

Nama : Naufal Aulia Nuchrizal

NPM : 23552011366 – 223 TIF RP 23 CID B

UTS – ESSAY

1. Apa fungsi setOnClickListener?

Fungsi setOnClickListener adalah untuk **mendengarkan event klik** pada suatu komponen tampilan (View) seperti Button, ImageView, TextView, dll. Ketika pengguna menekan komponen tersebut, kode di dalam onClick() akan dijalankan.

2. Syarat pemanggilan method setOnClickListener

Buat view contohnya button yang sudah diinisialisasi lalu kasih akses ke objek view.

Contoh XML:

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

```
import android.os.Bundle
import android.widget.Button
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

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

        val button = findViewById<Button>(R.id.button_click)

        button.setOnClickListener {
            Toast.makeText(this, "Button ditekan!", Toast.LENGTH_SHORT).show()
        }
    }
}
```

3. Disini findViewById dipanggil sebelum layout dimuat

```
class MainActivity : AppCompatActivity() {  
  
    lateinit var myButton: Button  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        // findViewById dipanggil sebelum layout dimuat  
        myButton = findViewById(R.id.my_button) // Akan NULL  
  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        myButton.setOnClickListener {  
            myButton.text = "Clicked!"  
        }  
    }  
}
```

Hasil logcat:

```
FATAL EXCEPTION: main  
Process: com.example.essay_uts, PID: 31589  
java.lang.RuntimeException: Unable to start activity ComponentInfo{com.example.essay_uts/c  
    at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:3740)  
    at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:3898)  
    at android.app.servertransaction.LaunchActivityItem.execute(LaunchActivityItem.java:85)  
    at android.app.servertransaction.TransactionExecutor.executeCallbacks(TransactionExecu  
    at android.app.servertransaction.TransactionExecutor.execute(TransactionExecutor.java:  
    at android.app.ActivityThread$H.handleMessage(ActivityThread.java:2305)  
    at android.os.Handler.dispatchMessage(Handler.java:106)  
    at android.os.Looper.loop(Looper.java:257)  
    at android.app.ActivityThread.main(ActivityThread.java:8192) <1 internal line>  
    at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:626)  
    at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:1015)  
Caused by: java.lang.NullPointerException: findViewById(...) must not be null  
    at com.example.essay_uts.MainActivity.onCreate(MainActivity.kt:14)  
    at android.app.Activity.performCreate(Activity.java:8119)  
    at android.app.Activity.performCreate(Activity.java:8103)  
    at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1359)  
    at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:3713)<11 more
```

4. Code menampilkan pesan NullPointerException

```
1 package com.example.essay_uts  
2  
3 import android.os.Bundle  
4 import android.widget.TextView  
5 import androidx.appcompat.app.AppCompatActivity  
6  
7 class MainActivity : AppCompatActivity() {  
8     override fun onCreate(savedInstanceState: Bundle?) {  
9         super.onCreate(savedInstanceState)  
10        setContentView(R.layout.activity_main)  
11  
12        // Memaksa NullPointerException tanpa try-catch  
13        val textView: TextView? = null  
14        textView!!.text = "Ini menyebabkan NullPointerException"  
15    }  
16 }
```

Hasil logcat:

```
FATAL EXCEPTION: main
Process: com.example.essay_uts, PID: 29742
java.lang.RuntimeException: Unable to start activity ComponentInfo{com.example.essay_uts/com.example.essay_uts.MainActivity}: java.lang.NullPointerException: Attempt to invoke virtual method 'void com.example.essay_uts.MainActivity.onCreate()' on a null object reference
    at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:3740)
    at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:3898)
    at android.app.servertransaction.LaunchActivityItem.execute(LaunchActivityItem.java:85)
    at android.app.servertransaction.TransactionExecutor.executeCallbacks(TransactionExecutor.java:135)
    at android.app.servertransaction.TransactionExecutor.execute(TransactionExecutor.java:95)
    at android.app.ActivityThread$H.handleMessage(ActivityThread.java:2305)
    at android.os.Handler.dispatchMessage(Handler.java:106)
    at android.os.Looper.loop(Looper.java:257)
    at android.app.ActivityThread.main(ActivityThread.java:8192) <1 internal line>
    at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:626)
    at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:1015)
Caused by: java.lang.NullPointerException
    at com.example.essay_uts.MainActivity.onCreate(MainActivity.kt:14)
    at android.app.Activity.performCreate(Activity.java:8119)
    at android.app.Activity.performCreate(Activity.java:8103)
    at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1359)
    at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:3713) <11 more...>
```

Studi kasus : buat SplashScreen, Login, Register dan ListChat.
SplashScreen(MainActivity.kt)

```
package com.example.essay_uts

import android.content.Intent
import android.os.Bundle
import android.os.Handler
import android.os.Looper
import androidx.appcompat.app.AppCompatActivity

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

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

Disini menampilkan layout utama dengan menunda selama 2 detik sebelum pindah ke *LoginActivity*, menggunakan Handler

LoginActivity.kt

```
class LoginActivity : AppCompatActivity() {
    private val gson = Gson()

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

Disini membuat instance dari library Gson untuk mengolah data JSON dan *onCreate* adalah method yang dipanggil saat activity dimulai. *setContentView* menampilkan layout *activity_login.xml*.

```

val usernameInput = findViewById<EditText>(R.id.usernameInput)
val passwordInput = findViewById<EditText>(R.id.passwordInput)
val loginBtn = findViewById<Button>(R.id.loginBtn)
val registerBtn = findViewById<Button>(R.id.registerRedirectBtn)

```

Menghubungkan elemen UI di layout dengan kode Kotlin.

```

val sharedPref = getSharedPreferences( name: "user_pref", Context.MODE_PRIVATE)
val jsonUsers = sharedPref.getString( key: "users", defValue: "{}")
val type = object : TypeToken<Map<String, String>>() {}.type
val userMap: Map<String, String> = gson.fromJson(jsonUsers, type)

```

Mengambil data user dari *SharedPreferences* (disimpan saat registras) dan mengubahnya dari JSON ke Map *username-password*.

```

loginBtn.setOnClickListener {
    val username = usernameInput.text.toString().trim()
    val password = passwordInput.text.toString().trim()

    if (userMap[username] == password) {
        Toast.makeText( context: this, text: "Login berhasil", Toast.LENGTH_SHORT).show()
        startActivity(Intent( packageContext: this, ListChatActivity::class.java))
        finish()
    } else {
        Toast.makeText( context: this, text: "Username atau password salah", Toast.LENGTH_SHORT).show()
    }
}

registerBtn.setOnClickListener {
    startActivity(Intent( packageContext: this, RegisterActivity::class.java))
}
}

```

Saat tombol login diklik, ambil input *username* dan *password*, cocokkan dengan data di *SharedPreferences*. Jika cocok, lanjut ke *ListChatActivity*. Lalu saat tombol register diklik. Navigasi ke *RegisterActivity*.

RegisterActivity.kt

Tampilkan layout registrasi

```

class RegisterActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_register)
    }
}

```

Hubungkan elemen UI.

```

val profileImage = findViewById<ImageView>(R.id.profileImage)
val usernameInput = findViewById<EditText>(R.id.usernameInput)
val passwordInput = findViewById<EditText>(R.id.passwordInput)
val registerBtn = findViewById<Button>(R.id.registerBtn)

```

Ambil data user yang sudah ada dan ubah ke dalam bentuk *MutableMap*

```
val sharedPref = getSharedPreferences(name: "user_pref", Context.MODE_PRIVATE)
val jsonUsers = sharedPref.getString(key: "users", defValue: "{}")
val type = object : TypeToken<MutableMap<String, String>>() {}.type
val userMap: MutableMap<String, String> = Gson().fromJson(jsonUsers, type)
```

Validasi input, jika OK maka simpan user ke *SharedPreferences* dan kembali ke login.

```
registerBtn.setOnClickListener {
    val username = usernameInput.text.toString().trim()
    val password = passwordInput.text.toString().trim()

    if (username.isEmpty() || password.isEmpty()) {
        Toast.makeText(context: this, text: "Semua field harus diisi", Toast.LENGTH_SHORT).show()
    } else if (userMap.containsKey(username)) {
        Toast.makeText(context: this, text: "Username sudah digunakan", Toast.LENGTH_SHORT).show()
    } else {
        userMap[username] = password
        sharedPref.edit().putString("users", Gson().toJson(userMap)).apply()
        Toast.makeText(context: this, text: "Registrasi berhasil", Toast.LENGTH_SHORT).show()
        startActivity(Intent(packageContext: this, LoginActivity::class.java))
        finish()
    }
}
```

ListChatActivity.kt

Disini pada *setContentview* menampilkan UI dari *activity_list_chat.xml*, mengakses *RecyclerView* dari *layout* dan mengatur agar tampil secara vertikal dengan *LinearLayoutManager*, membuat daftar contoh pengguna dan membuat instance *chatAdapter* dan menghubungkannya ke *RecyclerView* agar data muncul di layar.

```
package com.example.essay_uts

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView

class ListChatActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_list_chat)

        val recyclerView = findViewById<RecyclerView>(R.id.chatRecyclerView)
        recyclerView.layoutManager = LinearLayoutManager(context: this)

        val chatList = listOf(
            ChatItem(s: "Alice", st: "https://randomuser.me/api/portraits/women/1.jpg"),
            ChatItem(s: "Bob", st: "https://randomuser.me/api/portraits/men/2.jpg"),
            ChatItem(s: "Clara", st: "https://randomuser.me/api/portraits/women/3.jpg"),
            ChatItem(s: "David", st: "https://randomuser.me/api/portraits/men/4.jpg")
        )

        val adapter = ChatAdapter(chatList)
        recyclerView.adapter = adapter
    }
}
```

ChatItem.kt

Mendefinisikan data class Bernama ChatItem

```
package com.example.essay_uts

class ChatItem(s: String, s1: String) {
    val username: String = s
    val imageUrl: String = s1
}
```

ChatAdapter.kt

Membuat adapter untuk *RecyclerView*, menerima daftar *ChatItem*,
Lalu pada *ChatViewHolder* menyimpan referensi ke elemen – elemen UI
dalam *Item_chat.xml* agar bisa diisi datanya (teks dan gambar). Kemudian
membuat tampilan item dari layout *item_chat.xml* dan membungkus dalam
ChatViewHolder dan terakhir menampilkan data *ChatItem* ke dalam
tampilan

```
class ChatAdapter(private val chatList: List<ChatItem>) :
    RecyclerView.Adapter<ChatAdapter.ChatViewHolder>() {

    class ChatViewHolder(view: View) : RecyclerView.ViewHolder(view) {
        val usernameText: TextView = view.findViewById(R.id.usernameText)
        val profileImage: ImageView = view.findViewById(R.id.profileImage)
    }

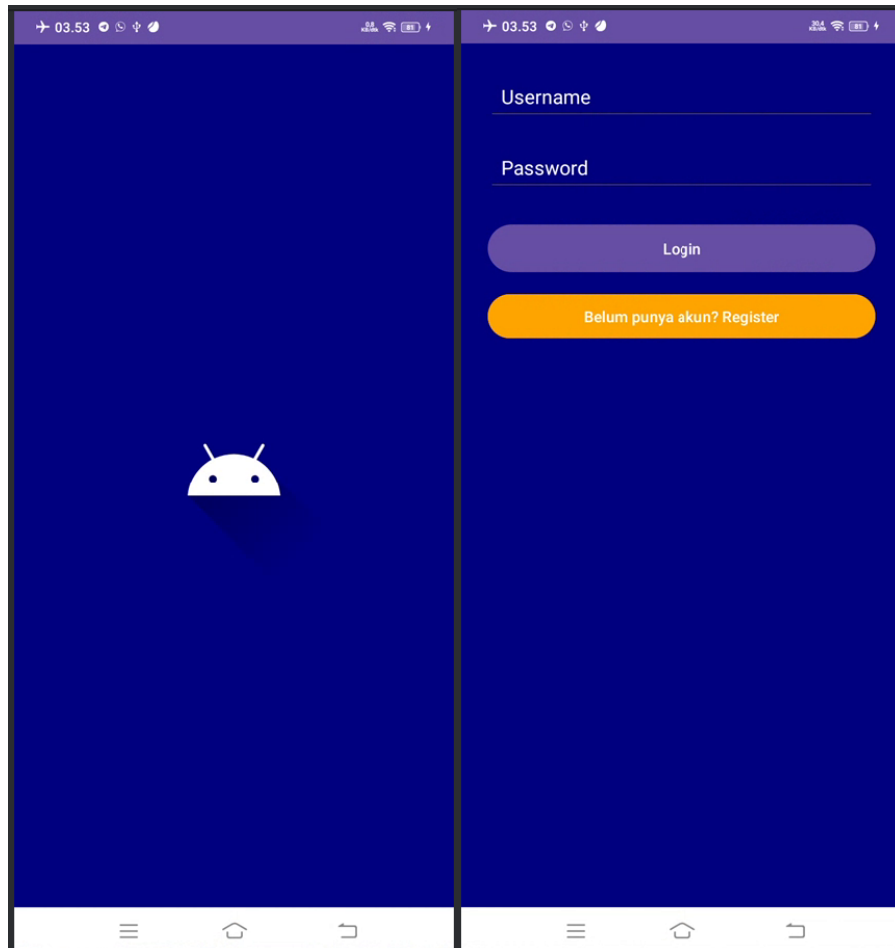
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ChatViewHolder {
        val view = LayoutInflater.from(parent.context)
            .inflate(R.layout.item_chat, parent, attachToRoot: false)
        return ChatViewHolder(view)
    }

    override fun onBindViewHolder(holder: ChatViewHolder, position: Int) {
        val item = chatList[position]
        holder.usernameText.text = item.username
        Glide.with(holder.profileImage.context)
            .load(item.imageUrl)
            .into(holder.profileImage)
    }

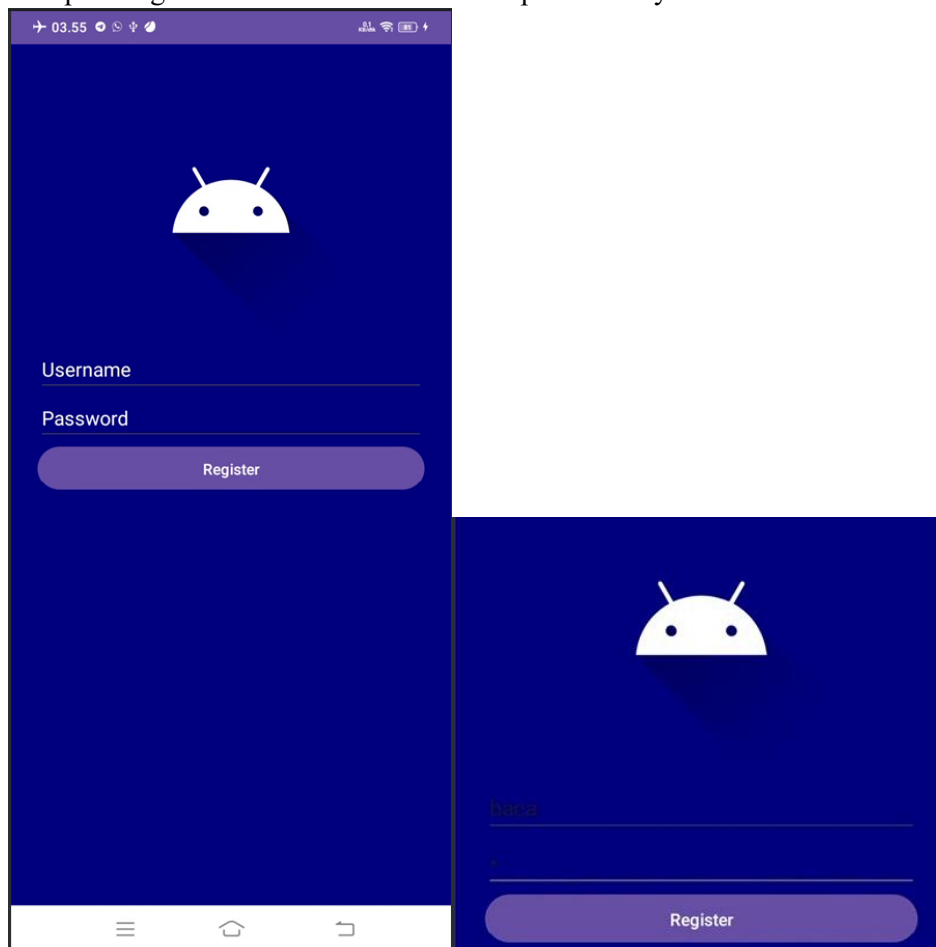
    override fun getItemCount(): Int = chatList.size
}
```

Pengujian:

Pertama masuk ke bagian splashscreen, lalu masuk ke bagian login bila belum memiliki akun bisa pergi ke register

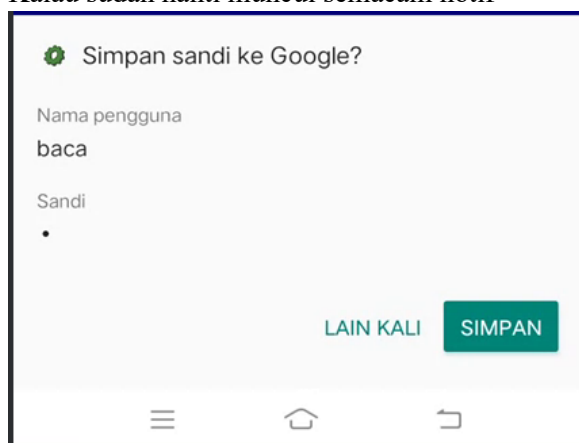


Nah pada register kita isikan username dan password nya



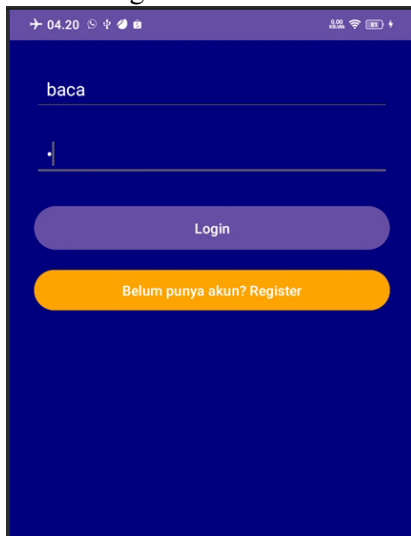
The image displays two side-by-side screenshots of an Android application's registration screen. Both screens have a dark blue background with a white Android robot icon at the top. The left screenshot shows the registration form with two input fields: 'Username' and 'Password', and a purple 'Register' button below them. The right screenshot shows the same form, but the 'Username' field is filled with the text 'baca' and the 'Password' field is filled with the text 'a'. The 'Register' button is also present on the right screenshot.

Kalau sudah nanti muncul semacam notif

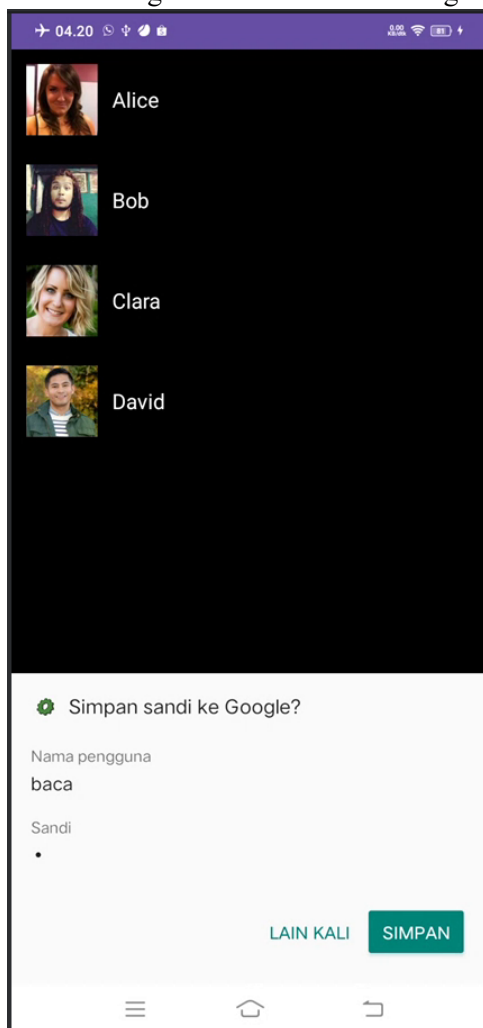


The image shows a screenshot of an Android notification dialog. The dialog has a light gray background and a title bar with a green gear icon and the text 'Simpan sandi ke Google?'. Below the title bar, there are two labels: 'Nama pengguna' and 'Sandi'. The 'Nama pengguna' field is filled with the text 'baca'. The 'Sandi' field is filled with a single dot, indicating a masked password. At the bottom of the dialog, there are two buttons: 'LAIN KALI' and 'SIMPAN'. The 'SIMPAN' button is highlighted in green.

Kembali ke login ke isi username dan password yang telah dibuat lalu klik tombol Login.



Nah sekarang kita sudah masuk ke bagian List chat nya.



Link Github : <https://github.com/naufal-aulia-nuchrizal/Uts-223-CID-B/tree/main/folder>