

UTS PEMROGRAMAN MOBILE 1

DOSEN PENGAMPU : Nova Agustina, ST., M.Kom.



NAMA : KARINA NUR FADILLA (23552011012)

KELAS : TIF RP 223 CNS-C

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

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)

JAWABAN

1. `getExtra` berfungsi untuk mengambil data yang dikirim dari Activity lain. Data tersebut sebelumnya dikirim menggunakan `putExtra`. Misalnya, jika kita mengirim nama pengguna dari Activity pertama, di Activity kedua kita bisa mengambil data itu dengan `getStringExtra("nama")`. Jadi, `getExtra` digunakan untuk menerima/mengakses data tambahan yang dibawa oleh intent.

2. Cara menerima data dari Activity pertama ke Activity kedua menggunakan intent

Proses pengiriman data dari Activity pertama ke Activity kedua terdiri dari 2 langkah utama:

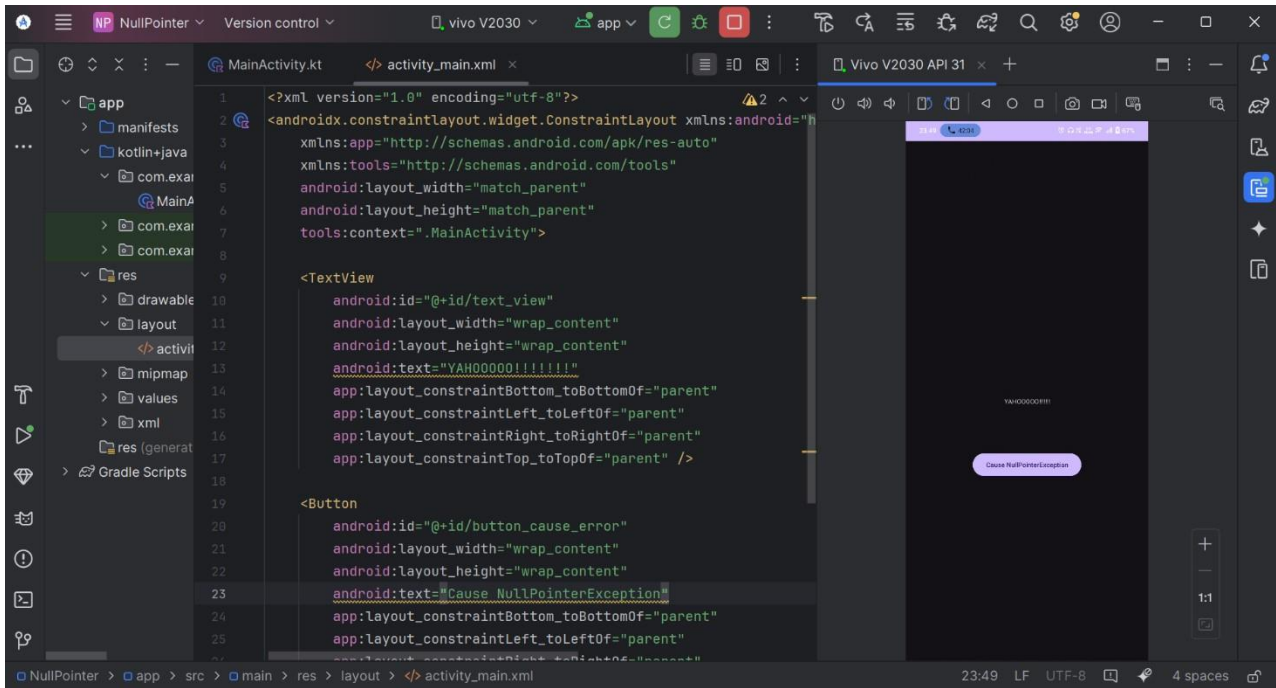
Langkah 1: Pada Activity pertama (pengirim)	Langkah 2: Pada Activity kedua (penerima)
<pre>Intent intent = new Intent(ActivityPertama.this, ActivityKedua.class); intent.putExtra("nama", "Andi"); startActivity(intent);</pre>	<pre>tring nama = getIntent().getStringExtra("nama");</pre>

Jadi, data dikirim dengan `putExtra` dan diterima dengan `getIntent().getStringExtra(...)`.

3. Error yang terjadi jika salah menginisialisasi `findViewById` atau objek XML belum diinisialisasi

Jika kita menggunakan `findViewById` tapi ID yang dicari tidak ada di layout atau belum dipasang layout yang benar dengan `setContentView`, maka akan terjadi error seperti `NullPointerException`. Artinya objek yang kita ambil nilainya null, dan ketika dipakai (misalnya untuk set text atau klik), aplikasi akan crash.

4. Buat sebuah contoh program untuk menampilkan pesan error NullPointerException! Screenshot logcat-nya!



BAGIAN II.

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)

JAWABAN

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

Source code AndroidManifest.xml
<pre><?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:label="CookingApp" android:icon="@drawable/pinky" android:roundIcon="@drawable/ain_cantik" android:theme="@style/Theme.AppCompat.Light.NoActionBar"> <!-- SplashScreen sebagai launcher --> <activity android:name=".SplashScreenActivity" android:exported="true"> <intent-filter> <action android:name="android.intent.action.MAIN" /> <category android:name="android.intent.category.LAUNCHER" /> </intent-filter> </activity> <!-- Daftar activity lainnya --> <activity android:name=".LoginActivity" /> <activity android:name=".RegisterActivity" /> <activity android:name=".GalleryActivity" /> </application> </manifest></pre>

File ini mendeskripsikan struktur dasar aplikasi Android bernama **CookingApp**, dengan splash screen sebagai tampilan awal dan beberapa aktivitas seperti login, pendaftaran, dan galeri.

Gallery Activity.kt
<pre>package com.example.utskarinanrfa import android.os.Bundle import android.widget.ArrayAdapter import android.widget.GridView import androidx.appcompat.app.AppCompatActivity class GalleryActivity : AppCompatActivity() { private val recipes = listOf(</pre>

```

        "Nasi Goreng", "Sate Ayam", "Rendang", "Bakso", "Mie Goreng",
        "Soto", "Gado-Gado", "Ayam Geprek", "Tumis Kangkung"
    )

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

        val gridView = findViewById<GridView>(R.id.gridView)
        gridView.adapter = ArrayAdapter(this, R.layout.item_gallery, R.id.tvItem, recipes)
    }
}

```

Activity ini menampilkan **daftar resep dalam bentuk GridView**. Data resepnya ditentukan langsung di dalam kode (hardcoded) dan ditampilkan menggunakan ArrayAdapter dalam layout yang sudah disiapkan.

Login Activity
<pre> package com.example.utskarinanrfa import android.content.Intent import android.os.Bundle import android.widget.Button import android.widget.TextView import androidx.appcompat.app.AppCompatActivity class LoginActivity : AppCompatActivity() { override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState) setContentView(R.layout.activity_login) val buttonLogin = findViewById<Button>(R.id.btnLogin) val registerLink = findViewById<TextView>(R.id.tvRegister) buttonLogin.setOnClickListener { startActivity(Intent(this, GalleryActivity::class.java)) } registerLink.setOnClickListener { startActivity(Intent(this, RegisterActivity::class.java)) } } } </pre>

Kode LoginActivity.kt menampilkan layar login. Saat tombol **Login** ditekan, pengguna diarahkan ke **GalleryActivity**, dan saat teks **Register** ditekan, pengguna diarahkan ke **RegisterActivity**.

Register Activity
<pre> package com.example.utskarinanrfa 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 class RegisterActivity : AppCompatActivity() { override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState) setContentView(R.layout.activity_register) val btnRegister = findViewById<Button>(R.id.btnRegister) btnRegister.setOnClickListener { val name = findViewById<EditText>(R.id.etName).text.toString() Log.d("RegisterActivity", "Register clicked: \$name") Toast.makeText(this, "Registering \$name", Toast.LENGTH_SHORT).show() startActivity(Intent(this, LoginActivity::class.java)) } } } </pre>

Kode RegisterActivity.kt berfungsi sebagai halaman **pendaftaran pengguna** dalam aplikasi. Saat activity ini dibuka, layout activity_register akan ditampilkan. Di dalamnya terdapat tombol bernama btnRegister yang akan merespons saat ditekan.

Ketika tombol **Register** ditekan, aplikasi akan:

1. Mengambil teks yang diketik pengguna dari EditText bernama etName.
2. Mencetak nama tersebut ke **log** (menggunakan Log.d) untuk tujuan debugging.
3. Menampilkan **pesan Toast** ke pengguna yang menyatakan bahwa proses pendaftaran sedang dilakukan.
4. Mengarahkan pengguna kembali ke halaman **LoginActivity** menggunakan Intent.

Namun, kode ini belum menyimpan data pengguna secara permanen. Artinya, informasi yang diinput tidak akan tersimpan setelah aplikasi ditutup atau berpindah halaman. Ini hanya simulasi proses registrasi secara sederhana.

Splash Screen Activity
<pre> package com.example.utskarinanrfa import android.content.Intent import android.os.Bundle import android.os.Handler import android.os.Looper </pre>

```

import androidx.appcompat.app.AppCompatActivity

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

        Handler(Looper.getMainLooper()).postDelayed({
            val intent = Intent(this, LoginActivity::class.java)
            startActivity(intent)
            finish()
        }, 3000)
    }
}

```

Kode SplashScreenActivity.kt berfungsi sebagai **halaman pembuka (splash screen)** untuk aplikasi. Saat activity ini dijalankan, ia akan menampilkan layout activity_splash_screen, yang biasanya berisi logo atau tampilan awal aplikasi.

Bagian inti dari aktivitas ini menggunakan Handler dengan Looper.getMainLooper() untuk menjalankan **penundaan selama 3 detik (3000 milidetik)**. Setelah waktu tersebut habis, aplikasi akan secara otomatis:

1. Membuat Intent untuk berpindah ke LoginActivity (halaman login).
2. Memulai aktivitas login dengan startActivity(intent).
3. Menutup SplashScreenActivity menggunakan finish() agar pengguna tidak bisa kembali ke halaman splash dengan tombol back.

Dengan begitu, splash screen hanya tampil sekali di awal, lalu pengguna langsung diarahkan ke halaman login setelah beberapa detik. Ini adalah pola umum pada banyak aplikasi untuk memberi kesan profesional atau waktu loading awal.

activity_gallery.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <GridView xmlns:android="http://schemas.android.com/apk/res/android"
        android:background="@drawable/karin_cantik"
        android:id="@+id/gridView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="8dp"
        android:numColumns="2"
        android:verticalSpacing="8dp"
        android:horizontalSpacing="8dp" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

Layout ini dirancang untuk menampilkan daftar item (seperti resep masakan) dalam bentuk grid 2 kolom dengan latar belakang gambar. Setiap item akan ditampilkan menggunakan layout terpisah yang disiapkan dalam ArrayAdapter. Meskipun ConstraintLayout digunakan sebagai root, penggunaannya tidak terlalu diperlukan karena hanya ada satu komponen di dalamnya.

Activity_login.xml
<pre><?xml version="1.0" encoding="utf-8"?> <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout_width="match_parent" android:layout_height="match_parent"> <LinearLayout android:background="@drawable/karin_cantik" android:orientation="vertical" android:gravity="center" android:layout_width="match_parent" android:layout_height="match_parent" android:padding="24dp"> <EditText android:id="@+id/etEmail" android:layout_width="match_parent" android:layout_height="wrap_content" android:hint="Email" android:minHeight="48dp" /> <EditText android:id="@+id/etPassword" android:layout_width="match_parent" android:layout_height="wrap_content" android:hint="Password" android:inputType="textPassword" android:minHeight="48dp" /> <Button android:id="@+id/btnLogin" android:text="Login" android:layout_width="match_parent" android:layout_height="wrap_content"/> <TextView android:id="@+id/tvRegister" android:text="Belum punya akun? Daftar" android:textColor="@android:color/holo_blue_dark" android:layout_marginTop="16dp" android:layout_width="wrap_content" android:layout_height="wrap_content"/> </LinearLayout></pre>


```
</androidx.constraintlayout.widget.ConstraintLayout>
```

Layout ini menyusun form login dengan sederhana dan jelas, terdiri dari input email, password, tombol login, dan tautan untuk pindah ke halaman pendaftaran. Semua komponen ditata dengan rapi dalam LinearLayout vertikal, dan tampil dengan latar belakang gambar yang menarik.

activity_register.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ScrollView
        android:background="@drawable/ain_cantik"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:padding="24dp"
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <ImageView
                android:layout_gravity="center"
                android:src="@drawable/pinky"
                android:layout_width="100dp"
                android:layout_height="100dp"/>

            <TextView
                android:text="Register"
                android:textSize="20sp"
                android:textStyle="bold"
                android:layout_marginTop="16dp"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"/>

            <EditText
                android:id="@+id/etName"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="Full Name"
                android:minHeight="48dp" />

            <EditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
```

```

        android:hint="Email"
        android:minHeight="48dp" />

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Password"
            android:inputType="textPassword"
            android:minHeight="48dp" />

        <Button
            android:id="@+id/btnRegister"
            android:text="Register"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"/>
    </LinearLayout>
</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>

```

Layout ini menyusun tampilan halaman pendaftaran pengguna dengan rapi dan responsif. Menggunakan ScrollView memastikan kenyamanan pada semua ukuran layar. Gaya visual diperkuat dengan gambar latar belakang dan ikon, sementara struktur formulir tetap sederhana dan mudah digunakan. Sangat cocok untuk aplikasi dengan tema menarik dan mudah dioperasikan oleh pengguna baru.

activity_splash_screen.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:background="@drawable/ain_cantik"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <ImageView
            android:src="@drawable/pinky"
            android:layout_centerInParent="true"
            android:layout_width="100dp"
            android:layout_height="100dp"/>

        <TextView
            android:text="Cooking"
            android:textSize="24sp"
            android:textStyle="bold"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="16dp"

```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    </RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Kode XML di atas adalah layout untuk halaman splash screen atau tampilan awal aplikasi. Menggunakan RelativeLayout di dalam ConstraintLayout, dengan elemen-elemen berikut:

- **Background:** Gambar latar @drawable/ain_cantik.
- **ImageView:** Gambar logo @drawable/pinky, yang diposisikan di tengah layar.
- **TextView:** Teks "Cooking" yang juga diposisikan di tengah horizontal dengan ukuran font 24sp dan gaya bold.

Layout ini dirancang untuk menampilkan logo dan nama aplikasi secara sederhana di tengah layar.

Link Github : <https://github.com/karinanrfa17/UTS-s4>