

UTS

PEMROGRAMAN MOBILE 1

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



TIF RP 23 CID B

Dikdik Nawa Cendekia Agung

23552011240

DEPARTEMEN TEKNIK INFORMATIKA

FAKULTAS INDUSTRI KREATIF

UNIVERSITAS TEKNOLOGI BANDUNG

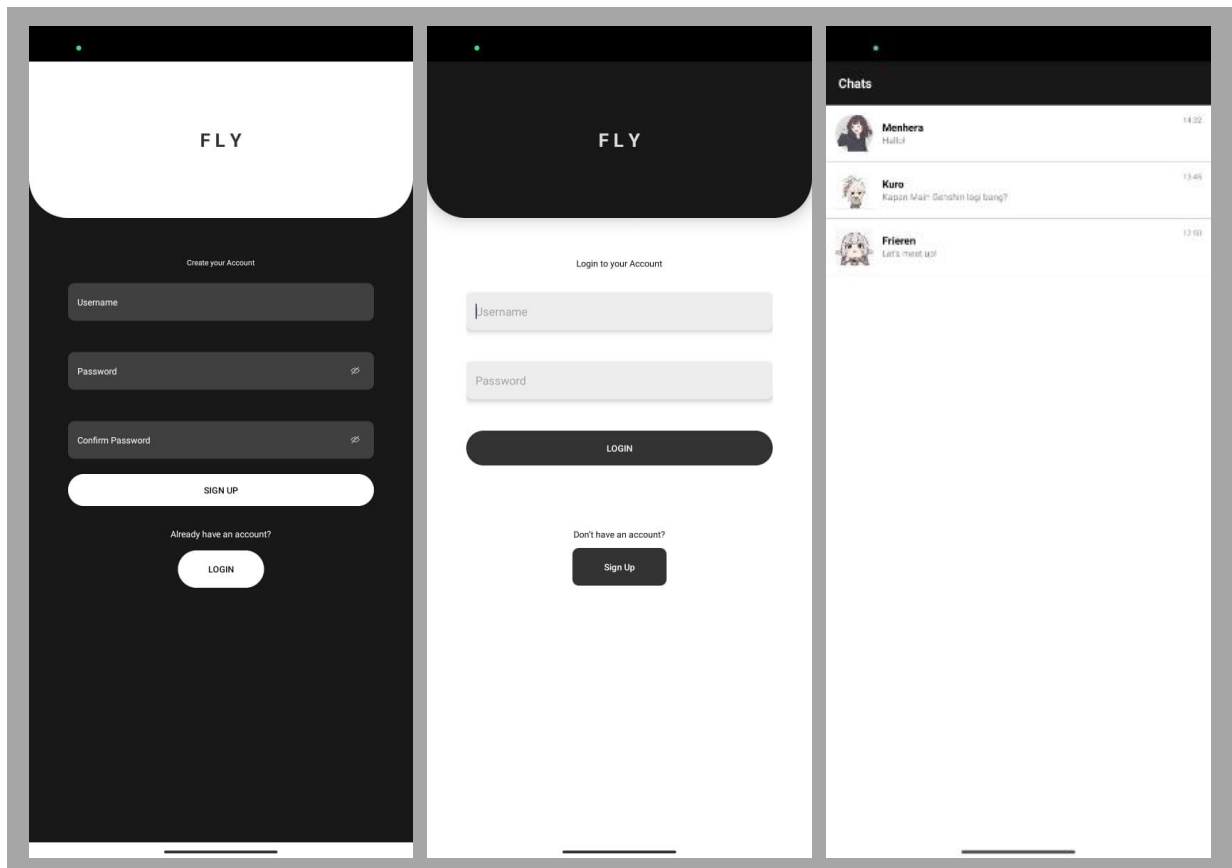
2025

NOTE : Untuk Essay dan penjelasan project ada di halaman berikutnya

Link Github : https://github.com/DikdikNawa/UTS_Pemrograman-Mobile_Dikdik-Nawa-Cendekia-Agung_23552011240.git

Ringkasan Tugas Project:

1. Terdapat Form Login dan Register yang bisa diakses oleh pengguna/user.
2. Dalam Form Login User dapat memasukkan Username “admin” dan Password “admin” yang sudah disediakan dalam ArrayList sebagai contoh yang dapat digunakan user untuk mencoba proses login agar berhasil.
3. Jika user ingin membuat akun baru, user dapat menekan button “Sign Up” yang akan mengarahkan user ke Form Register, dan user akan diminta untuk memasukkan Username dan Password. Jika proses Register berhasil dan muncul “Akun Berhasil Dibuat!”, dan` user dapat menggunakan Username dan Password yang tadi diinputkan dalam Form Login.
4. Jika proses Login berhasil user akan diarahkan ke menu ListChatting yang berisi contoh percakapan dengan beberapa karakter yang sudah disediakan oleh program yang dibuat.



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)

Jawaban

1. SetOnClickListener adalah sebuah method yang dapat dipakai di Android Studio menggunakan bahasa pemrograman (Java/Kotlin) yang digunakan untuk menangani aktivitas saat user mengklik pada suatu elemen User Interface(UI) yang tampil pada layar seperti button.
2. syarat pemanggilan method:
 - Komponen yang dipanggil harus di-inisialisasi dengan findViewById
 - Fungsi SetOnClikListener dilakukan dalam method onCreate() setelah setContentView()
 - Contoh:

```
btnLogin.setOnClickListener {  
    val Username = editTextUsername.text.toString()  
    val password = editTextPassword.text.toString()  
  
    val user = UserStorage.users.find { it.Username == Username && it.password == password }  
    if (user != null) {  
        val intent = Intent( packageContext: this, ListChattingActivity::class.java)  
        startActivity(intent)  
        overridePendingTransition(R.anim.slide_in_top, R.anim.slide_out_bottom)  
        finish()  
    } else {  
        Toast.makeText( context: this, text: "Email atau password salah", Toast.LENGTH_SHORT).show()  
    }  
}
```

- btnLogin adalah komponen yang dipanggil supaya ketika user klik button tersebut program akan melakukan perintah yang sudah ditetapkan
- saat button login di klik, program akan memeriksa Username dan Password yang diinputkan oleh pengguna, jika Username dan Password sesuai dengan data yang ada pada ArrayList maka pengguna dapat melanjutkan kedalam aplikasinya.
- OverridePendingTransition adalah tambahan yang digunakan agar ketika perpindahan activity/layout ada animasi/transisi supaya perpindahannya menjadi lebih rapih dan untuk menambah nilai estetika aplikasi.

3. Jika salah memanggil findViewById atau ID tidak ada di file layout.xml maka akan muncul error (java.lang.NullPointerException: Attempt to invoke virtual method on a null object reference) dan program tidak memunculkan layout pada layar.

```
Accessing hidden method Landroid/view/ViewGroup;~>makeOptionalFitSystemWindows(JV (unsupported, reflection, allowed)
Shutting down VM
FATAL EXCEPTION: main
Process: com.example.contohnull, PID: 13875
java.lang.RuntimeException: Unable to start activity ComponentInfo{com.example.contohnull/com.example.contohnull.MainActivity}: java.lang.NullPointerException
    at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:3780)
    at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:3947)
    at android.app.servertransaction.LaunchActivityItem.execute(LaunchActivityItem.java:106)
    at android.app.servertransaction.TransactionExecutor.executeCallbacks(TransactionExecutor.java:135)
    at android.app.servertransaction.TransactionExecutor.execute(TransactionExecutor.java:95)
    at android.app.ActivityThread$H.handleMessage(ActivityThread.java:2325)
    at android.os.Handler.dispatchMessage(Handler.java:106)
    at android.os.Looper.loopOnce(Looper.java:210)
    at android.os.Looper.loop(Looper.java:299)
    at android.app.ActivityThread.main(ActivityThread.java:8280) <1 internal line>
    at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:576)
    at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:1073)
Caused by: java.lang.NullPointerException
    at com.example.contohnull.MainActivity.onCreate(MainActivity.kt:16)
    at android.app.Activity.performCreate(Activity.java:8163)
    at android.app.Activity.performCreate(Activity.java:8130)
    at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1329)
    at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:3753) <12 more...>
```

4. Contoh Program yang menghasilkan error NullPointerException

```
1 package com.example.contohnull
2
3 > import ...
4
5
6
7
8
9
10 </> class MainActivity : AppCompatActivity() {
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         setContentView(R.layout.activity_main)
14
15         val myTextView: TextView? = null
16         myTextView!!.text = "Ini akan menyebabkan NullPointerException"
17     }
18 }
```

Penjelasan Kode Kotlin

1. SplashScreenActivity

```
1 package com.example.uts_pemrograman_mobile
2
3 import android.content.Intent
4 import android.os.Bundle
5 import android.os.Handler
6 import android.os.Looper
7 import android.view.WindowManager
8 import androidx.appcompat.app.AppCompatActivity
9
10 class SplashScreenActivity : AppCompatActivity() {
11
12     override fun onCreate(savedInstanceState: Bundle?) {
13         super.onCreate(savedInstanceState)
14         window.setFlags(
15             WindowManager.LayoutParams.FLAG_FULLSCREEN,
16             WindowManager.LayoutParams.FLAG_FULLSCREEN
17         )
18         setContentView(R.layout.activity_splash_screen)
19
20         Handler(Looper.getMainLooper()).postDelayed({
21             val intent = Intent(packageContext, LoginActivity::class.java)
22             startActivity(intent)
23             finish()
24         }, delayMillis = 2000)
25     }
26 }
```

Package menunjukkan file yang digunakan untuk menyimpan project

Import menunjukkan library yang digunakan dalam project

RegisterActivity adalah activity yang digunakan yang akan memperluas class

AppCompatActivity adalah halaman yang akan ditampilkan di android

Fungsi onCreate adalah fungsi yang dipanggil saat activity dibuat pertama kali

Class SplashScreenActivity menunjukkan kelas dengan nama SplashScreenActivity yang berisi logika yang digunakan agar kode dapat berjalan.

Fungsi Window.setFlags digunakan agar tampilan aplikasi menjadi Full Screen, mengapa ada kode yang dicoret seperti ~~FLAG_FULLSCREEN~~, itu dikarenakan ada cara yang lebih modern yang dapat digunakan untuk menerapkan fungsi tersebut, namun saya menggunakan fungsi window ini karena lebih mudah diimplementasikan dan banyak contohnya juga.

SetContentView adalah fungsi untuk memanggil layout agar terhubung ke file kotlin dan dapat muncul pada layar pengguna.

Fungsi Handler adalah untuk menentukan berapa lama layout tersebut akan ditampilkan sebelum diarahkan ke layout berikutnya (dalam kasus diatas setelah deay 2000 milisecond akan langsung diarahkan ke layout Login)

2. LoginActivity

```
1 package com.example.uts_pemrograman_mobile
2
3 > import ...
12
13 <> class LoginActivity : AppCompatActivity() {
14     =
15     @ override fun onCreate(savedInstanceState: Bundle?) {
16         super.onCreate(savedInstanceState)
17         setContentView(R.layout.activity_login)
18
19         window.setFlags(
20             WindowManager.LayoutParams.FLAG_FULLSCREEN,
21             WindowManager.LayoutParams.FLAG_FULLSCREEN
22         )
23
24         window.decorView.systemUiVisibility =
25             View.SYSTEM_UI_FLAG_LAYOUT_HIDE_NAVIGATION or
26             View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN or
27             View.SYSTEM_UI_FLAG_LAYOUT_STABLE
28
29         window.navigationBarColor = Color.TRANSPARENT
30         window.statusBarColor = Color.TRANSPARENT
31
32         val editTextUsername = findViewById<EditText>(R.id.editTextUsername)
33         val editTextPassword = findViewById<EditText>(R.id.editTextPassword)
34         val btnLogin = findViewById<Button>(R.id.btnLogin)
35         val btnSignUp = findViewById<Button>(R.id.btnSignUp)
36
37         btnLogin.setOnClickListener {
38             val Username = editTextUsername.text.toString()
39             val password = editTextPassword.text.toString()
40
41             val user = UserStorage.users.find { it.Username == Username && it.password == password }
42             if (user != null) {
43                 val intent = Intent(packageContext, this, ListChattingActivity::class.java)
44                 startActivity(intent)
45                 overridePendingTransition(R.anim.slide_in_top, R.anim.slide_out_bottom)
46                 finish()
47             } else {
48                 Toast.makeText(context, this, text: "Username atau password salah", Toast.LENGTH_SHORT).show()
49             }
50         }
51
52         btnSignUp.setOnClickListener {
53             val intent = Intent(packageContext, this, RegisterActivity::class.java)
54             startActivity(intent)
55             overridePendingTransition(R.anim.slide_in_top, R.anim.slide_out_bottom)
56         }
57     }
58 }
59
```

Fungsi `Window.setFlags` digunakan agar tampilan aplikasi menjadi Full Screen, mengapa ada kode yang dicoret seperti ~~`FLAG_FULLSCREEN`~~, itu dikarenakan ada cara yang lebih modern yang dapat digunakan untuk menerapkan fungsi tersebut, namun saya menggunakan fungsi `window` ini karena lebih mudah diimplementasikan dan banyak contohnya juga.

`Window.navigation` berfungsi untuk membantu agar pengguna yang menggunakan navigasi gerakan agar layar yang ada pada bagian bawah menjadi transparan.

`Val editTextUsername, editTextPassword` adalah logika agar data yang dimasukkan pengguna dalam textbox username dan password dapat di periksa oleh logika dalam program kotlin.

`Val btn` adalah deklarasi agar button dapat berfungsi dan terhubung dengan logika yang ada pada file kotlin

`BtnLogin.SetOnClickListener` adalah method yang digunakan agar program dapat memeriksa kolom textbox/textfield yang sudah dimasukkan oleh user untuk dibandingkan dengan data yang tersimpan dalam program. Jika usernam dan passwordnya sesuai maka proses login akan berhasil, dan jika salah satu atau keduanya salah maka akan muncul pesan "Username atau Password salah"

Fungsi `overridePendingTransition` adalah animasi yang menangani ketika terjadi perpindahan antar layout. Dalam kasus diatas animasi yang digunakan adalah `slide_in_top`, dan `slide_out_bottom` yang sebelumnya sudah dibuat dan disimpan dalam folder `res/anim`.

`BtnSignUp.SetOnClickListener` adalah method yang digunakan untuk menghubungkan layout Login dan layout Register, jadi saat pengguna klik button "Sign Up" maka pengguna akan diarahkan ke layout Register.

`Val intent` adalah variabel yang digunakan untuk memulai tindakan.

3. RegisterActivity

```
10 class RegisterActivity : AppCompatActivity() {
11     override fun onCreate(savedInstanceState: Bundle?) {
12         window.setFlags(
13             WindowManager.LayoutParams.FLAG_FULLSCREEN,
14             WindowManager.LayoutParams.FLAG_FULLSCREEN
15         )
16         setContentView(R.layout.activity_register)
17
18         val editTextUsername = findViewById<EditText>(R.id.editTextUsername)
19         val editTextPassword = findViewById<EditText>(R.id.editTextPassword)
20         val editTextConfirmPassword = findViewById<EditText>(R.id.editTextConfirmPassword)
21         val togglePassword = findViewById<ImageView>(R.id.togglePassword)
22         val toggleConfirmPassword = findViewById<ImageView>(R.id.toggleConfirmPassword)
23         val btnSignUp = findViewById<Button>(R.id.btnSignUp)
24         val btnLogin = findViewById<Button>(R.id.btnLogin)
25
26         var isPasswordVisible = false
27         var isConfirmPasswordVisible = false
28
29         togglePassword.setOnClickListener {
30             isPasswordVisible = !isPasswordVisible
31             if (isPasswordVisible) {
32                 editTextPassword.inputType = InputType.TYPE_TEXT_VARIATION_VISIBLE_PASSWORD
33                 togglePassword.setImageResource(R.drawable.eyepass_white)
34             } else {
35                 editTextPassword.inputType =
36                     InputType.TYPE_CLASS_TEXT or InputType.TYPE_TEXT_VARIATION_PASSWORD
37                 togglePassword.setImageResource(R.drawable.eyepass_white)
38             }
39             editTextPassword.setSelection(editTextPassword.text.length)
40         }
41
42         toggleConfirmPassword.setOnClickListener {
43             isConfirmPasswordVisible = !isConfirmPasswordVisible
44             if (isConfirmPasswordVisible) {
45                 editTextConfirmPassword.inputType = InputType.TYPE_TEXT_VARIATION_VISIBLE_PASSWORD
46                 toggleConfirmPassword.setImageResource(R.drawable.eyepass_white)
47             } else {
48                 editTextConfirmPassword.inputType =
49                     InputType.TYPE_CLASS_TEXT or InputType.TYPE_TEXT_VARIATION_PASSWORD
50                 toggleConfirmPassword.setImageResource(R.drawable.eyepass_white)
51             }
52             editTextConfirmPassword.setSelection(editTextConfirmPassword.text.length)
53         }
54
55         btnSignUp.setOnClickListener {
56             val Username = editTextUsername.text.toString()
57             val password = editTextPassword.text.toString()
58             val confirmPassword = editTextConfirmPassword.text.toString()
59
60             if (password == confirmPassword) {
61                 UserStorage.users.add(User(Username, password))
62                 Toast.makeText(context, this, text, "Akun berhasil dibuat!", Toast.LENGTH_SHORT).show()
63                 val intent = Intent(packageContext, this, LoginActivity::class.java)
64                 startActivity(intent)
65                 overridePendingTransition(R.anim.slide_in_top, R.anim.slide_out_bottom)
66                 finish()
67             } else {
68                 Toast.makeText(context, this, text, "Password tidak sesuai", Toast.LENGTH_SHORT).show()
69             }
70         }
71
72         btnLogin.setOnClickListener {
73             val intent = Intent(packageContext, this, LoginActivity::class.java)
```


RegisterActivity adalah activity yang digunakan yang akan memperluas class

AppCompatActivity adalah halaman yang akan ditampilkan di android

Fungsi onCreate adalah fungsi yang dipanggil saat activity dibuat pertama kali

Val Toggle adalah variabel yang digunakan pada fungsi menampilkan password ketika pengguna klik logo "mata" pada bagian kanan di dalam textbox password dan confirm password yang akan menampilkan atau menyembunyikan password

Jika pengguna klik toggle ini maka password akan ditampilkan, dan jika pengguna mengklik lagi maka password akan disembunyikan

Dan jika pengguna memasukkan password yang berbeda didalam password dan confirm password maka akan muncul pesan "Password tidak sesuai", dan jika pengguna berhasil membuat akun maka akan muncul "Akun berhasil dibuat"

setImageResources adalah untuk menghubungkan agar gambar yang telah disimpan pada res dapat digunakan dan ditampilkan

setSelection memastikan agar kursor tetap di akhir teks

method btnSignUp adalah fungsi untuk menyimpan data kedalam arraylist sekaligus mengembalikan pengguna ke halaman login.

4. ListChatting Activity

```
1 package com.example.uts_pemrograman_mobile
2
3 import android.os.Bundle
4 import android.view.WindowManager
5 import android.widget.ListView
6 import android.widget.Toast
7 import androidx.appcompat.app.AppCompatActivity
8
9 class ListChattingActivity : AppCompatActivity() {
10
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         window.setFlags(
14             WindowManager.LayoutParams.FLAG_FULLSCREEN,
15             WindowManager.LayoutParams.FLAG_FULLSCREEN
16         )
17
18         setContentView(R.layout.activity_list_chatting)
19
20         val listViewChats: ListView = findViewById(R.id.listViewChats)
21
22         val chatList = mutableListOf(
23             Chat(name: "Menhera", lastMessage: "Hallo!", time: "14:32", R.drawable.anime_ok),
24             Chat(name: "Kuro", lastMessage: "Kapan Main Genshin lagi bang?", time: "13:45", R.drawable.kazuha),
25             Chat(name: "Frieren", lastMessage: "Let's meet up!", time: "12:50", R.drawable.frieren)
26         )
27
28         val chatAdapter = ChatAdapter(context: this, chatList)
29         listViewChats.adapter = chatAdapter
30
31         listViewChats.setOnItemClickListener { _, _, position, _ ->
32             val selectedChat = chatList[position]
33             Toast.makeText(context: this, text: "Chat dengan ${selectedChat.name}", Toast.LENGTH_SHORT).show()
34         }
35     }
36 }
37 | Ctrl+I to Command
```

Inisialisasi listView untuk menghubungkan antara kotlin dan komponen ListView dari layout

Data chatList Menhera, Kuro, dan Frieren adalah contoh chat yang ada didalam program dan ini juga merupakan data class yang digunakan untuk menyimpan nama, pesan, waktu, dan gambar.

Val chatAdapter adalah variabel untuk memanggil class ChatAdapter dan menampilkannya kepada pengguna

5. ChatAdapter

```
1 package com.example.Uts_pemrograman_mobile
2
3 import android.content.Context
4 import android.view.LayoutInflater
5 import android.view.View
6 import android.view.ViewGroup
7 import android.widget.BaseAdapter
8 import android.widget.ImageView
9 import android.widget.TextView
10
11 class ChatAdapter(private val context: Context, private val chatList: List<Chat>) : BaseAdapter() {
12
13     override fun getCount(): Int = chatList.size
14
15     override fun getItem(position: Int): Any = chatList[position]
16
17     override fun getItemId(position: Int): Long = position.toLong()
18
19     override fun getView(position: Int, convertView: View?, parent: ViewGroup?): View {
20         val chat = getItem(position) as Chat
21
22         val inflater = context.getSystemService(Context.LAYOUT_INFLATER_SERVICE) as LayoutInflater
23         val view = inflater.inflate(R.layout.item_chat, root: null)
24
25         val tvName: TextView = view.findViewById(R.id.tvName)
26         val tvLastMessage: TextView = view.findViewById(R.id.tvLastMessage)
27         val tvTime: TextView = view.findViewById(R.id.tvTime)
28         val imgProfile: ImageView = view.findViewById(R.id.imgProfile)
29
30         tvName.text = chat.name
31         tvLastMessage.text = chat.lastMessage
32         tvTime.text = chat.time
33         imgProfile.setImageResource(chat.profileImage)
34
35         return view
36     }
37 }
38 Ctrl+I to Command
```

chatAdapter adalah adapter yang mewarisi BaseAdapter

digunakan untuk menampilkan data dalam ListView

chatlist adalah list data yang ada dari objek chat

override fun getCount berfungsi untuk mengembalikan jumlah item kedalam list

override fun getItem untuk mengambil chat pada posisi tertentu

override fun getItemId untuk mengembalikan ID unik berdasarkan posisinya

tvName untuk menampilkan nama

imgprofile.setImageResource untuk mengambil dan menampilkan gambar profil dari ID resources

6. Chat

```
1 package com.example.uts_pemrograman_mobile
2
3 data class Chat(
4     val name: String,
5     val lastMessage: String,
6     val time: String,
7     val profileImage: Int
8 )
9
```

Data chat adalah class khusus pada kotlin yang dirancang untuk menyimpan data

Kotlin secara otomatis membuat method seperti hashCode() untuk pemetaan datanya.

7. User

```
1 package com.example.uts_pemrograman_mobile
2
3 data class User(val Username : String, val password : String)
4
5 object UserStorage {
6     val users = ArrayList<User>().apply {
7         add(User(Username: "admin", password: "admin"))
8     }
9 }
```

Data class user ini digunakan untuk menyimpan data saat pengguna membuat akun baru, atau digunakan sebagai data validasi ketika pengguna melakukan login.