

UJIAN TENGAH SEMESTER

Pemograman Mobile 1

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



Disusun Oleh :

Nama: Muhamad Akbar Ergiansyah

NPM: 23552011411

Kelas: TIF RP -223 CNS B

**FAKULTAS INDUSTRI KREATIF
DAPERTEMEN TEKNIK INFORMATIKA
UNIVERSITAS TEKNOLOGI BANDUNG
2025**

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. Fungsi `setOnClickListener` adalah untuk menetapkan sebuah listener (pendengar) pada sebuah view (tampilan) di Android, sehingga kode tertentu dapat dieksekusi ketika view tersebut diklik oleh pengguna.
2. Syarat pemanggilan method `setOnClickListener`:
 - **Inisialisasi View:** Instance `btnLogin` (asumsi: `Button`) harus telah diinisialisasi dengan benar melalui proses *view binding* atau `findViewById()`. Di Tempat yang Tepat:
 - **Lingkup Eksekusi:** Kode `setOnClickListener` harus dieksekusi dalam lingkup `Activity` atau `Fragment` yang valid.
 - **Implementasi `OnClickListener`:** Lambda expression `{ ... }` secara implisit mengimplementasikan interface `View.OnClickListener`, menyediakan *callback* untuk event klik.

Contoh:

```
btnLogin.setOnClickListener {  
    val email = emailInput.text.toString()  
    val password = passwordInput.text.toString()  
  
    // cek akun dummy bawaan  
    val dummyAccounts = mapOf(  
        "admin@gmail.com" to "admin123",  
        "user@gmail.com" to "user123"  
    )  
}
```

3. eror yang akan muncul ketika salah menginisialisasikan findViewById adalah Unresolved Reference dalam contoh password input memanggil id colum_passwod, yang seharusnya memiliki penamaan colum_password

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_login)  
  
    val emailInput = findViewById<EditText>(R.id.colum_email)  
    val passwordInput = findViewById<EditText>(R.id.colum_passwod)  
    val btnLogin = findViewById<Button>(R.id.btn_login)  
    val txtRegister = findViewById<TextView>(R.id.txt_up)
```

```
<EditText  
    android:id="@+id/colum_password"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:background="@drawable/colomininput"  
    android:hint="Enter Password"  
    android:inputType="textPassword"  
    android:textColor="#000000"  
    android:textColorHint="#000000"  
    android:layout_below="@+id/password"/>
```

Build Sync x Build Output x Build Analyzer x

Build failed At 10/05/2025 14:25 with 3 errors 12 sec, 741 ms

Download info

:app:compileDebugKotlin 1 error 3 sec, 311 ms

LoginActivity.kt app\src\main\java\com\example\uts_apkbookingtiketbioskop 1 error

Unresolved reference 'colum_passwod'. :18

org.jetbrains.kotlin.gradle.tasks.CompilationErrorException: Compilation error. See log for more details

Compilation error

4. Di program yang saya buat ini val btnlogin dan val txtRegister saya simpan di atas override yang menyebabkan eror NullPointerException karena mencoba mengakses objek yang bernilai null

```
class LoginActivity : AppCompatActivity() {
    val btnLogin = findViewById<Button>(R.id.btn_login)
    val txtRegister = findViewById<TextView>(R.id.txt_up)
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_login)

        val emailInput = findViewById<EditText>(R.id.colum_email)
        val passwordInput = findViewById<EditText>(R.id.colum_password)

        // akses SharedPreferences
        val sharedPref = getSharedPreferences("UserPref", MODE_PRIVATE)

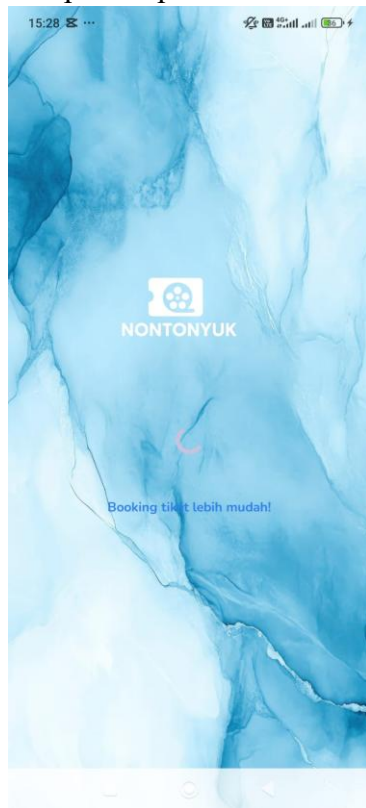
        btnLogin.setOnClickListener {
            val email = emailInput.text.toString()
            val password = passwordInput.text.toString()
        }
    }
}
```

```
at android.app.ActivityThread.main(ActivityThread.java:9593) <1 internal line>
at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:593)
at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:936)
Caused by: java.lang.NullPointerException: Attempt to read from field 'int android.content.pm.ApplicationInfo.targetSdkVersion' on a null object reference
at android.view.ContextThemeWrapper.getTheme(ContextThemeWrapper.java:174)
at android.content.Context.obtainStyledAttributes(Context.java:1059)
at androidx.appcompat.app.AppCompatActivityDelegateImpl.createSubDecor(AppCompatActivityDelegateImpl.java:898)
at androidx.appcompat.app.AppCompatActivityDelegateImpl.ensureSubDecor(AppCompatActivityDelegateImpl.java:865)
at androidx.appcompat.app.AppCompatActivityDelegateImpl.findViewById(AppCompatActivityDelegateImpl.java:667)
at androidx.appcompat.app.AppCompatActivity.findViewById(AppCompatActivity.java:264)
at com.example.uts_apkbookingtiketbioskop.LoginActivity.<init>(LoginActivity.kt:12)
at java.lang.Class.newInstance(Native Method)
```

STUDI KASUS

a. SplashScreenActivity

1) Tampilan Aplikasi



2) Codingan dan Penjelasan

```
package com.example.uts_apkbookingtiketbioskop
import android.content.Intent
import android.os.Bundle
import android.os.Handler
import android.view.animation.AnimationUtils
import android.widget.ImageView
import androidx.appcompat.app.AppCompatActivity
class SplashScreenActivity {

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

        val logo = findViewById<ImageView>(R.id.logo1)
        val animation = AnimationUtils.loadAnimation(context, R.anim.fade_in)
        logo.startAnimation(animation)

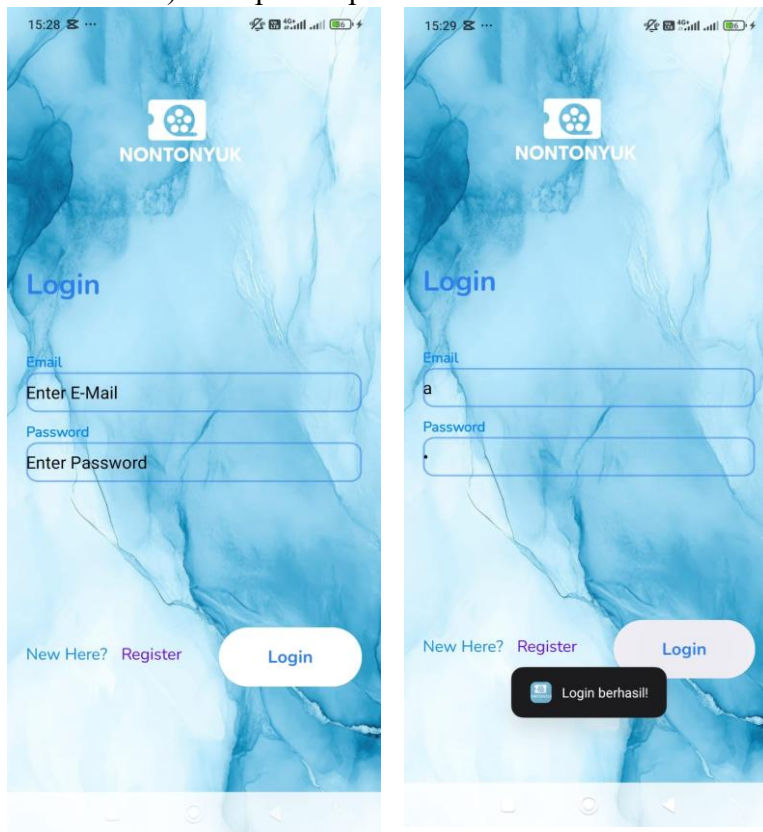
        Handler().postDelayed({
            startActivity(Intent(context, LoginActivity::class.java))
            finish()
        }, delayMillis: 3000) // 3 detik
    }
}
```

- Pada kode ini, pertama-tama saya mendefinisikan package dengan nama com.example.uts_apkbookingtiketbioskop. Package ini digunakan untuk mengelompokkan file-file program dalam satu aplikasi.
- Selanjutnya, saya melakukan import beberapa library penting seperti Intent, Bundle, Handler, AnimationUtils, ImageView, dan AppCompatActivity. Library-library ini diperlukan untuk mendukung fungsi intent berpindah activity, animasi, penanganan delay, serta penggunaan komponen ImageView.
- Kemudian saya membuat class utama bernama SplashScreenActivity yang merupakan turunan dari AppCompatActivity. Class ini berfungsi sebagai activity untuk menampilkan splash screen.

	<ul style="list-style-type: none"> • Di dalam class SplashScreenActivity, saya mengoverride method onCreate. Method ini dijalankan pertama kali ketika activity dibuat. • Di dalam onCreate, saya memanggil super.onCreate agar proses inisialisasi bawaan tetap berjalan. Lalu, saya menggunakan setContentView(R.layout.activity_splash) • untuk menampilkan layout activity_splash.xml ke layar. • Saya mendeklarasikan variabel logo yang dihubungkan ke komponen ImageView dengan ID logo1 menggunakan findViewById. • Setelah itu, saya memuat animasi fade_in dari resource anim menggunakan AnimationUtils.loadAnimation dan menyimpannya ke variabel animation. • Animasi fade_in kemudian dijalankan pada logo dengan perintah logo.startAnimation(animation). • Saya menggunakan Handler().postDelayed untuk memberikan jeda selama 3 detik (3000 milidetik) sebelum berpindah ke activity berikutnya. • Setelah jeda 3 detik, saya memanggil startActivity dengan intent menuju LoginActivity agar berpindah dari splash screen ke halaman login. Kemudian saya memanggil finish() untuk menutup splash screen agar activity ini tidak kembali saat tombol back ditekan.
--	--

b. Login Activity

1) Tampilan Aplikasi



2) Codingan dan Penjelasan

```
class LoginActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        val passwordInput = findViewById<EditText>(R.id.colum_password)
        val btnLogin = findViewById<Button>(R.id.btn_login)
        val txtRegister = findViewById<TextView>(R.id.txt_up)

        // akses SharedPreferences
        val sharedPref = getSharedPreferences("UserPref", MODE_PRIVATE)

        btnLogin.setOnClickListener {
            val email = emailInput.text.toString()
            val password = passwordInput.text.toString()

            // cek akun dummy bawaan
            val dummyAccounts = mapOf(
                "admin@gmail.com" to "admin123",
                "user@gmail.com" to "user123"
            )

            val savedPassword = sharedPref.getString(email, defaultValue = null)?: dummyAccounts[email]

            if (savedPassword != null && savedPassword == password) {
                Toast.makeText(context, "Login berhasil!", Toast.LENGTH_SHORT).show()
                startActivity(Intent(context, MainActivity::class.java))
            } else {
                Toast.makeText(context, "Login gagal! Periksa email atau password.", Toast.LENGTH_SHORT).show()
            }
        }

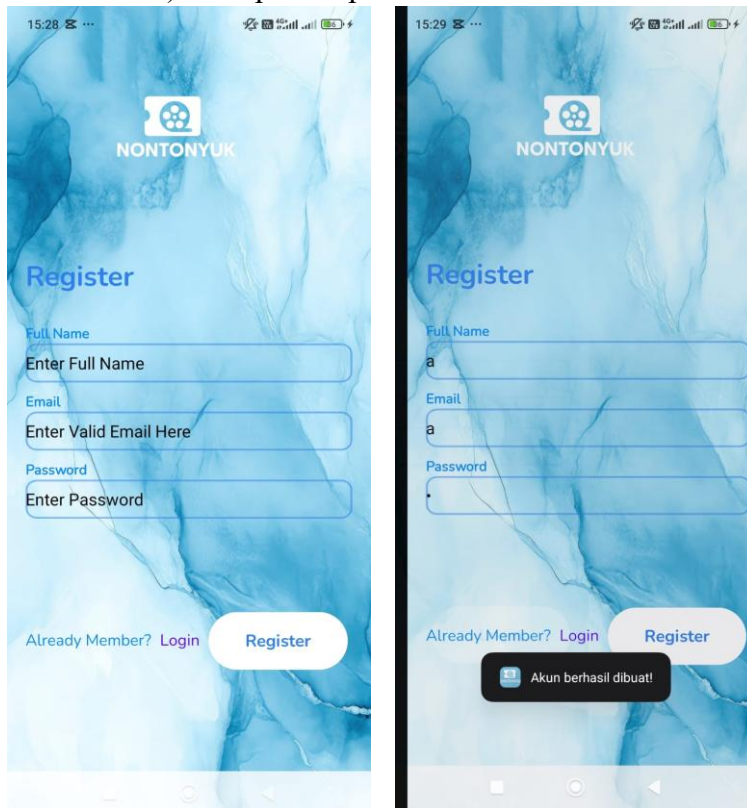
        txtRegister.setOnClickListener {
            val intent = Intent(context, RegisterActivity::class.java)
            startActivity(intent)
            overridePendingTransition(android.R.anim.slide_out_right, android.R.anim.slide_out_right)
        }
    }
}
```

- Pada kode ini, saya membuat class LoginActivity yang merupakan turunan dari AppCompatActivity. Class ini berfungsi sebagai halaman login di aplikasi.
- Di dalam class, saya mengoverride method onCreate yang akan dijalankan pertama kali ketika activity ini dibuat.
- Di dalam onCreate, saya mendefinisikan tiga variabel untuk menghubungkan komponen pada layout, yaitu passwordInput untuk EditText dengan ID colum_password, btnLogin untuk Button dengan ID btn_login, dan txtRegister untuk TextView dengan ID txt_up.
- Selanjutnya saya mengakses SharedPreferences dengan nama UserPref menggunakan mode MODE_PRIVATE.

	<p>SharedPreferences ini digunakan untuk menyimpan data akun secara lokal.</p> <ul style="list-style-type: none"> • Saya menambahkan event <code>setOnClickListener</code> pada tombol login (<code>btnLogin</code>) yang akan menangkap input email dan password ketika tombol ditekan. Email dan password diambil dari komponen <code>emailInput</code> dan <code>passwordInput</code>, lalu dikonversi menjadi string. • Saya membuat map <code>dummyAccounts</code> yang berisi akun email dan password secara hardcoded, misalnya <code>admin@gmail.com</code> dengan password <code>admin123</code>, dan <code>user@gmail.com</code> dengan password <code>user123</code>. Akun dummy ini digunakan sebagai akun bawaan untuk uji coba. • Kemudian saya mengambil data password yang tersimpan di <code>SharedPreferences</code> berdasarkan input email, dengan default value mengambil dari <code>dummyAccounts</code> jika data belum ada. • Setelah itu saya membuat logika pengecekan: jika <code>savedPassword</code> tidak null dan sama dengan password yang diinput, maka akan menampilkan pesan “Login berhasil!” menggunakan <code>Toast</code> dan pindah ke <code>MainActivity</code> menggunakan intent. • Jika email atau password salah, maka akan muncul <code>Toast</code> dengan pesan “Login gagal! Periksa email atau password.” • Saya juga menambahkan event <code>setOnClickListener</code> pada <code>txtRegister</code> yang jika diklik akan memindahkan user ke halaman <code>RegisterActivity</code> menggunakan intent. Transisi antar activity menggunakan animasi <code>slide out right</code>.
--	---

c. Register Activity

1) Tampilan Aplikasi



2) Codengan dan Penjelasan

```
package com.example.uts_apkbookingtiketbioskop

import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

class RegisterActivity : AppCompatActivity() {

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

        val inputEmail = findViewById<EditText>(R.id.colum_email)
        val inputPassword = findViewById<EditText>(R.id.colum_password)
        val btnRegister = findViewById<Button>(R.id.btn_register)
        val txtlogin = findViewById<TextView>(R.id.txt_login)

        // Simpan data di SharedPreferences
        val sharedPref = getSharedPreferences("UserPref", MODE_PRIVATE)

        btnRegister.setOnClickListener {
            val email = inputEmail.text.toString()
            val password = inputPassword.text.toString()

            if (email.isNotEmpty() && password.isNotEmpty()) {
                val editor = sharedPref.edit()
                editor.putString(email, password)
                editor.apply()

                Toast.makeText(context, this, "Akun berhasil dibuat!", Toast.LENGTH_SHORT).show()

                val intent = Intent(packageContext, LoginActivity::class.java)
                startActivity(intent)
                overridePendingTransition(android.R.anim.slide_in_left, android.R.anim.slide_out_right)
                finish()
            } else {
                Toast.makeText(context, this, "Kolom tidak boleh kosong!", Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```

- Pada kode ini, saya membuat class RegisterActivity yang merupakan turunan dari AppCompatActivity. Class ini berfungsi sebagai halaman registrasi (pendaftaran akun) pada aplikasi.
- Di dalam method onCreate, saya memanggil setContentView untuk menampilkan layout activity_register ke dalam activity ini.
- Saya mendefinisikan beberapa variabel untuk menghubungkan komponen UI dengan ID-nya masing-masing, yaitu inputEmail untuk EditText dengan ID colum_email, inputPassword untuk EditText dengan ID colum_password, btnRegister untuk Button dengan ID btn_register, dan txtlogin untuk TextView dengan ID txt_login.
- Selanjutnya, saya mengakses **SharedPreferences** dengan nama UserPref menggunakan mode MODE_PRIVATE untuk menyimpan data akun yang diinput user.

```

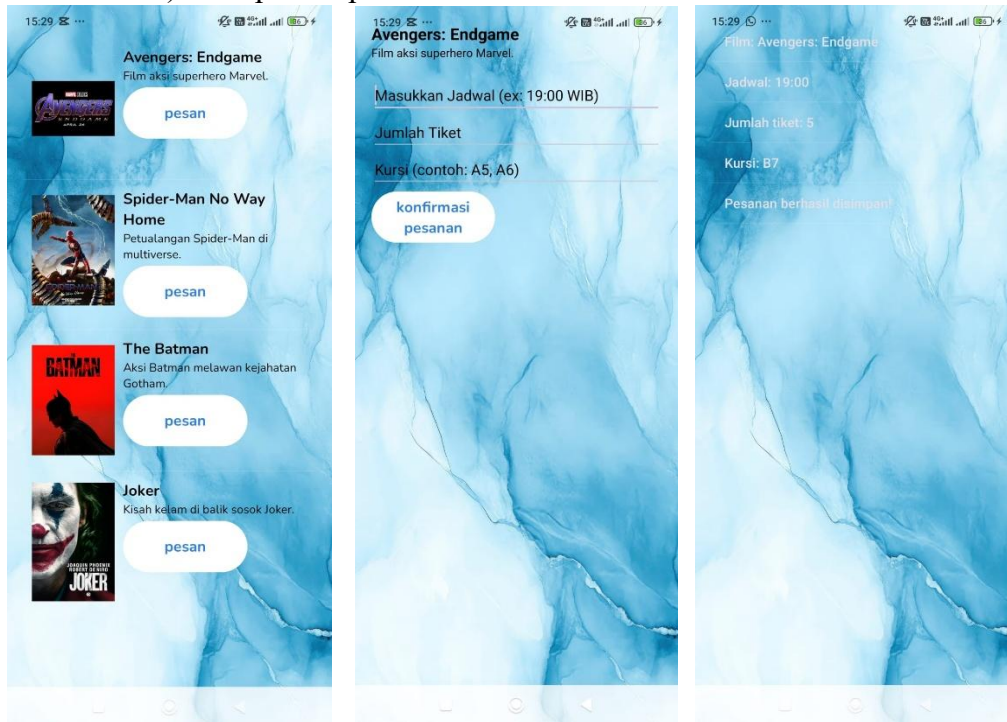
    }
    txtLogin.setOnClickListener {
        val intent = Intent(packageContext, LoginActivity::class.java)
        startActivity(intent)
        overridePendingTransition(android.R.anim.slide_in_left, android.R.anim.slide_out_right)
    }
}

```

- Saya menambahkan event **setOnClickListener** pada btnRegister. Ketika tombol register ditekan, akan diambil input email dan password dari inputEmail dan inputPassword lalu dikonversi ke string.
- Saya membuat logika pengecekan: jika input email dan password tidak kosong, maka data email dan password disimpan ke **SharedPreferences** menggunakan editor.putString() lalu editor.apply().
- Setelah berhasil disimpan, akan ditampilkan pesan Toast “Akun berhasil dibuat!” kemudian aplikasi langsung berpindah ke halaman **LoginActivity** dengan intent. Selain itu, saya menambahkan animasi transisi slide_in_left dan slide_out_right ketika pindah activity, lalu menutup halaman register dengan finish().
- Jika input email atau password kosong, maka akan ditampilkan Toast dengan pesan “Kolom tidak boleh kosong!” tanpa menyimpan data.
- Saya juga menambahkan event
- **setOnClickListener** pada txtlogin, sehingga jika user menekan teks “login”, maka akan langsung diarahkan ke **LoginActivity** dengan intent dan transisi animasi yang sama.

d. List Chating

1) Tampilan Aplikasi



2) Codingan dan Penjelasan

Didalam Listchat ini saya memiliki beberapa codingan kotlin terdiri dari:

ListChat, InputData, FilmAdapter, Dan Film

```
package com.example.uts_apkbookingtiketbioskop

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.ArrayAdapter
import android.widget.ListView

class ListChatActivity : AppCompatActivity() {

    private lateinit var listChat: ListView

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

        listChat = findViewById(R.id.listChat)

        val judul = intent.getStringExtra("judul")
        val jadwal = intent.getStringExtra("jadwal")
        val tiket = intent.getStringExtra("tiket")
        val kursi = intent.getStringExtra("kursi")

        val pesanChat = ArrayList<String>()
        pesanChat.add("Film: $judul")
        pesanChat.add("Jadwal: $jadwal")
        pesanChat.add("Jumlah tiket: $tiket")
        pesanChat.add("Kursi: $kursi")
        pesanChat.add("Pesanan berhasil disimpan!")

        val adapter = ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, pesanChat)
        listChat.adapter = adapter
    }
}
```

- Pada kode ini, saya membuat class **ListChatActivity** yang merupakan turunan dari **AppCompatActivity**. Class ini berfungsi untuk menampilkan ringkasan pesanan tiket dalam bentuk daftar/list.
- Di dalam method **onCreate**, pertama saya memanggil **setContentView** untuk menampilkan layout activity_list_chat ke dalam activity ini.
- Saya mendeklarasikan variabel **listChat** untuk menghubungkan komponen **ListView** dari layout dengan ID listChat.
- Selanjutnya, saya mengambil data yang dikirimkan melalui **intent** dari activity sebelumnya, yaitu data **judul film**, **jadwal tayang**, **jumlah tiket**, dan **nomor kursi**. Data ini diambil menggunakan **intent.getStringExtra()**.
- Saya membuat sebuah **ArrayList<String>** bernama **pesanChat** yang berisi string-string untuk

menampilkan informasi pesanan. Data-data yang sudah diambil dari intent kemudian ditambahkan ke pesanChat dengan format string seperti: "Film: (judul)", "Jadwal: (jadwal)", "Jumlah tiket: (tiket)", "Kursi: (kursi)", Pesan tambahan: "Pesanan berhasil disimpan!"

- Kemudian saya membuat sebuah **ArrayAdapter** untuk menghubungkan pesanChat ke komponen ListView. Adapter ini menggunakan layout standar Android `android.R.layout.simple_list_item_1` untuk menampilkan setiap item di dalam list.
- Terakhir, saya menetapkan adapter tersebut ke listChat dengan `listChat.adapter = adapter`, sehingga semua item dari pesanChat akan tampil di layar.

```
package com.example.uts_apkbookingtiketbioskop

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.*

class InputActivity : AppCompatActivity() {

    private lateinit var txtJudulFilm: TextView
    private lateinit var txtDeskripsiFilm: TextView
    private lateinit var inputJadwal: EditText
    private lateinit var inputTiket: EditText
    private lateinit var inputKursi: EditText
    private lateinit var btnKonfirmasi: Button

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_input)

        txtJudulFilm = findViewById(R.id.txtJudulFilm)
        txtDeskripsiFilm = findViewById(R.id.txtDeskripsiFilm)
        inputJadwal = findViewById(R.id.inputJadwal)
        inputTiket = findViewById(R.id.inputTiket)
        inputKursi = findViewById(R.id.inputKursi)
        btnKonfirmasi = findViewById(R.id.btnKonfirmasi)

        // ambil data dari MainActivity
        val judul = intent.getStringExtra("judul")
        val deskripsi = intent.getStringExtra("deskripsi")

        txtJudulFilm.text = judul
        txtDeskripsiFilm.text = deskripsi

        btnKonfirmasi.setOnClickListener {
            val jadwal = inputJadwal.text.toString()
            val tiket = inputTiket.text.toString()
            val kursi = inputKursi.text.toString()

            if (jadwal.isEmpty() || tiket.isEmpty() || kursi.isEmpty()) {
                Toast.makeText(context, "Semua field harus diisi!", Toast.LENGTH_SHORT).show()
                return@setOnClickListener
            }

            val intent = Intent(packageContext, ListChatActivity::class.java)
            intent.putExtra("judul", judul)
            intent.putExtra("jadwal", jadwal)
            intent.putExtra("tiket", tiket)
            intent.putExtra("kursi", kursi)
            startActivity(intent)
        }
    }
}
```

- Pada kode ini, saya membuat class `InputActivity` yang merupakan turunan dari `AppCompatActivity`. Class ini berfungsi untuk mengambil input detail pemesanan tiket dari user, seperti jadwal, jumlah tiket, dan nomor kursi. Data ini kemudian akan dikirimkan ke activity lain untuk ditampilkan.
- Di dalam method `onCreate`, pertama saya memanggil `setContentView` untuk menampilkan layout activity input ke dalam activity ini. Layout ini berisi komponen-komponen UI seperti `TextView` untuk menampilkan judul dan deskripsi film, `EditText` untuk input, dan `Button` untuk konfirmasi.
- Saya mendeklarasikan variabel-variabel untuk menghubungkan komponen-komponen UI (`TextView`, `EditText`, `Button`) dari layout dengan ID yang sesuai.
- Selanjutnya, saya mengambil data judul dan deskripsi film yang dikirimkan melalui intent dari activity sebelumnya (misalnya, activity daftar film). Data ini diambil menggunakan `intent.getStringExtra()`.
- Saya menampilkan judul dan deskripsi film yang sudah diambil ke dalam `TextView` yang sesuai.
- Kemudian, saya menambahkan `OnClickListener` pada Button "Konfirmasi". Ketika tombol ini ditekan:

	<ul style="list-style-type: none"> • Saya mengambil data yang diinput oleh user dari EditText (jadwal, jumlah tiket, nomor kursi). • Saya melakukan validasi untuk memastikan semua field telah diisi. Jika ada yang kosong, saya menampilkan pesan error. • Jika semua field terisi, saya membuat Intent untuk berpindah ke activity ListChatActivity. • Saya menambahkan data yang telah diinput (jadwal, jumlah tiket, nomor kursi) ke dalam Intent. Data ini akan dikirimkan ke ListChatActivity. • Terakhir, saya memulai ListChatActivity menggunakan startActivity(intent).
<pre> package com.example.uts_apkbookingtiketbioskop import android.content.Context import android.content.Intent import android.view.LayoutInflater import android.view.View import android.view.ViewGroup import android.widget.* import java.util.ArrayList class FilmAdapter(private val context: Context, private val listFilm: ArrayList<Film>) : BaseAdapter() { override fun getCount(): Int = listFilm.size override fun getItem(position: Int): Any = listFilm[position] override fun getItemId(position: Int): Long = position.toLong() override fun getView(position: Int, convertView: View?, parent: ViewGroup?): View { val view: View val holder: ViewHolder if (convertView == null) { view = LayoutInflater.from(context).inflate(R.layout.item_film, parent, attachToRoot = false) holder = ViewHolder(view.findViewById(R.id.imgFilm), view.findViewById(R.id.txtJudul), view.findViewById(R.id.txtDeskripsi), view.findViewById(R.id.btnPesan)) view.tag = holder } else { view = convertView holder = view.tag as ViewHolder } val film = listFilm[position] holder.imgFilm.setImageResource(film.gambar) holder.txtJudul.text = film.judul holder.txtDeskripsi.text = film.deskripsi } } </pre>	<ul style="list-style-type: none"> • Pada kode ini, saya membuat class FilmAdapter yang merupakan turunan dari BaseAdapter. Class ini berfungsi sebagai adapter untuk menampilkan daftar film dalam sebuah ListView atau GridView. Adapter ini bertugas untuk menghubungkan data film (dari ArrayList<Film>) dengan tampilan (layout item_film.xml). • Konstruktor FilmAdapter menerima dua parameter: context (konteks dari activity atau fragment yang menggunakan adapter ini) dan listFilm (daftar film yang akan ditampilkan). • Method-method yang di-override dari BaseAdapter: • getCount(): Mengembalikan jumlah total item dalam daftar film. • getItem(position): Mengembalikan objek film pada posisi tertentu dalam daftar. • getItemId(position): Mengembalikan ID unik untuk item pada posisi tertentu (biasanya indeks posisi). • getView(position, convertView, parent): Method terpenting. Bertanggung jawab untuk membuat atau menggunakan kembali tampilan (View) untuk setiap item dalam daftar.


```

holder.btnPesan.setOnClickListener {
    val intent = Intent(context, InputActivity::class.java)
    intent.putExtra(name: "judul", film.judul)
    intent.putExtra(name: "deskripsi", film.deskripsi)
    context.startActivity(intent)
}

return view
}

private data class ViewHolder(
    val imgFilm: ImageView,
    val txtJudul: TextView,
    val txtDeskripsi: TextView,
    val btnPesan: Button
)
}

```

- Di dalam getView(), dilakukan optimasi dengan menggunakan pola ViewHolder:
- Pertama, diperiksa apakah convertView (tampilan yang sudah ada) bernilai null. Jika null, berarti tampilan baru perlu dibuat.
- Jika convertView null:
- Layout item_film.xml di-inflate (dibuat menjadi objek View) menggunakan LayoutInflater.
- Objek ViewHolder dibuat. ViewHolder berfungsi untuk menyimpan referensi ke komponen-komponen UI dalam layout item_film.xml (ImageView untuk gambar film, TextView untuk judul dan deskripsi, Button untuk pesan).
- Referensi ke ViewHolder disimpan dalam tag View (view.tag = holder).
- Jika convertView tidak null:
- Tampilan yang sudah ada (convertView) digunakan kembali.
- Objek ViewHolder diambil dari tag View (holder = view.tag as ViewHolder).
- Data film pada posisi position diambil dari listFilm.
- Data film (gambar, judul, deskripsi) diatur ke komponen-komponen UI yang sesuai di dalam ViewHolder.
- OnClickListener ditambahkan ke Button "Pesan". Ketika tombol ini ditekan:
- Intent dibuat untuk berpindah ke InputActivity.
- Data judul dan deskripsi film ditambahkan ke Intent.
- InputActivity dimulai menggunakan context.startActivity(intent).
- ViewHolder (data class): Class ini berfungsi untuk menyimpan referensi ke komponen-komponen UI di dalam layout item_film.xml. Tujuannya adalah untuk menghindari pemanggilan findViewById() yang mahal setiap kali tampilan item di-render.

```
package com.example.uts_apkbookingtiketbioskop  
data class Film(val gambar: Int, val judul: String, val deskripsi: String)
```

- Film adalah *data class* yang menyimpan informasi film: gambar (resource ID), judul, dan deskripsi.

e. Link github

https://github.com/MuhamadAkbarErgiansyah/UTS_Muhamad-Akbar-Ergiansyahh