

**LAPORAN UJIAN TENGAH SEMESTER
MATA KULIAH PEMROGRAMAN MOBILE 1**

Dosen pengampu : Nova Agustina, M.Kom



Disusun oleh :

Nama : Arika Azhar

NPM : 23552011408

Kelas : TIF RP 23 CNS A

**FAKULTAS INDUSTRI KREATIF
DEPARTEMEN TEKNIK INFORMATIKA
UNIVERSITAS TEKNOLOGI BANDUNG
TAHUN 2025**

Essay :

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

Jawaban Essay :

1. Fungsi findViewById dalam pemrograman Android digunakan untuk menghubungkan elemen tampilan (View) di file XML dengan kode Java/Kotlin dalam activity atau fragment

Contoh :

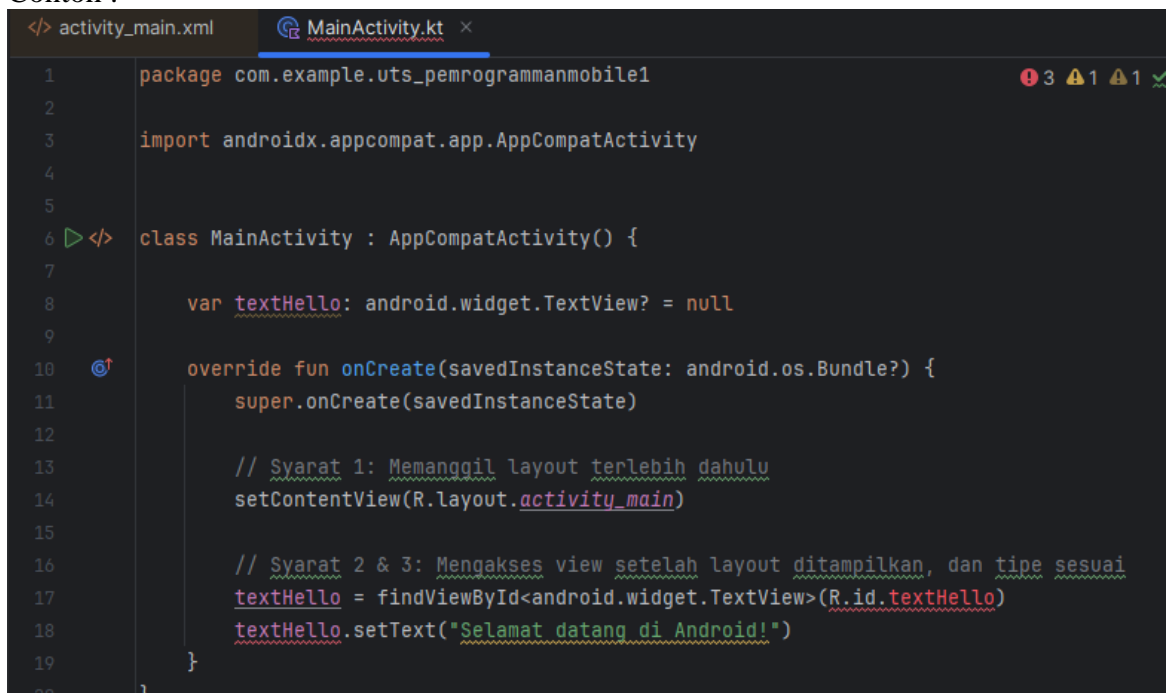
```
TextView textView = findViewById(R.id.textViewWelcome);
textView.setText ("Selamat Datang!");
```

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

Syarat – syarat nya :

1. Harus dipanggil setelah layout di-*inflate* (ditampilkan)
Biasanya dipanggil setelah setContentView(R.layout.nama_layout) di Activity
2. Harus di jalankan dalam konteks (kontext) yang memiliki akses terhadap tampilan misalnya class Activity

Contoh :



```
<?xml version="1.0" encoding="utf-8"?>
<activity_main.xml>
</activity_main.xml>
MainActivity.kt
1 package com.example.uts_pemrogrammanmobile1
2
3 import androidx.appcompat.app.AppCompatActivity
4
5
6 class MainActivity : AppCompatActivity() {
7
8     var textHello: android.widget.TextView? = null
9
10    override fun onCreate(savedInstanceState: android.os.Bundle?) {
11        super.onCreate(savedInstanceState)
12
13        // Syarat 1: Memanggil layout terlebih dahulu
14        setContentView(R.layout.activity_main)
15
16        // Syarat 2 & 3: Mengakses view setelah layout ditampilkan, dan tipe sesuai
17        textHello = findViewById<android.widget.TextView>(R.id.textHello)
18        textHello.setText("Selamat datang di Android!")
19    }
20 }
```

3. Error apa yang terjadi jika file kotlin salah menginisialisasi findViewById atau objek pada xml belum diinisialisasi?

Dalam pengembangan Android menggunakan Kotlin, jika findViewById digunakan secara tidak tepat atau objek belum dihubungkan dengan benar ke elemen di XML, maka aplikasi bisa mengalami error saat dijalankan (runtime) maupun kesalahan saat kompilasi (compile-time).

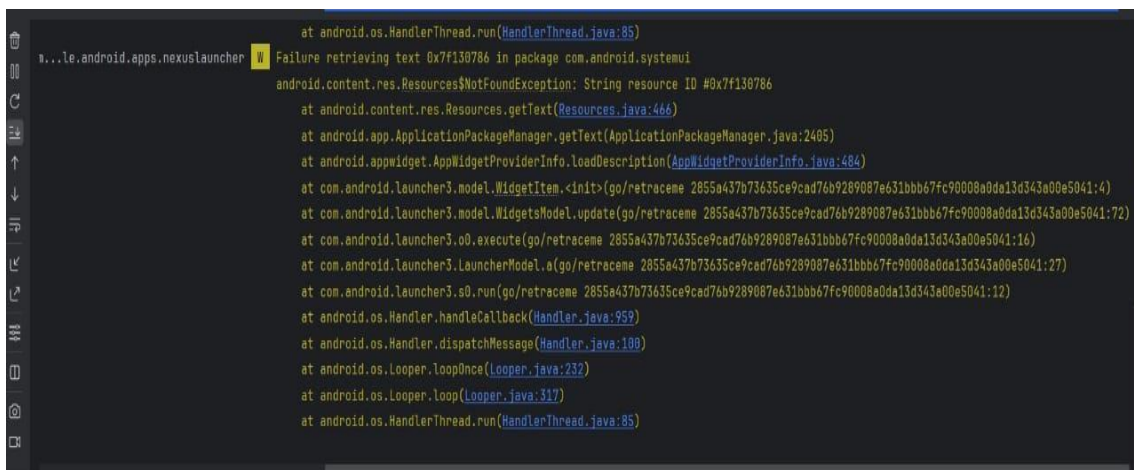
Jenis jenis error dan penyebab nya :

- NullPointerException

Penyebab:

- findViewById dipanggil sebelum setContentView(...)
- findViewById mengembalikan null karena ID tidak cocok dengan komponen yang ada di layout
- Kamu mengakses method/property dari objek yang sebenarnya

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



```
package com.example.uts_pemrogrammanmobile1
```

```
import android.content.res.Resources
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
```

```
class MainActivity : AppCompatActivity() {
```

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

```
        // Memicu Resources.NotFoundException karena string ID tidak ada
        try {
            val errorString = getString(R.string.string_tidak_ada) // Tidak
            // didefinisikan di strings.xml
            Log.d("TestLog", errorString)
        } catch (e: Resources.NotFoundException) {
            Log.e("ResourceError", "Terjadi Resources.NotFoundException", e)
        }
    }
}
```

```
<!-- res/values/strings.xml -->
<resources>
    <string name="app_name">UTS
    Pemrogramman Mobile 1</string>
</resources>
```

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
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. Pada List Chating terdapat data yang ditampilkan dalam listview
6. Upload project di Github.
7. **Jelaskan fungsi setiap baris source code pada file kotlin dan submit dalam bentuk pdf pada Elearning**

JAWABAN :

Link Github : https://github.com/Arika321/UTS_PEMOGRAMMAN-MOBILE1_ARIKA-AZHAR_23552011408

- Judul aplikasi : Soto Bandung

Aplikasi soto bandung adalah aplikasi yang memberikan kemudahan kepada pembeli untuk memudahkan pembelian soto dengan membeli secara digital/online

Aplikasi "Soto Bandung" dapat memiliki berbagai tujuan, antara lain:

1. **Pesan Makanan:** Memudahkan pengguna untuk memesan Soto Bandung dan menu lainnya secara online.
2. **Informasi Menu:** Menyediakan informasi lengkap tentang menu, termasuk harga dan deskripsi makanan.
3. **Portal Komunikasi:** Menghubungkan pelanggan dengan restoran melalui fitur chat untuk menjawab pertanyaan atau menampung permintaan khusus.
4. **Pendaftaran Pengguna:** Memungkinkan pengguna untuk membuat akun, sehingga mereka dapat mengelola pesanan dan profil mereka.
5. **Interaksi Sosial:** Menciptakan komunitas bagi penggemar Soto Bandung untuk berbagi pengalaman dan rekomendasi.

Penjelasan code FILE KOTLIN :

SplashScreen	<pre> package com.example.uts_arika05 import android.content.Intent import android.os.Bundle import android.os.Handler import androidx.appcompat.app.AppCompatActivity class SplashScreenActivity : AppCompatActivity() { override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState) setContentView(R.layout.splash_screen) Handler().postDelayed({ val intent = Intent(this, LoginActivity::class.java) startActivity(intent) finish() }, 3000) // 3 seconds delay } } <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout_width="match_parent" android:layout_height="match_parent" android:orientation="vertical" android:gravity="center" android:background="#FFEB3B"> <ImageView android:layout_width="wrap_content" android:layout_height="wrap_content" android:src="@drawable/your_image" /> <!-- Replace with your image --> <TextView android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="Welcome!!!" android:textSize="24sp" android:textColor="#000000" /> </LinearLayout> </pre>
Register	<pre> <?xml version="1.0" encoding="utf-8"?> <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent" android:layout_height="match_parent" android:background="@drawable/register" tools:ignore="ExtraText"> <TextView android:id="@+id/textView" </pre>

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/register_title" <!-- Use string resource -->
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.5"
app:layout_constraintVertical_bias="0.1" />
```

<!-- Add EditTexts for Username, Password, and Email -->

<EditText

```
    android:id="@+id/username"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Username"
    app:layout_constraintTop_toBottomOf="@+id/textView"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="16dp"
    android:padding="10dp"/>
```

<EditText

```
    android:id="@+id/password"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Password"
    android:inputType="textPassword"
    app:layout_constraintTop_toBottomOf="@+id/username"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="16dp"
    android:padding="10dp"/>
```

<EditText

```
    android:id="@+id/email"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Email"
    app:layout_constraintTop_toBottomOf="@+id/password"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
```

	<pre> android:layout_marginTop="16dp" android:padding="10dp"/> </androidx.constraintlayout.widget.ConstraintLayout> </pre>
Login	<pre> package com.example.uts_arika05 import android.content.Intent import android.os.Bundle import android.widget.Button import android.widget.EditText import android.widget.Toast import androidx.appcompat.app.AppCompatActivity class LoginActivity : AppCompatActivity() { private lateinit var usernameEditText: EditText private lateinit var passwordEditText: EditText private lateinit var loginButton: Button override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState) setContentView(R.layout.login) usernameEditText = findViewById(R.id.username) passwordEditText = findViewById(R.id.password) loginButton = findViewById(R.id.loginButton) loginButton.setOnClickListener { val username = usernameEditText.text.toString() val password = passwordEditText.text.toString() if (username.isNotEmpty() && password.isNotEmpty()) { val intent = Intent(this, MainActivity::class.java) startActivity(intent) } else { Toast.makeText(this, "Please enter username and password", Toast.LENGTH_SHORT).show() } } } } </pre>

	<pre> <?xml version="1.0" encoding="utf-8"?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout_width="match_parent" android:layout_height="match_parent" android:orientation="vertical" android:padding="16dp" android:background="@drawable/list_chat"> <TextView android:id="@+id/message_user" android:layout_width="wrap_content" android:layout_height="wrap_content" android:background="#e1ffc7" android:padding="10dp" android:layout_gravity="end" android:layout_marginBottom="8dp" android:layout_marginTop="8dp" android:visibility="gone"/> <TextView android:id="@+id/message_bot" android:layout_width="wrap_content" android:layout_height="wrap_content" android:background="#d1e3ff" android:padding="10dp" android:layout_gravity="start" android:layout_marginBottom="8dp" android:layout_marginTop="8dp" android:visibility="gone"/> </LinearLayout> </pre>
List_Chat	<pre> package com.example.uts_arika05 import android.os.Bundle import android.widget.AdapterView import android.widget.ImageView import android.widget.ListView import android.widget.TextView import androidx.appcompat.app.AppCompatActivity class list_view : AppCompatActivity() { </pre>


```

private lateinit var listView: ListView

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

    listView = findViewById(R.id.listView)

    val menuItems = arrayListOf(
        MenuItem("Soto Spesial", "25.000", R.drawable.soto_spesial), //
        Ganti dengan nama gambar Anda
        MenuItem("Soto Ayam", "20.000", R.drawable.soto_ayam), //
        Ganti dengan nama gambar Anda
        MenuItem("Soto", "15.000", R.drawable.soto) // Ganti
        dengan nama gambar Anda
    )

    val adapter = MenuAdapter(this, menuItems)
    listView.adapter = adapter
}

private class MenuAdapter(
    private val context: list_view,
    private val menuItems: List<MenuItem>
) : ArrayAdapter<MenuItem>(context, 0, menuItems) {

    override fun getView(position: Int, convertView:
    android.view.View?, parent: android.view.ViewGroup):
    android.view.View {
        val itemView = convertView ?:
        android.view.LayoutInflater.from(context).inflate(R.layout.list_item,
        parent, false)

        val menuItem = menuItems[position]
        val menuName: TextView =
        itemView.findViewById(R.id.menuName)
        val menuPrice: TextView =
        itemView.findViewById(R.id.menuPrice)
        val imageView: ImageView =
        itemView.findViewById(R.id.imageView)

```

```
        menuName.text = menuItem.name
        menuPrice.text = menuItem.price
        imageView.setImageResource(menuItem.imageResId)

        return itemView
    }
}

data class MenuItem(val name: String, val price: String, val
imageResId: Int)
}
```

