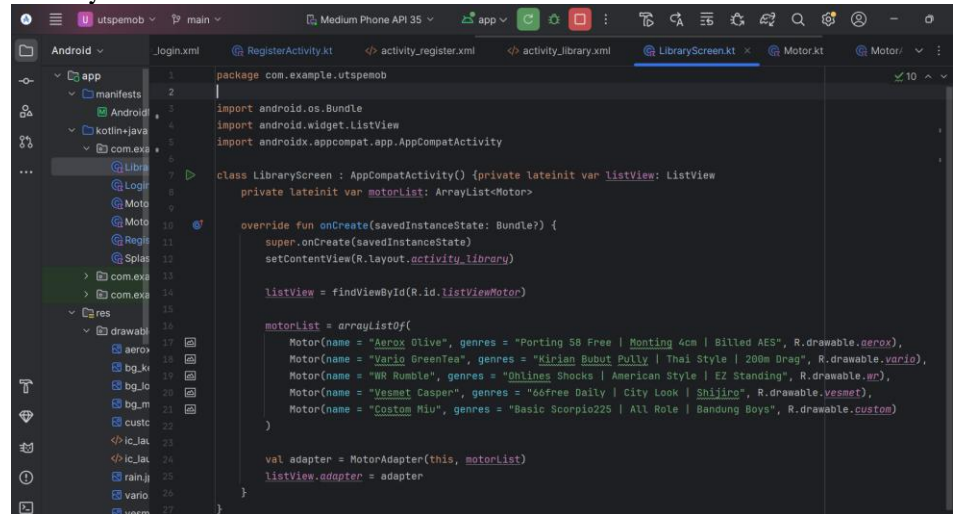
```

- **private lateinit var** perintah untuk mendeklarasikan ketiga input untuk username, password dan konfirmasi password.
- **Variabel etRegUsername, etRegPassword, etConfirmPassword**
Digunakan untuk mewakili input dari username, password, dan konfirmasi password dari EditText.
- **override fun onCreate** digunakan untuk membuka page, yang dimana dalam RegisterActivity akan membuka activity_register.xml.
- **btnRegister.setOnClickListener** Ketika tombol ditekan maka akan mengambil teks dari input yang dilakukan oleh user.
Logika dari perintah tersebut adalah:
 - Ambil input user, pass, confirm.
 - Jika ada yang kosong maka akan menampilkan Toast “Isi dulu semuanya ya”.
 - Jika password tidak sama dengan yang password awal dimasukkan maka konfirmasi dan akan menampilkan Toast “Passwordnya salah”.

- Jika semua sudah benar maka buat Intent ke LoginScreen, kirim username & password lewat intent,
- tampilkan Toast “Daftarnya Berhasil”, lalu pindah ke LoginScreen dan tutup halaman register.

D. List Chatting beserta penjelasan:

- **LibraryScreen.kt**



- **private lateinit var listView** digunakan untuk menampilkan daftar.
- **private lateinit var motorList** digunakan untuk menampilkan object dari motor.
- **override fun onCreate** digunakan untuk membuka page, yang dimana dalam LibraryScreen akan membuka activity_library.xml yang berisi ListView.
- **listView = findViewById** akan mengambil **id: listViewMotor** yang akan digunakan untuk menghubungkan listView dengan komponen ListView yang ada di layout xml.
- **motorList = arrayListOf** digunakan untuk membuat daftar motor yang nantinya akan ditarik ke xml dan ditampilkan pada output yang berisikan nama, fitur dan gambar motor dengan cara mengambil **name**, **genres** dan gambar dari **drawable**.
- **val adapter = MotorAdapter** digunakan untuk mengatur tampilan item motor di ListView. Adapter ini dikombinasikan dengan xml untuk mengatur bagaimana posisi layout motor akan ditampilkan.

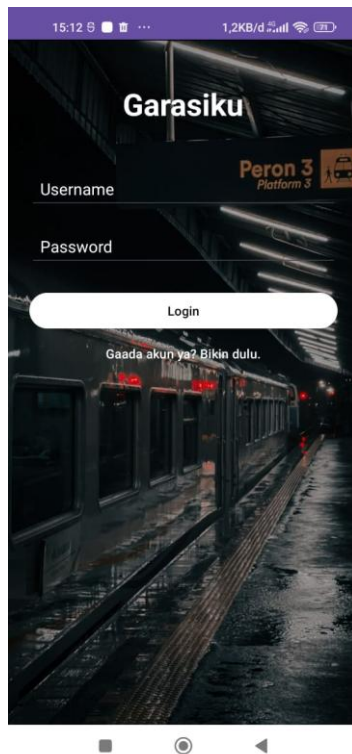
❖ **Link Github:** <https://github.com/RizalAge/UTSPemob>

➤ Hasil Output

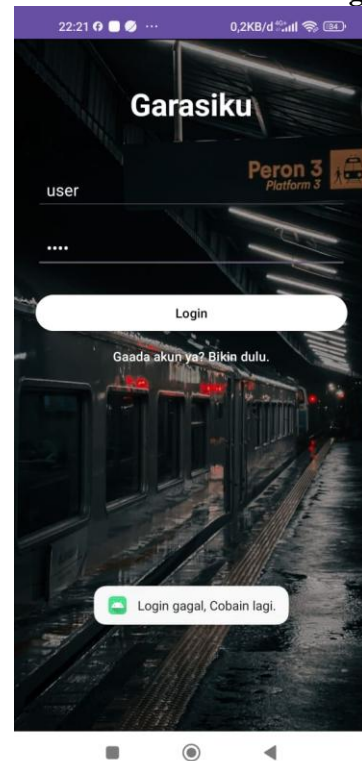
Splash Screen



Login Screen



Ketika belum melakukan registrasi



Register Screen

15:12 20,4KB/d

Peron 3 Platform 3

Garasiku

Username

Password

Confirm Password

Register

Ketika berhasil registrasi

22:32 1,4KB/d

Garasiku

Peron 3 Platform 3

Username

Password

Login

Gaada akun ya? Bikin dulu.

Daftarnya Berhasil

List Screen

