

PEMOGRAMAN MOBILE 1

Disusun dalam rangka memenuhi tugas

Dosen Pengampu : Nova Agustina, S.T., M.Kom.



Disusun oleh:

Sheva Nadhif Gazzauhar

(23552011018)

TIF RP 223PA

**PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS TEKNOLOGI BANDUNG
2025**

Soal :

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.

Contoh :

```
Button btnLogin = findViewById(R.id.btnLogin);

btnLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Kode yang akan dijalankan saat tombol diklik
        Toast.makeText(getApplicationContext(), "Button clicked!",
        Toast.LENGTH_SHORT).show();
    }
});
```

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

Syarat Pemanggilan setOnClickListener:

1. View yang dapat diklik: Seperti Button, ImageView, dll.
2. Referensi ke View: Menggunakan findViewById().
3. Menetapkan OnClickListener: Membuat listener untuk menangani klik.

```
Button btnLogin = findViewById(R.id.btnLogin);
btnLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(MainActivity.this, "Button clicked!",
        Toast.LENGTH_SHORT).show();
    }
});
```

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

```
Caused by: java.lang.NullPointerException: Attempt to invoke virtual method 'void android.widget.Button.setOnClickListener(android.view.View.OnClickListener)' on a null object reference
at com.example.uts.LoginActivity.onCreate(LoginActivity.kt:47)
at android.app.Activity.performCreate(Activity.java:9155)
at android.app.Activity.performCreate(Activity.java:9133)
at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1321)
at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:4262)33 more...
```

Akan terjadi error `NullPointerException`, karena ID tersebut tidak dikenali oleh compiler karena tidak ditemukan di `R.id`.

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

Contoh :

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        String myString = null; // Variabel yang diset ke null
        // Mengakses metode pada objek null akan menyebabkan NullPointerException
        int length = myString.length(); // Ini akan menyebabkan NullPointerException
    }
}
```

Penjelasan:

- Pada kode di atas, variabel `myString` diset ke `null`.
- Kemudian, saat kita mencoba memanggil `myString.length()`, aplikasi akan melemparkan `NullPointerException` karena `myString` tidak memiliki nilai (`null`).

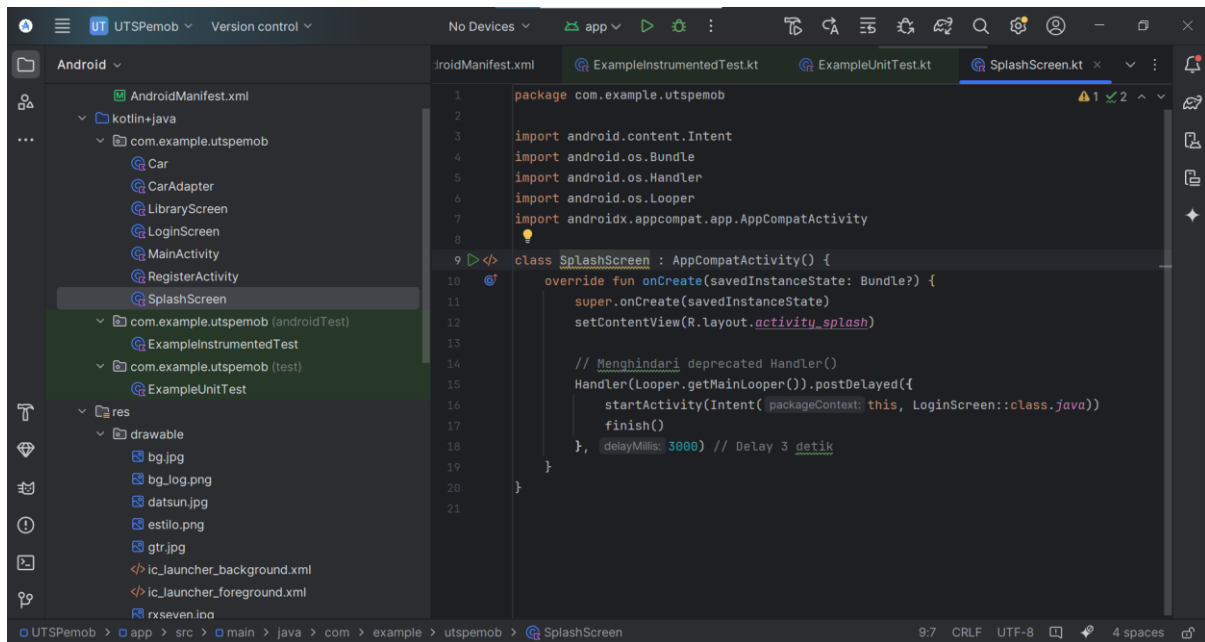
Studi Kasus

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

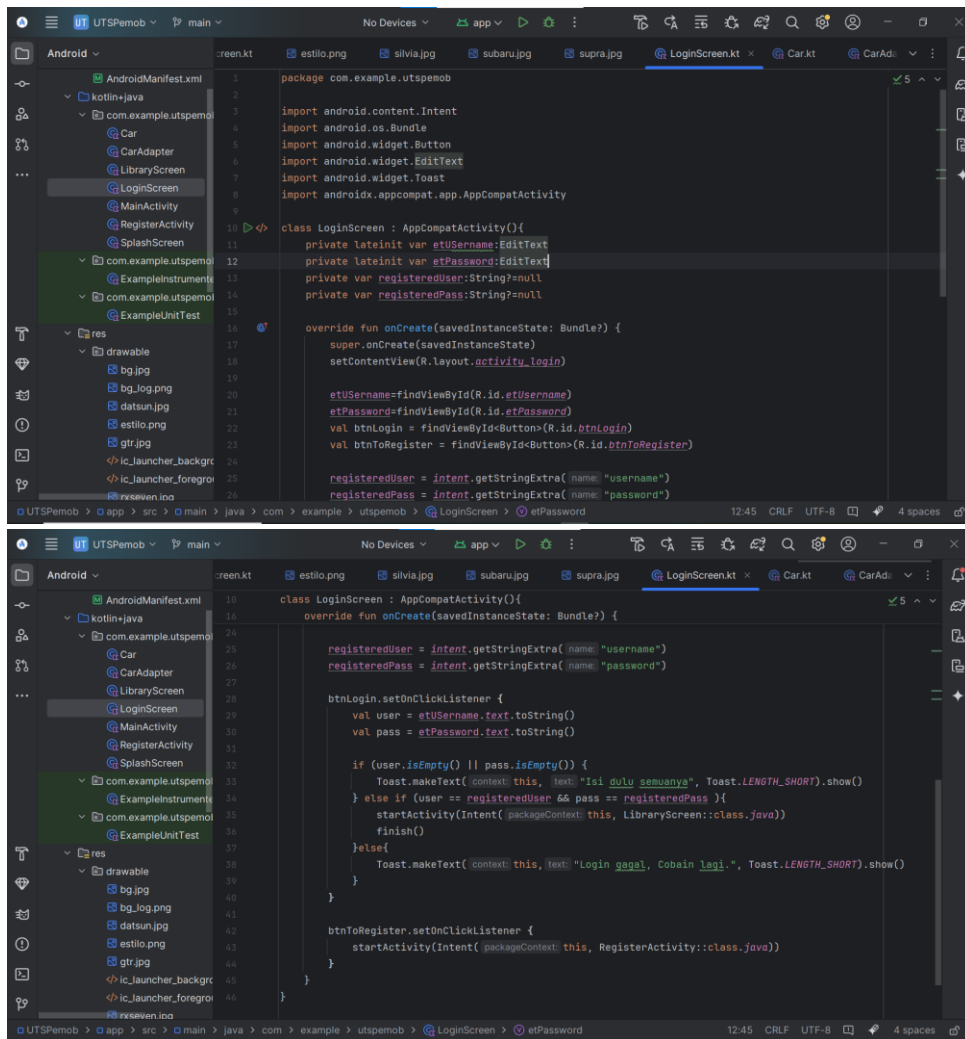
Link Github : <https://github.com/nadhifsheva/UTS-PEMOB>

- A. SplashScreen Activity



- Intent: Digunakan untuk berpindah dari satu activity ke activity lainnya.
- Bundle: Digunakan untuk menyimpan dan mengembalikan data yang dapat dipertahankan antara lifecycle activity (misalnya, menyimpan status aplikasi).
- Handler dan Looper: Digunakan untuk mengatur penundaan eksekusi di thread utama (UI thread).
- AppCompatActivity: Kelas dasar untuk activity yang memberikan dukungan untuk elemen UI dan kompatibilitas dengan berbagai versi Android.
- Handler(Looper.getMainLooper()): Membuat objek Handler yang terhubung dengan **main thread** (UI thread). Looper.getMainLooper() digunakan untuk memastikan eksekusi terjadi pada thread utama aplikasi, yang mana diperlukan untuk update UI.
- postDelayed(...): Menunda eksekusi kode dalam lambda expression selama 3 detik (3000 milidetik).
- startActivity(Intent(this, LoginActivity::class.java)): Membuat objek Intent untuk berpindah ke activity LoginActivity. Setelah 3 detik, activity ini akan berpindah ke LoginActivity.
- finish(): Setelah berpindah ke LoginActivity, finish() dipanggil untuk menutup activity SplashScreen sehingga pengguna tidak bisa kembali ke splash screen dengan tombol kembali (back button).

B. Login Activity



- etUsername dan etPassword: Variabel untuk menyimpan referensi ke elemen input teks untuk nama pengguna dan kata sandi.
- registeredUser dan registeredPass: Variabel untuk menyimpan informasi nama pengguna dan kata sandi yang terdaftar (akan diambil dari activity lain).
- onCreate() adalah metode lifecycle yang dipanggil saat activity pertama kali dibuat. setContentView(R.layout.activity_login) menetapkan layout yang akan ditampilkan, yang dalam hal ini adalah activity_login.xml.
- etUsername dan etPassword: Mencari input teks untuk nama pengguna dan kata sandi.
- btnLogin dan btnToRegister: Mencari tombol untuk login dan untuk menuju halaman registrasi

C. Register Activity

```

1 package com.example.utspemob
2
3 import android.annotation.SuppressLint
4 import android.content.Intent
5 import android.os.Bundle
6 import android.widget.Button
7 import android.widget.EditText
8 import android.widget.Toast
9 import androidx.appcompat.app.AppCompatActivity
10
11 class RegisterActivity : AppCompatActivity() {
12     private lateinit var etRegUsername: EditText
13     private lateinit var etRegPassword: EditText
14     private lateinit var etConfirmPassword: EditText
15
16     override fun onCreate(savedInstanceState: Bundle?) {
17         super.onCreate(savedInstanceState)
18         setContentView(R.layout.activity_register)
19
20         etRegUsername = findViewById(R.id.etRegUsername)
21         etRegPassword = findViewById(R.id.etRegPassword)
22         etConfirmPassword = findViewById(R.id.etConfirmPassword)
23         val btnRegister = findViewById<Button>(R.id.btnRegister)
24
25         btnRegister.setOnClickListener {

```

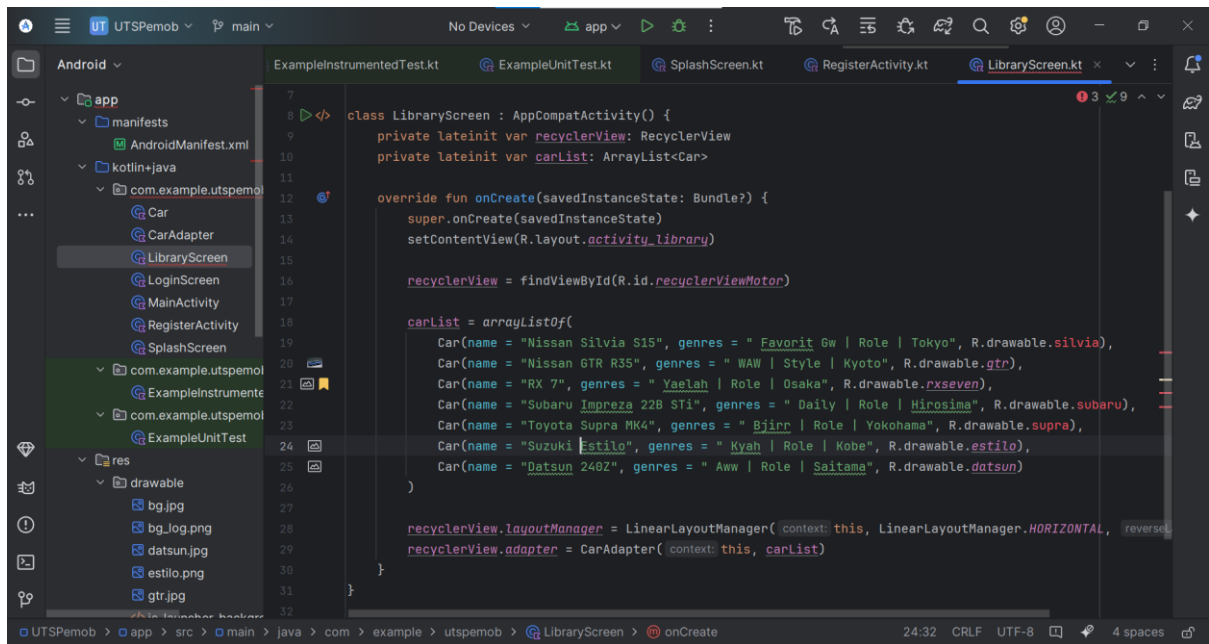
```

26             val user = etRegUsername.text.toString()
27             val pass = etRegPassword.text.toString()
28             val confirm = etConfirmPassword.text.toString()
29
30             if (user.isEmpty() || pass.isEmpty() || confirm.isEmpty()) {
31                 Toast.makeText(context, "Isi dulu semuanya ya.", Toast.LENGTH_SHORT).show()
32             } else if (pass != confirm) {
33                 Toast.makeText(context, "Passwordnya Salah", Toast.LENGTH_SHORT).show()
34             } else {
35                 val intent = Intent(packageContext, LoginActivity::class.java)
36                 intent.putExtra("name", "username", user)
37                 intent.putExtra("name", "password", pass)
38                 Toast.makeText(context, "Daftarnya Berhasil", Toast.LENGTH_SHORT).show()
39                 startActivity(intent)
40                 finish()
41             }
42         }
43     }
44 }
45

```

- etRegUsername: Input teks untuk nama pengguna yang ingin didaftarkan.
- etRegPassword: Input teks untuk kata sandi yang ingin didaftarkan.
- etConfirmPassword: Input teks untuk konfirmasi kata sandi.
- etRegUsername, etRegPassword, etConfirmPassword: Mencari input teks untuk nama pengguna, kata sandi, dan konfirmasi kata sandi.
- btnRegister: Mencari tombol untuk mendaftar.

D. List Chating



- LibraryScreen sebagai sebuah Activity yang akan ditampilkan ke pengguna.
- recyclerView: Komponen UI untuk menampilkan daftar mobil.
- carList: List berisi data mobil bertipe Car.
- ArrayList dari objek Car yang berisi data mobil: nama, deskripsi (genre), dan gambar (R.drawable.* menunjuk ke file gambar di folder res/drawable).
- LinearLayoutManager.HORIZONTAL: Arah scroll horizontal.
- false: Tidak dibalik (dari kiri ke kanan).

Output Aplikasi :

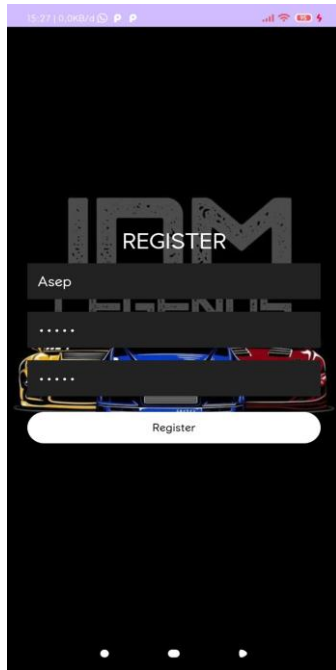
Splash Screen



Login Screen



Regist Screen :



List Screen

