

TUGAS UTS MAHASISWA

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



Disusun oleh:

Nama : Naufal Eka Putra Pratama

Kelas : TIF RP 23 CID B

NPM :23552011445

**PROGRAM STUSY TEKNIK INFORMATIKA
UNIVERSITAN TEKOLOGI BANDUNG TAHUN
2024**

1. Apa fungsi setOnClickListener?

Jawaban : setOnClickListener adalah method yang digunakan dalam Android untuk memberikan aksi atau tindakan pada komponen UI seperti Button, ImageView, atau TextView ketika diklik oleh pengguna.

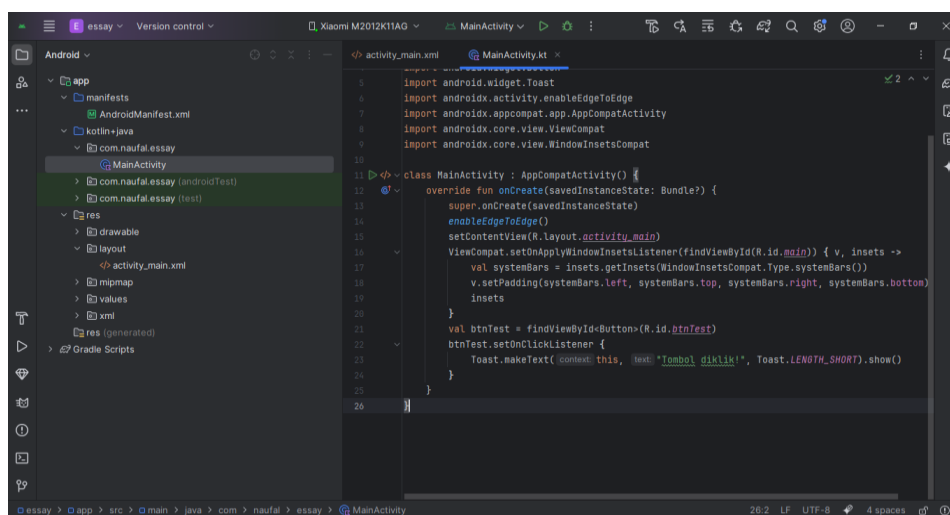
2. Apa syarat pemanggilan setOnClickListener? Berikan contoh dan screenshot kode

Jawaban : Syaratnya adalah

- Komponen yang dipanggil **sudah terhubung** ke kode Kotlin melalui findViewById.
- Komponen UI seperti Button atau ImageView memiliki id di XML.

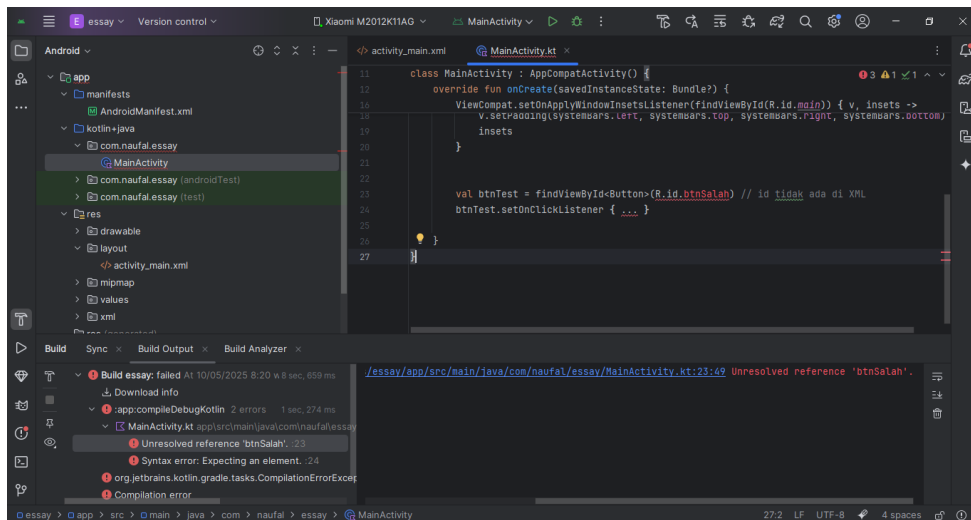
```
<Button
    android:id="@+id/btnTest"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Klik Saya" />
```

```
val btnTest = findViewById<Button>(R.id.btnTest)
btnTest.setOnClickListener {
    Toast.makeText(this, "Tombol diklik!", Toast.LENGTH_SHORT).show()
}
```



3. Error apa yang terjadi jika salah menginisialisasi findViewById atau objek XML belum ada?

Jawaban : **Error:** NullPointerException atau Unresolved reference.



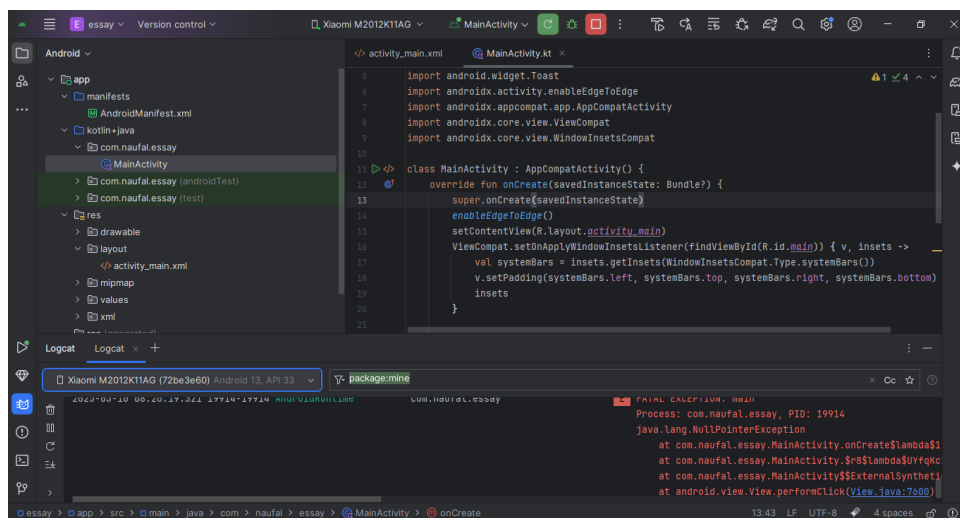
4. Buat contoh program yang menampilkan pesan error NullPointerException

Jawaban : codenya

```
val btnTest = findViewById<Button>(R.id.btnTest)
btnTest.setOnClickListener {
    val nama: String? = null
    val panjang = nama!!.length
    Toast.makeText(this, "Panjang nama: $panjang",
    Toast.LENGTH_SHORT).show()
}
```

<Button

```
android:id="@+id/btnTest"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Tes NullPointerException" />
```



5. Jelaskan Fungsi Setiap Baris source code pada file kotlin

Jawaban : file kotlin No 4

```
package com.naufal.essay
```

Menentukan package tempat file ini berada (biasanya mengikuti struktur folder project).

```
import android.os.Bundle
import android.widget.Button
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
```

Mengimpor berbagai class dan fungsi dari Android SDK:

- Bundle untuk menyimpan state.
- Button dan Toast untuk UI dan notifikasi.
- enableEdgeToEdge, ViewCompat, WindowInsetsCompat digunakan untuk menyesuaikan layout terhadap area sistem (fitur baru di Android 13+).

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

Mendeklarasikan MainActivity sebagai turunan dari AppCompatActivity agar bisa digunakan sebagai layar (Activity) dalam aplikasi.

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

Fungsi lifecycle utama. Dipanggil saat activity pertama kali dibuat.

```
super.onCreate(savedInstanceState)
```

Memanggil method onCreate dari superclass (AppCompatActivity) agar proses bawaan tetap berjalan.

```
enableEdgeToEdge()
```

Mengaktifkan mode **layar penuh** modern (menggunakan seluruh layar hingga ke status bar).

```
setContentView(R.layout.activity_main)
```

Menentukan tampilan activity ini dengan file layout activity_main.xml.

```
ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets  
->
```

Memasang listener untuk menyesuaikan padding layout agar tidak tertimpa status bar atau navigation bar.

```
val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())  
v.setPadding(systemBars.left, systemBars.top, systemBars.right,  
systemBars.bottom)  
insets
```

Mengatur padding tampilan agar menghindari area sistem (status bar, nav bar).

```
val nama: String? = null
```

Mendeklarasikan variabel nama bertipe String? (nullable) dan memberi nilai null.

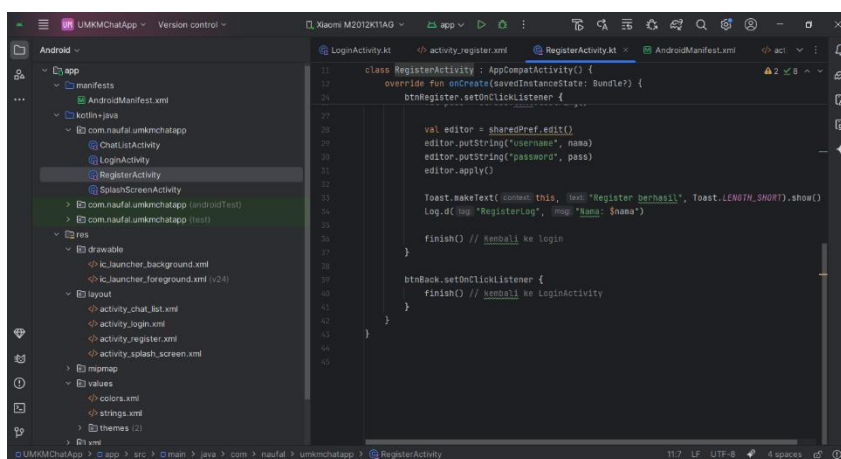
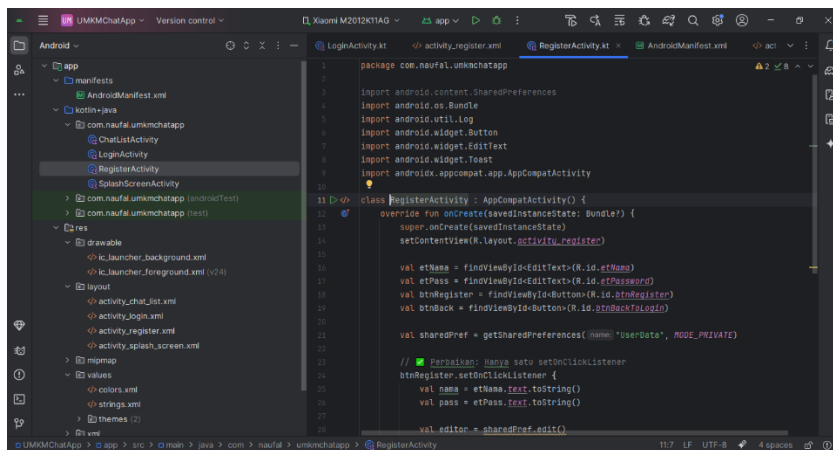
```
val panjang = nama!!.length
```

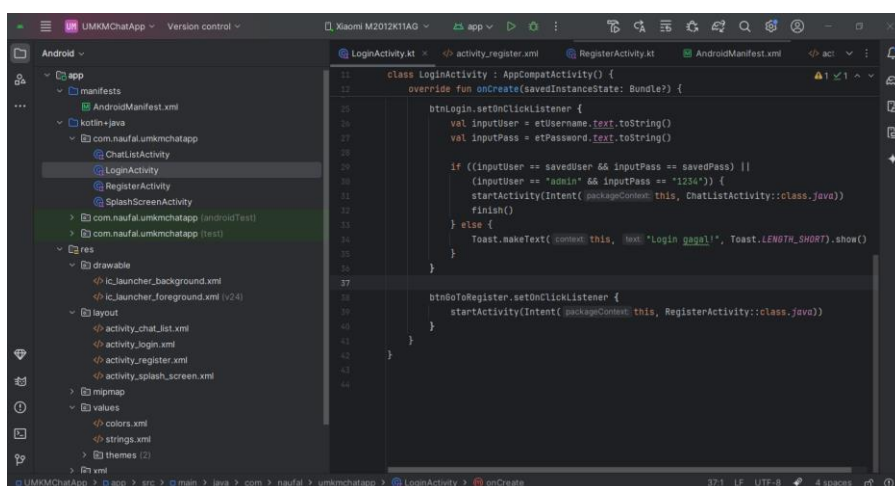
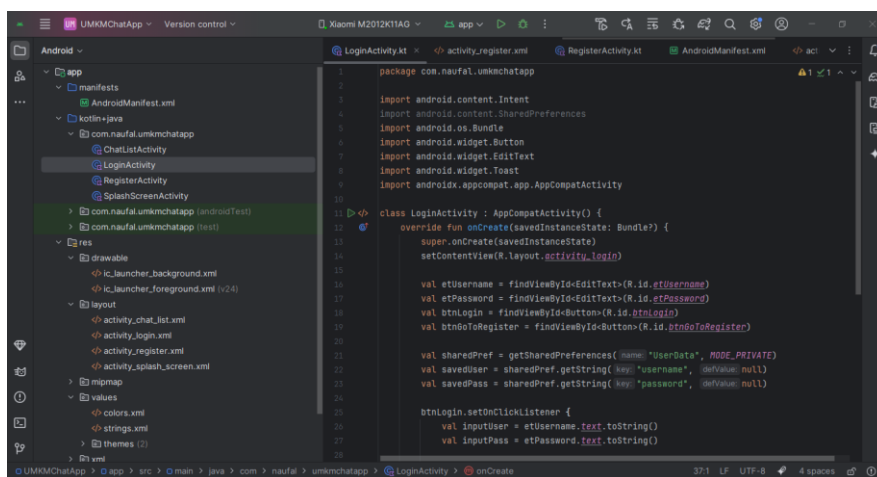
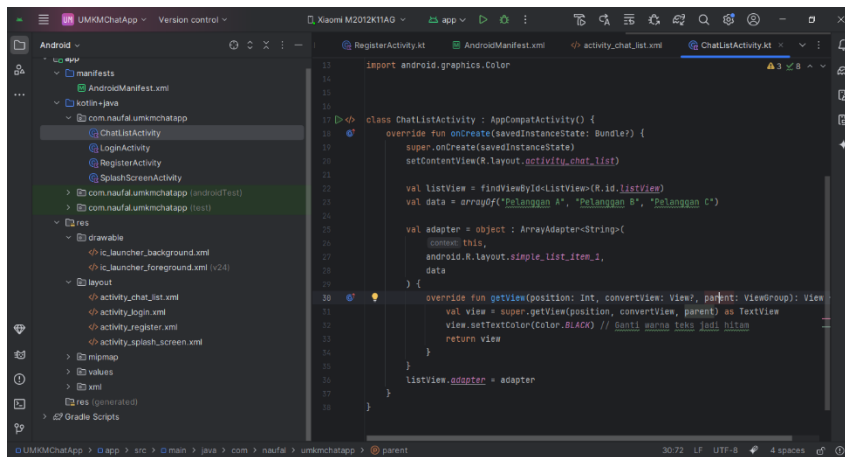
Memaksa mengakses .length dari nama yang null. Karena menggunakan !!, maka akan langsung crash dan melempar NullPointerException.

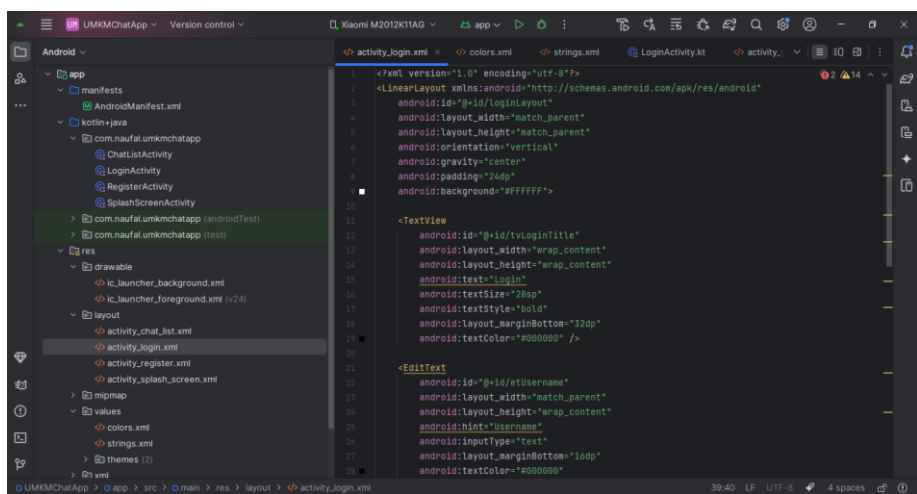
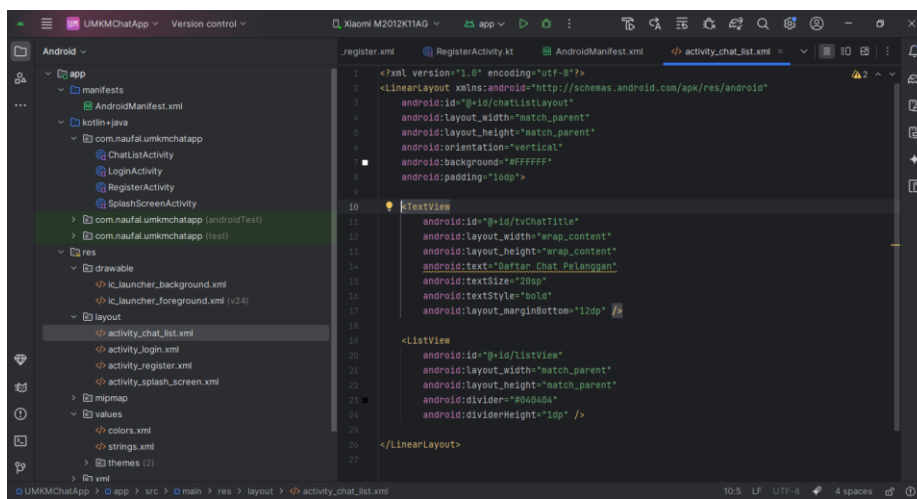
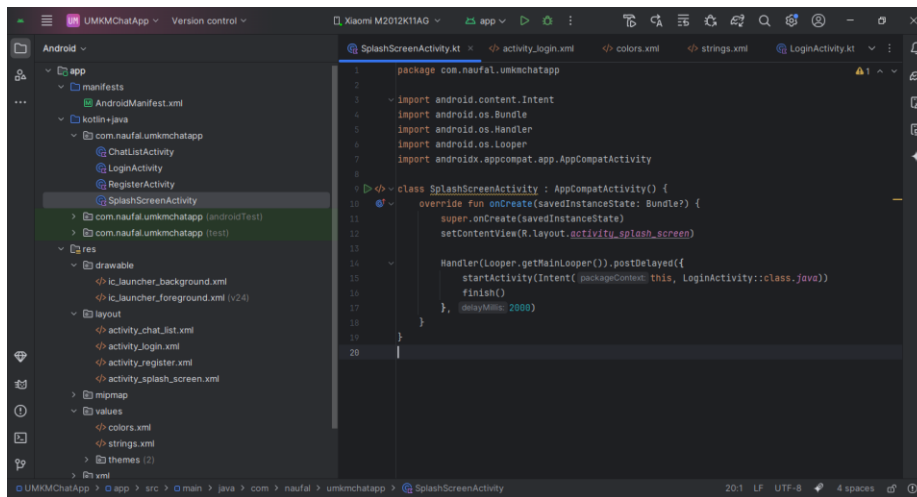
Studi Kasus

Aplikasi **UMKM Chat App** adalah aplikasi Android sederhana berbasis **Kotlin + XML** yang dirancang untuk mendukung usaha kecil (UMKM) dalam komunikasi dengan pelanggan.

Ini adalah Screenshoot code aplikasi





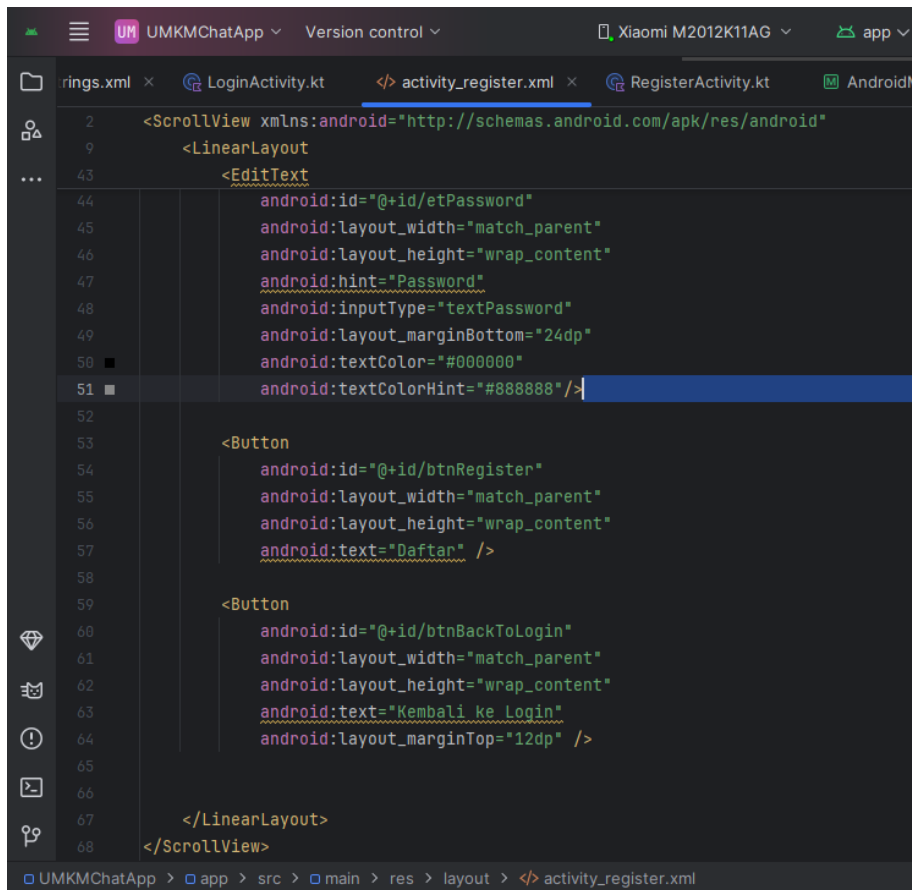


```
UMKMChatApp Version control Xiaomi M2012K11AG app
rings.xml LoginActivity.kt activity_register.xml RegisterActivity.kt AndroidManifest.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
3     android:id="@+id/registerScrollView"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:padding="24dp"
7     android:background="#FFFFFF">
8
9     <LinearLayout
10         android:layout_width="match_parent"
11         android:layout_height="wrap_content"
12         android:orientation="vertical"
13         android:gravity="center_horizontal">
14
15         <TextView
16             android:id="@+id/tvRegisterTitle"
17             android:layout_width="wrap_content"
18             android:layout_height="wrap_content"
19             android:text="Register Akun"
20             android:textSize="24sp"
21             android:textStyle="bold"
22             android:layout_marginBottom="24dp"
23             android:textColor="#000000" />
24
25         <ImageView
26             android:id="@+id/imgLogo"
27             android:layout_width="120dp"
28             android:layout_height="120dp"

```

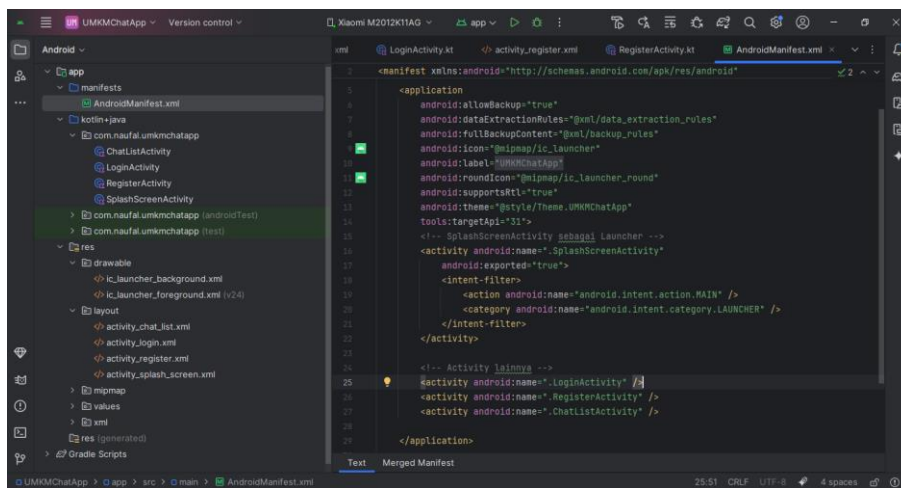
```
UMKMChatApp Version control Xiaomi M2012K11AG app
rings.xml LoginActivity.kt activity_register.xml RegisterActivity.kt AndroidManifest.xml
2 <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
9 <LinearLayout
25 <ImageView
26     android:id="@+id/imgLogo"
27     android:layout_width="120dp"
28     android:layout_height="120dp"
29     android:src="@mipmap/ic_launcher"
30     android:layout_marginBottom="24dp" />
31
32 <EditText
33     android:id="@+id/etNama"
34     android:layout_width="match_parent"
35     android:layout_height="wrap_content"
36     android:hint="Nama"
37     android:inputType="textPersonName"
38     android:layout_marginBottom="16dp"
39     android:textColor="#000000"
40     android:textColorHint="#888888" />
41
42
43 <EditText
44     android:id="@+id/etPassword"
45     android:layout_width="match_parent"
46     android:layout_height="wrap_content"
47     android:hint="Password"
48     android:inputType="textPassword"
49     android:layout_marginBottom="24dp"
50     android:textColor="#AAAAAA"

```

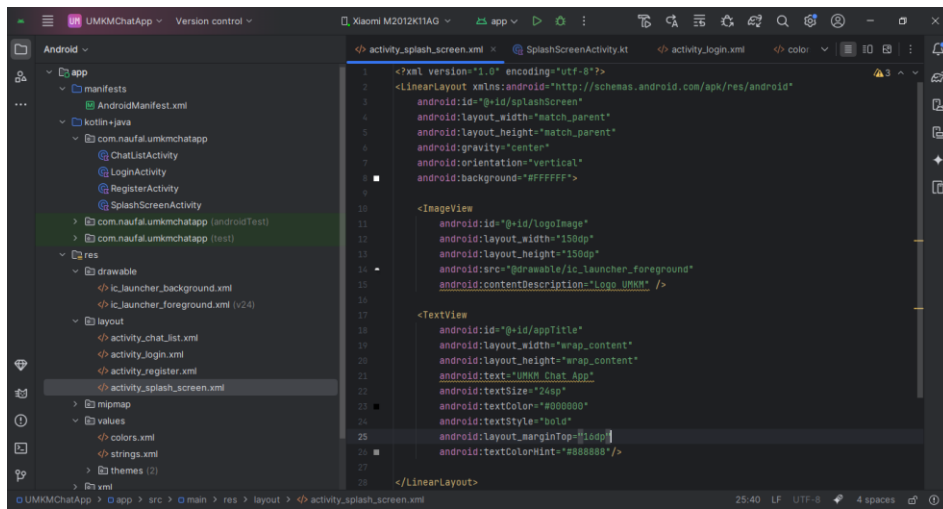
```
2 <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
9 <LinearLayout
43 <EditText
44     android:id="@+id/etPassword"
45     android:layout_width="match_parent"
46     android:layout_height="wrap_content"
47     android:hint="Password"
48     android:inputType="textPassword"
49     android:layout_marginBottom="24dp"
50     android:textColor="#000000"
51     android:textColorHint="#888888" />
52
53 <Button
54     android:id="@+id/btnRegister"
55     android:layout_width="match_parent"
56     android:layout_height="wrap_content"
57     android:text="Daftar" />
58
59 <Button
60     android:id="@+id/btnBackToLogin"
61     android:layout_width="match_parent"
62     android:layout_height="wrap_content"
63     android:text="Kembali ke Login"
64     android:layout_marginTop="12dp" />
65
66
67 </LinearLayout>
68 </ScrollView>
```

UMKMChatApp > app > src > main > res > layout > activity_register.xml

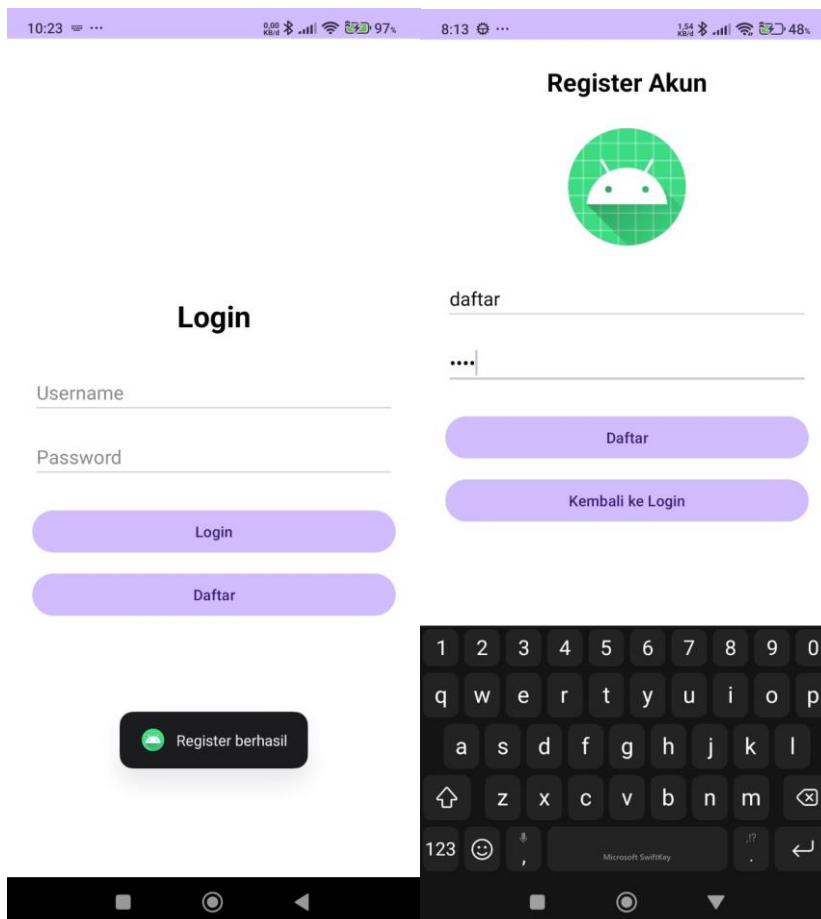


```
7 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
8
9 <application
10     android:allowBackup="true"
11     android:dataExtractionRules="@xml/data_extraction_rules"
12     android:fullBackupContent="@xml/backup_rules"
13     android:icon="@mipmap/ic_launcher"
14     android:label="@string/app_name"
15     android:roundIcon="@mipmap/ic_launcher_round"
16     android:supportRtl="true"
17     android:theme="@style/Theme.UMKMChatApp"
18     tools:targetApi="31"
19     <!-- SplashScreenActivity sebagai Launcher -->
20     <activity android:name=".SplashScreenActivity"
21         android:exported="true"
22         <intent-filter>
23             <action android:name="android.intent.action.MAIN" />
24             <category android:name="android.intent.category.LAUNCHER" />
25         </intent-filter>
26     </activity>
27
28     <!-- Activity lainnya -->
29     <activity android:name=".LoginActivity" />
30     <activity android:name=".RegisterActivity" />
31     <activity android:name=".ChatListActivity" />
32
33 </application>
```

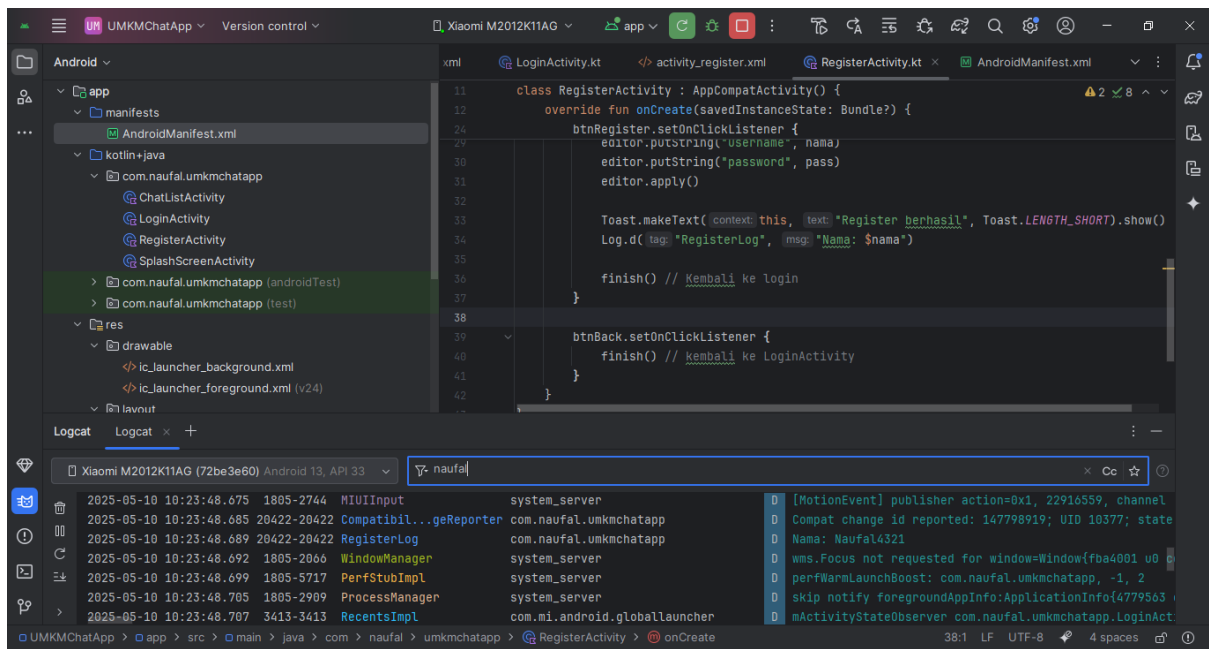
UMKMChatApp > app > src > main > AndroidManifest.xml



Evenlog saat mengklik daftar



Logcat saat klik tombol daftar dengan username Naufal4321



Penjelasan tiap baris code pada file kotlin

SplashScreenActivity.kt

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

Membuat class SplashScreenActivity sebagai Activity awal

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

Fungsi onCreate dipanggil saat activity pertama kali dibuat.

```
        super.onCreate(savedInstanceState)
```

Memanggil implementasi dari superclass (*AppCompatActivity*).

```
        setContentView(R.layout.activity_splash_screen)
```

Menampilkan tampilan dari activity_splash_screen.xml.

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

Menunda 2 detik sebelum berpindah ke LoginActivity dan menutup splash screen.

LoginActivity.kt

```
val etUsername = findViewById<EditText>(R.id.etUsername)
val etPassword = findViewById<EditText>(R.id.etPassword)
val btnLogin = findViewById<Button>(R.id.btnLogin)
val btnGoToRegister = findViewById<Button>(R.id.btnGoToRegister)
```

Menghubungkan komponen UI dari XML ke kode Kotlin.

```
val sharedPref = getSharedPreferences("UserData", MODE_PRIVATE)
val savedUser = sharedPref.getString("username", null)
val savedPass = sharedPref.getString("password", null)
```

Mengambil data username dan password yang sebelumnya disimpan di SharedPreferences.

```
btnLogin.setOnClickListener {
```

Menentukan aksi saat tombol login ditekan.

```
    val inputUser = etUsername.text.toString()
    val inputPass = etPassword.text.toString()
```

Mengambil inputan user.

```
    if ((inputUser == savedUser && inputPass == savedPass) || ...)
```

Mengecek apakah login cocok dengan data register atau admin default.

```
startActivity(Intent(this, ChatListActivity::class.java))  
finish()
```

Jika benar, pindah ke halaman ChatListActivity.

```
Toast.makeText(this, "Login gagal!", Toast.LENGTH_SHORT).show()
```

Jika salah, tampilkan notifikasi gagal.

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

Jika ditekan, pindah ke RegisterActivity.

RegisterActivity.kt

```
val etNama = findViewById<EditText>(R.id.etNama)  
val etPass = findViewById<EditText>(R.id.etPassword)  
val btnRegister = findViewById<Button>(R.id.btnRegister)  
val btnBack = findViewById<Button>(R.id.btnBackToLogin)
```

Menghubungkan input nama, password, dan tombol-tombol ke Kotlin.

```
val sharedPref = getSharedPreferences("UserData", MODE_PRIVATE)
```

Membuka file penyimpanan lokal.

```
btnRegister.setOnClickListener {  
    val nama = etNama.text.toString()  
    val pass = etPass.text.toString()
```

Mengambil input dari user.

```
val editor = sharedPref.edit()  
editor.putString("username", nama)  
editor.putString("password", pass)  
editor.apply()
```

Menyimpan data ke SharedPreferences.

```
Toast.makeText(this, "Register berhasil", Toast.LENGTH_SHORT).show()  
Log.d("RegisterLog", "Nama: $nama")
```

Menampilkan notifikasi dan log debug.

```
finish()
```

Menutup halaman register (kembali ke login).

```
btnBack.setOnClickListener {  
    finish()  
}
```

Tombol kembali juga menutup halaman.

ChatListActivity.kt

```
val listView = findViewById<ListView>(R.id.listView)
```

Menghubungkan ListView dari layout ke Kotlin.

```
val data = arrayOf("Pelanggan A", "Pelanggan B", "Pelanggan C")
```

Data dummy chat pelanggan.

```
val adapter = object : ArrayAdapter<String>(  
    this,  
    android.R.layout.simple_list_item_1,  
    data  
)
```

Membuat objek ArrayAdapter khusus (anonymous class) untuk menampilkan array data ke dalam ListView, menggunakan layout bawaan simple_list_item_1.

```
override fun getView(position: Int, convertView: View?, parent: ViewGroup): View {
```

Override method `getView` agar bisa mengubah tampilan masing-masing item list (baris).

```
val view = super.getView(position, convertView, parent) as TextView
```

Memanggil method asli `getView()` untuk mendapatkan tampilan item, lalu memastikan bahwa itu adalah `TextView`.

```
view.setTextColor(Color.BLACK)
```

Mengatur warna teks item menjadi hitam agar terlihat jelas di background putih.

```
return view
```

Mengembalikan tampilan `TextView` yang sudah diubah warnanya ke dalam `ListView`.

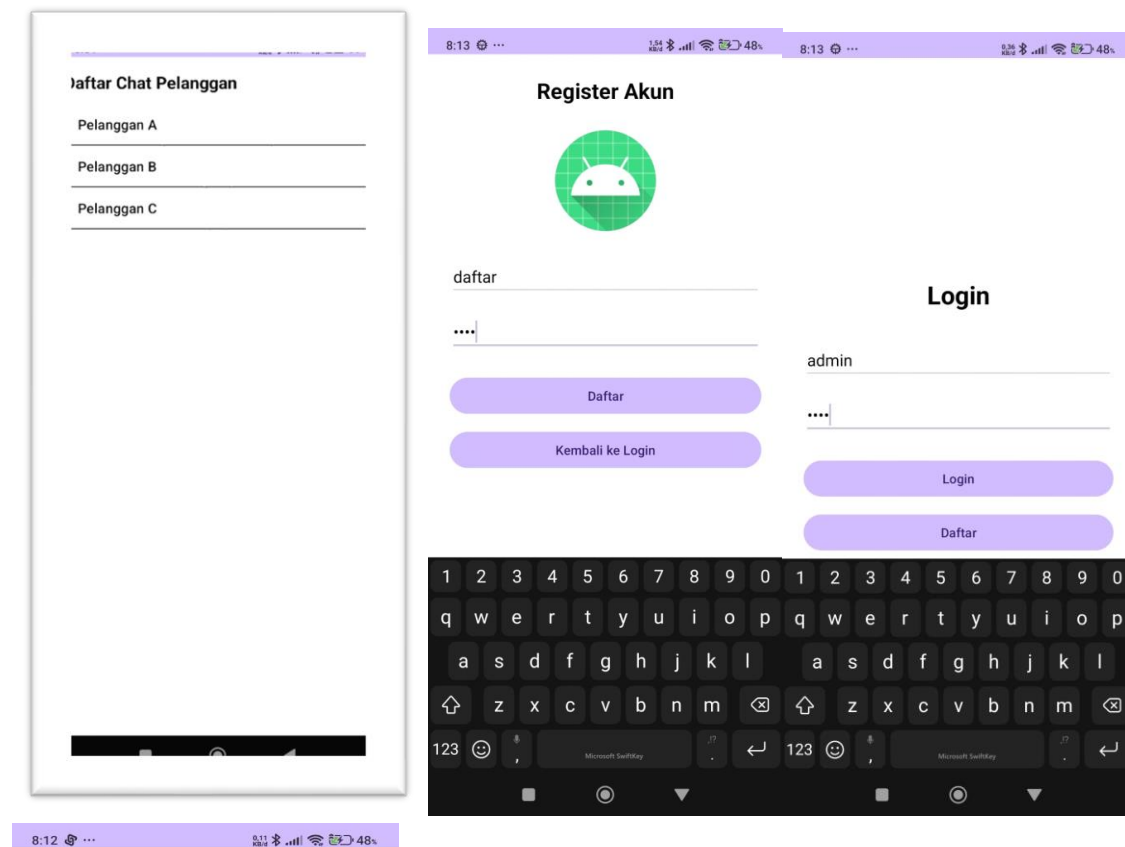
```
listView.adapter = adapter
```

Mengatur adapter yang sudah dikustomisasi ke dalam komponen `ListView`.

Link Github :

https://github.com/NaufalEka4371/Tugas_Uts_Mobile

Tampilan Aplikasi




UMKM Chat App

