MOBILE PROGRAMING 1

UTS

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

Disusun oleh:

Nama : Muhammad Nabil Hernada

NPM : 23552011064

Kelas : 223 CID B – Teknik Informatika

PROGRAM STUDI TEKNIK INFORMATIKA

UNIVERSITAS TEKNOLOGI BANDUNG

2025

Essay

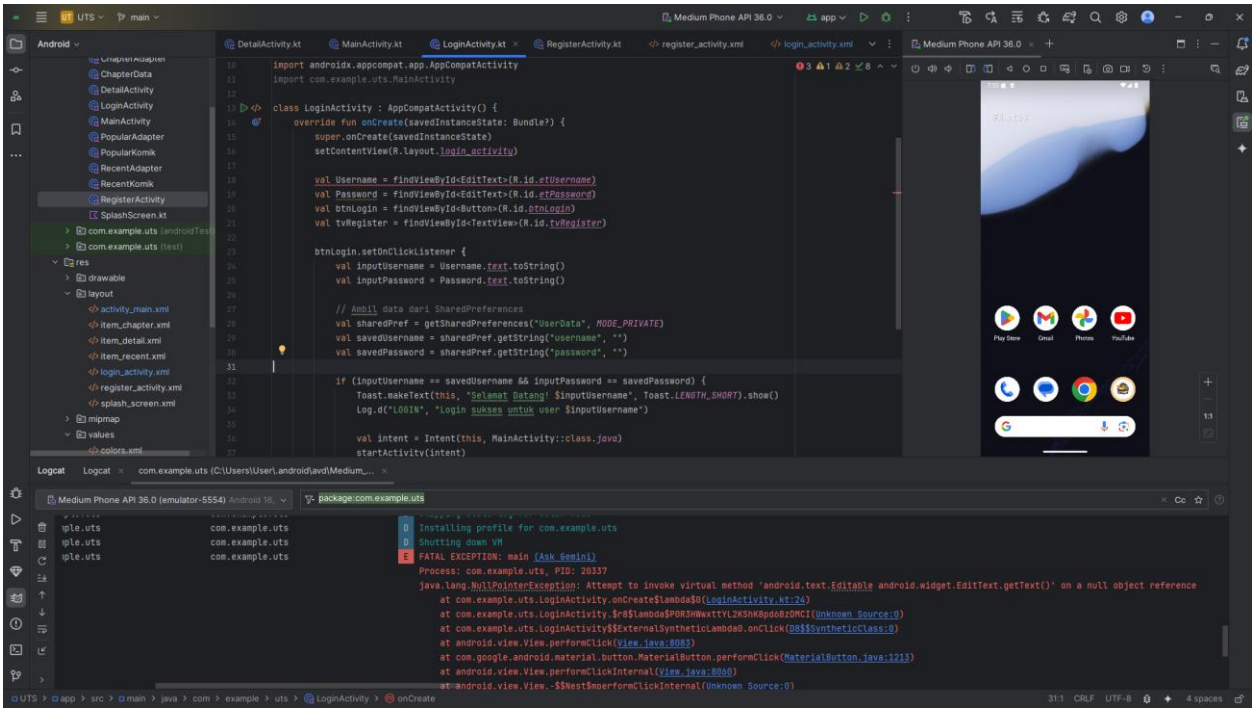
1. Apa fungsi setOnClickListener?
2. Apa syarat pemanggilan method setOnClickListener? Buat contohnya dan screenshot source code nya!
3. Error apa yang terjadi jika file kotlin salah menginisialisasi findViewById atau objek pada xml belum diinisialisasi? Screenshot logcat-nya!
4. Buat sebuah contoh program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

Jawaban :

1. setOnClickListener adalah metode yang digunakan untuk memberikan perintah pada elemen user interface, yang dimana ketika interface tersebut di klik/disentuh maka akan memberikan suatu aksi.
2.
```
val logoutButton = findViewById<Button>(R.id.logoutButton)
```
```
logoutButton.setOnClickListener {
    val intent = Intent(this, LoginActivity::class.java)
    startActivity(intent)
    finish()

    Toast.makeText(this, "Sampai Jumpa $savedUsername !", Toast.LENGTH_SHORT).show()
    Log.d("LOGOUT", "Logout $savedUsername")
}
```
Syaratnya pemanggilan method setOnClickListener yaitu memiliki target objek yang ditandai dengan id, disini contohnya adalah sebuah button dengan id logoutButton.
3. 

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

4. 

   Disini saya menggunakan ID etUsername, tetapi ID tersebut tidak terletak di file xml yang sesuai. ID etUsername seharusnya menyambung pada file **login_activity.xml** tetapi saya menyimpannya di file **activity_main.xml**, maka terjadilah error NullPointerException

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 (MainActivity)

Link Github : https://github.com/Hernada/UTS-Mobile1

- **LOGINACTIVITY**

```kotlin
class LoginActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.login_activity)

        val Username = findViewById<EditText>(R.id.etUsername)
        val Password = findViewById<EditText>(R.id.etPassword)
        val btnLogin = findViewById<Button>(R.id.btnLogin)
        val tvRegister = findViewById<TextView>(R.id.tvRegister)

        btnLogin.setOnClickListener {
            val inputUsername = Username.text.toString()
            val inputPassword = Password.text.toString()

            // Ambil data dari SharedPreferences
            val sharedPref = getSharedPreferences("UserData", MODE_PRIVATE)
            val savedUsername = sharedPref.getString("username", "")
            val savedPassword = sharedPref.getString("password", "")

            if (inputUsername == savedUsername && inputPassword == savedPassword) {
                Toast.makeText(this, "Selamat Datang! $inputUsername", Toast.LENGTH_SHORT).show()
                Log.d("LOGIN", "Login sukses untuk user $inputUsername")

                val intent = Intent(this, MainActivity::class.java)
                startActivity(intent)
                finish()
            } else {
                Toast.makeText(this, "Username atau Password salah", Toast.LENGTH_SHORT).show()
                Log.d("LOGIN", "Login gagal untuk user $inputUsername")
            }
        }

        tvRegister.setOnClickListener {
            val intent = Intent(this, RegisterActivity::class.java)
            startActivity(intent)
        }
    }
}
```

val Username = findViewById<EditText>(R.id.etUsername)

val Password = findViewById<EditText>(R.id.etPassword)

val btnLogin = findViewById<Button>(R.id.btnLogin)

val tvRegister = findViewById<TextView>(R.id.tvRegister)

Menghubungkan komponen di layout XML ke variabel di Kotlin supaya bisa diakses/dimanipulasi lewat kode.

SharedPreferences di sini dipakai buat menyimpan data login saat register. Parameter "UserData" adalah nama file-nya.

- **REGISTERACTIVITY**

```
class RegisterActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.register_activity)

        val buttonRegister = findViewById<Button>(R.id.buttonRegister)
        val editTextUsername = findViewById<EditText>(R.id.editTextUsername)
        val editTextPassword = findViewById<EditText>(R.id.editTextPassword)
        val backLogin = findViewById<Button>(R.id.backToLogin)

        buttonRegister.setOnClickListener {
            val username = editTextUsername.text.toString()
            val password = editTextPassword.text.toString()
            val sharedPref = getSharedPreferences("UserData", MODE_PRIVATE)
            val editor = sharedPref.edit()

            if (username.isEmpty() || password.isEmpty()) {
                Toast.makeText(this, "Username dan Password tidak boleh kosong", Toast.LENGTH_SHORT).show()
                return@setOnClickListener
            } else if (password.length < 8) {
                Toast.makeText(this, "Password harus memiliki minimal 8 karakter", Toast.LENGTH_SHORT).show()
                return@setOnClickListener
            }

            editor.putString("username", username)
            editor.putString("password", password)
            editor.apply()

            Log.d("RegisterActivity", "Username: $username, Password: $password") // Log
            Toast.makeText(this, "Register Berhasil", Toast.LENGTH_SHORT).show() // Toast

            val intent = Intent(this, LoginActivity::class.java)
            startActivity(intent)
            finish()
        }

        backLogin.setOnClickListener {
            val intent = Intent(this, LoginActivity::class.java)
            startActivity(intent)
            finish()
        }
    }
}
```

backLogin.setOnClickListener {

val intent = Intent(this, LoginActivity::class.java)

startActivity(intent)

finish()

}

Jika user klik tombol "Back to Login", langsung pindah ke LoginActivity tanpa menyimpan apapun.

- **SPLASHSCREEN**

```
class SplashActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.splash_screen)

        Handler(Looper.getMainLooper()).postDelayed({
            goToLoginActivity()

        }, 3000L)

    }

    private fun goToLoginActivity() {
        Intent(this, LoginActivity::class.java).also{
            startActivity(it)
            finish();
        }
    }
}
```

enableEdgeToEdge() → bikin konten tampil dari ujung ke ujung layar.

setContentView() → pakai layout splash_screen.xml buat splash screen-nya.

Handler + Looper.getMainLooper() → memastikan kode dijalankan di main thread (UI thread).

postDelayed() → menunda eksekusi selama 3000 milidetik (3 detik).

Setelah delay, panggil fungsi goToLoginActivity().

Panggil **startActivity()** buat pindah halaman.

finish() → tutup SplashActivity agar tidak bisa balik ke splash pakai tombol back.

- **MAINACTIVITY**

```
class MainActivity : AppCompatActivity() {

    private lateinit var recentRecycler: RecyclerView
    private lateinit var popularRecycler: RecyclerView

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

        // Binding RecyclerView
        recentRecycler = findViewById(R.id.recentRecycler)
        popularRecycler = findViewById(R.id.popularRecycler)
        val logoutButton = findViewById<Button>(R.id.logoutButton)

        val sharedPref = getSharedPreferences("UserData", MODE_PRIVATE)
        val savedUsername = sharedPref.getString("username", "")

        // Layout Manager
        recentRecycler.layoutManager = LinearLayoutManager(this, LinearLayoutManager.HORIZONTAL, false)
        popularRecycler.layoutManager = GridLayoutManager(this, 4)

        // Data List
        val recentList = listOf(
            RecentKomik(R.drawable.komika,"Komik A", "Action, Comedy", "Kisah penuh aksi mendebarkan ya
            RecentKomik(R.drawable.komikb,"Komik B", "Romance, Drama", "Sebuah kisah cinta yang penuh l
            RecentKomik(R.drawable.komikc,"Komik C", "Fantasy", "Jelajahi dunia penuh keajaiban, di man
        )

        val popularList = listOf(
            PopularKomik(R.drawable.komikx,"Komik X", "Adventure, Sci-Fi", "Petualangan seru melintasi
            PopularKomik(R.drawable.komiky,"Komik Y", "Thriller, Horror", "Sebuah kisah mencekam yang g
            PopularKomik(R.drawable.komikz,"Komik Z", "Fantasy, Romance", "Di dunia penuh keajaiban, du
            PopularKomik(R.drawable.komikm,"Komik M", "Fantasy, Comedy", "Petualangan di dunia fantasi
            PopularKomik(R.drawable.komikn,"Komik N", "Fantasy, Action", "Sebuah kisah petualangan epik
            PopularKomik(R.drawable.komico,"Komik O", "Drama, Action", "Kisah penuh emosi yang dipenuhi
            PopularKomik(R.drawable.komikp,"Komik P", "Romance, Action", "Dua hati yang saling mencinta
            PopularKomik(R.drawable.komikq,"Komik Q", "Thriller, Action", "Sebuah kisah mencekam penuh
        )

        // Set Adapter
        recentRecycler.adapter = RecentAdapter(recentList)
        popularRecycler.adapter = PopularAdapter(popularList)
```

```
        logoutButton.setOnClickListener {
            val intent = Intent(this, LoginActivity::class.java)
            startActivity(intent)
            finish()

            Toast.makeText(this, "Sampai Jumpa $savedUsername !", Toast.LENGTH_SHORT).show()
            Log.d("LOGOUT", "Logout $savedUsername")
        }
    }
}
```

val sharedPref = getSharedPreferences("UserData", MODE_PRIVATE)

val savedUsername = sharedPref.getString("username", "")

Mengambil data username yang sudah disimpan di SharedPreferences saat register atau login.

Recent Komik → pakai **LinearLayoutManager horizontal**.

Popular Komik → pakai **GridLayoutManager 4 kolom**.(Menampilkan 4 item per-baris)

val recentList = listOf(

RecentKomik(R.drawable.komika,"Komik A", ...)

...

)

List data komik yang ada di bagian recent ditampilkan secara horizontal.

val popularList = listOf(

PopularKomik(R.drawable.komikx,"Komik X", ...)

...

)

List data komik yang ada di bagian popular ditampilkan dalam grid dan per-baris ada 4 item ada.

recentRecycler.adapter = RecentAdapter(recentList)

popularRecycler.adapter = PopularAdapter(popularList)

Pasang adapter ke masing-masing RecyclerView agar bisa menampilkan data list-nya.

logoutButton.setOnClickListener {

val intent = Intent(this, LoginActivity::class.java)

startActivity(intent)

finish()

Toast.makeText(this, "Sampai Jumpa $savedUsername!",

Toast.LENGTH_SHORT).show()

Log.d("LOGOUT", "Logout $savedUsername")

}

Ketika tombol Logout ditekan:

- Pindah ke LoginActivity.
- Tampilkan pesan Toast perpisahan.
- Log.d untuk loging ke Logcat.
- Tutup MainActivity agar tidak bisa kembali pakai tombol Back.

- **DETAILACTIVITY**

```
class DetailActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.item_detail)

        val imgResource = intent.getIntExtra("coverImage", 0)
        val name = intent.getStringExtra("name")
        val genres = intent.getStringExtra("genres")
        val desc = intent.getStringExtra("desc")

        val chapterRecycler: RecyclerView = findViewById(R.id.chapterRecycler)
        chapterRecycler.layoutManager = LinearLayoutManager(this)

        val chapterList = listOf(
            ChapterData("Chapter 1: Awal Petualangan"),
            ChapterData("Chapter 2: Rahasia Terungkap"),
            ChapterData("Chapter 3: Lah Terungkap"),
            ChapterData("Chapter 4: Astaga Terungkap"),
            ChapterData("Chapter 5: Loh Terungkap"),
            ChapterData("Final Chapter 6: Pertempuran Besar")
        )

        val chapterAdapter = ChapterAdapter(chapterList)
        chapterRecycler.adapter = chapterAdapter


        val cover: ImageView = findViewById(R.id.coverImage)
        val txtName: TextView = findViewById(R.id.txtDetailName)
        val txtGenres: TextView = findViewById(R.id.txtDetailGenres)
        val txtDesc: TextView = findViewById(R.id.txtDetailDesc)

        cover.setImageResource(imgResource)
        txtName.text = name
        txtGenres.text = genres
        txtDesc.text = desc
    }
}
```

val imgResource = intent.getIntExtra("coverImage", 0)

val name = intent.getStringExtra("name")

val genres = intent.getStringExtra("genres")

val desc = intent.getStringExtra("desc")

Ambil data yang dilempar dari activity sebelumnya (dari AdapterChapter/Popular pas item diklik). Key-nya harus sama seperti yang dikirim waktu Intent dibuat.

Bikin list chapter manual pakai data class ChapterData.

cover.setImageResource(imgResource)

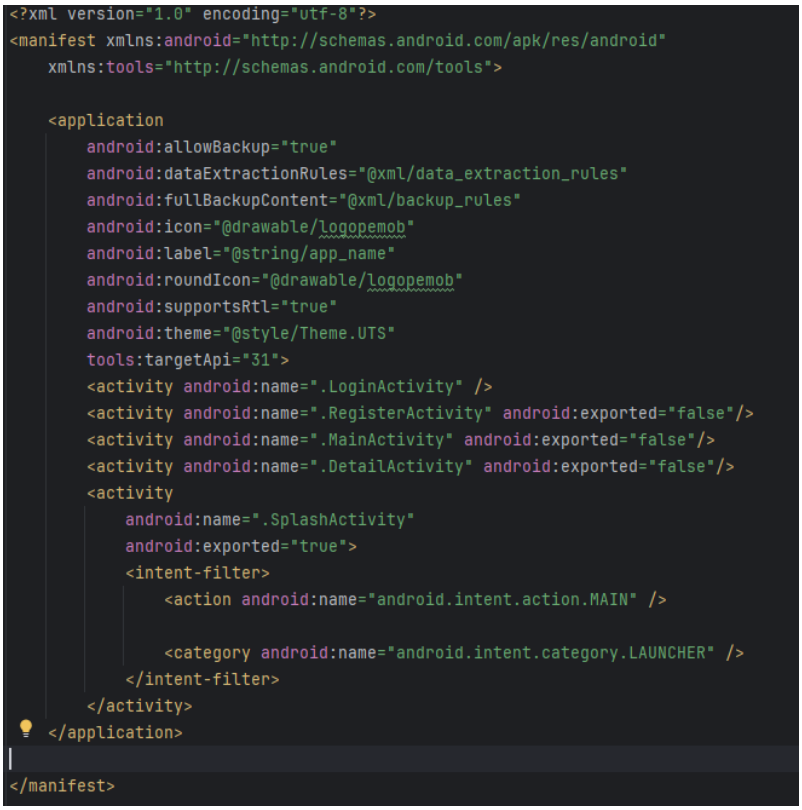
txtName.text = name

txtGenres.text = genres

txtDesc.text = desc

Untuk set nilai-nilai yang tadi diambil dari Intent ke masing-masing view.

- **AndroidManifest.xml**



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@drawable/logopemob"
        android:label="@string/app_name"
        android:roundIcon="@drawable/logopemob"
        android:supportsRtl="true"
        android:theme="@style/Theme.UTS"
        tools:targetApi="31">
        <activity android:name=".LoginActivity" />
        <activity android:name=".RegisterActivity" android:exported="false"/>
        <activity android:name=".MainActivity" android:exported="false"/>
        <activity android:name=".DetailActivity" android:exported="false"/>
        <activity
            android:name=".SplashActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

<activity

android:name=".SplashActivity"

android:exported="true">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

- Activity pertama yang dijalankan saat aplikasi dibuka (MAIN & LAUNCHER)
- <Activity> berfungsi untuk deklarasi komponen halaman di Android
- **android:exported="true"** → diharuskan di API 31+ untuk activity yang bisa di-launcher dari luar (misal icon aplikasi)
- **splashActivity** → Launcher pertama.
- **exported="true"** → wajib untuk activity yang bisa diakses dari luar aplikasi (icon launcher).
- **exported="false"** → aman, hanya internal.
- Icon dan tema aplikasi diatur di sini.