

UJIAN TENGAH SEMESTER

PEMOGRAMAN MOBILE 1

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



Disusun oleh:

Nama: Mochammad Raffi Fadhillah

NPM: 23552011066

Kelas: TIF RP 23 CID B

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

2025

A. Essay

1. Apa fungsi setOnClickListener?

Jawab:

untuk menetapkan listener atau pendengar yang akan dipicu saat pengguna mengklik tampilan (View) tertentu

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

Jawab:

harus memanggil findViewById() atau metode binding lainnya untuk mendapatkan referensi View

Contoh:

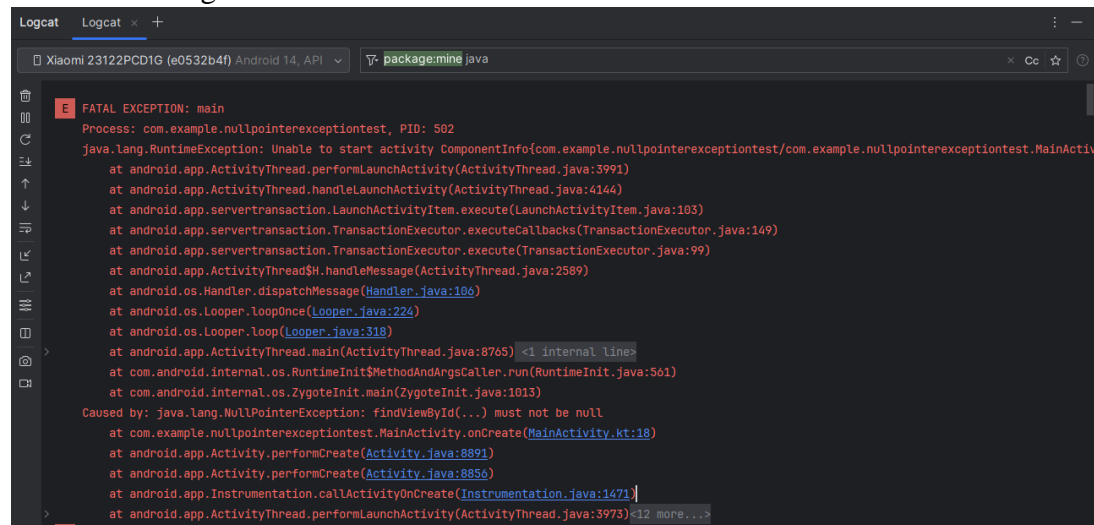
```
val buttonClick: Button = findViewById(R.id.buttonClick)
buttonClick.setOnClickListener {
    Toast.makeText(context: this, text: "Clicked!", Toast.LENGTH_SHORT).show()
}
```

3. Error apa yang terjadi jika file kotlin salah menginisialisasi findViewById atau objek pada xml belum diinisialisasi? Screenshot logcat-nya!

Jawab:

Aplikasi akan crash/force close dan mendapatkan error “NullPointerException”, karena android tidak dapat menemukan view tersebut dalam layout, sehingga findViewById mengembalikan null

Screenshoot logcat:



4. Buat sebuah contoh program untuk menampilkan pesan error NullPointerException!

Jawab:

```
package com.example.nullpointerexceptiontest
```

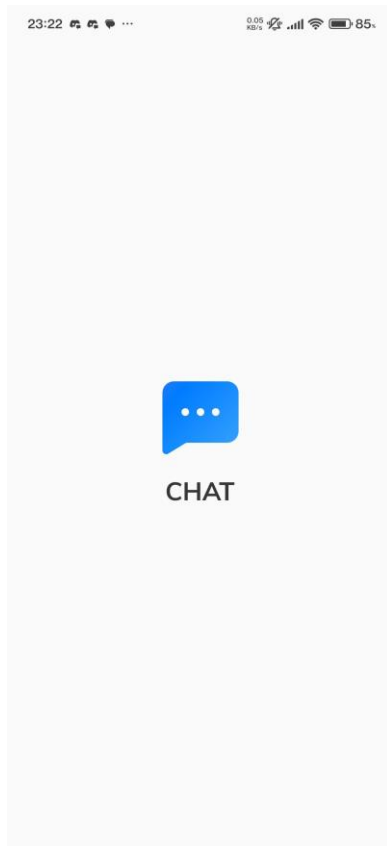
```
import android.os.Bundle
import android.widget.Button
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

class MainActivity : AppCompatActivity() {
    private var nullableString: String? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        val btn: Button = findViewById(R.id.buttonTrigger)
        btn.setOnClickListener {
            try {
                val length = nullableString!!.length
                Toast.makeText(this, "Length: $length",
                    Toast.LENGTH_SHORT).show()
            } catch (e: NullPointerException) {
                Toast.makeText(
                    this,
                    "Terjadi error: ${e::class.simpleName} -
                    ${e.message}",
                    Toast.LENGTH_LONG
                ).show()
            }
        }
    }
}
```

B. Studi kasus

a. SplashScreen Activity



Menampilkan animasi SplashScreen saat aplikasi diluncurkan pertama kali sebelum beralih ke activity_main.xml (Pilihan untuk Login dan register)

Penjelasan:

- Durasi tampilan: 1 detik
- setelah delay, pindah ke layar Login dan menutup SplashScreen.
- Implementasi:

```
package com.example.utsmobile

import android.content.Intent
import android.os.Bundle
import android.os.Handler
import android.os.Looper
import android.util.Log
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

class SplashActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
```

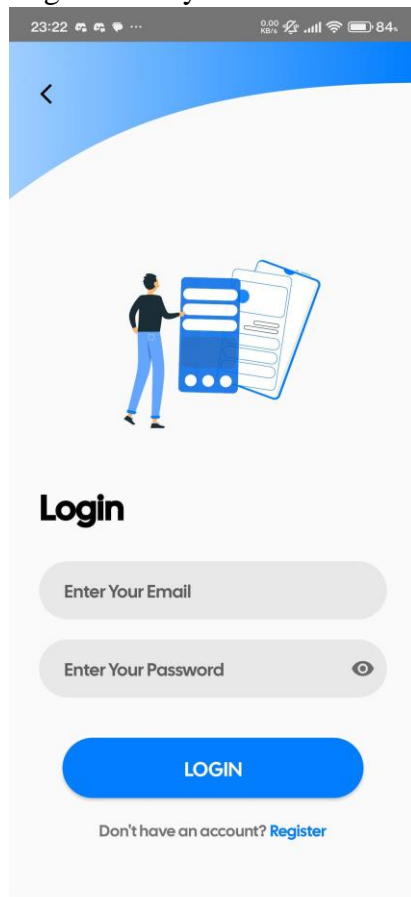
```

super.onCreate(savedInstanceState)
enableEdgeToEdge()
setContentView(R.layout.activity_splash)

ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.m
ain)) { v, insets ->
    val systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars())
    v.setPadding(systemBars.left, systemBars.top,
systemBars.right, systemBars.bottom)
    insets
}
// Delay 3 detik, kemudian buka Main_activity
Handler(Looper.getMainLooper()).postDelayed({
    Log.d("SplashActivity", "Handler: Splash
Screen...")
    startActivity(Intent(this,
MainActivity::class.java))
    finish()
}, 1000)
}
}

```

b. Login Activity



Form autentikasi pengguna menggunakan email dan password

Penjelasan:

- EditText: untuk input email dan password
- login_button: untuk mengecek validasi pada akun yang telah di register dan jika form-nya kosong
- Implementasi:

```
package com.example.utsmobile

import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.widget.ImageView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import com.example.utsmobile.databinding.ActivityLoginBinding
import com.google.firebase.auth.FirebaseAuth

class LoginActivity : AppCompatActivity() {
    lateinit var binding : ActivityLoginBinding //
    inisialisasi activity_login
    lateinit var firebaseAuth: FirebaseAuth // inisialisasi
    firebase authentication
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
        ActivityLoginBinding.inflate(layoutInflater)
        setContentView(binding.root)

        // Mendapatkan instance Firebase Authentication untuk
        operasi pendaftaran.
        firebaseAuth = FirebaseAuth.getInstance()

        // Ketika teks "Register" diklik, user akan
        dipindahkan ke RegisterActivity
        binding.txtRegister.setOnClickListener {
            Log.d("LoginActivity", "setOnClickListener:
            berhasil ke menu Register")
            val intent = Intent(this,
            RegisterActivity::class.java)
            startActivity(intent)
        }

        binding.btnLogin.setOnClickListener {
            // Mengambil Input dari EditText
            val email =
            binding.editLoginEmail.text.toString()
            val password =
```

```

binding.editLoginPassword.text.toString()

        // Cek apakah field kosong
        if (email.isNotEmpty() && password.isNotEmpty())
        {
            // memanggil firebase untuk Login dengan
            email dan password

            firebaseAuth.signInWithEmailAndPassword(email,
            password).addOnCompleteListener {
                if (it.isSuccessful) {
                    val intent = Intent(this,
                    HomeActivity::class.java)

                    startActivity(intent)
                    // tampilkan pesan toast jika
                    login berhasil

                    Log.d("LoginActivity",
                    "setOnClickListener: Login berhasil")
                    Toast.makeText(this, "Selamat
                    datang, $email", Toast.LENGTH_SHORT).show()
                } else {
                    // tampilkan pesan toast jika
                    input email dan password salah

                    Log.d("LoginActivity",
                    "setOnClickListener: Email atau password salah")
                    Toast.makeText(this, "Email atau
                    password salah. Silakan coba lagi.",
                    Toast.LENGTH_SHORT).show()
                }
            }
        } else {
            // tampilkan pesan toast jika input email dan
            password tidak diisi

            Log.d("LoginActivity", "setOnClickListener:
            Semua field harus diisi")
            Toast.makeText(this, "Semua field harus
            diisi", Toast.LENGTH_SHORT).show()
        }
    }

    // Tombol kembali untuk ke main_activity
    val backButton: ImageView =
    findViewById(R.id.btn_loginBack)
    backButton.setOnClickListener {
        Log.d("LoginActivity", "setOnClickListener:
        Berhasil kembali ke MainActivity")
        finish()
    }
}

ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.m
ain)) { v, insets ->
    val systemBars =
    insets.getInsets(WindowInsetsCompat.Type.systemBars())
    v.setPadding(systemBars.left, systemBars.top,

```

```

systemBars.right, systemBars.bottom)
        insets
    }
}
}
}

```

c. Register Activity

Form pendaftaran untuk akun baru dengan Email, Password, dan ConfirmPassword
Penjelasan:

- Semua textfield wajib diisi termasuk Email, Password, dan ConfirmPassword dan akan muncul pesan toast “semua field harus diisi”

Semua field harus diisi

Implementasi:

```

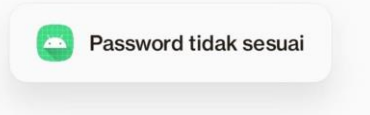
if (email.isNotEmpty() && password.isNotEmpty() &&
confirmPassword.isNotEmpty()) {
} else {
    // tampilkan pesan toast jika semua field harus diisi
    Log.d("RegisterActivity", "setOnClickListener: Semua
field harus diisi")
    Toast.makeText(this, "Semua field harus diisi",
Toast.LENGTH_SHORT).show()
}

```


Event log:

```
2025-05-07 21:27:49.935 25252-25252 RegisterActivity
com.example.utsmobile D setOnClickListener: Semua
field harus diisi
```

- Jika password tidak sesuai maka akan muncul toast “password tidak sesuai”



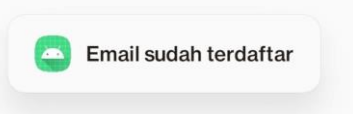
Implementasi:

```
Toast.makeText(this, "Password tidak sesuai",
Toast.LENGTH_SHORT).show()
```

Event log:

```
2025-05-07 21:41:02.024 28927-28927 RegisterActivity
com.example.utsmobile D setOnClickListener: Password
tidak sesuai
```

- Jika email sudah terdaftar maka akan muncul toast “email sudah terdaftar”



Implementasi:

```
Toast.makeText(this, "Email sudah terdaftar",
Toast.LENGTH_SHORT).show()
```

Event log:

```
2025-05-07 21:50:08.614 28927-28927 RegisterActivity
com.example.utsmobile D setOnClickListener: Email
sudah terdaftar
```

- Jika sudah register akan langsung masuk ke LoginActivity, lalu muncul pesan toast “registrasi berhasil”



Implementasi:

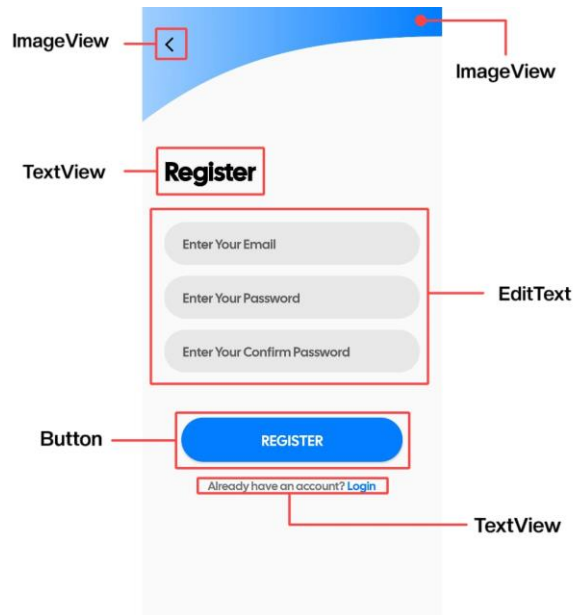
```
Toast.makeText(this, "Registrasi berhasil",
Toast.LENGTH_SHORT).show()
```

Event log:

```
2025-05-07 21:45:49.144 28927-28927 RegisterActivity
com.example.utsmobile D setOnClickListener: register
berhasil
```

- Data yang sudah di register akan disimpan di firebase server

- Di bagian RegisterActivity terdapat beberapa objek yaitu TextView, EditText, Button, dan ImageView



- Implementasi:

```
package com.example.utsmobile

import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.widget.ImageView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import com.example.utsmobile.databinding.ActivityRegisterBinding
import com.google.firebase.auth.FirebaseAuth

class RegisterActivity : AppCompatActivity() {

    lateinit var binding: ActivityRegisterBinding //
    inisialisasi activity_register
    lateinit var firebaseAuth: FirebaseAuth // inisialisasi
    firebase authentication
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
        ActivityRegisterBinding.inflate(layoutInflater)
        setContentView(binding.root)

        // Mendapatkan instance Firebase Authentication untuk
```

```

operasi pendaftaran.
    firebaseAuth = FirebaseAuth.getInstance()

    // Ketika teks "Login" diklik, user akan dipindahkan
ke LoginActivity
    binding.txtLogin.setOnClickListener {
        Log.d("LoginActivity", "setOnClickListener:
berhasil ke menu Login")
        val intent = Intent(this,
LoginActivity::class.java)
        startActivity(intent)
    }

    binding.btnRegister.setOnClickListener {
        // Mengambil Input dari EditText
        val email = binding.editEmail.text.toString()
        val password =
binding.editPassword.text.toString()
        val confirmPassword =
binding.editConfirmpassword.text.toString()
        // Cek apakah field kosong
        if (email.isNotEmpty() && password.isNotEmpty()
&& confirmPassword.isNotEmpty()) {
            if (password == confirmPassword) {
                // memanggil firebase untuk membuat akun
firebaseAuth.createUserWithEmailAndPassword(email, password)
                .addOnCompleteListener {
                    if (it.isSuccessful) {
                        val intent = Intent(this,
LoginActivity::class.java)
                        startActivity(intent)
                        // tampilkan pesan toast jika
registrasi berhasil
                        Log.d("RegisterActivity",
"setOnClickListener: register berhasil")
                        Toast.makeText(this,
"Registrasi berhasil", Toast.LENGTH_SHORT)
                            .show()
                    } else {
                        // tampilkan pesan toast jika
email telah terdaftar
                        Log.d("RegisterActivity",
"setOnClickListener: Email sudah terdaftar")
                        Toast.makeText(this, "Email
sudah terdaftar", Toast.LENGTH_SHORT)
                            .show()
                    }
                }
            } else {
                // tampilkan pesan toast jika password
tidak sesuai
                Log.d("RegisterActivity",
"setOnClickListener: Password tidak sesuai")
            }
        }
    }
}

```

```

// Toast.makeText(this, "Password tidak
sesuai", Toast.LENGTH_SHORT).show()
    }
    } else {
        // tampilkan pesan toast jika semua field
harus diisi
        Log.d("RegisterActivity",
"setOnClickListener: Semua field harus diisi")
        Toast.makeText(this, "Semua field harus
diisi", Toast.LENGTH_SHORT).show()
    }
    }
    // Tombol kembali ke main_activity
    val backButton: ImageView =
findViewById(R.id.btn_registerBack)
    backButton.setOnClickListener {
        Log.d("RegisterActivity", "setOnClickListener:
Berhasil kembali ke MainActivity")
        finish()
    }

ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.m
ain)) { v, insets ->
    val systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars())
    v.setPadding(systemBars.left, systemBars.top,
systemBars.right, systemBars.bottom)
    insets
}
}
}

```

d. List Chating



Menampilkan daftar percakapan menggunakan RecyclerView dengan GridLayoutManager, dan memanfaatkan Glide untuk memuat gambar avatar
Penjelasan:

- Berisi list objek Chat berisi nama kontak dan pesan terakhir
- Implementasi:

```
package com.example.utsmobile

import android.content.Intent
import android.graphics.Color
import android.os.Bundle
import android.util.Log
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import android.view.Menu
import android.view.MenuItem
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.widget.Toolbar
import androidx.core.graphics.drawable.DrawableCompat
import androidx.recyclerview.widget.GridLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.google.firebase.auth.FirebaseAuth
```

```

class HomeActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_home)
        val toolbar: Toolbar = findViewById(R.id.toolbar)
        setSupportActionBar(toolbar)
        supportActionBar?.title = "Chat"
        // mengganti warna toolbar menu menjadi warna putih
        toolbar.overflowIcon?.let { drawable ->
            val wrapped = DrawableCompat.wrap(drawable)
            DrawableCompat.setTint(wrapped.mutate(),
Color.WHITE)
            toolbar.overflowIcon = wrapped
        }

        val recyclerView =
findViewById<RecyclerView>(R.id.recycler_view)
        recyclerView.layoutManager = GridLayoutManager(this,
1) // Mengatur RecyclerView menampilkan item dalam satu kolom
        recyclerView.setHasFixedSize(true) // Memberi tahu
        bahwa ukuran RecyclerView tidak berubah berdasarkan konten

        // menyimpan daftar itemList yang berisi objek
        ItemList(judul, subJudul, imageUrl)
        val itemList = listOf(
            ItemList("Vivian Banshee", "Lorem ipsum dolor sit
amet",
                "https://static.wikia.nocookie.net/zenless-
zone-
zero/images/3/31/Avatar_Vivian_Banshee.png/revision/latest?cb=20250423022832"),
            ItemList("Hoshimi Miyabi", "Lorem ipsum dolor sit
amet",
                "https://static.wikia.nocookie.net/zenless-
zone-
zero/images/7/79/Avatar_Hoshimi_Miyabi.png/revision/latest?cb=20241220224623"),
            ItemList("Tsukishiro Yanagi", "Lorem ipsum dolor
sit amet",
                "https://static.wikia.nocookie.net/zenless-
zone-
zero/images/a/a3/Avatar_Tsukishiro_Yanagi.png/revision/latest?cb=20241117203653"),
            ItemList("Astra Yao", "Lorem ipsum dolor sit
amet",
                "https://static.wikia.nocookie.net/zenless-
zone-
zero/images/2/27/Avatar_Astra_Yao.png/revision/latest?cb=20250123000846"),
            ItemList("Evelyn Chevalier", "Lorem ipsum dolor
sit amet",
                "https://static.wikia.nocookie.net/zenless-
zone-

```

```

zero/images/9/94/Avatar_Evelyn_Chevalier.png/revision/latest?
cb=20250212183017"),
        ItemList("Zhu Yuan", "Lorem ipsum dolor sit
amet",
                "https://static.wikia.nocookie.net/zenless-
zone-
zero/images/5/53/Avatar_Zhu_Yuan.png/revision/latest?cb=20241
117203557"),
        ItemList("Jane Doe", "Lorem ipsum dolor sit
amet",
                "https://static.wikia.nocookie.net/zenless-
zone-
zero/images/5/59/Avatar_Jane_Doe.png/revision/latest?cb=20241
117203544"),
        ItemList("Ellen Joe", "Lorem ipsum dolor sit
amet",
                "https://static.wikia.nocookie.net/zenless-
zone-
zero/images/2/21/Avatar_Ellen_Joe.png/revision/latest?cb=2024
1117203511"),

    )
    val adapter = AdapterList(itemList)
    recyclerView.adapter = adapter
}

// Menu Toolbar (Logout)
override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    // Inflate menu_main.xml ke dalam toolbar
    menuInflater.inflate(R.menu.menu_main, menu)
    return true
}

    override fun onOptionsItemSelected(item: MenuItem):
Boolean {
        return when (item.itemId) {
            R.id.action_logout -> {
                showLogoutConfirmation()
                true
            }
            else -> super.onOptionsItemSelected(item)
        }
    }
}

// AlertDialog Konfirmasi Logout
private fun showLogoutConfirmation() {
    AlertDialog.Builder(this)
        .setTitle("Konfirmasi Logout")
        .setMessage("Apakah Anda yakin ingin logout?")
        .setPositiveButton("Ya") { dialog, _ ->
            dialog.dismiss()
            performLogout()
            Log.d("HomeActivity", "AlertDialog: logout
berhasil")
        }
}

```

```

        .setNegativeButton("Tidak") { dialog, _ ->
            dialog.dismiss()
            Log.d("HomeActivity", "AlertDialog: batal
logout")
        }
        .setCancelable(true)
        .show()
    }

    // Proses Logout
    private fun performLogout() {
        FirebaseAuth.getInstance().signOut()
        // Kembali ke LoginActivity
        val intent = Intent(this, LoginActivity::class.java)
        intent.flags = Intent.FLAG_ACTIVITY_NEW_TASK or
Intent.FLAG_ACTIVITY_CLEAR_TASK
        startActivity(intent)
        finish()
        Log.d("HomeActivity", "performLogout: Kembali ke
LoginActivity")
    }
}

```

- AdapterList.kt: bertanggung jawab menghubungkan data model ItemList ke tampilan item pada RecyclerView
Implementasi:

```

package com.example.utsmobile

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.annotation.NonNull
import androidx.recyclerview.widget.RecyclerView
import com.bumptech.glide.Glide

class AdapterList (private val itemList:
kotlin.collections.List<ItemList>):RecyclerView.Adapter<Adapt
erList.ViewHolder>() {
    // Menyimpan referensi ke view pada layout item_data.xml
    class ViewHolder (@NonNull itemView: View) :
RecyclerView.ViewHolder(itemView) {
        val imageView: ImageView = itemView.
findViewById(R.id.item_image)
        val judul: TextView =
itemView.findViewById(R.id.title)
        val subJudul: TextView =
itemView.findViewById(R.id.sub_title)
    }
    // Membuat instance ViewHolder baru saat RecyclerView
membutuhkan tampilan baru
    override fun onCreateViewHolder(parent: ViewGroup,
viewType: Int): AdapterList.ViewHolder {

```



```

        val view =
LayoutInflater.from(parent.context).inflate(R.layout.item_data, parent, false)
        return ViewHolder (view)
    }
    // Mengikat (bind) data dari itemList[position] ke tampilan di holder.
    override fun onBindViewHolder(holder:
AdapterList.ViewHolder, position: Int) {
        val item = itemList [position]
        holder.judul.text = item.judul
        holder.subJudul.text = item.subJudul

Glide.with(holder.imageView.context).load(item.imageUrl).into
(holder.imageView)
    }

    override fun getItemCount(): Int {
        return itemList.size
    }
}

```

- ItemList.kt: data class yang berfungsi sebagai model data untuk setiap item yang terdiri dari judul, subJudul, dan imageUrl

Implementasi:

```

package com.example.utsmobile

// Untuk menyimpan data judul, subJudul, dan imageUrl
data class ItemList (
    var judul: String,
    var subJudul: String,
    var imageUrl: String
)

```