

MOBILE PROGRAMMING

UTS



OLEH :

GITA AULIA HAFID ( 23552011050 )

**Program Studi Teknik Informatika**

**Fakultas Industri Kreatif**

**Universitas Teknologi Bandung**

**2025**

## Essay

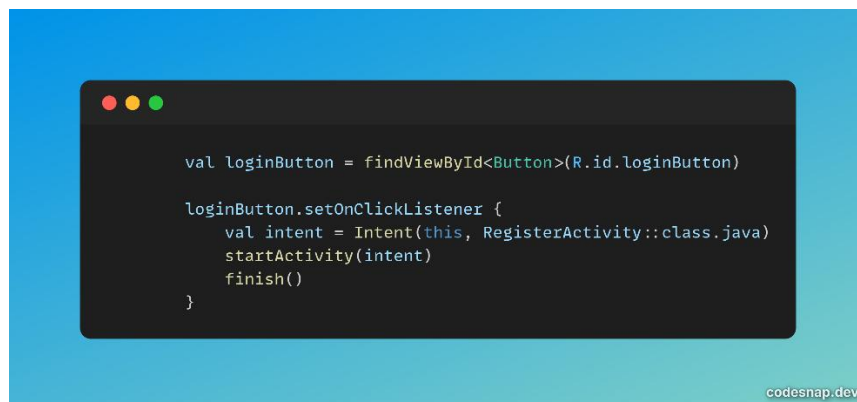
### 1. Apa fungsi setOnClickListener?

setOnClickListener berfungsi apabila pengguna menekan pada suatu tombol navigasi contohnya button dan akan menghasilkan sebuah kejadian seperti redirect ke halaman lain.

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

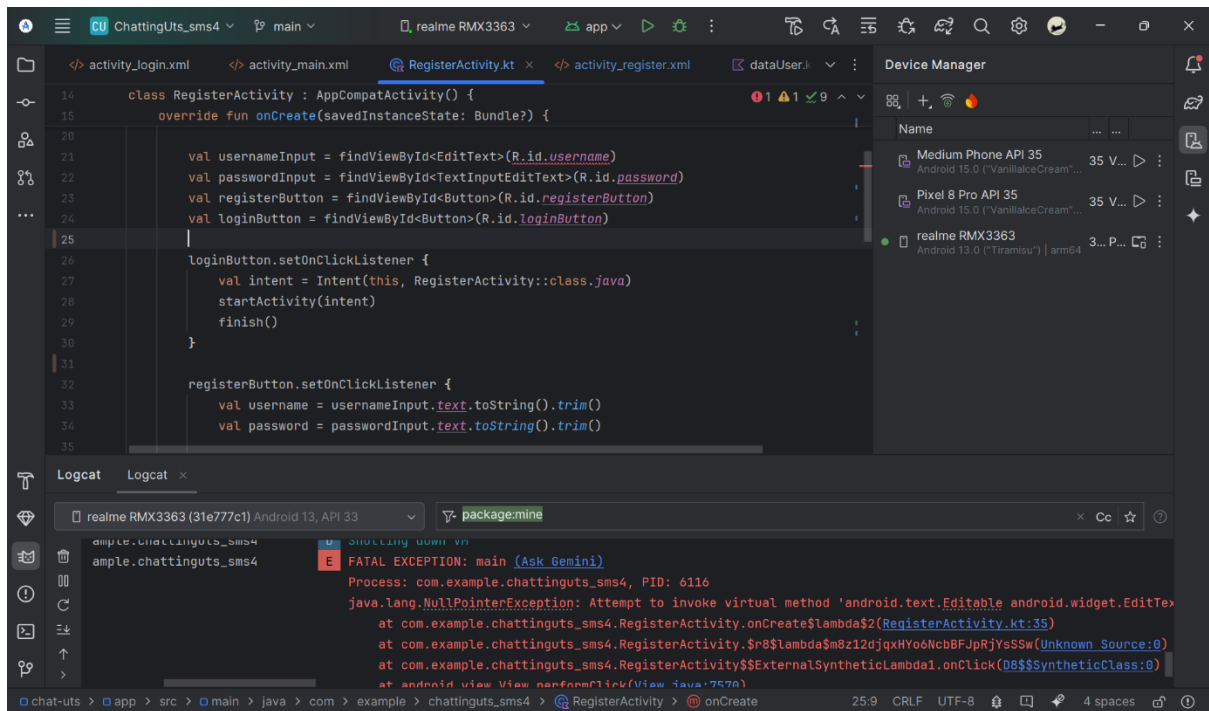
Syarat pemanggilan method setOnClickListener yaitu memiliki target dengan menandai nya sebagai id yang akan diberikan method tersebut, lalu pengguna harus melakukan interaksi dengan target tersebut seperti menekan.

Contoh source code nya :



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

Sebagai contoh saya memasukkan id yang salah di RegisterActivity. Seharusnya ini username namun saya salahkan menjadi username



Error nya `NullPointerException` karena id username tidak ada di xml dan tidak ada inisialisasi sebelumnya.

4. Buat sebuah contoh program untuk menampilkan pesan error `NullPointerException`! Hasilnya sama dengan nomor 3.

## Studi Kasus

### 1. Kebutuhan tambahan

#### 1.A File dataUser

Pada kode dibawah ini saya membuat sebuah data class User yang menampung 2 atribut yaitu username dan password lalu saya membuat sebuah list yang dapat diubah untuk menyimpan data user dalam bentuk objek user.

```
object UserDatabase {  
    val users = mutableListOf<User>()  
}  
data class User(val username: String, val password: String)
```

#### 1.B file dataMessage

Pada kode dibawah ini saya membuat sebuah data class user yang menampung 3 atribut yaitu from , message dan time lalu saya membuat sebuah list yang dapat diubah untuk menyimpan data pesan dalam bentuk objek messageSystem.

```
object dataMessage{  
    val message = mutableListOf<messageSystem>()  
}  
data class messageSystem(val from: String , val message: String, val time: String)
```

## A. SPLASHSCREEN ACTIVITY

Penjelasan disertakan dalam baris komentar pada source code dibawah ini .

```
class SplashActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_splash)

        // HANDLER UNTUNK MENJEDA RUNNING FUNGSI goToLoginActivity Selama 3 detik
        Handler(Looper.getMainLooper()).postDelayed({
            goToLoginActivity()
        }, 3000L)

    }
    // FUNGSI INI UNTUK MENAMPILKAN HALAMAN UTAMA
    private fun goToLoginActivity() {
        Intent(this, LoginActivity::class.java).also{
            startActivity(it)
            finish();
        }
    }
}
```

## B. LOGIN ACTIVITY

Penjelasan ada dalam source code dibawah ini .

```
class LoginActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()

        // kode logika ini berelasi dengan layout activity_login
        setContentView(R.layout.activity_login)

        // ambil id dari komponen tampilan dari layout activity_login
        val usernameInput = findViewById<EditText>(R.id.username)
        val passwordInput = findViewById<TextInputEditText>(R.id.password)
        val loginButton = findViewById<Button>(R.id.loginButton)
        val registerButton = findViewById<Button>(R.id.registerButton)

        // aksi yang akan dijalankan ketika mengklik RegisterButton
        registerButton.setOnClickListener {
            // intent ini akan mengarahkan ke halaman registerActivity.
            val intent = Intent(this, RegisterActivity::class.java)
            startActivity(intent)
            finish()
        }

        // aksi yang akan dijalankan ketika mengklik loginButton.
        loginButton.setOnClickListener {
            // ambil inputan dari user dan hapus spasi di awal atau akhir
            val username = usernameInput.text.toString().trim()
            val password = passwordInput.text.toString().trim()

            // kondisi ini akan mengecek apakah username dan password tdidak kosong
            if(username.isNotEmpty() && password.isNotEmpty()){
                // ini akan mengecek data di data class userDatabase apakah ada username dan password yang di inputkan user.
                val userMatch = UserDatabase.users.find{ it.username == username && it.password == password}

                // jika Variable userMatch ini terisi maka data ditemukan
                if(userMatch != null){
                    // akan redirect ke halaman mainActivity
                    val intent = Intent(this, MainActivity::class.java)
                    // dengan menyisipkan username ke intent dengan key Username_login.
                    intent.putExtra("Username_login" , username)
                    startActivity(intent)
                    finish()
                }
                // jika password dan username tidak terdaftar akan memunculkan toast.
            } else {
                Toast.makeText(this, "Username Atau Password Salah" , Toast.LENGTH_SHORT).show()
            }
        }
        // jika password dan username kosong memunculkan toast peringatan.
    } else {
        Toast.makeText(this, "Username dan Password Harus Diisi!" , Toast.LENGTH_SHORT).show()
    }
}
}
```

## C. REGISTER ACTIVITY

Penjelasan disertakan dalam bentuk komentar dalam source code.

```
class RegisterActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()

        // kode ini berelasi dengan layout activity_register
        setContentView(R.layout.activity_register)

        // ambil id dari komponen tampilan dari layout activity_register
        val usernameInput = findViewById<EditText>(R.id.username)
        val passwordInput = findViewById<TextInputEditText>(R.id.password)
        val registerButton = findViewById<Button>(R.id.registerButton)
        val loginButton = findViewById<Button>(R.id.loginButton)

        // aksi yang akan dijalankan ketika mengklik loginButton
        loginButton.setOnClickListener {
            // intent ini akan mengarahkan ke halaman LoginActivity yang dimana berelasi dengan activity_login.
            val intent = Intent(this, LoginActivity::class.java)
            startActivity(intent)
            finish()
        }

        // pada saat registerButton Di klik akan menjalankan kondisi dibawah ini
        registerButton.setOnClickListener {
            // ambil inputan dari user dan hapus spasi di awal atau akhir
            val username = usernameInput.text.toString().trim()
            val password = passwordInput.text.toString().trim()

            // kondisi ini akan mengecek apakah username dan password tidak kosong
            if (username.isNotEmpty() && password.isNotEmpty()) {
                // ini akan mengecek di data class userDatabase apakah ada username yang diinputkan user
                val isUsernameTaken = UserDatabase.users.any { it.username == username }

                // jika username tidak ada ( artinya unik belum tersimpan di data class )
                if (!isUsernameTaken) {
                    // maka akan langsung disimpan di data class User Database
                    val newUser = User(username, password)
                    UserDatabase.users.add(newUser)

                    // selanjutnya redirect ke halaman loginActivity
                    val intent = Intent(this, LoginActivity::class.java)
                    // logcat
                    Log.d("RegisterActivity", "Username Diterima : $username")
                    startActivity(intent)
                    finish()
                }
                // jika username sudah ada di data class akan menjalankan kondisi ini
            } else {
                // toast dan logcat.
                Toast.makeText(this, "Username sudah Terdaftar", Toast.LENGTH_SHORT).show()
                Log.w("RegisterActivity", "Username Sudah Terdaftar : $username")
            }
        }

        // jika user name dan password kosong akan menjalankan kondisi dibawah
        else {
            // menjalankan toast dan logcat.
            Log.e("RegisterActivity", "Username dan Password Kosong!")
            Toast.makeText(this, "Username dan Password tidak boleh kosong!", Toast.LENGTH_SHORT).show()
        }
    }
}
```

## Ketentuan minimal pada activity register XML

```
<ImageView
    android:id="@+id/logoChat"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:src="@drawable/logo1"
/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="JUST CHATTING"
    android:textSize="40sp" />
<EditText
    android:id="@+id/username"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Username"
    android:layout_marginTop="16dp"
    android:minHeight="48dp" />
<Button
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:background="@color/black"
    android:text="LOGIN"
    android:id="@+id/loginButton"
    android:layout_gravity="right"
    android:elevation="0dp" />
```



## D. LIST CHATING

Dibutuhkan adapter untuk menjebatani antara data dan tampilan ui di list\_item.

- Adapter

Penjelasan ada dalam source code dibawah ini.

```
class adapterList(private val context: Context, private val data: List<messageSystem> ): BaseAdapter() {

    // fungsi ini mmeberikan jumlah item yang akan tampil dalam list
    override fun getCount(): Int = data.size

    // fungsi ini mengembalikan data pada posisi tertentu.
    override fun getItem(position: Int): Any = data[position]

    // fungsi ini mengembalikan id unik dari setiap item.
    override fun getItemId(position: Int): Long = position.toLong()

    // fungsi ini menampilkan tampilan tiap item
    override fun getView(position: Int, convertView: View?, parent: ViewGroup?): View{

        val view : View = convertView ?: LayoutInflater.from(context).inflate(R.layout.list_item, parent, false)

        // var item akan mengambil data dari dataMessage di posisi POSITION
        // untuk ditampilkan pada list item.
        val item = data[position]

        // ambil id dari komponen tampilan dari layout list_item.xml
        val fromText = view.findViewById<TextView>(R.id.fromTextView)
        val messageText = view.findViewById<TextView>(R.id.messageTextView)
        val timeText = view.findViewById<TextView>(R.id.timeTextView)
        val imageView = view.findViewById<CircleImageView>(R.id.profileImage)

        // isi nilai elemen dengan data dari dataMessage.
        fromText.text = item.from
        messageText.text = item.message
        timeText.text = item.time

        // Mendeklarasikan tipe array dan mengisinya dengan nilai image yang ada di drawable
        val profileImages: Array<Int> = arrayOf(R.drawable.profile, R.drawable.profile2, R.drawable.profile3)

        // pilih gambar secara acak menggunakan random.nextInt
        val randomImage = profileImages[Random.nextInt(profileImages.size)]

        //ambil nilai dari random image render ke image view di list item.
        imageView.setImageResource(randomImage)

        return view
    }
}
```

- Halaman MainActivity

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        // kode logika ini berelasi dengan layout activity_main
        setContentView(R.layout.activity_main)

        // ambil id dari komponen tampilan dari layout activity_main
        val username = findViewById<TextView>(R.id.usernameLogin)
        val logout = findViewById<Button>(R.id.logoutButton)
        val listView = findViewById<ListView>(R.id.listView)

        // ambil sisipan username yang tadi telah dibuat dengan key Username_Login
        val loginUser = intent.getStringExtra("Username_Login")

        //jika variabel loginUser ini tidak kosong maka akan menjalankan kode dibawah ini
        if(!loginUser.isNullOrEmpty()) {
            // mengisi variabel username dengan loginUser.
            username.text = loginUser

            //menambahkan data kedalam list dataMessage.message
            dataMessage.message.add(messageSystem("Camilla" , "hello" , "8.34"))
            dataMessage.message.add(messageSystem("Becky" , "hello Whatsup" , "8.00"))
            dataMessage.message.add(messageSystem("Natti" , "dare you" , "6.00"))

            // menghubungkan adapter ke listView agar data muncul di halaman.
            val adapter = adapterList(this, dataMessage.message)
            listView.adapter = adapter
        }
        // jika variable loginUser kosong maka akan redirect ke halaman
        // LoginActivity.
        else{
            val intent = Intent(this, LoginActivity::class.java)
            startActivity(intent)
            finish()
        }

        // aksi yang akan terjadi jika mengklik button logout
        logout.setOnClickListener {
            // redirect ke halaman LoginActivity.
            val intent = Intent(this, LoginActivity::class.java)
            startActivity(intent)
            finish()
        }
    }
}
```

Link Github : <https://github.com/gitaaulia05/chat-uts>