

**UTS PEMROGRAMAN MOBILE**



**UTB**  
**UNIVERSITAS**  
**TEKNOLOGI BANDUNG**

**Nama: Adi Esa Putra**

**NIM: 23552011230**

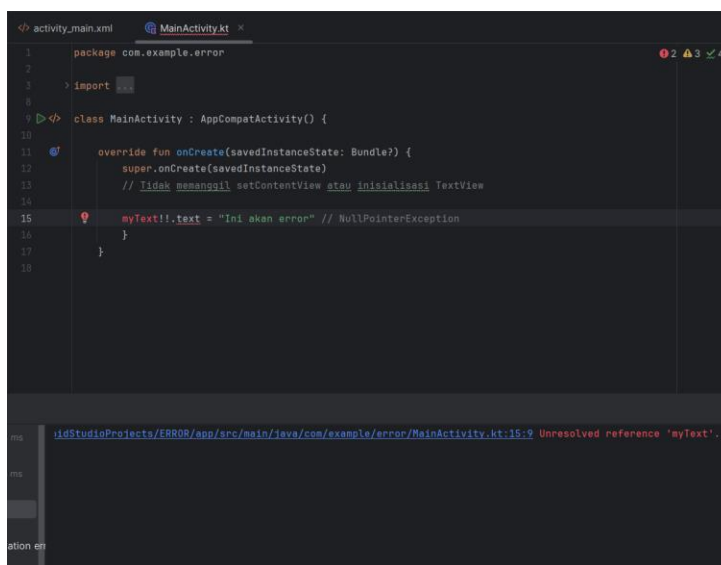
**Kelas: RP-223-PB**

## 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!

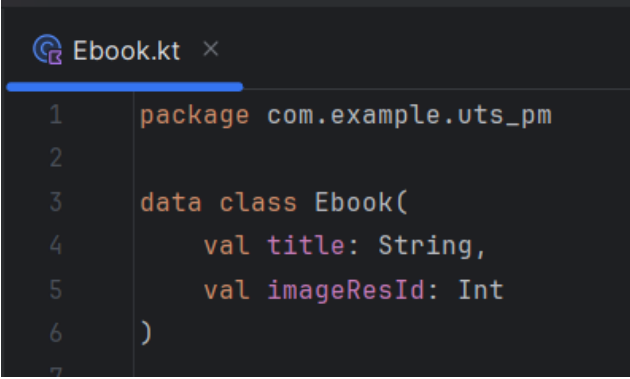
### Jawaban:

1. Fungsi setOnClickListener digunakan untuk menangani event ketika pengguna menekan (klik) sebuah komponen UI seperti Button, TextView dan ImageView. Ketika diklik, kode yang didefinisikan dalam setOnClickListener akan dijalankan.
2.
  - a. Komponen UI seperti Button harus sudah terhubung ke Kotlin melalui findViewById atau View Binding.
  - b. Pemanggilan dilakukan dalam konteks Activity atau Fragment.
  - c. Listener menggunakan lambda atau anonymous class.
3. Jika memanggil findViewById dengan ID yang tidak ada di layout atau menggunakan view sebelum layout ditetapkan, maka biasanya terjadi NullPointerException.



4.

5. Jelaskan fungsi setiap baris source code pada **file kotlin** dan submit dalam bentuk pdf pada Elearning

a. The screenshot shows a code editor window titled 'Ebook.kt'. The code is as follows:  
1 package com.example.uts\_pm  
2  
3 data class Ebook(  
4 val title: String,  
5 val imageResId: Int  
6 )  
7

- Menentukan **package** tempat file ini berada.
- Package digunakan untuk mengelompokkan file Kotlin agar lebih terorganisir.
- Dalam hal ini, file `Ebook.kt` termasuk ke dalam package `com.example.uts_pm`.

b.

```
EbookAdapter.kt
1 package com.example.uts_pm
2
3 import android.content.Context
4 import android.view.LayoutInflater
5 import android.view.View
6 import android.view.ViewGroup
7 import android.widget.ArrayAdapter
8 import android.widget.ImageView
9 import android.widget.TextView
10
11 class EbookAdapter(
12     private val context: Context,
13     private val ebooks: List<Ebook>
14 ) : ArrayAdapter<Ebook>(context, resource: 0, ebooks) {
15
16     override fun getView(position: Int, convertView: View?, parent: ViewGroup): View {
17         val ebook = ebooks[position]
18         val view = convertView ?: LayoutInflater.from(context).inflate(R.layout.item_ebook, parent, attachToRoot: false)
19
20         val imgEbook = view.findViewById<ImageView>(R.id.imgEbook)
21         val tvEbookTitle = view.findViewById<TextView>(R.id.tvEbookTitle)
22
23         imgEbook.setImageResource(ebook.imageResId)
24         tvEbookTitle.text = ebook.title
25
26         return view
27     }
28 }
```

```
import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ArrayAdapter
import android.widget.ImageView
import android.widget.TextView
```

Membuat layout

Mengelola tampilan

Menangani adapter dan widget seperti ImageView, TextView, dll.

```
class EbookAdapter(...) : ArrayAdapter<Ebook>:
```

Melewatkan Context (konteks activity)

Mengambil List<Ebook> (daftar eBook)

Mewarisi ArrayAdapter dengan tipe data Ebook

```
override fun getView(...):
```

`position`: posisi item dalam list

`convertView`: view lama yang bisa digunakan kembali

`parent`: `parent ViewGroup` (biasanya `ListView`)

```
val ebook = ebooks[position]
```

Mengambil objek `Ebook` dari list berdasarkan `position`.

```
val view = convertView ?:
```

```
LayoutInflater.from(context).inflate(R.layout.item_ebook, parent, false)
```

Jika `convertView` null, maka akan membuat (inflate) layout baru dari file XML `item_ebook.xml`.

```
val imgEbook = view.findViewById<ImageView>(R.id.imgEbook)
```

```
val tvEbookTitle = view.findViewById<TextView>(R.id.tvEbookTitle)
```

Menghubungkan komponen layout dengan variabel Kotlin.

```
imgEbook.setImageResource(ebook.imageResId)
```

```
tvEbookTitle.text = ebook.title
```

Menampilkan **gambar** dan **judul** eBook di tampilan item list.

```
return view
```

Mengembalikan `View` untuk ditampilkan pada posisi tertentu di `ListView`.

```
HomeActivity.kt x
1 package com.example.uts_pm
2
3 import android.os.Bundle
4 import android.util.Log
5 import android.widget.ListView
6 import android.widget.Toast
7 import androidx.appcompat.app.AppCompatActivity
8
9 class HomeActivity : AppCompatActivity() {
10
11     private lateinit var ebookListView: ListView
12
13     override fun onCreate(savedInstanceState: Bundle?) {
14         super.onCreate(savedInstanceState)
15         setContentView(R.layout.activity_home)
16
17         ebookListView = findViewById(R.id.ebookListView)
18         val ebookList = listOf(
19             Ebook(title: "Belajar Kotlin Dasar", R.drawable.ic_book),
20             Ebook(title: "Panduan Android Studio", R.drawable.ic_book),
21             Ebook(title: "Kumpulan Cerpen Nusantara", R.drawable.ic_book),
22             Ebook(title: "Tips UI/UX Aplikasi Mobile", R.drawable.ic_book),
23             Ebook(title: "Teknik Membuat Aplikasi Sukses", R.drawable.ic_book),
24             Ebook(title: "Algoritma dan Struktur Data", R.drawable.ic_book),
25             Ebook(title: "Filsafat Ilmu Modern", R.drawable.ic_book),
26             Ebook(title: "Pemrograman Java Lanjutan", R.drawable.ic_book),
27             Ebook(title: "Design Thinking untuk Pemula", R.drawable.ic_book),
28             Ebook(title: "Panduan Freelance Developer", R.drawable.ic_book),
29             Ebook(title: "Machine Learning Sederhana", R.drawable.ic_book),
30             Ebook(title: "Literasi Digital Generasi Z", R.drawable.ic_book),
31             Ebook(title: "Menenal UI Jetpack Compose", R.drawable.ic_book),
32             Ebook(title: "Cerita Rakyat Nusantara", R.drawable.ic_book),
33             Ebook(title: "Kumpulan Puisi Romantis", R.drawable.ic_book)
```

```
import android.os.Bundle
import android.util.Log
import android.widget.ListView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
```

ListView untuk menampilkan daftar

Toast untuk notifikasi ringan

Log untuk mencetak ke Logcat

AppCompatActivity untuk membuat activity

```
class HomeActivity : AppCompatActivity()
```

Mendefinisikan **Activity** bernama **HomeActivity**, yang merupakan halaman utama dari aplikasi.

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

Menyambungkan Activity ini dengan layout `activity_home.xml`.

```
ebookListView = findViewById(R.id.ebookListView)
```

Menghubungkan variabel Kotlin `ebookListView` dengan komponen `ListView` dari XML.

```
val ebookList = listOf(  
  
    Ebook("Belajar Kotlin Dasar", R.drawable.ic_book),  
  
    ...  
  
)
```

Membuat list dari objek `Ebook`, setiap objek memiliki `title` dan `imageResId`.

Ini adalah data yang akan ditampilkan di daftar buku.

```
val adapter = EbookAdapter(this, ebookList)  
ebookListView.adapter = adapter
```

Membuat objek `EbookAdapter`, lalu dipasangkan ke `ListView`.

Adapter ini yang akan menampilkan setiap item (judul + gambar) ke dalam daftar.

```
ebookListView.setOnItemClickListener { _, _, position, _ ->  
    val selectedBook = ebookList[position]  
    Toast.makeText(this, "Membuka: ${selectedBook.title}",  
        Toast.LENGTH_SHORT).show()  
    Log.d("HomeActivity", "Klik: ${selectedBook.title}")  
}
```

Menambahkan listener ketika item daftar diklik.

Menampilkan pesan toast dan mencatat ke Logcat judul buku yang dipilih.

```
LoginActivity.kt ×
1 package com.example.uts_pm
2
3 import android.content.Intent
4 import android.os.Bundle
5 import android.util.Log
6 import android.widget.Button
7 import android.widget.EditText
8 import android.widget.Toast
9 import androidx.activity.enableEdgeToEdge
10 import androidx.appcompat.app.AppCompatActivity
11
12 class LoginActivity : AppCompatActivity() {
13     override fun onCreate(savedInstanceState: Bundle?) {
14         super.onCreate(savedInstanceState)
15         setContentView(R.layout.activity_login)
16
17         val etEmail = findViewById<EditText>(R.id.etEmail)
18         val etPassword = findViewById<EditText>(R.id.etPassword)
19         val btnLogin = findViewById<Button>(R.id.btnLogin)
20         val btnToRegister = findViewById<Button>(R.id.btnToRegister)
21
22         btnLogin.setOnClickListener {
23             Log.d("LoginActivity", "Tombol Login ditekan")
24
25             val email = etEmail.text.toString()
26             val password = etPassword.text.toString()
27
28             val sharedPreferences = getSharedPreferences("UserData", MODE_PRIVATE)
29             val savedEmail = sharedPreferences.getString("email", null)
30             val savedPassword = sharedPreferences.getString("password", null)
31
32             if (email == savedEmail && password == savedPassword) {
33                 startActivity(Intent(this, MainActivity::class.java))
34             } else {
35                 Toast.makeText(this, "Email atau password salah", Toast.LENGTH_SHORT).show()
36             }
37         }
38     }
39 }
```

```
import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
```

Intent: untuk berpindah antar activity.



Log: mencetak pesan ke Logcat.

Toast: menampilkan pesan pop-up singkat.

EditText dan Button: komponen UI.

AppCompatActivity: superclass untuk activity dengan kompatibilitas ke Android lama.

```
class LoginActivity : AppCompatActivity()
```

Mendefinisikan sebuah Activity untuk **login pengguna**.

```
setContentView(R.layout.activity_login)
```

Menetapkan layout yang digunakan (`activity_login.xml`).

```
val etEmail = findViewById<EditText>(R.id.etEmail)
val etPassword = findViewById<EditText>(R.id.etPassword)
val btnLogin = findViewById<Button>(R.id.btnLogin)
val btnToRegister = findViewById<Button>(R.id.btnToRegister)
```

Menghubungkan variabel di Kotlin ke elemen XML agar bisa digunakan.

```
btnLogin.setOnClickListener {
    ...
}
```

Logika tombol login

```
Log.d("LoginActivity", "Tombol Login ditekan")
```

Log tombol ditekan

```
val email = etEmail.text.toString()
val password = etPassword.text.toString()
```

Ambil input user

```
val sharedPref = getSharedPreferences("UserData", MODE_PRIVATE)
val savedEmail = sharedPref.getString("email", null)
val savedPassword = sharedPref.getString("password", null)
```

Ambil data tersimpan dari SharedPreferences

```
if (email == savedEmail && password == savedPassword) {  
    startActivity(Intent(this, HomeActivity::class.java))  
    finish()  
} else {  
    Toast.makeText(this, "Email atau Password salah", Toast.LENGTH_SHORT).show()  
}
```

```

1 package com.example.uts_pm
2
3 > import ...
4
10
11 class RegisterActivity : AppCompatActivity() {
12     override fun onCreate(savedInstanceState: Bundle?) {
13         super.onCreate(savedInstanceState)
14         setContentView(R.layout.activity_register)
15
16         val etName = findViewById<EditText>(R.id.etName)
17         val etEmail = findViewById<EditText>(R.id.etEmail)
18         val etPassword = findViewById<EditText>(R.id.etPassword)
19         val etConfirmPassword = findViewById<EditText>(R.id.etConfirmPassword)
20         val btnRegister = findViewById<Button>(R.id.btnRegister)
21         val btnBackToLogin = findViewById<Button>(R.id.btnBackToLogin)
22
23         btnRegister.setOnClickListener {
24             Log.d(tag: "RegisterActivity", msg: "Tombol Register ditekan")
25             Toast.makeText(context: this, text: "Register ditekan", Toast.LENGTH_SHORT).show()
26
27             val name = etName.text.toString()
28             val email = etEmail.text.toString()
29             val password = etPassword.text.toString()
30             val confirm = etConfirmPassword.text.toString()
31
32             if (name.isNotEmpty() && email.isNotEmpty() && password == confirm) {
33                 val sharedPref = getSharedPreferences(name: "UserData", MODE_PRIVATE)
34                 with(sharedPref.edit()) {
35                     putString("email", email)
36                     putString("password", password)
37                     apply()
38                 }
39                 Toast.makeText(context: this, text: "Berhasil daftar, silakan login", Toast.LENGTH_SHORT).show()

```

```
package com.example.uts_pm
```

Kode ini menunjukkan bahwa file berada di dalam package `com.example.uts_pm`.

```

import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

```

Intent: untuk berpindah antar activity.

Log: mencetak log ke Logcat.

Toast: menampilkan pesan singkat kepada pengguna.

EditText, Button: elemen UI.

AppCompatActivity: kelas dasar untuk activity Android.

```
class RegisterActivity : AppCompatActivity()
```

RegisterActivity adalah activity yang digunakan untuk proses **pendaftaran user baru**.

```
override fun onCreate(savedInstanceState: Bundle?)
```

Method ini dijalankan saat activity pertama kali dibuat. Di sini, komponen UI diinisialisasi dan event listener ditetapkan.

```
val etName = findViewById<EditText>(R.id.etName)
val etEmail = findViewById<EditText>(R.id.etEmail)
val etPassword = findViewById<EditText>(R.id.etPassword)
val etConfirmPassword = findViewById<EditText>(R.id.etConfirmPassword)
val btnRegister = findViewById<Button>(R.id.btnRegister)
val btnBackToLogin = findViewById<Button>(R.id.btnBackToLogin)
```

Menghubungkan variabel dengan komponen pada layout `activity_register.xml`.

```
Log.d("RegisterActivity", "Tombol Register ditekan")
Toast.makeText(this, "Register ditekan", Toast.LENGTH_SHORT).show()
```

Log & Toast

```
val name = etName.text.toString()
val email = etEmail.text.toString()
val password = etPassword.text.toString()
val confirm = etConfirmPassword.text.toString()
```

Ambil Input

```
if (name.isNotEmpty() && email.isNotEmpty() && password == confirm) {
```

Validasi Input

```
val sharedPref = getSharedPreferences("UserData", MODE_PRIVATE)
with(sharedPref.edit()) {
```

```
putString("email", email)
putString("password", password)
apply()
}
```

Simpan ke SharedPreferences

```
Toast.makeText(this, "Berhasil daftar, silakan login", Toast.LENGTH_SHORT).show()
startActivity(Intent(this, LoginActivity::class.java))
finish()
```

Navigasi ke Login

```
SplashActivity.kt x
1 package com.example.uts_pm
2
3 > import ...
9
10 class SplashActivity : AppCompatActivity() {
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         enableEdgeToEdge()
14         setContentView(R.layout.activity_splash)
15
16         Log.d( tag: "SplashActivity", msg: "Splash screen tampil")
17
18         Handler().postDelayed({
19             startActivity(Intent( packageContext: this, LoginActivity::class.java))
20             finish()
21         }, delayMillis: 2000) // 2 detik
22
23     }
24 }
```

```
import android.content.Intent
import android.os.Bundle
import android.util.Log
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import android.os.Handler
```

Intent: Untuk berpindah dari splash screen ke activity berikutnya (Login).

Bundle: Untuk menangani data saat onCreate.

Log: Untuk mencetak log debugging.

enableEdgeToEdge(): Untuk membuat tampilan menyatu hingga ke tepi layar (fit modern UI).

Handler: Untuk menjalankan aksi dengan delay.

```
class SplashActivity : AppCompatActivity()
```

Mendefinisikan class SplashActivity, yaitu activity pertama yang muncul saat aplikasi dijalankan.

```
override fun onCreate(savedInstanceState: Bundle?)
```

Fungsi ini dijalankan saat activity pertama kali dibuka.

```
enableEdgeToEdge()
```

Aktifkan Edge-to-Edge Display

```
setContentView(R.layout.activity_splash)
```

Tampilkan Layout

```
Log.d("SplashActivity", "Splash screen tampil")
```

Tampilkan Log

```
Handler().postDelayed({  
    startActivity(Intent(this, LoginActivity::class.java))  
    finish()  
}, 2000)
```

Delay Sebelum Pindah Halaman