

UJIAN TENGAH SEMESTER

Mata Kuliah Pemrograman Mobile 1
Dosen Pengampu : Nova Agustina, ST., M.Kom.



Disusun oleh :

Farhan Permana

23552011325

Kelas : TIF RP – CID 23B

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INDUSTRI KREATIF
UNIVERSITAS TEKNOLOGI BANDUNG
2025**

Perhatikan setiap butir soal berikut dengan teliti. Pastikan Anda sudah memenuhi semua kriteria sebelum disubmit!

Essay

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

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**
8. **Link Github harus tercantum pada pdf (point 7)**

I. ESSAY

1. `setOnClickListener` berfungsi untuk mengatur tindakan yang akan dijalankan ketika sebuah view seperti tombol, teks, gambar diklik oleh pengguna.
2. Syarat pemanggilan `setOnClickListener` yaitu komponen viewnya sudah terhubung dengan layout XML menggunakan Id dan objeknya sudah tidak null.

Contoh :

```
btnDatePicker = findViewById(R.id.btnDatePicker)
btnToast = findViewById(R.id.btnToast)

btnDatePicker.setOnClickListener { it: View!
    val calendar = Calendar.getInstance()
    val year = calendar.get(Calendar.YEAR)
    val month = calendar.get(Calendar.MONTH)
    val day = calendar.get(Calendar.DAY_OF_MONTH)

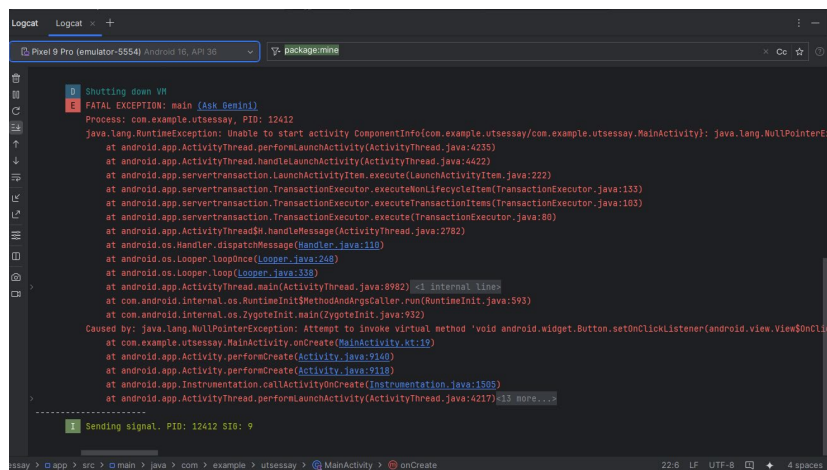
    val datePickerDialog = DatePickerDialog(
        context: this,
        { _, selectedYear, selectedMonth, selectedDay ->
            val selectedDate = "$selectedDay/$selectedMonth + 1/$selectedYear"
            Toast.makeText(context: this, text: "Tanggal dipilih: $selectedDate", Toast.LENGTH_LONG).show()
        },
        year, month, day
    )

    datePickerDialog.show()
}

btnToast.setOnClickListener { it: View!
    Toast.makeText(context: this, text: "Ini adalah Toast Message", Toast.LENGTH_SHORT).show()
}
```

Komponen view tersebut sudah terhubung dengan id `btnDatePicker` dan `btnToast`.

- Pada `btnDatePicker.setOnClickListener`, saat tanggal dipilih maka akan memunculkan toast message berupa text tanggal yang dipilih tersebut.
 - Pada `btnToast.setOnClickListener`, saat tombol tersebut di klik akan memunculkan toast message yang berisi text “Ini adalah Toast Message”.
3. Apabila file kotlin salah menginisialisasi `findViewById` atau objek pada xml belum diinisialisasi maka akan terjadi error, dikarenakan `findViewById` mengembalikan null.



4.

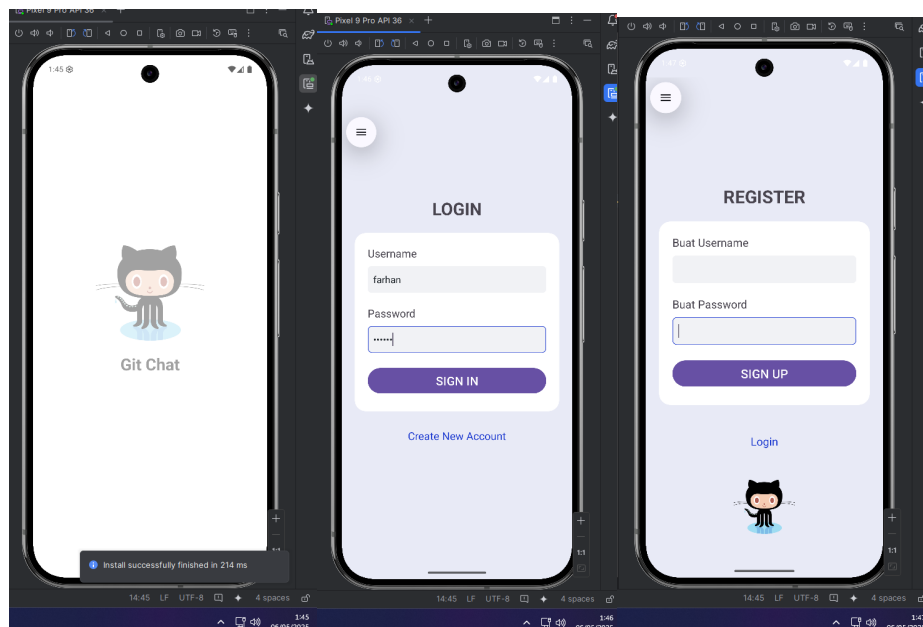
```
activity_main.xml MainActivity.kt
1 package com.example.utsessay
2
3 > import ...
11
12 class MainActivity : AppCompatActivity() {
13     override fun onCreate(savedInstanceState: Bundle?) {
14         super.onCreate(savedInstanceState)
15         setContentView(R.layout.activity_main)
16
17         val nama: String? = null
18         val panjangNama = nama!!.length |
19     }
20 }
```

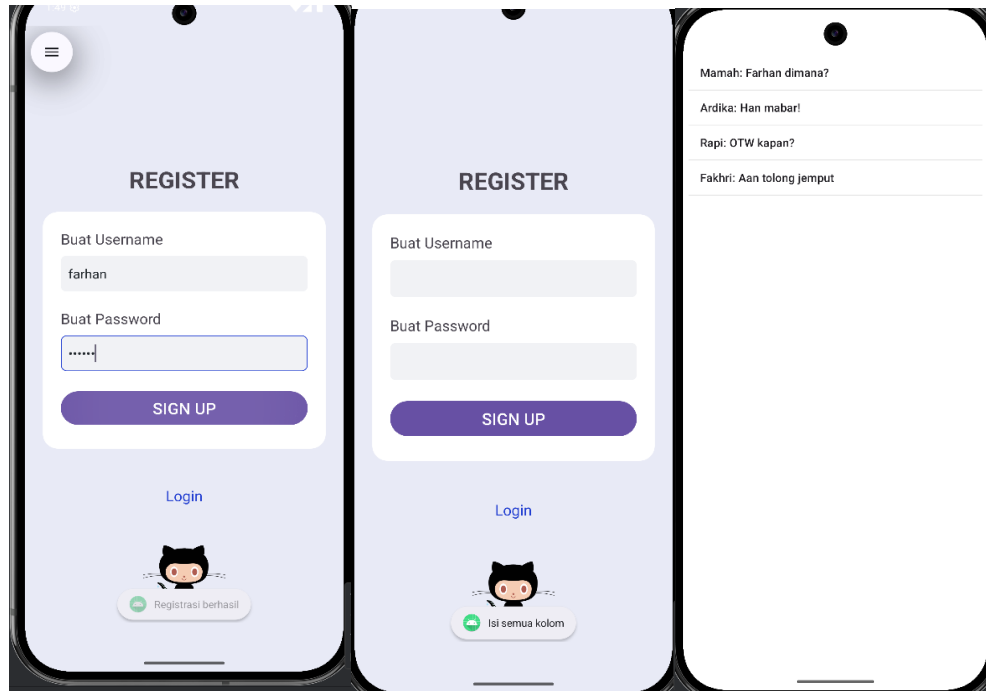
FATAL EXCEPTION: main (Ask Gemini)
Process: com.example.utsessay, PID: 12574
java.lang.RuntimeException: Unable to start activity ComponentInfo{com.example.utsessay/com.example.utsessay.MainActivity}: java.lang.NullPointerException

Kode tersebut mencoba mengakses properti length dari variabel nama yang bernilai null menggunakan operator !!, maka hasilnya akan menyebabkan NullPointerException.

II. STUDI KASUS

- Link Github: https://github.com/farhanperm/uts_project
- Tampilan:





- Penjelasan Source Code:

1. SplashActivity.kt

```

14 class SplashActivity : AppCompatActivity() {
15     override fun onCreate(savedInstanceState: Bundle?) {
16         super.onCreate(savedInstanceState)
17         enableEdgeToEdge()
18         setContentView(R.layout.activity_splash)

```

Class SplashActivity mewarisi AppCompatActivity lalu onCreate untuk mengatur layout

```

20     val fade_in = AnimationUtils.loadAnimation(context, R.anim.fade_in)
21     val logo = findViewById<ImageView>(R.id.logo)
22     val app_desc = findViewById<TextView>(R.id.app_name)
23
24     logo.startAnimation(fade_in)
25     app_desc.startAnimation(fade_in)

```

Menjalankan animasi pada logo dan teks

```

28     Handler(Looper.getMainLooper()).postDelayed({
29         goToLoginActivity()
30     }, delayMillis: 3000L)

```

Delay splashscreen selama 3 detik sebelum pindah ke LoginActivity

```

31         private fun goToLoginActivity(){
32             Intent( packageContext: this, LoginActivity::class.java).also { it: Intent
33                 startActivity(it)
34                 finish()
35             }
36         }

```

Fungsi untuk berpindah ke LoginActivity setelah delay selesai

2. LoginActivity.kt

```

11 <img alt="Kotlin icon" data-bbox="215 265 235 285"/> class LoginActivity : AppCompatActivity() {
12     override fun onCreate(savedInstanceState: Bundle?) {
13         super.onCreate(savedInstanceState)
14         setContentView(R.layout.activity_login)

```

Class LoginActivity mewarisi AppCompatActivity lalu onCreate untuk mengatur layout

```

16         val etUsername = findViewById<EditText>(R.id.etUsername)
17         val etPassword = findViewById<EditText>(R.id.etPassword)
18         val btnLogin = findViewById<Button>(R.id.btnLogin)
19         val tvCreateAccount = findViewById<TextView>(R.id.create_new_account)

```

Membuat value untuk menghubungkan komponen dari layout

```

23         btnLogin.setOnClickListener { it: View!
24             val username = etUsername.text.toString()
25             val password = etPassword.text.toString()

```

Mengatur tombol untuk mengambil input dari username dan password

```

26         val sharedPref = getSharedPreferences( name: "UserPrefs", MODE_PRIVATE)
27         val savedUsername = sharedPref.getString( key: "username", defValue: "")
28         val savedPassword = sharedPref.getString( key: "password", defValue: "")

```

Mengambil data yang tersimpan di SharedPreferences

```

30         if (username == savedUsername && password == savedPassword) {
31             startActivity(Intent( packageContext: this, ListChattingActivity::class.java))
32         } else {
33             Toast.makeText( context: this, text: "Login gagal", Toast.LENGTH_SHORT).show()
34         }
35     }

```

Membaca inputan username dan password apabila sesuai maka masuk ke ListChattingActivity dan apabila tidak sesuai akan menampilkan toast

```

37         tvCreateAccount.setOnClickListener { it: View!
38             startActivity(Intent( packageContext: this, RegisterActivity::class.java))
39         }

```

Mengatur apabila tulisan di klik akan masuk ke RegisterActivity

3. RegisterActivity.kt

```

1  </> class RegisterActivity : AppCompatActivity() {
2      override fun onCreate(savedInstanceState: Bundle?) {
3          super.onCreate(savedInstanceState)
4          setContentView(R.layout.activity_register)

```

Class RegisterActivity mewarisi AppCompatActivity lalu onCreate untuk mengatur layout

```

16         val etUsernameReg = findViewById<EditText>(R.id.etUsernameReg)
17         val etPasswordReg = findViewById<EditText>(R.id.etPasswordReg)
18         val btnRegister = findViewById<Button>(R.id.btnRegister)
19         val tvBackToLogin = findViewById<TextView>(R.id.back_to_login)

```

Membuat value untuk menghubungkan komponen dari layout

```

21         btnRegister.setOnClickListener { it: View!
22             val username = etUsernameReg.text.toString()
23             val password = etPasswordReg.text.toString()

```

Mengambil nilai input saat klik tombol register

```

25         if (username.isNotEmpty() && password.isNotEmpty()) {
26             val sharedPref = getSharedPreferences( name: "UserPrefs", MODE_PRIVATE)
27             val editor = sharedPref.edit()
28             editor.putString("username", username)
29             editor.putString("password", password)
30             editor.apply()

```

Menyimpan username dan password ke SharedPreferences

```

32         Toast.makeText( context: this, text: "Registrasi berhasil", Toast.LENGTH_SHORT).show()
33         startActivity(Intent( packageContext: this, LoginActivity::class.java))
34     } else {
35         Toast.makeText( context: this, text: "Isi semua kolom", Toast.LENGTH_SHORT).show()

```

Memunculkan toast jika berhasil atau gagal

```

38         tvBackToLogin.setOnClickListener { it: View!
39             startActivity(Intent( packageContext: this, LoginActivity::class.java))

```

Mengatur apabila tulisan di klik akan masuk ke LoginActivity

4. ListChatting.kt

```
8  </> class ListChattingActivity : AppCompatActivity() {  
9  @Override fun onCreate(savedInstanceState: Bundle?) {  
10      super.onCreate(savedInstanceState)  
11      setContentView(R.layout.activity_list_chatting)
```

Class ListChattingActivity mewarisi AppCompatActivity lalu onCreate untuk mengatur layout

```
13      val chatListView = findViewById<ListView>(R.id.chatListView)  
14      val chatData = listOf(  
15          "Mamah: Farhan dimana?",  
16          "Ardika: Han mabar!",  
17          "Rapi: OTW kapan?",  
18          "Fakhri: Aan tolong jemput"  
19      )
```

Mengatur value chatListView dan isi datanya

```
21      val adapter = ArrayAdapter<Context>(this, android.R.layout.simple_list_item_1, chatData)  
22      chatListView.adapter = adapter
```

Menghubungkan data ke ListView dengan array adapter.