

Disusun Untuk Memenuhi UTS Mata Kuliah

Pemograman Mobile 1

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



Disusun Oleh :

Khoirul umam 23552011428

TIF RP 23 CNS

PROGRAM STUDI TEKNIK INFORMATIKA

UNIVERSITAS TEKNOLOGI BANDUNG

TAHUN 2025

## Essay

1. Apa fungsi getExtra pada intent?
2. Bagaimana cara menerima data dari Activity pertama ke Activity kedua menggunakan intent?
3. Error apa yang terjadi jika file java salah menginisialisasi findViewById atau objek pada xml belum diinisialisasi?
4. Buat sebuah contoh program untuk menampilkan pesan error NullPointerException! Screenshot logcat-nya!
5. Kumpulkan dalam bentuk pdf di Elearning. Soal essay digabung dengan soal studi kasus cek point 7)

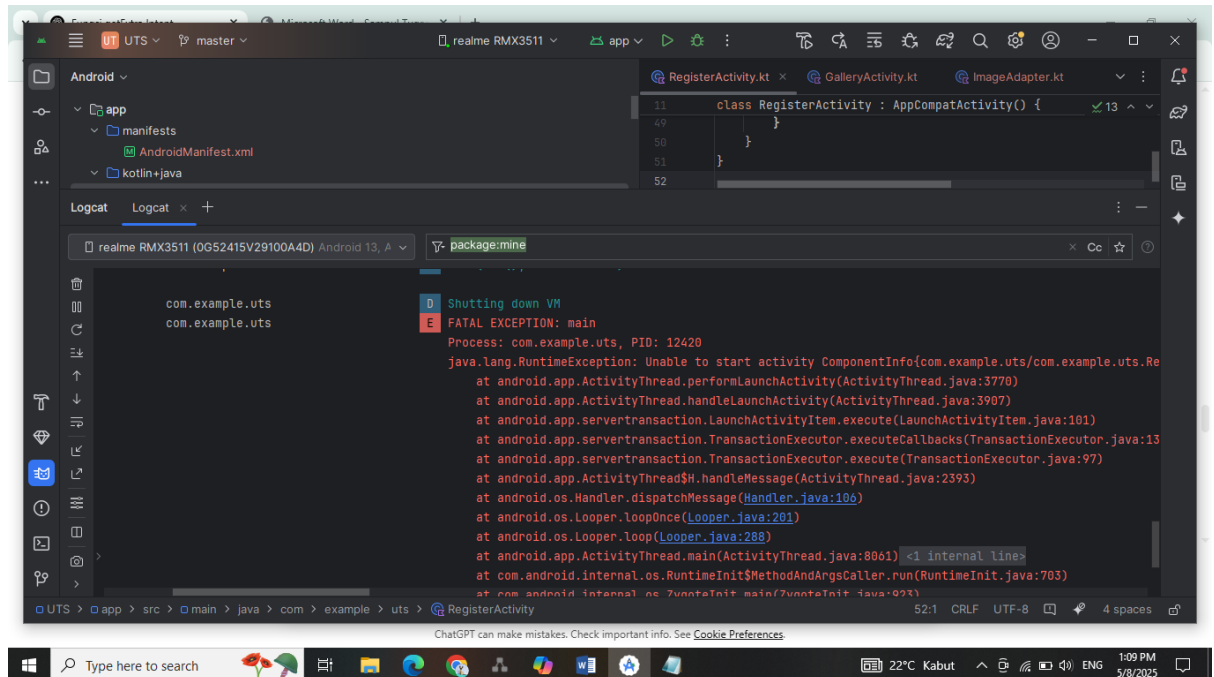
## Studi Kasus

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 Gallery Activity
2. Ketentuan: Silahkan membuat splashscreen dengan baik.
3. Pada Register Activity, minimal terdapat objek: TextView, EditText, Button, ImageView!
4. Tampilkan event Log, Toast dan Toast pada saat Button Register di klik.
5. Gallery Activity terdapat data yang ditampilkan dalam gridview
6. Upload project di Github.
7. **Jelaskan fungsi setiap baris source code pada file kotlin dan submit dalam bentuk pdf pada Elearning**
8. **Link Github harus tercantum pada pdf (point 7)**

## Essay

1. untuk **mengambil data tambahan** (extra data) yang sebelumnya dikirim bersama Intent.
2. Untuk mengirim data antar Activity, gunakan putExtra saat membuat Intent di Activity pertama, lalu ambil data tersebut di Activity kedua dengan getIntent().getXXXExtra(). Pastikan key yang digunakan sama di kedua Activity.

3. Jika `findViewById` salah atau objek XML belum diinisialisasi, akan terjadi **`NullPointerException`** karena objek bernilai `null` saat diakses.
- 4.



#### A. SplashScreenActivity

```
import android.content.Intent
import android.os.Bundle
import android.os.Handler
import android.os.Looper
import androidx.appcompat.app.AppCompatActivity
```

mengimpor kelas-kelas yang dibutuhkan

```
class SplashScreenActivity : AppCompatActivity() {
```

class SplashScreenActivity yang merupakan subclass dari AppCompatActivity. activity pertama yang dijalankan saat aplikasi dibuka.

```
private val splashTime: Long = 3000 // 3 detik
```

waktu splashscreen selama 3 detik

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

memanggil implementasi onCreate dari superclass. Dan menentukan xml yang digunakan

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

Kode menjadwalkan eksekusi setelah delay 3 detik menggunakan `Handler(Looper.getMainLooper()).postDelayed(...)`. Setelah delay,

LoginActivity dimulai dengan startActivity(...) dan SplashScreenActivity ditutup dengan finish() agar tidak bisa kembali ke splash screen.

## B. LoginActivity

```
import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
```

mengimpor kelas yang dibutuhkan

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

Mendefinisikan class LoginActivity yang mewarisi dari AppCompatActivity.

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

Fungsi onCreate() dijalankan saat activity pertama kali dibuat.

```
val btnLogin = findViewById<Button>(R.id.btnLogin)
val editEmail = findViewById<EditText>(R.id.editLoginEmail)
val editPassword = findViewById<EditText>(R.id.editLoginPassword)
val textRegister = findViewById<TextView>(R.id.textRegister)
```

Menghubungkan elemen UI dari XML ke variabel Kotlin agar bisa digunakan dalam kode.

```
btnLogin.setOnClickListener {
    val inputEmail = editEmail.text.toString()
    val inputPassword = editPassword.text.toString()
```

Mengambil input dari pengguna untuk email dan password, lalu mengubahnya menjadi String.

```
if (inputEmail.isNotEmpty() && inputPassword.isNotEmpty()) {
```

input tidak kosong sebelum memproses login.

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

Mengambil data email dan password yang tersimpan sebelumnya di SharedPreferences dengan nama "UserData".

```
if (inputEmail == savedEmail && inputPassword == savedPassword) {
    Toast.makeText(this, "Login Berhasil!", Toast.LENGTH_SHORT).show()
    val intent = Intent(this, GalleryActivity::class.java)
    startActivity(intent)
    finish()
```

Jika data yang diinput cocok dengan data yang tersimpan:

- Tampilkan pesan login berhasil.

- Pindah ke `GalleryActivity`.
- Tutup `LoginActivity` agar tidak bisa dikembalikan dengan tombol back.

```
• else {
    Toast.makeText(this, "Email atau Password salah!",
    Toast.LENGTH_SHORT).show()
}
```

Jika data tidak cocok, tampilkan pesan kesalahan login.

```
else {
    Toast.makeText(this, "Email & Password wajib diisi!",
    Toast.LENGTH_SHORT).show()
}
```

Jika ada input kosong, tampilkan pesan bahwa semua kolom wajib diisi.

```
textRegister.setOnClickListener {
    startActivity(Intent(this, RegisterActivity::class.java))
    finish()
}
```

Ketika teks "Register" ditekan, pindah ke `RegisterActivity` dan tutup halaman login saat ini.

### C. RegisterActivity

```
D. 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
```

mengimpor package yang dibutuhkan

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

Mendeklarasikan `RegisterActivity`, yaitu halaman untuk pengguna mendaftar.

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

- Fungsi `onCreate` dijalankan saat activity dibuat.
- `setContentView(...)` menampilkan layout `activity_register.xml`.

```

• val btnRegister = findViewById<Button>(R.id.btnRegister)
  val editName = findViewById<EditText>(R.id.editName)
  val editEmail = findViewById<EditText>(R.id.editEmail)
  val editPassword = findViewById<EditText>(R.id.editPassword)

```

Menghubungkan elemen-elemen dari layout ke kode Kotlin agar bisa digunakan.

```
btnRegister.setOnClickListener {
```

Menentukan aksi yang terjadi saat tombol **Register** ditekan.

```

val name = editName.text.toString().trim()
val email = editEmail.text.toString().trim()
val password = editPassword.text.toString().trim()

```

Mengambil input dari pengguna dan menghapus spasi di awal/akhir string.

```

if (name.isEmpty() || email.isEmpty() || password.isEmpty()) {
    Toast.makeText(this, "Semua kolom harus diisi!",
    Toast.LENGTH_SHORT).show()
    return@setOnClickListener
}

```

Mengecek apakah ada input yang kosong. Jika iya, tampilkan toast dan hentikan proses.

```

val sharedPref = getSharedPreferences("UserData", MODE_PRIVATE)
with(sharedPref.edit()) {
    putString("name", name)
    putString("email", email)
    putString("password", password)
    apply()
}

```

Menyimpan data user ke SharedPreferences dengan nama "UserData". Ini adalah metode penyimpanan lokal sederhana di Android.

```
Log.d("REGISTER", "User Registered: $name | Email: $email")
```

Mencetak log ke Logcat untuk keperluan debugging. Bisa dilihat di Android Studio.

```

Toast.makeText(this, "Registrasi Berhasil!", Toast.LENGTH_SHORT).show()
Toast.makeText(this, "Selamat datang, $name!", Toast.LENGTH_SHORT).show()

```

Menampilkan dua pesan sambutan sekaligus, yang akan muncul secara berurutan.

```

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

```

Setelah registrasi berhasil, aplikasi berpindah ke halaman GalleryActivity dan menutup RegisterActivity.

#### D. GaleryActivity

```
E. import android.content.Intent
import android.os.Bundle
import android.widget.Button
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
```

mengimpor package yang dibutuhkan

```
class GalleryActivity : AppCompatActivity() {
```

Mendeklarasikan class GalleryActivity, halaman yang menampilkan galeri gambar setelah login atau registrasi.

```
private val imageIds = arrayOf(
    R.drawable.perang1,
    R.drawable.perang2,
    R.drawable.perang3,
    R.drawable.perang4,
    R.drawable.perang5
)
```

Array berisi ID resource gambar dari folder res/drawable.

```
private val imageDescriptions = arrayOf(
    "Invasi Rusia ke Ukraina dimulai tahun 2022",
    "Pasukan Ukraina bertahan di Kyiv",
    "Pengungsi meninggalkan wilayah timur",
    "Dukungan NATO terhadap Ukraina",
    "Kerusakan infrastruktur akibat perang"
)
```

Array deskripsi yang sesuai untuk setiap gambar dalam imageIds.

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

Fungsi onCreate dijalankan saat activity dimulai. Layout activity\_gallery.xml dipasang.

```
val recyclerView = findViewById<RecyclerView>(R.id.galleryRecycler)
recyclerView.layoutManager = LinearLayoutManager(this)
recyclerView.adapter = ImageAdapter(this, imageIds, imageDescriptions)
```

Mengatur **RecyclerView**, komponen yang digunakan untuk menampilkan daftar gambar dengan teks

```
val btnLogout = findViewById<Button>(R.id.btnLogout)
btnLogout.setOnClickListener {
    // Kembali ke LoginActivity
    val intent = Intent(this, LoginActivity::class.java)
    intent.flags = Intent.FLAG_ACTIVITY_CLEAR_TOP or
Intent.FLAG_ACTIVITY_NEW_TASK
    startActivity(intent)
    finish()
}
```

- Ketika tombol logout ditekan:
  - Membuat Intent untuk kembali ke LoginActivity.

E image adapter

```
import android.content.Context
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
```

mengimpor package yang dibutuhkan

```
class ImageAdapter(
    private val context: Context,
    private val images: Array<Int>,
    private val descriptions: Array<String>
) : RecyclerView.Adapter<ImageAdapter.ViewHolder>() {
```

ImageAdapter adalah kelas adapter khusus yang mengatur bagaimana data (gambar dan deskripsi) ditampilkan dalam RecyclerView

```
inner class ViewHolder(view: View) : RecyclerView.ViewHolder(view) {
    val imageView: ImageView = view.findViewById(R.id.imageView)
    val textView: TextView = view.findViewById(R.id.textView)
}
```

ViewHolder adalah inner class yang menyimpan referensi ke komponen dalam item layout (item\_image.xml).

```
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
ViewHolder {
    val view = LayoutInflater.from(context).inflate(R.layout.item_image,
parent, false)
    return ViewHolder(view)
}
```

Membuat dan meng-inflate layout item (dalam hal ini item\_image.xml).



```
override fun onBindViewHolder(holder: ViewHolder, position: Int) {  
    holder.imageView.setImageResource(images[position])  
    holder.textView.text = descriptions[position]  
}
```

Menghubungkan data (images dan descriptions) ke tampilan item pada posisi tertentu (position).

```
override fun getItemCount(): Int = images.size  
}
```

Memberi tahu RecyclerView berapa banyak item yang akan ditampilkan. Jumlah item sama dengan jumlah gambar.

Link Github : <https://github.com/HiuBodas/Ujian-Tengah-semester-Pemrograman-mobile-.git>