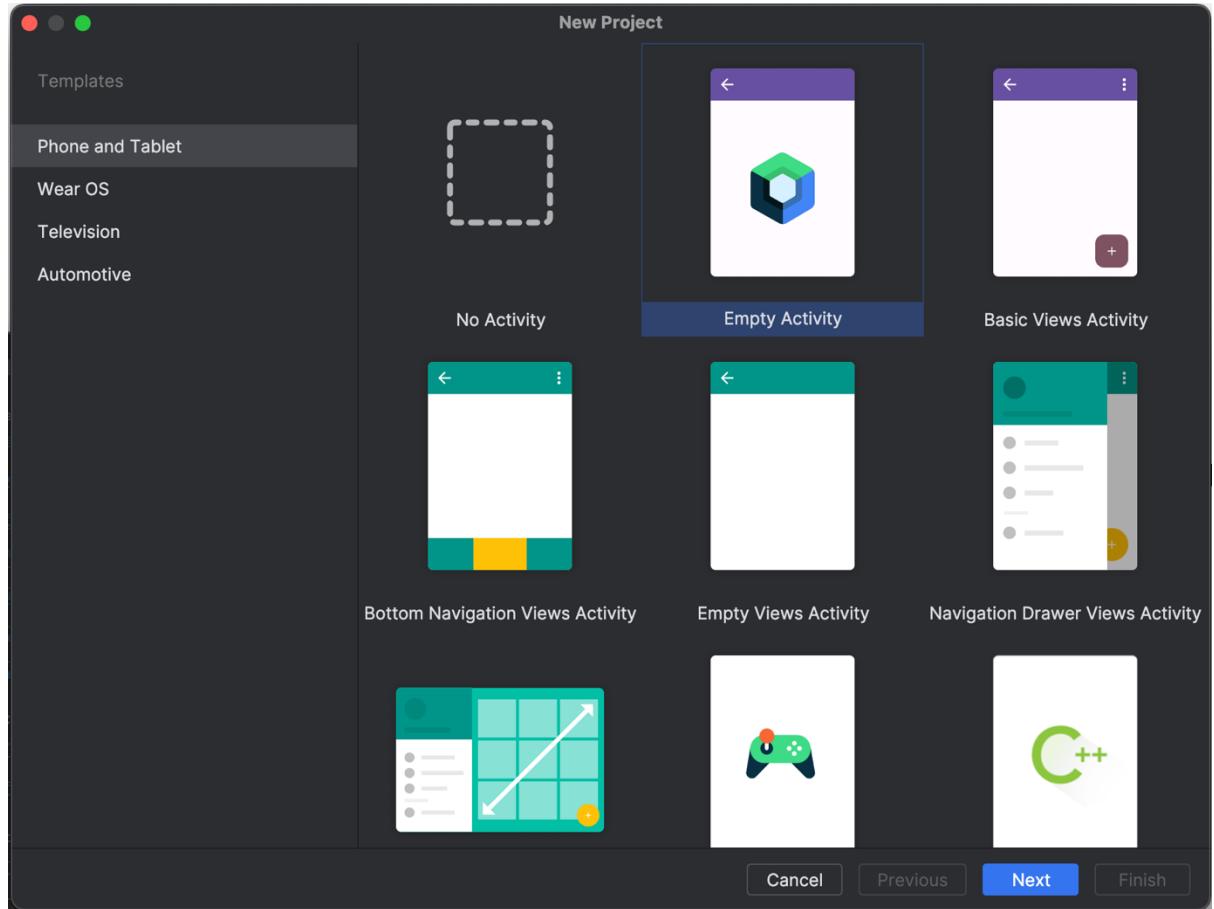
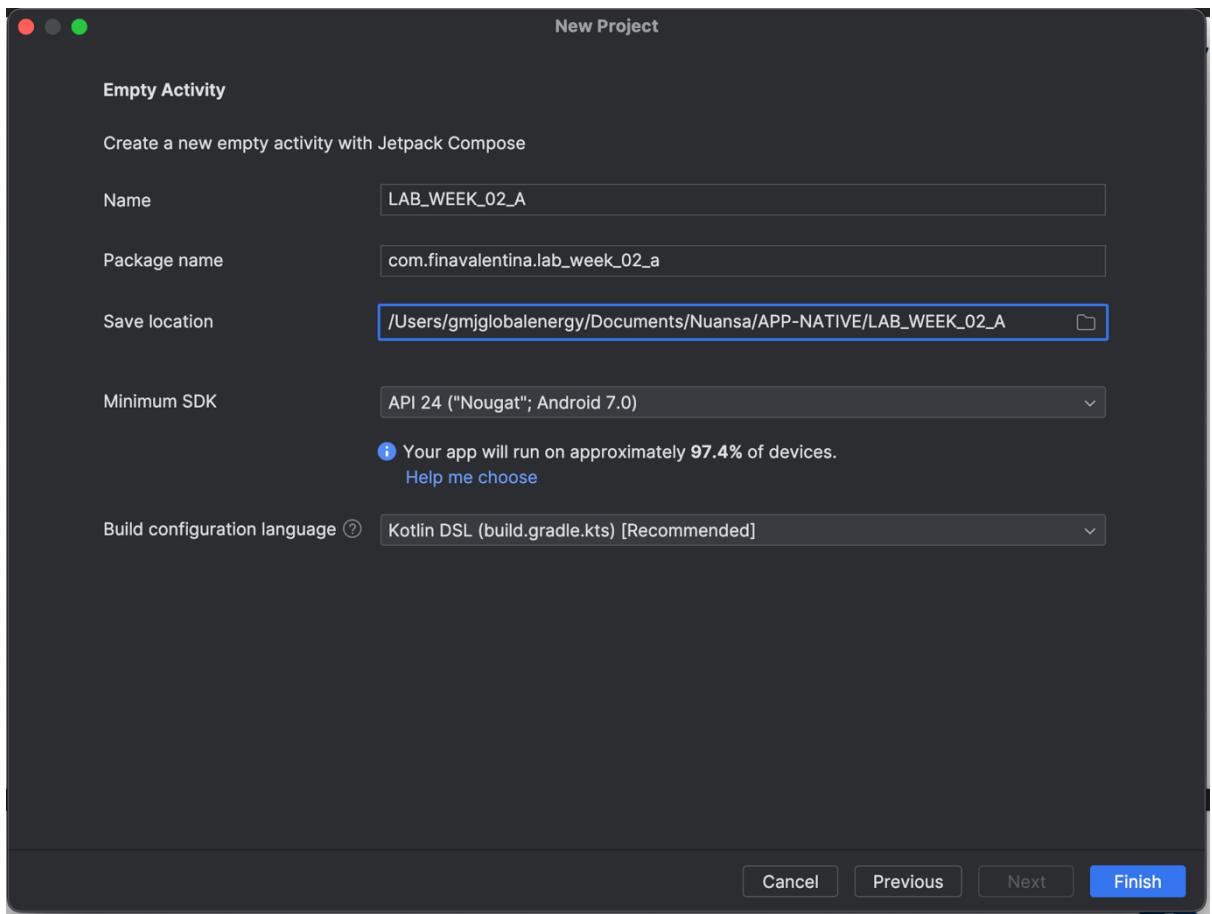


LANGKAH LANGKAH PENGERJAAN LAB WEEK 02 A

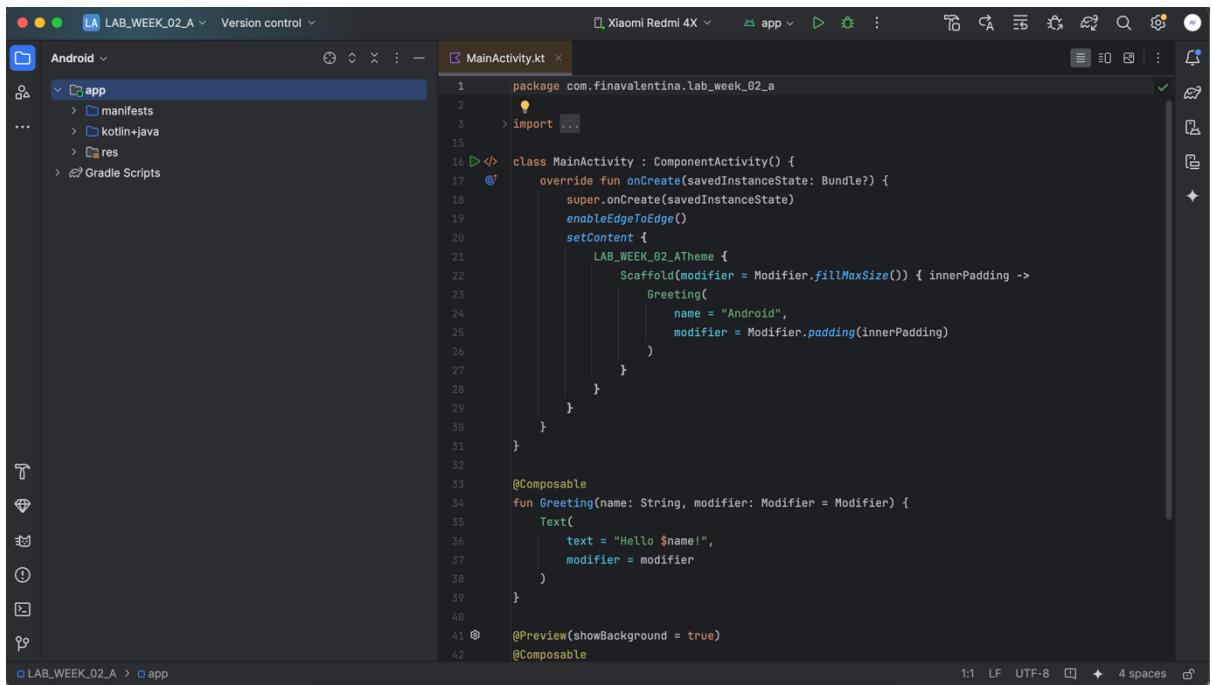
1. Buat Project, dan pilih Empty Activity dan klik “Next”



2. Masukan nama project dan atur SDK dll, lalu klik “Finish”



3. Maka akan tampil editor yang menampilkan MainActivity



4. Ubah code MainActivity dan import AppCompatActivity, untuk pertama kali pilih "Add dependency on androidx ..."

The screenshot shows the Android Studio interface with the code editor open to `MainActivity.kt`. The code is as follows:

```
1 package com.finavalentina.lab_week_02_a
2
3 > import ...
4
5
6 class MainActivity : AppCompatActivity() {
7 }
```

A red exclamation mark icon is present at the end of line 16, indicating an error. A tooltip message "Unresolved reference: AppCompatActivity" appears above the cursor. Below the code editor, the status bar shows the path "LAB_WEEK_02_A > app > src > main > java > com > finavalentina > lab_week_02_a > MainActivity".

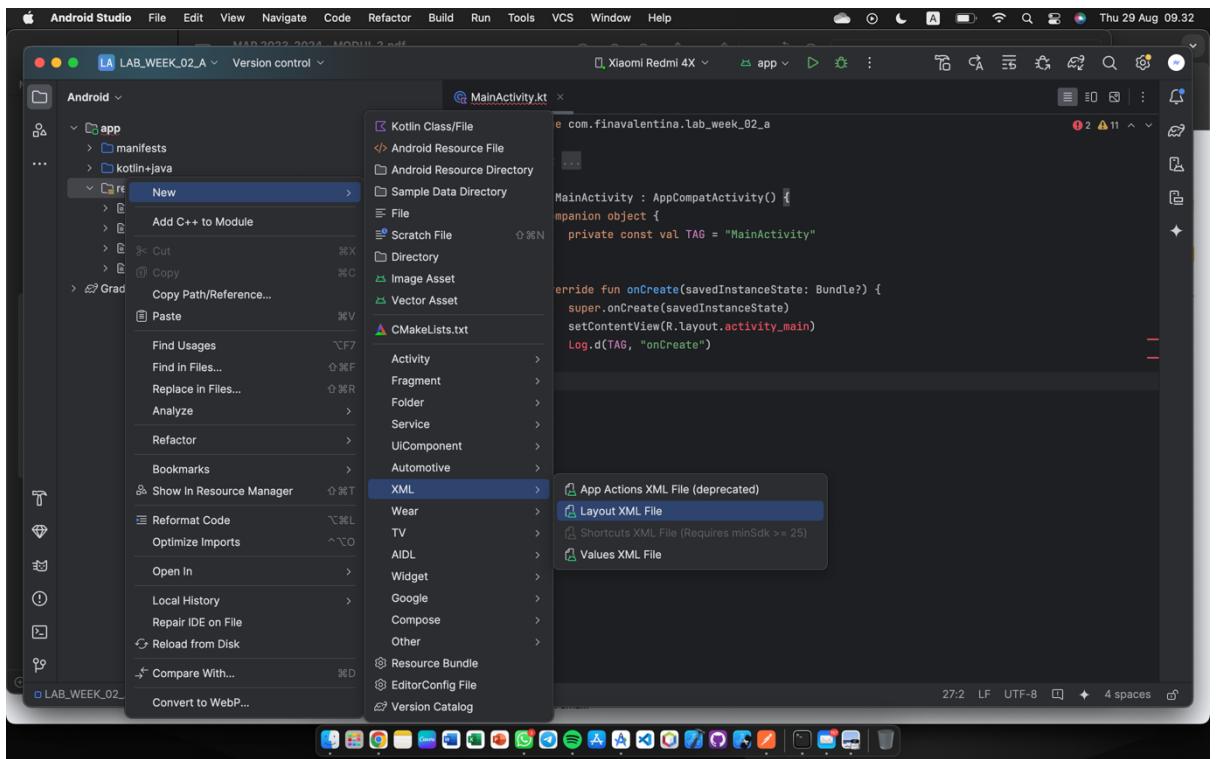
5. Lanjutkan edit file MainActivity seperti ini

The screenshot shows the Android Studio interface with the code editor open to `MainActivity.kt`. The code has been updated to:

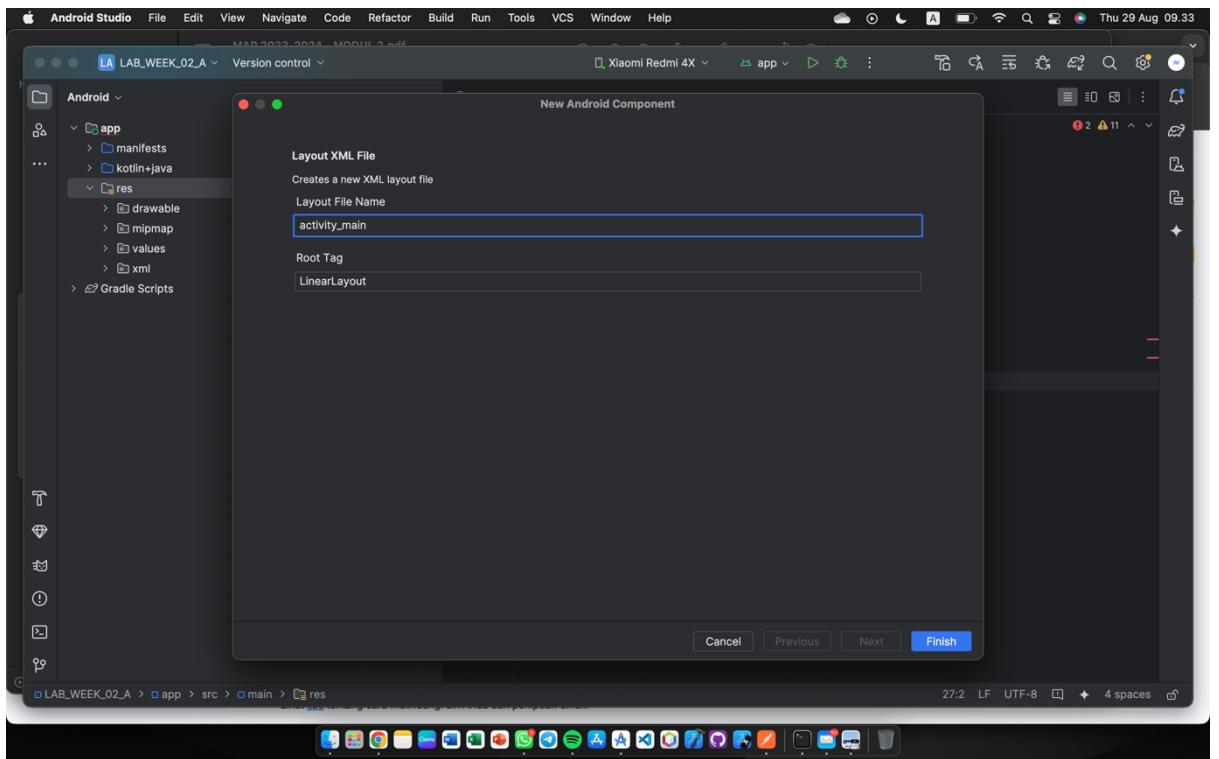
```
1 package com.finavalentina.lab_week_02_a
2
3 > import ...
4
5
6 class MainActivity : AppCompatActivity() {
7     companion object {
8         private const val TAG = "MainActivity"
9     }
10
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         setContentView(R.layout.activity_main)
14         Log.d(TAG, "onCreate")
15     }
16 }
```

The red exclamation mark icon from the previous screenshot is no longer visible. The status bar shows the path "LAB_WEEK_02_A > app > src > main > java > com > finavalentina > lab_week_02_a > MainActivity".

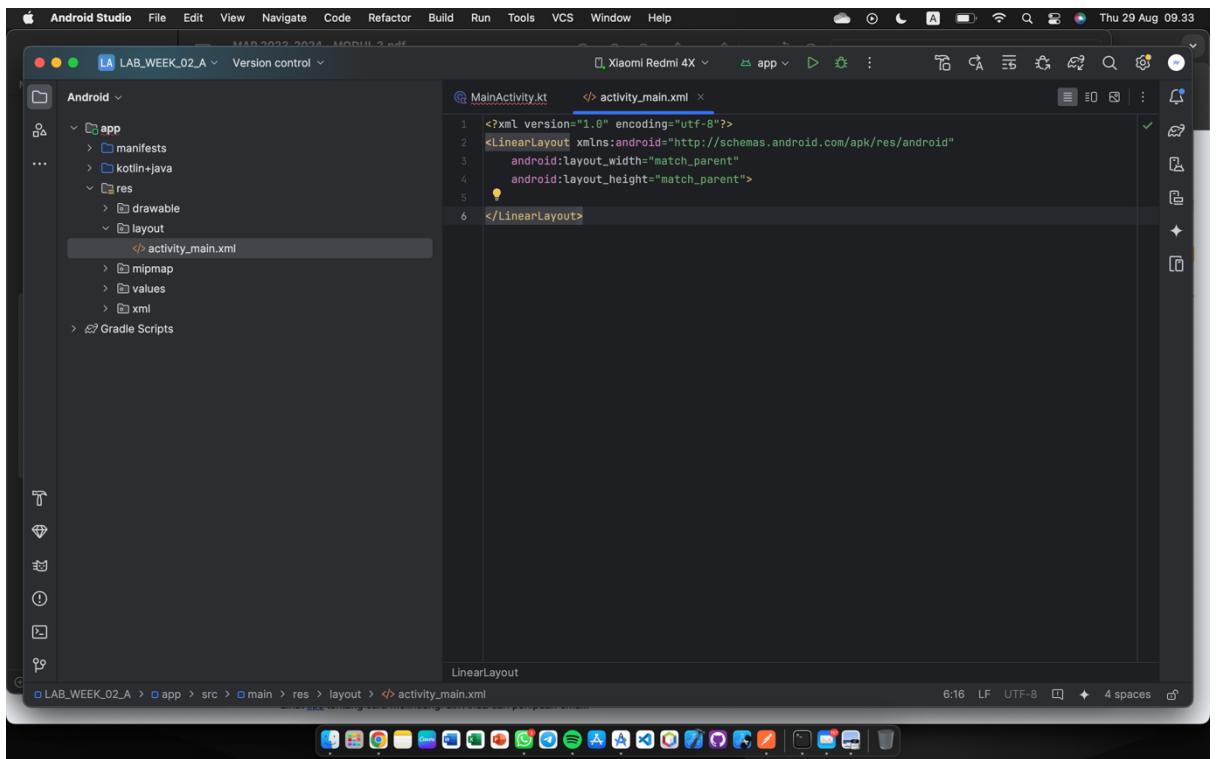
6. Lalu buat layout “activity_main” dengan klik kanan pada folder res dan ikuti gambar berikut



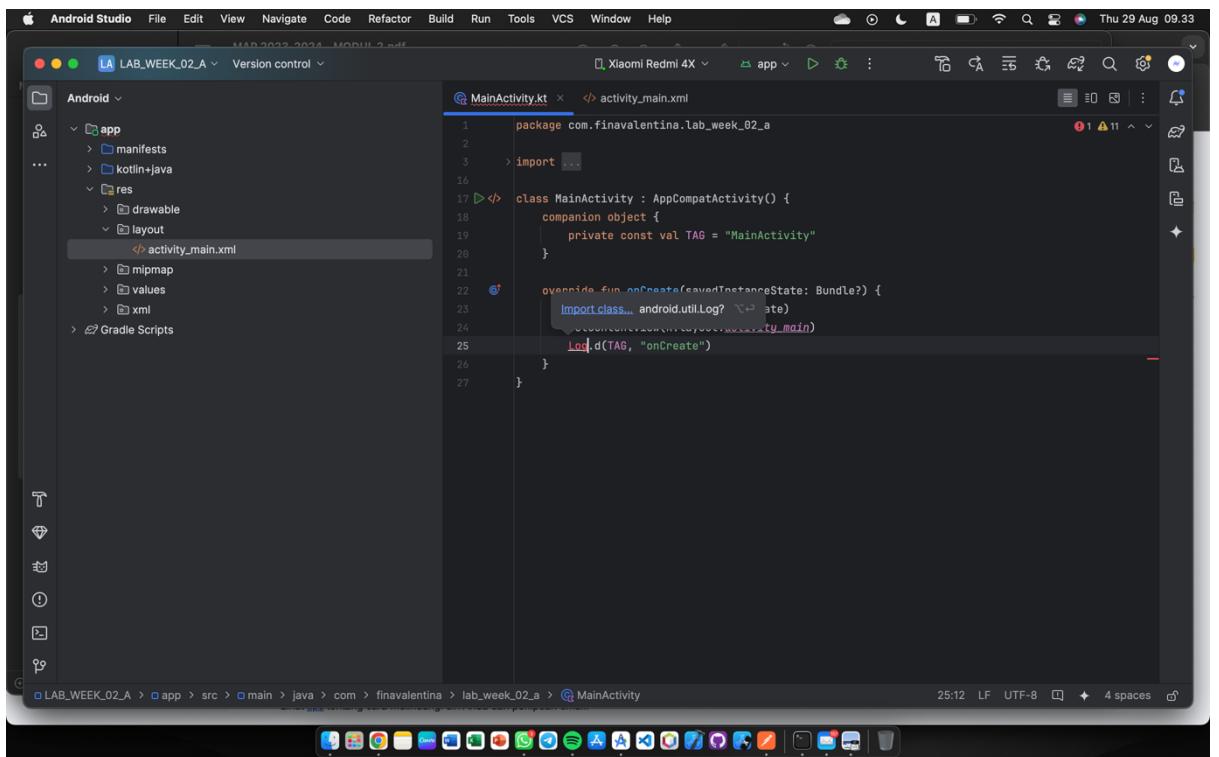
7. Masukan nama “activity_main” pada dialog, lalu klik “Finish”



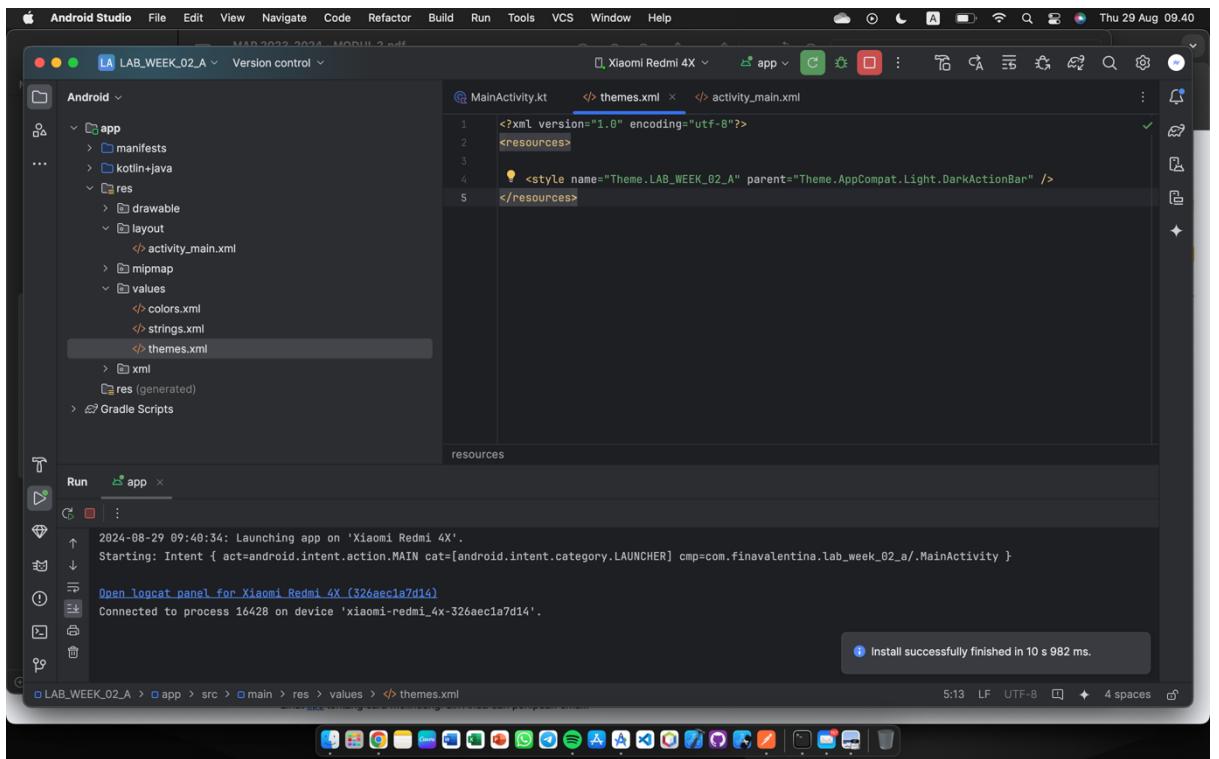
8. Maka akan tampil editor yang menampilkan “activity_main.xml”



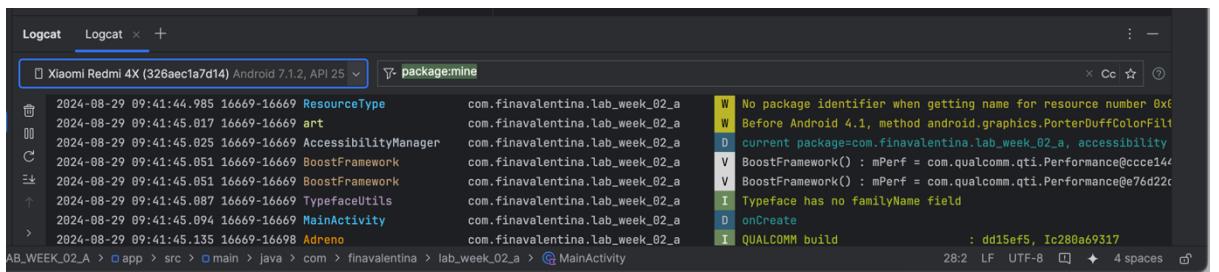
9. Pada file MainActivity, import Log, arahkan cursor mouse ke bagian yang merah lalu klik import class



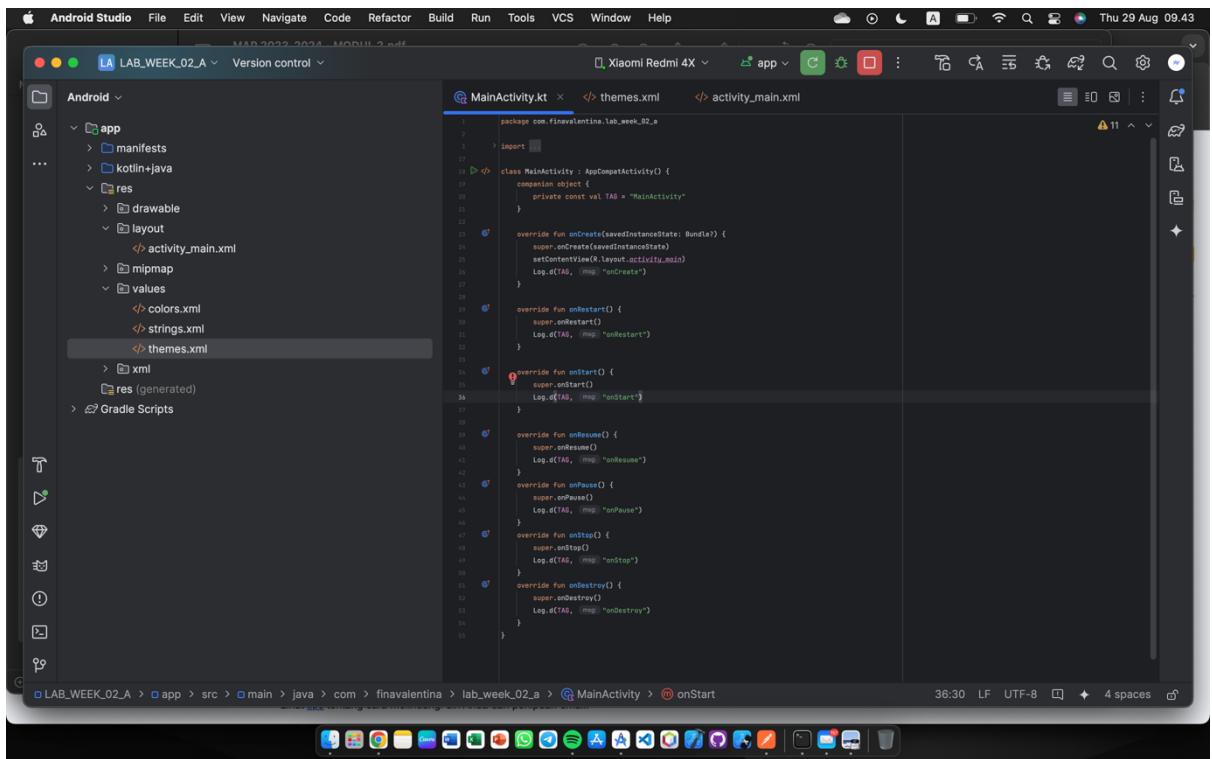
10. Selanjutnya, masuk ke file "themes.xml" ubah isi file tersebut menjadi ini



11. Coba jalankan aplikasinya dan lihat Logcat



12. Lengkapi file MainActivity menjadi seperti ini



```
package com.finavalentina.lab_week_02_a
import ...
class MainActivity : AppCompatActivity() {
    companion object {
        private const val TAB = "MainActivity"
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(layout.activity_main)
        Log.d(TAB, msg = "onCreate")
    }

    override fun onStart() {
        super.onStart()
        Log.d(TAB, msg = "onStart")
    }

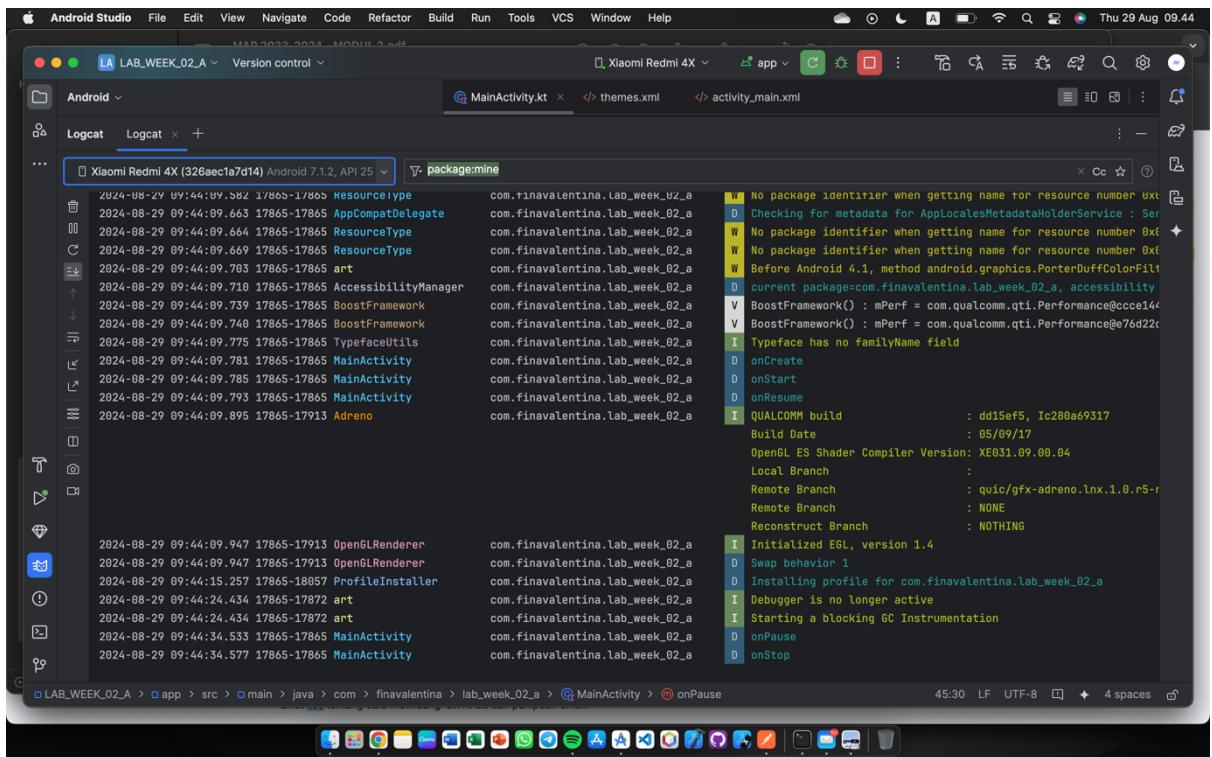
    override fun onResume() {
        super.onResume()
        Log.d(TAB, msg = "onResume")
    }

    override fun onPause() {
        super.onPause()
        Log.d(TAB, msg = "onPause")
    }

    override fun onStop() {
        super.onStop()
        Log.d(TAB, msg = "onStop")
    }

    override fun onDestroy() {
        super.onDestroy()
        Log.d(TAB, msg = "onDestroy")
    }
}
```

13. Lalu jalankan ulang dan lihat Logcat



```
2024-08-29 09:44:09.582 17865-17865 ResourceType com.finavalentina.lab_week_02_a
2024-08-29 09:44:09.663 17865-17865 AppCompatDelegate com.finavalentina.lab_week_02_a
2024-08-29 09:44:09.664 17865-17865 ResourceType com.finavalentina.lab_week_02_a
2024-08-29 09:44:09.669 17865-17865 ResourceType com.finavalentina.lab_week_02_a
2024-08-29 09:44:09.703 17865-17865 art com.finavalentina.lab_week_02_a
2024-08-29 09:44:09.710 17865-17865 AccessibilityManager com.finavalentina.lab_week_02_a
2024-08-29 09:44:09.739 17865-17865 BoostFramework com.finavalentina.lab_week_02_a
2024-08-29 09:44:09.746 17865-17865 BoostFramework com.finavalentina.lab_week_02_a
2024-08-29 09:44:09.775 17865-17865 TypefaceUtils com.finavalentina.lab_week_02_a
2024-08-29 09:44:09.781 17865-17865 MainActivity com.finavalentina.lab_week_02_a
2024-08-29 09:44:09.785 17865-17865 MainActivity com.finavalentina.lab_week_02_a
2024-08-29 09:44:09.793 17865-17865 MainActivity com.finavalentina.lab_week_02_a
2024-08-29 09:44:09.895 17865-17913 Adreno com.finavalentina.lab_week_02_a
2024-08-29 09:44:09.947 17865-17913 OpenGLRenderer com.finavalentina.lab_week_02_a
2024-08-29 09:44:09.947 17865-17913 OpenGLRenderer com.finavalentina.lab_week_02_a
2024-08-29 09:44:15.257 17865-18057 ProfileInstaller com.finavalentina.lab_week_02_a
2024-08-29 09:44:24.434 17865-17872 art com.finavalentina.lab_week_02_a
2024-08-29 09:44:24.434 17865-17872 art com.finavalentina.lab_week_02_a
2024-08-29 09:44:34.533 17865-17865 MainActivity com.finavalentina.lab_week_02_a
2024-08-29 09:44:34.577 17865-17865 MainActivity com.finavalentina.lab_week_02_a

I: No package identifier when getting name for resource number 0x1
W: Checking for metadata for AppLocalesMetadataHolderService : Service
W: No package identifier when getting name for resource number 0x1
W: No package identifier when getting name for resource number 0x1
W: Before Android 4.1, method android.graphics.PorterDuffColorFilter
D: current_package=com.finavalentina.lab_week_02_a, accessibility
V: BoostFramework(): mPerf = com.qualcomm.qti.Performance@ccce14c
V: BoostFramework(): mPerf = com.qualcomm.qti.Performance@76d22c
I: Typeface has no familyName field
D: onCreate
D: onStart
D: onResume
I: QUALCOMM build : dd15ef5, 1c280a69317
Build Date : 05/09/17
OpenGL ES Shader Compiler Version: XE031.09.00.04
Local Branch :
Remote Branch : quic/gfx-adreno.lnx.1.0.r5-r
Remote Branch : NONE
Reconstruct Branch : NOTHING
I: Initialized EGL, version 1.4
D: Swap behavior 1
D: Installing profile for com.finavalentina.lab_week_02_a
I: Debugger is no longer active
I: Starting a blocking GC Instrumentation
D: onPause
D: onStop
```

14. Ubah file AndroidManifest.xml menjadi seperti ini untuk mencegah Pemanggilan berulang ketika layar di ubah rotasinya

```

<manifest>
    <application>
        <activity android:name=".MainActivity" android:exported="true" android:configChanges="orientation|screenSize|screenLayout" android:label="@string/app_name" android:roundIcon="@mipmap/ic_launcher_round" android:supportsRtl="true" android:theme="@style/Theme.LAB_WEEK_02_A" tools:targetApi="31">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

The screenshot shows the Android Studio interface with the code editor open to the `AndroidManifest.xml` file. The manifest defines a single activity named `MainActivity` which is exported and supports orientation changes. It also includes a launcher category. The code editor has syntax highlighting and line numbers. Below the editor is the Logcat tab, which displays log messages from the application's runtime. The status bar at the bottom shows the date and time as `Thu 29 Aug 09.50`.

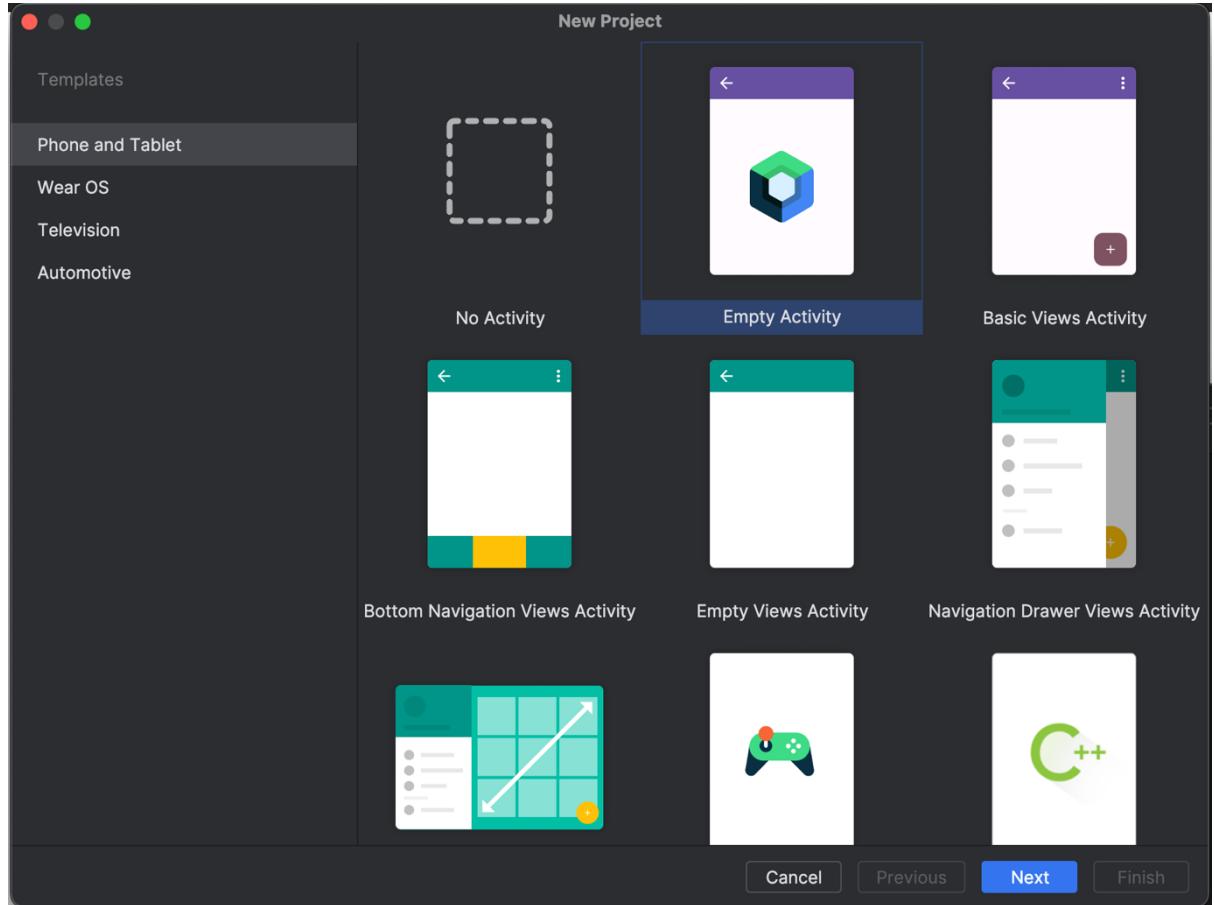
15. Jalankan kembali dan lihat Logcat

The screenshot shows the Android Studio interface with the Logcat tab expanded. The Logcat window displays numerous log entries from the application's runtime. These entries include various framework logs like BoostFramework, TypefaceUtils, and OpenGLRenderer, along with application-specific logs for `MainActivity` and `Adreno`. The Logcat interface includes filters, a search bar, and a timeline at the bottom.

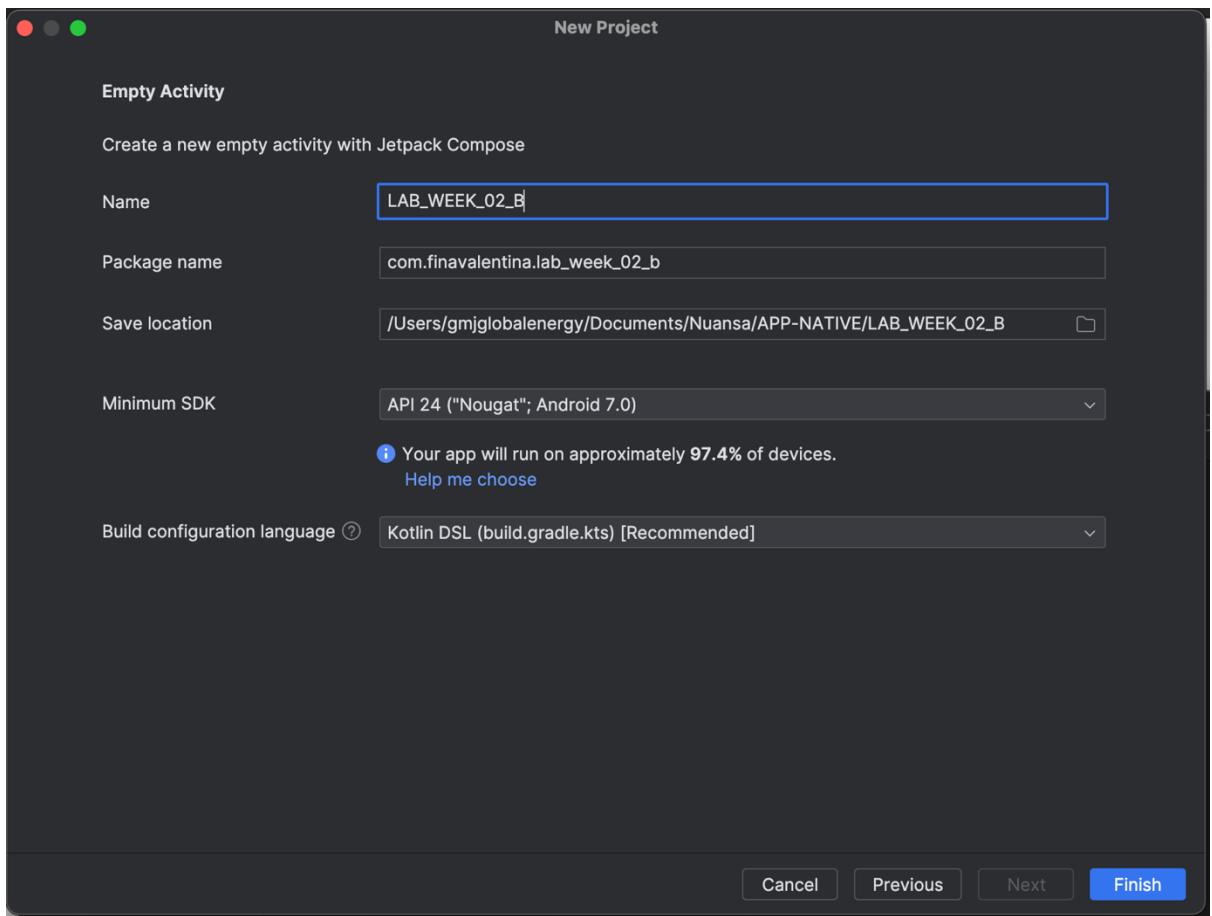
16. Selesai

LANGKAH LANGKAH PENGERJAAN LAB WEEK 02 B

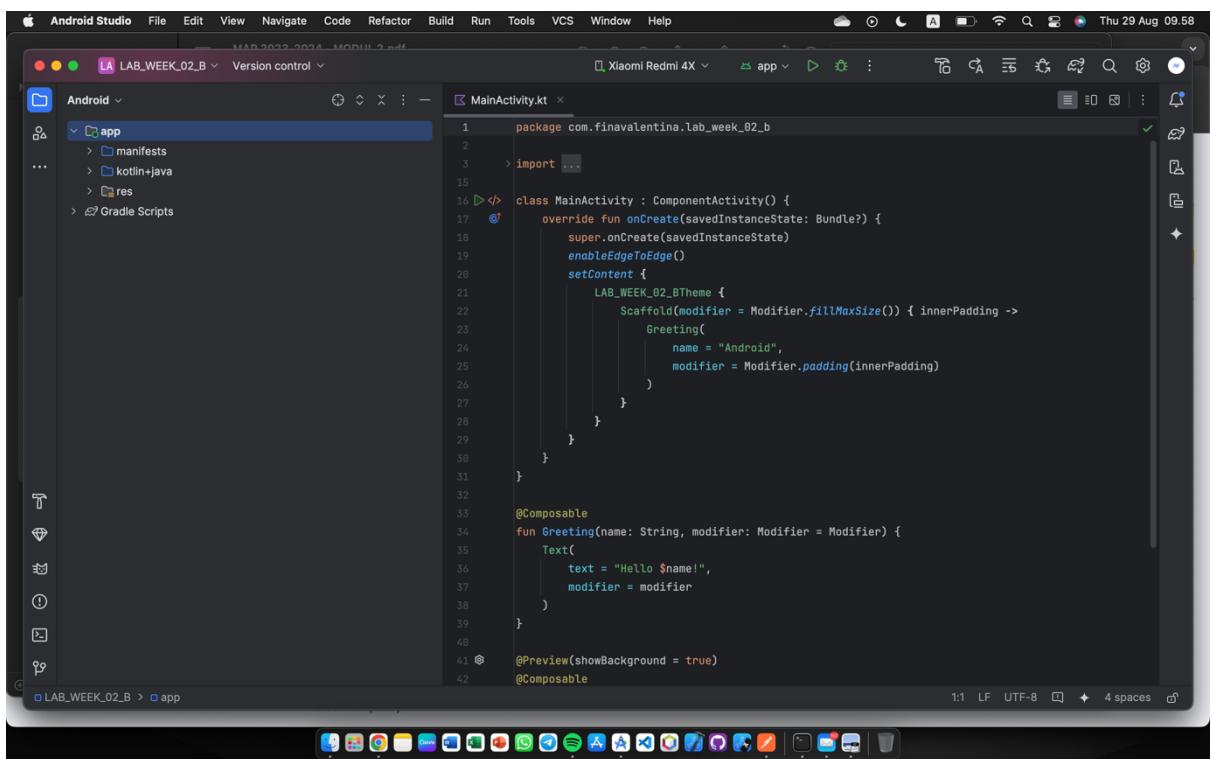
1. Buat project , dan pilih Empty Activity lalu klik “Next”



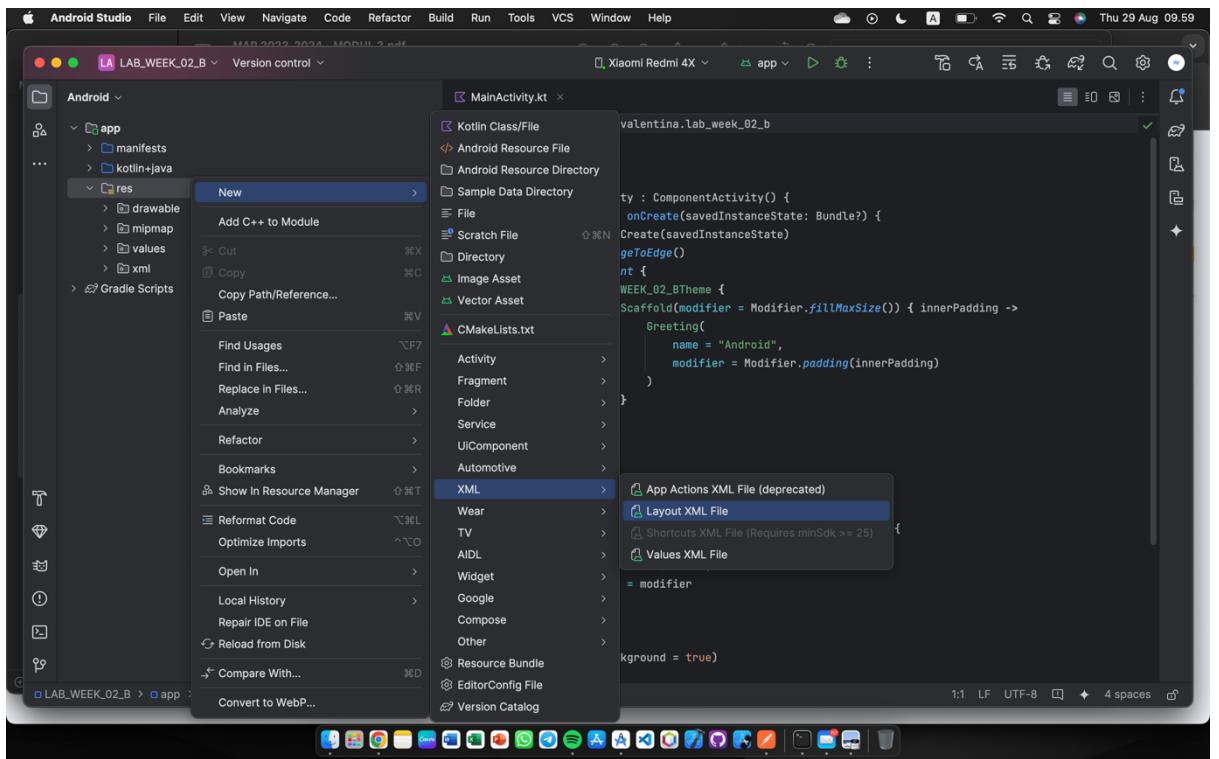
2. Isi nama project dan package name , sdk dll, lalu klik “Finisih”



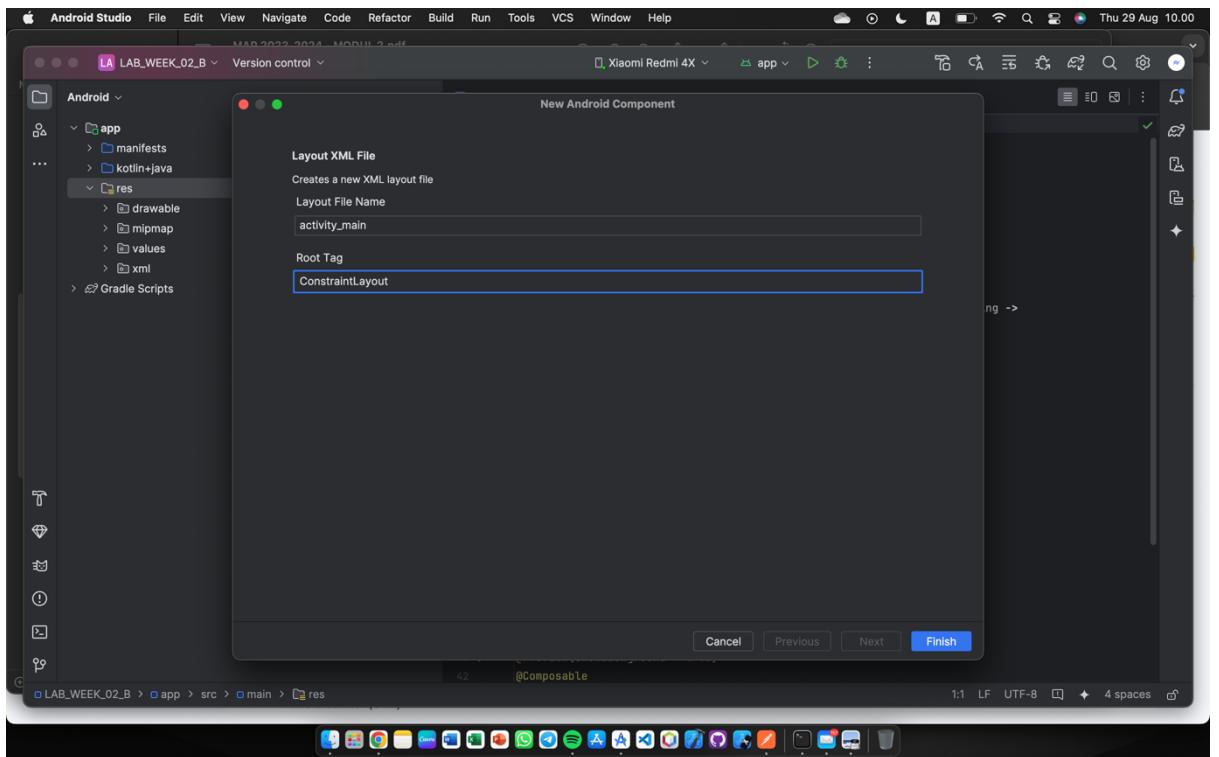
3. Maka akan tampil editor yang menampilkan MainActivity



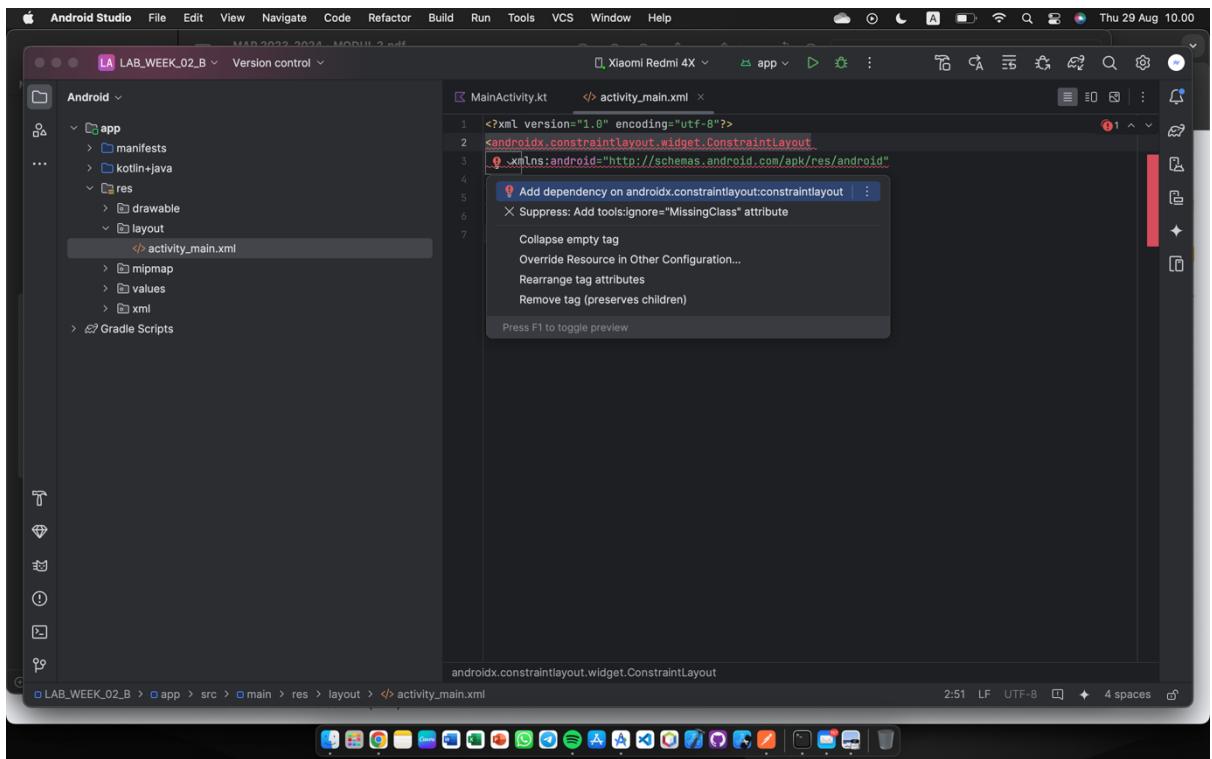
4. Buat layout "activity_main" di folder res,



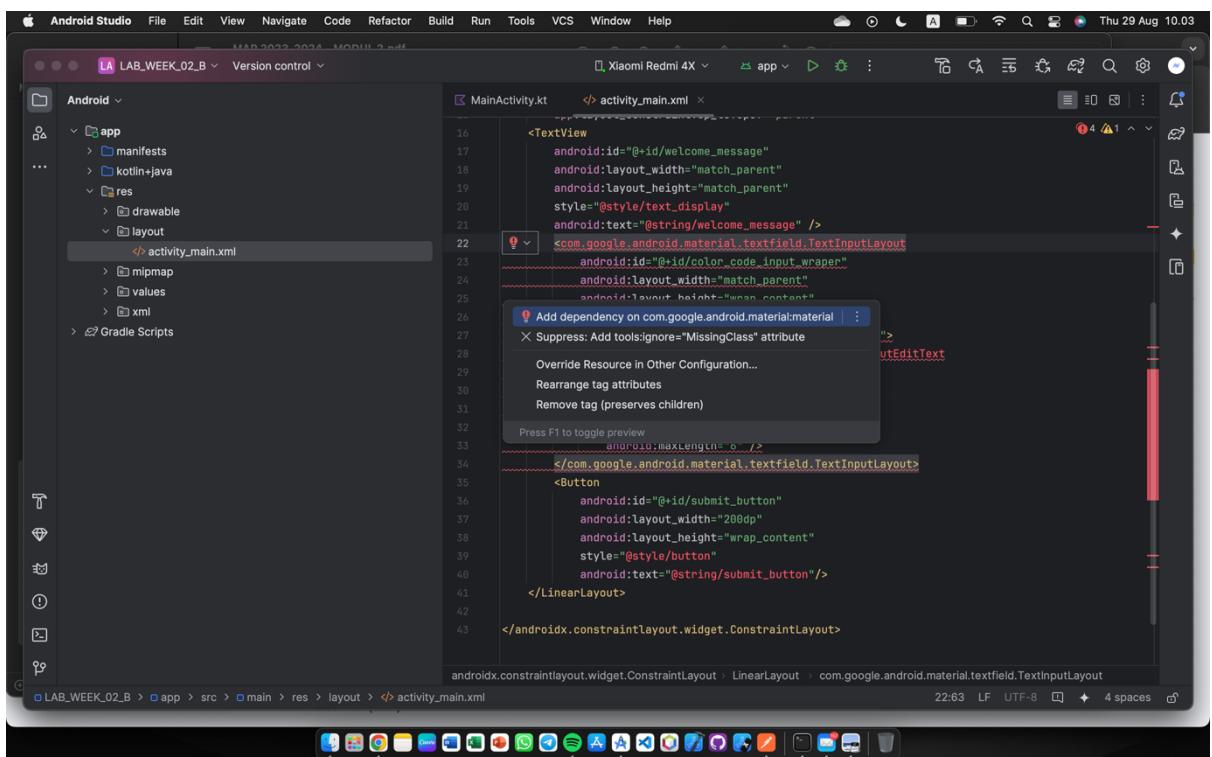
5. Masukan nama file "activity_main" pada dialog, lalu klik "Finish"



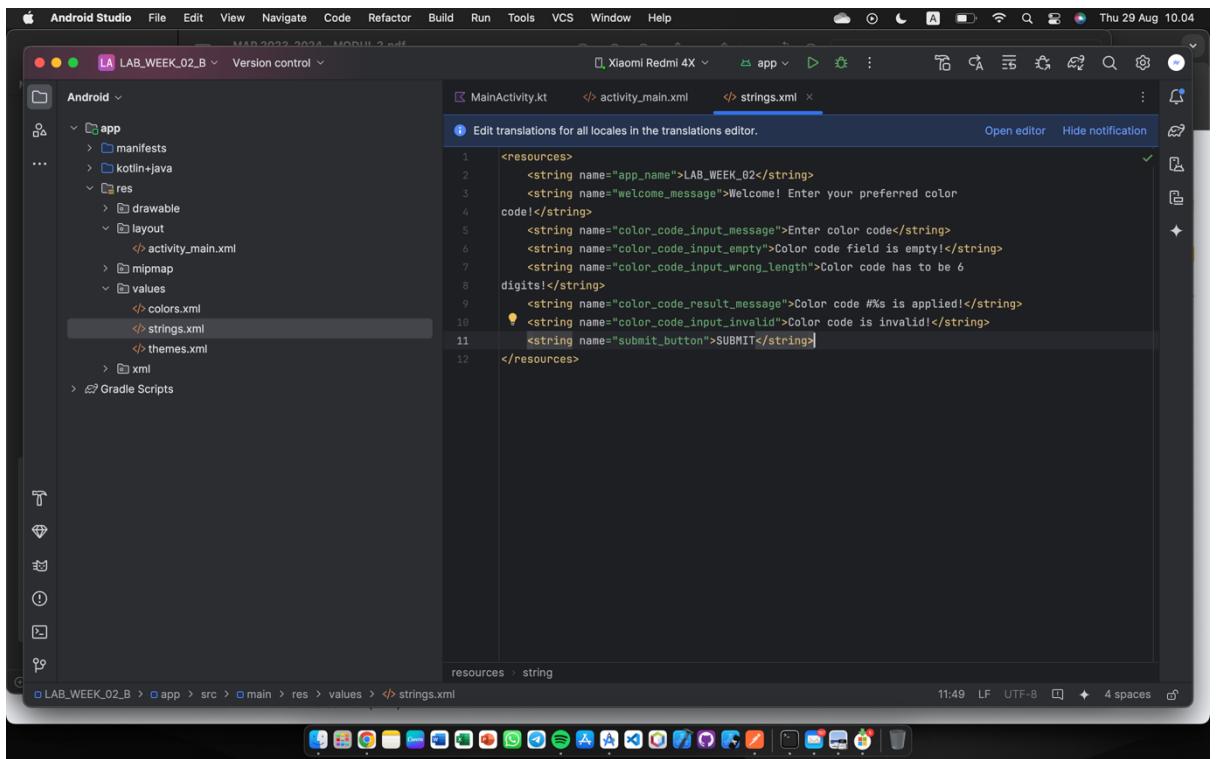
6. Maka akan tampil editor yang menampilkan "activity_main.xml" lalu ubah file itu menjadi seperti ini dulu, lalu klik "Add dependency"



7. Copy paste code yang ada di modul, lalu Add dependency pada baris yang berwarna merah



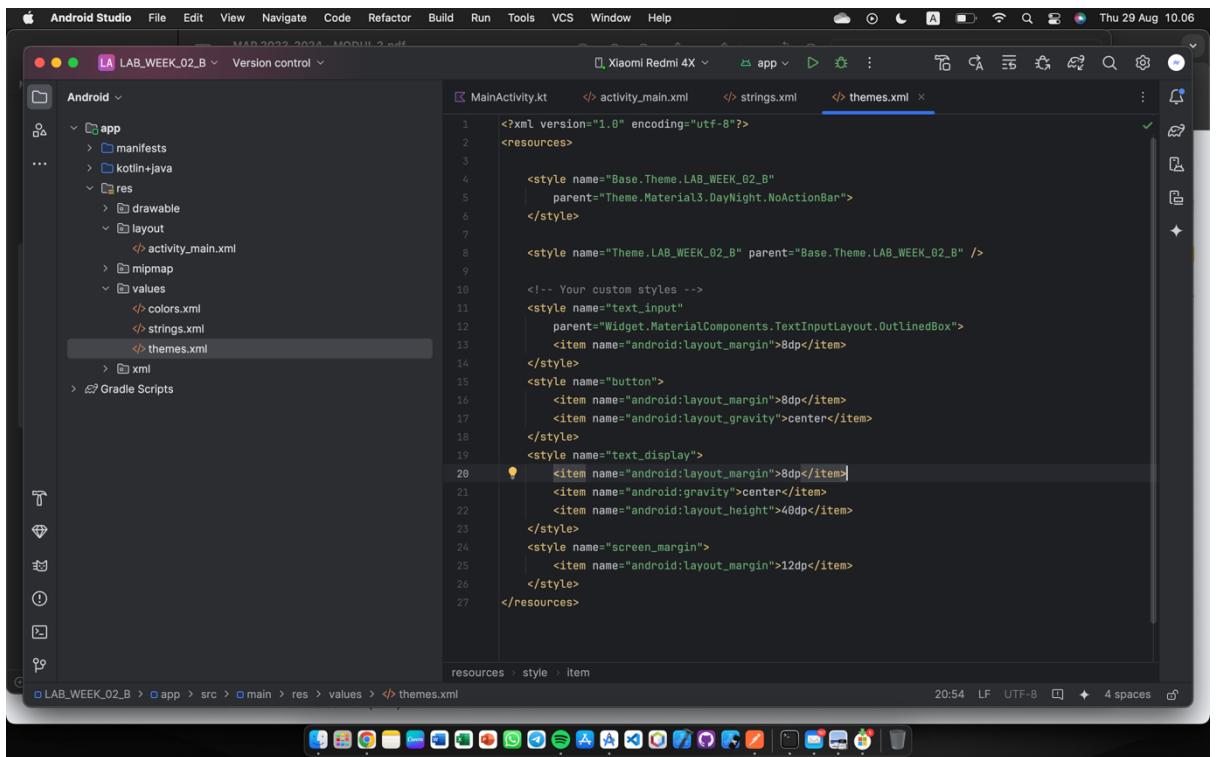
8. Buka file "strings.xml" lalu ubah isi file tersebut menjadi seperti ini



The screenshot shows the Android Studio interface with the project 'LAB_WEEK_02_B' open. The left sidebar shows the project structure under the 'Android' tab, with the 'app' module selected. In the main editor area, the 'strings.xml' file is open. The code in the editor is:

```
<resources>
    <string name="app_name">LAB_WEEK_02</string>
    <string name="welcome_message">Welcome! Enter your preferred color code:</string>
    <string name="color_code_input_message">Enter color code</string>
    <string name="color_code_input_empty">Color code field is empty!</string>
    <string name="color_code_input_wrong_length">Color code has to be 6 digits!</string>
    <string name="color_code_result_message">Color code #%s is applied!</string>
    <string name="color_code_input_invalid">Color code is invalid!</string>
    <string name="submit_button">SUBMIT</string>
</resources>
```

9. Lalu masuk ke file “themes.xml” dan ubah isi file tersebut menjadi seperti ini



The screenshot shows the Android Studio interface with the project 'LAB_WEEK_02_B' open. The left sidebar shows the project structure under the 'Android' tab, with the 'app' module selected. In the main editor area, the 'themes.xml' file is open. The code in the editor is:

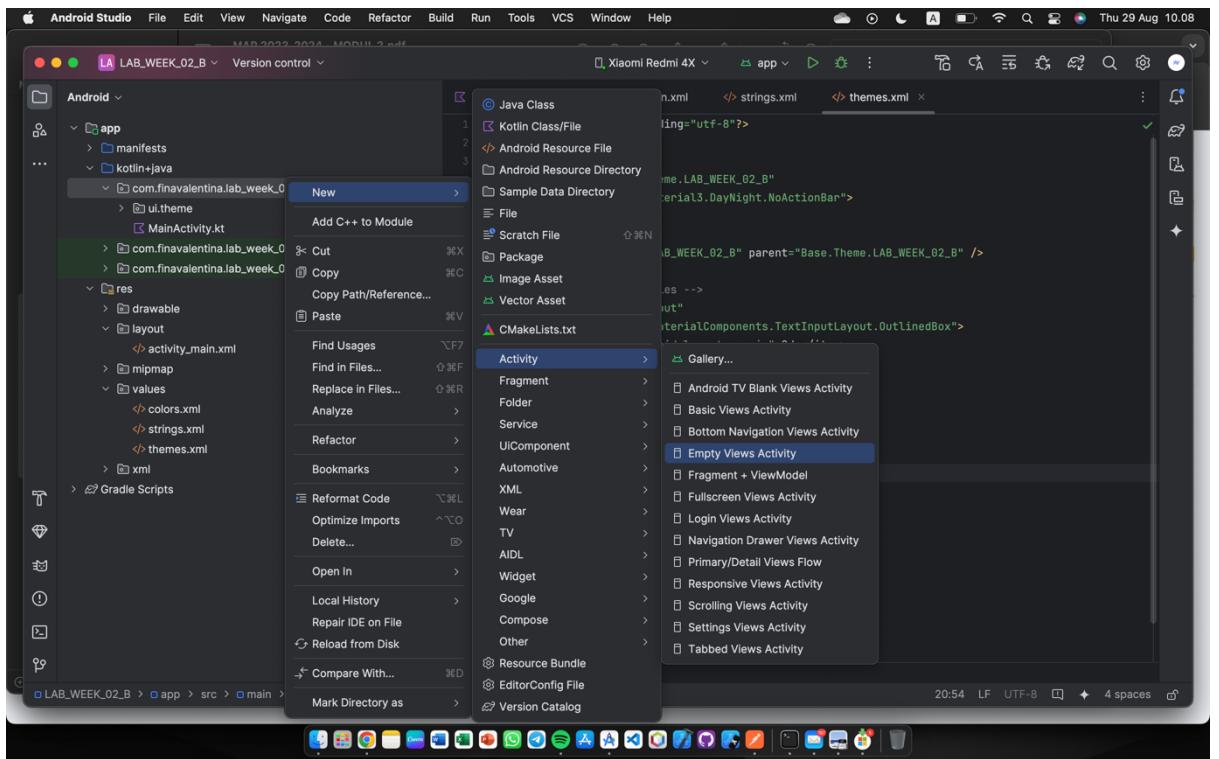
```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <style name="Base.Theme.LAB_WEEK_02_B"
        parent="Theme.Material3.DayNight.NoActionBar">
    </style>

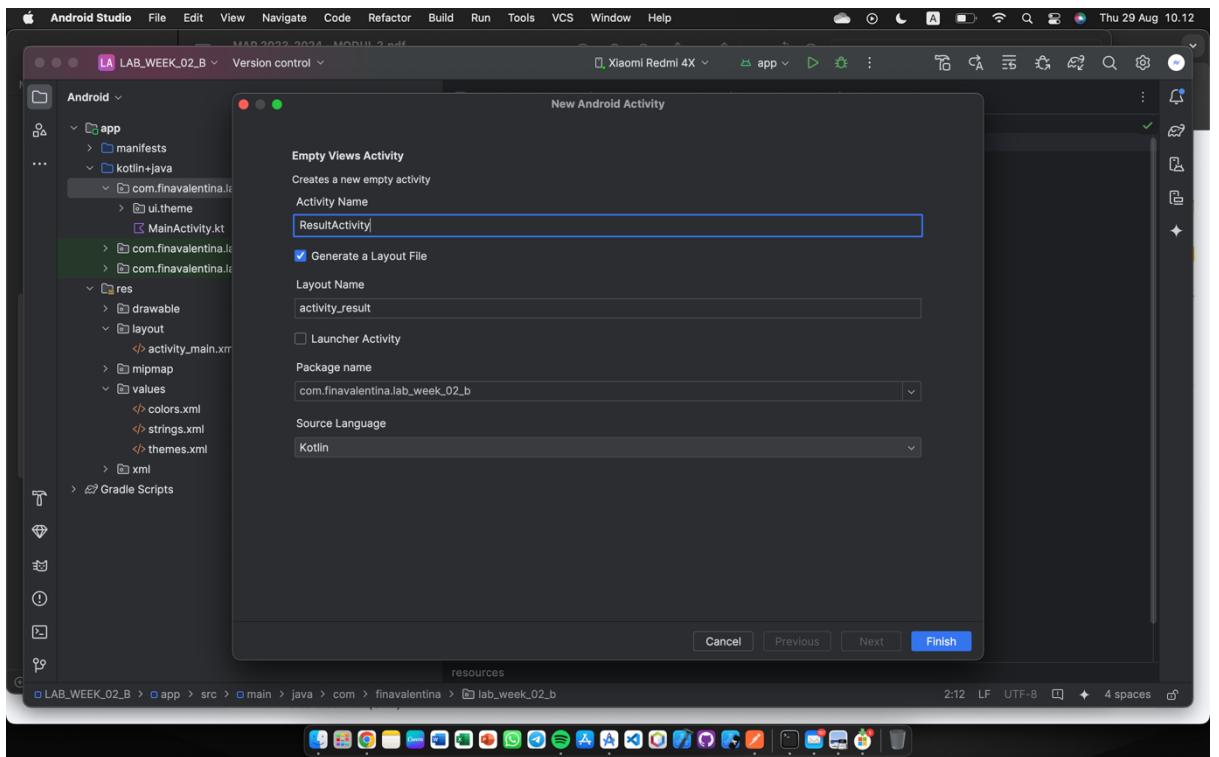
    <style name="Theme.LAB_WEEK_02_B" parent="Base.Theme.LAB_WEEK_02_B" />

    <!-- Your custom styles -->
    <style name="text_input"
        parent="Widget.MaterialComponents.TextInputLayout.OutlinedBox">
        <item name="android:layout_margin">8dp</item>
    </style>
    <style name="button">
        <item name="android:layout_margin">8dp</item>
        <item name="android:layout_gravity">center</item>
    </style>
    <style name="text_display">
        <item name="android:layout_margin">8dp</item>
        <item name="android:gravity">center</item>
        <item name="android:layout_height">40dp</item>
    </style>
    <style name="screen_margin">
        <item name="android:layout_margin">12dp</item>
    </style>
</resources>
```

10. Selanjutnya buat Activity baru di folder kotlin+java, dan pilih com. **packagename** yang pertama klik kana pada folder itu



11. Masukan "ResultActivity" dan layout name "activity_result", lalu klik "Finish"



12. Ubah file "activity_result" menjadi seperti ini

The screenshot shows the Android Studio interface with the project 'LAB_WEEK_02_B' open. The left sidebar displays the project structure under the 'Android' tab, including the app module with its sub-directories like manifests, kotlin+java, and res. In the main editor area, the file 'activity_result.xml' is open, showing XML code for a ConstraintLayout. The code includes a TextView with id '@+id/color_code_result_message' and text color set to white. The status bar at the bottom indicates the file is 18:53 LF UTF-8 with 4 spaces.

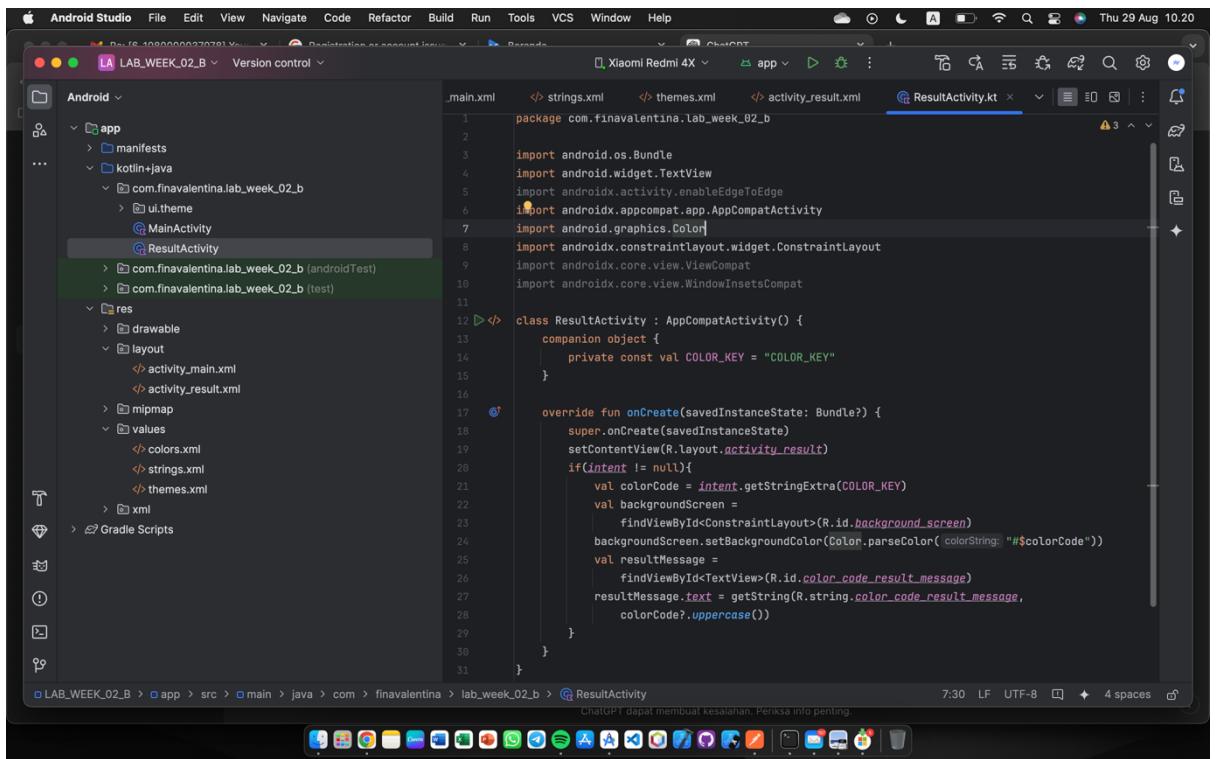
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ResultActivity"
    android:id="@+id/background_screen">
    <TextView
        android:id="@+id/color_code_result_message"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        style="@style/text_display"
        android:textColor="@color/white"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

13. Buka file MainActivity dan Copy Paste code dari Modul, dan lakukan import satu persatu di baris yang ditandai merah

The screenshot shows the Android Studio interface with the project 'LAB_WEEK_02_B' open. The left sidebar displays the project structure under the 'Android' tab, including the app module with its sub-directories like manifests, kotlin+java, and res. In the main editor area, the file 'MainActivity.kt' is open, showing the implementation of the MainActivity class. The code handles button click events and uses Toast to display messages. The status bar at the bottom indicates the file is 51:29 LF UTF-8 with 4 spaces.

```
package com.finavalentina.lab_week_02_b
import ...
class MainActivity : AppCompatActivity() {
    companion object {
        private const val COLOR_KEY = "COLOR_KEY"
    }
    private val submitButton: Button
        get() = findViewById(R.id.submit_button)
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        submitButton.setOnClickListener { view: View? ->
            val colorCode =
                findViewById(R.id.color_code_input_field).text.toString()
            if (colorCode.isNotEmpty()) {
                if (colorCode.length < 6) {
                    Toast
                        .makeText(
                            context,
                            getString(R.string.color_code_input_wrong_length),
                            Toast.LENGTH_LONG
                        )
                        .show()
                } else {
                    val ResultIntent = Intent(packageContext, ResultActivity::class.java)
                    ResultIntent.putExtra(COLOR_KEY, colorCode)
                    startActivity(ResultIntent)
                }
            } else {
                Toast
                    .makeText(
                        context,
                        getString(R.string.color_code_input_empty),
                        Toast.LENGTH_LONG
                    ).show()
            }
        }
    }
}
```

14. Buka file ResultActivity dan ubah isi nya menjadi seperti ini dan lakukan import satu persatu jika terjadi error



The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `ResultActivity.kt` file. The code is as follows:

```
1 package com.finavaleNTINA.lab_week_02_b
2
3 import android.os.Bundle
4 import android.widget.TextView
5 import androidx.activity.enableEdgeToEdge
6 import androidx.appcompat.app.AppCompatActivity
7 import android.graphics.Color
8 import androidx.constraintlayout.widget.ConstraintLayout
9 import android.core.view.ViewCompat
10 import android.core.view.WindowInsetsCompat
11
12 class ResultActivity : AppCompatActivity() {
13     companion object {
14         private const val COLOR_KEY = "COLOR_KEY"
15     }
16
17     override fun onCreate(savedInstanceState: Bundle?) {
18         super.onCreate(savedInstanceState)
19         setContentView(R.layout.activity_result)
20         if(intent != null){
21             val colorCode = intent.getStringExtra(COLOR_KEY)
22             val backgroundScreen =
23                 findViewById<ConstraintLayout>(R.id.background_screen)
24             backgroundScreen.setBackgroundColor(Color.parseColor("#$colorCode"))
25             val resultMessage =
26                 findViewById<TextView>(R.id.color_code_result_message)
27             resultMessage.text = getString(R.string.color_code_result_message,
28                                           colorCode?.uppercase())
29         }
30     }
31 }
```

15. Lalu, coba jalankan aplikasi dan tampilan aplikasinya seperti ini

10:25

0,1KB/d

Welcome! Enter your preferred color code!

Enter color code

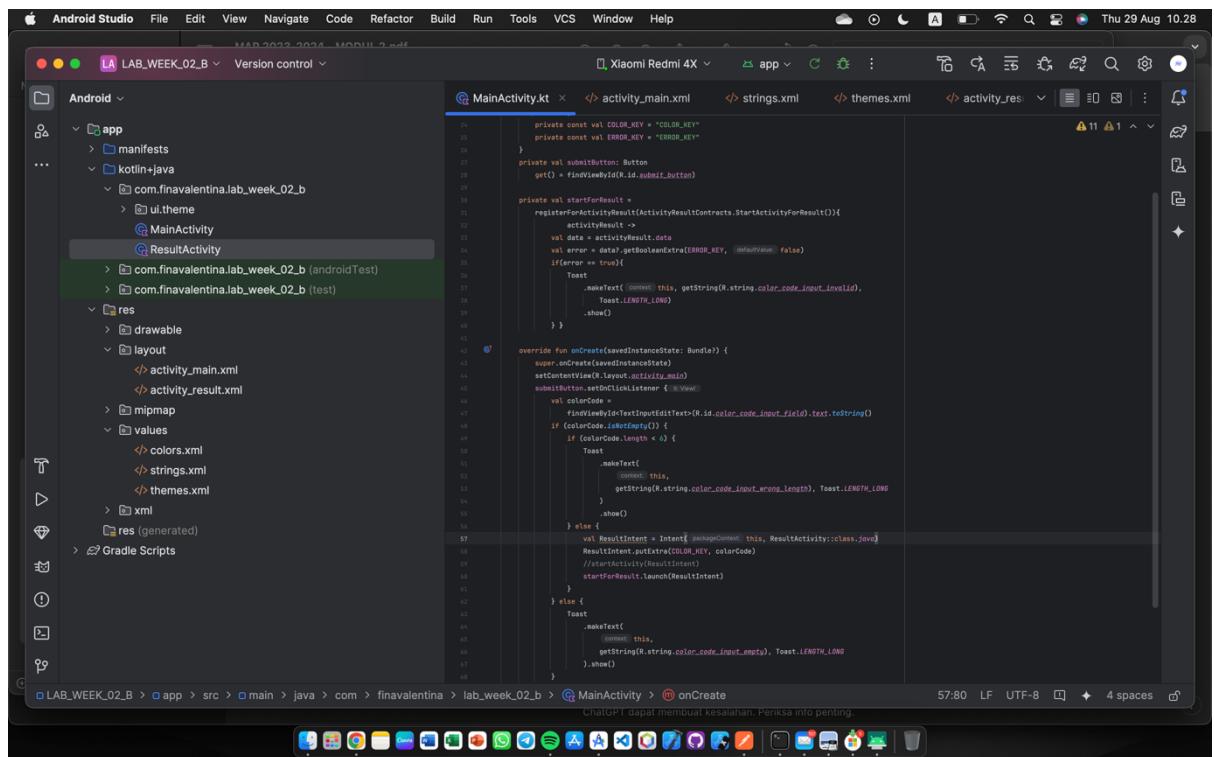
SUBMIT

10:25

0,2KB/d

Color code #213718 is applied!

16. Perbaiki file MainActivity untuk mencegah error ketika kembali dari halaman result



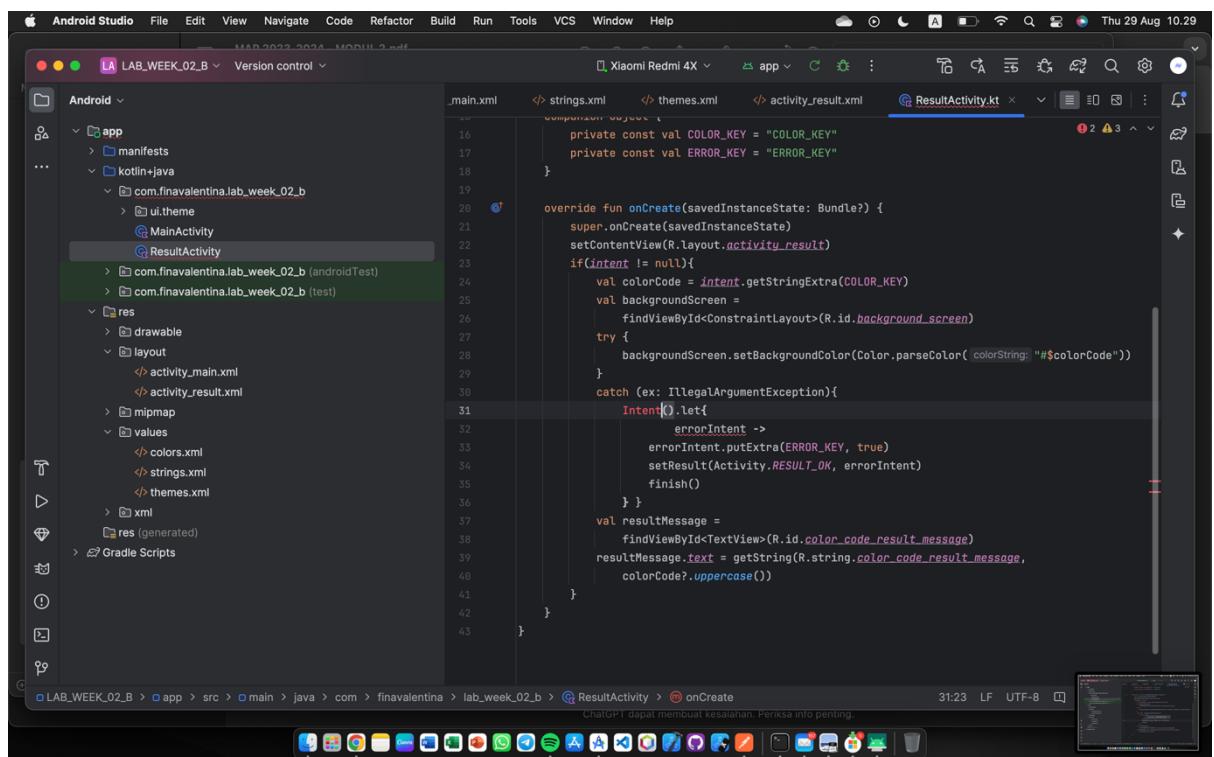
The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `MainActivity.kt` file. The code handles the `onCreate` method, which checks if the `COLOR_KEY` extra is present. If it is, it retrieves the color code and displays a toast message. If it's not present or is empty, it shows an error toast. The code also registers for activity results and starts the `ResultActivity`.

```
private const val COLOR_KEY = "COLOR_KEY"
private const val ERROR_KEY = "ERROR_KEY"

private val submitButton: Button
    get() = findViewById(R.id.submit_button)

private val startForResult =
    registerForActivityResult(ActivityResultContracts.StartActivityForResult()) {
        activityResult ->
        val data = activityResult.data
        val error = data?.getBooleanExtra(ERROR_KEY, false)
        if(error == true) {
            Toast.makeText(context, getString(R.string.color_code_input_invalid), Toast.LENGTH_LONG).show()
        } else {
            override fun onCreate(savedInstanceState: Bundle?) {
                super.onCreate(savedInstanceState)
                setContentView(R.layout.activity_main)
                submitButton.setOnClickListener {
                    val colorCode = findViewById<EditText>(R.id.color_code_input_field).text.toString()
                    if(colorCode.isNotEmpty()) {
                        if(colorCode.length < 3) {
                            Toast.makeText(
                                context, getString(R.string.color_code_input_wrong_length), Toast.LENGTH_LONG).show()
                        } else {
                            val resultIntent = Intent(packageName, this, ResultActivity::class.java)
                            resultIntent.putExtra(COLOR_KEY, colorCode)
                            startActivityForResult(resultIntent)
                            startForResult.launch(resultIntent)
                        }
                    } else {
                        Toast.makeText(
                            context, getString(R.string.color_code_input_empty), Toast.LENGTH_LONG).show()
                    }
                }
            }
        }
    }
```

17. Lakukan perubahan juga pada file ResultActivity



The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `ResultActivity.kt` file. It overrides the `onCreate` method to handle the result from the previous activity. It checks if the `COLOR_KEY` extra is present. If it is, it retrieves the color code, creates a toast message with the uppercase version of the color code, and sets the background color of the screen. If the extra is not present, it shows an error toast.

```
private const val COLOR_KEY = "COLOR_KEY"
private const val ERROR_KEY = "ERROR_KEY"

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_result)
    if(intent != null){
        val colorCode = intent.getStringExtra(COLOR_KEY)
        val backgroundScreen =
            findViewById<ConstraintLayout>(R.id.background_screen)
        try {
            backgroundScreen.setBackgroundColor(Color.parseColor("#$colorCode"))
        } catch (ex: IllegalArgumentException){
            Intent().let{
                errorIntent ->
                errorIntent.putExtra(ERROR_KEY, true)
                setResult(Activity.RESULT_OK, errorIntent)
                finish()
            }
        }
        val resultMessage =
            findViewById<TextView>(R.id.color_code_result_message)
        resultMessage.text = getString(R.string.color_code_result_message,
            colorCode?.uppercase())
    }
}
```

18. TUGAS: Buka file “strings.xml” lalu sesuaikan code nya menjadi seperti ini

```
<resources>
    <string name="app_name">LAB_WEEK_02</string>
    <string name="welcome_message">Welcome! Enter your preferred color code!</string>
    <string name="color_code_input_message">Enter color code</string>
    <string name="color_code_input_empty">Color code field is empty!</string>
    <string name="color_code_input_wrong_length">Color code has to be 6 digits!</string>
    <string name="color_code_result_message">Color code %s is applied!</string>
    <string name="color_code_input_invalid">Color code is invalid!</string>
    <string name="submit_button">SUBMIT</string>
    <string name="back_button">BACK</string>
</resources>
```

19. TUGAS: Lalu sesuaikan juga file “activity_result.xml” menjadi seperti ini

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ResultActivity"
    android:id="@+id/background_screen">

    <TextView
        android:id="@+id/color_code_result_message"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        style="@style/text_display"
        android:textColor="@color/white"/>

    <Button
        android:id="@+id/back_button"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        style="@style/button"
        android:text="String/back_button"
        app:layout_constraintTop_toBottomOf="@+id/color_code_result_message"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        tools:ignore="MissingConstraints" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

20. TUGAS: Lalu sesuaikan juga file “ResultActivity” menjadi seperti ini

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `ResultActivity.kt` file, which is a Kotlin Activity. The code handles the creation of the activity from an intent, retrieves a color code from the intent, and sets it as the background color of the screen. It also handles errors by putting them into the intent and setting the result message. The back button's click listener is set to finish the activity.

```
import android.graphics.Color
import android.widget.Button
import androidx.constraintlayout.widget.ConstraintLayout
import android.core.view.ViewCompat
import android.core.view.WindowInsetsCompat

class ResultActivity : AppCompatActivity() {
    companion object {
        private const val COLOR_KEY = "COLOR_KEY"
        private const val ERROR_KEY = "ERROR_KEY"
    }

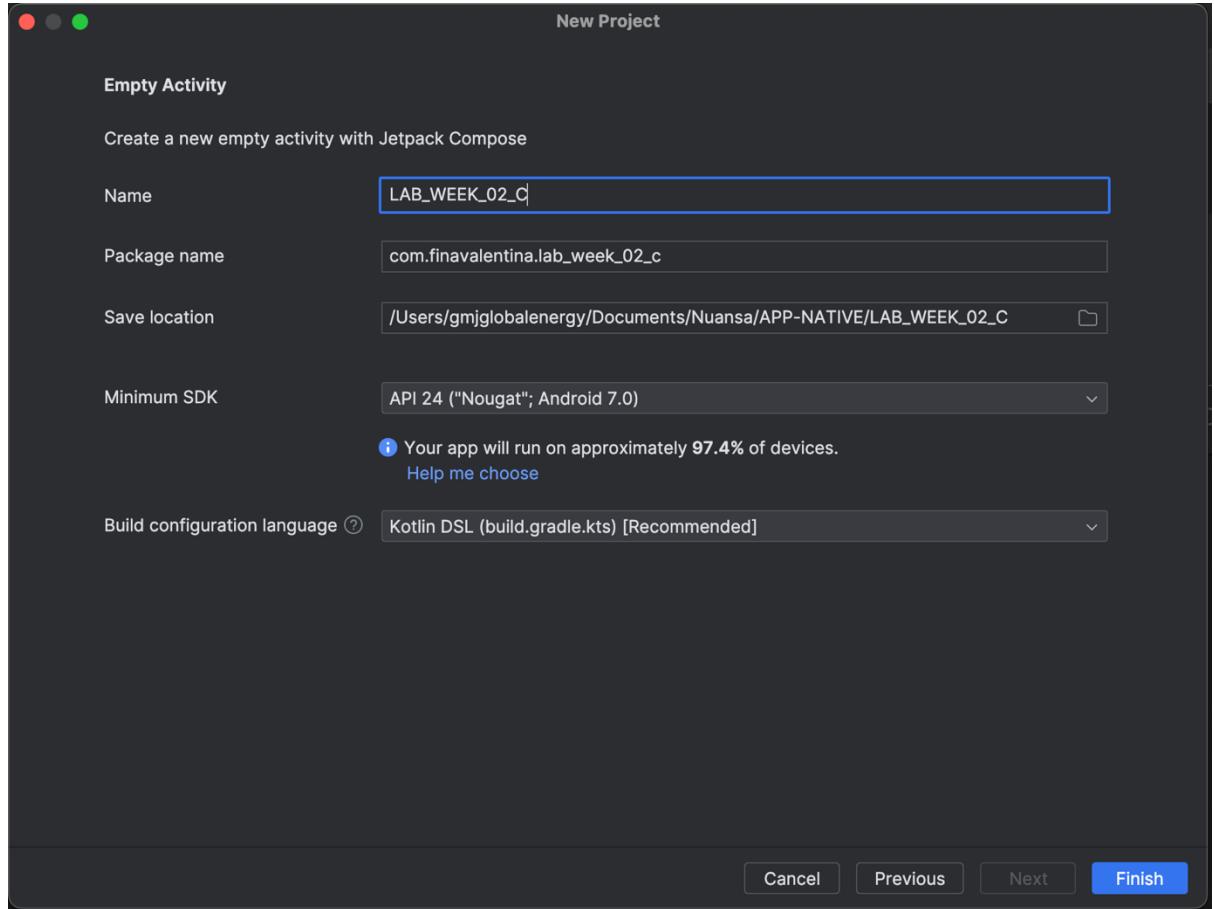
    private val backButton: Button
        get() = findViewById(R.id.back_button)

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_result)
        if(intent != null) {
            val colorCode = intent.getStringExtra(COLOR_KEY)
            val backGroundScreen = findViewById(R.id.background_screen)
            try {
                backGroundScreen.setBackgroundColor(Color.parseColor(colorCode))
            } catch (e: IllegalArgumentException) {
                e.printStackTrace()
                Intent().also {
                    errorIntent.putExtra(ERROR_KEY, true)
                    setResult(Activity.RESULT_CANCELED, errorIntent)
                    finish()
                }
            }
            val resultMessage = findViewById(R.id.color_code_result_message)
            resultMessage.text = getString(R.string.color_code_result_message,
                colorCode.toUpperCase())
        }
        backButton.setOnClickListener {
            finish()
        }
    }
}
```

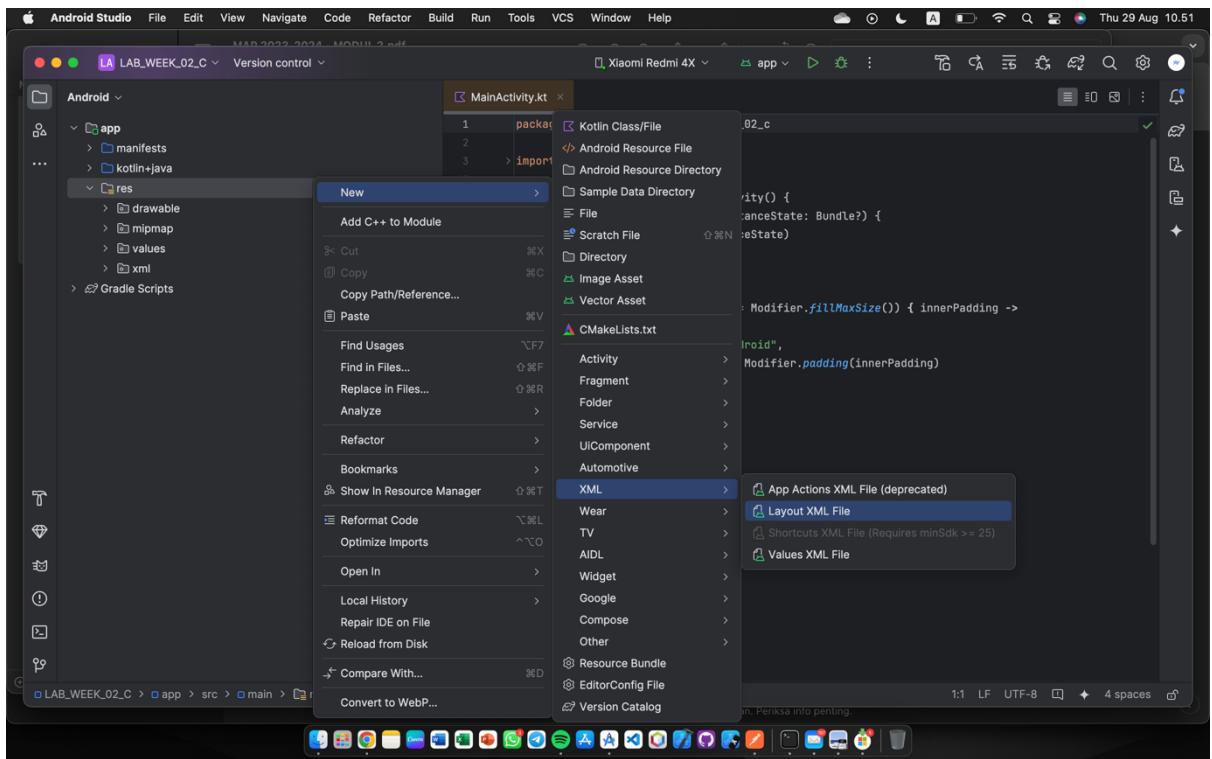
21. Selesai

LANGKAH LANGKAH PENGERJAAN LAB WEEK 02 C

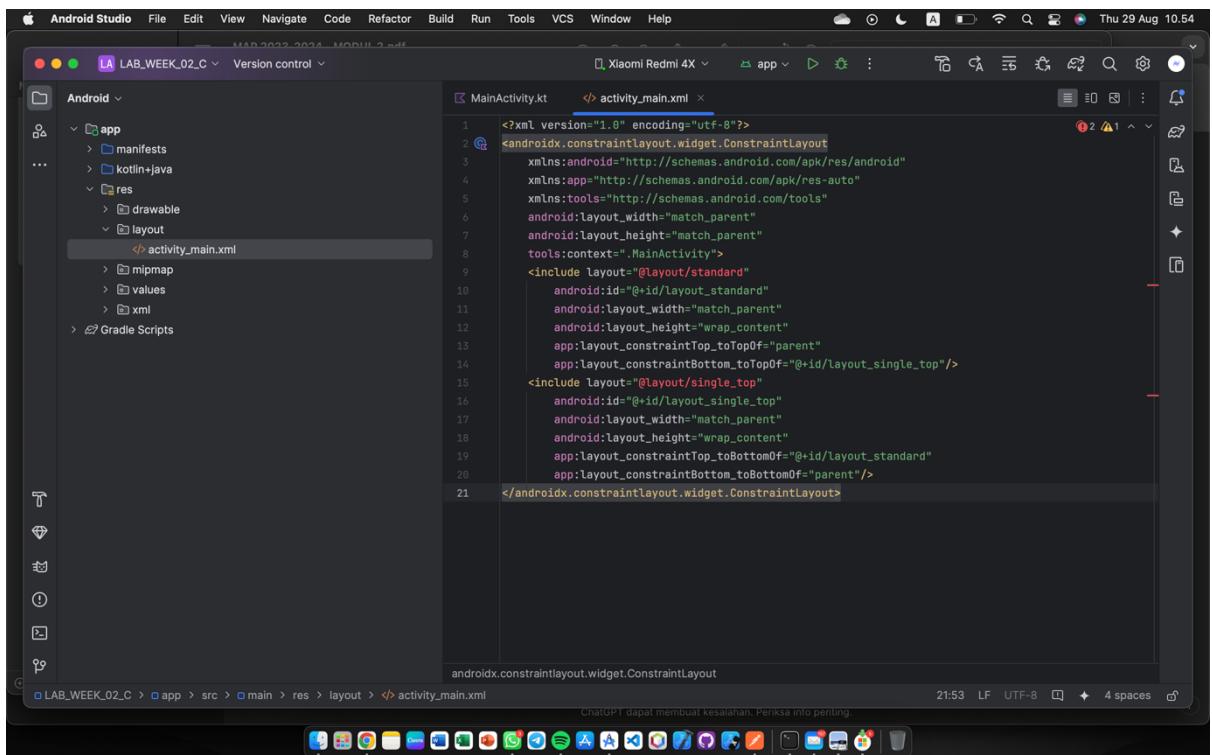
1. Buat project dan pilih “Empty Activity” lalu klik Next



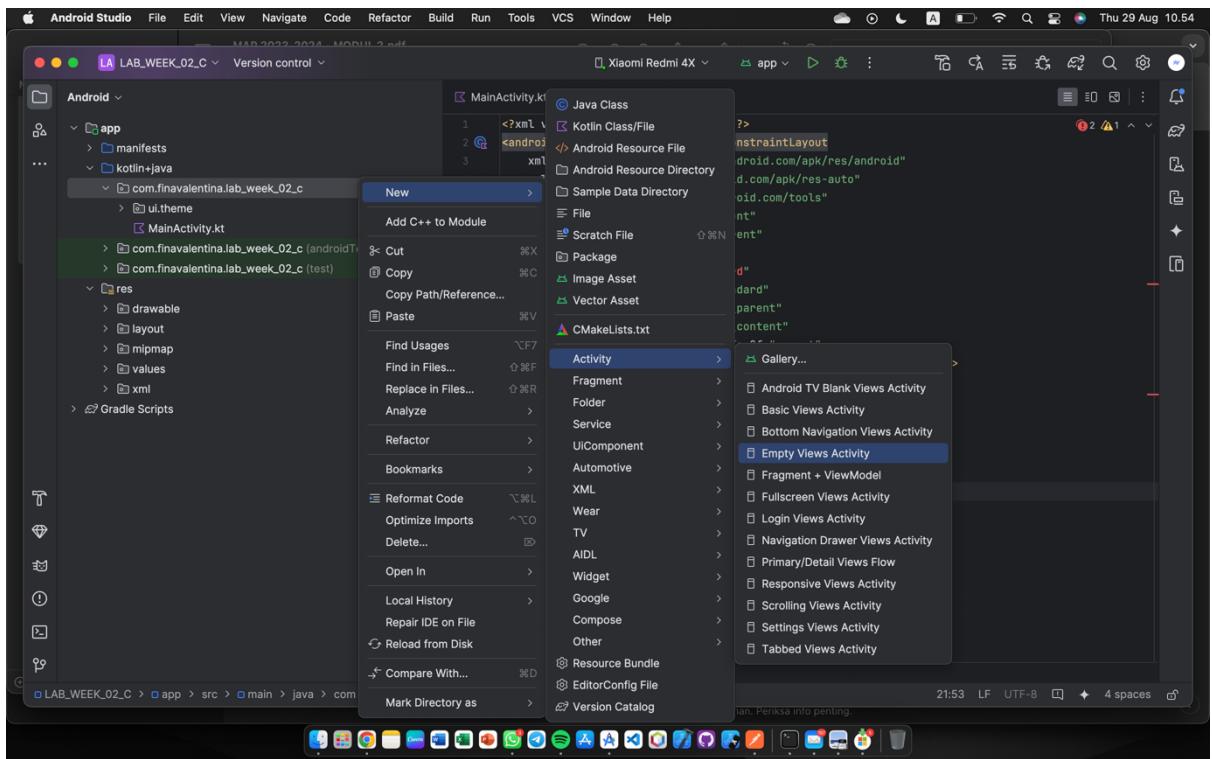
2. Buat layout “activity_main” di folder res



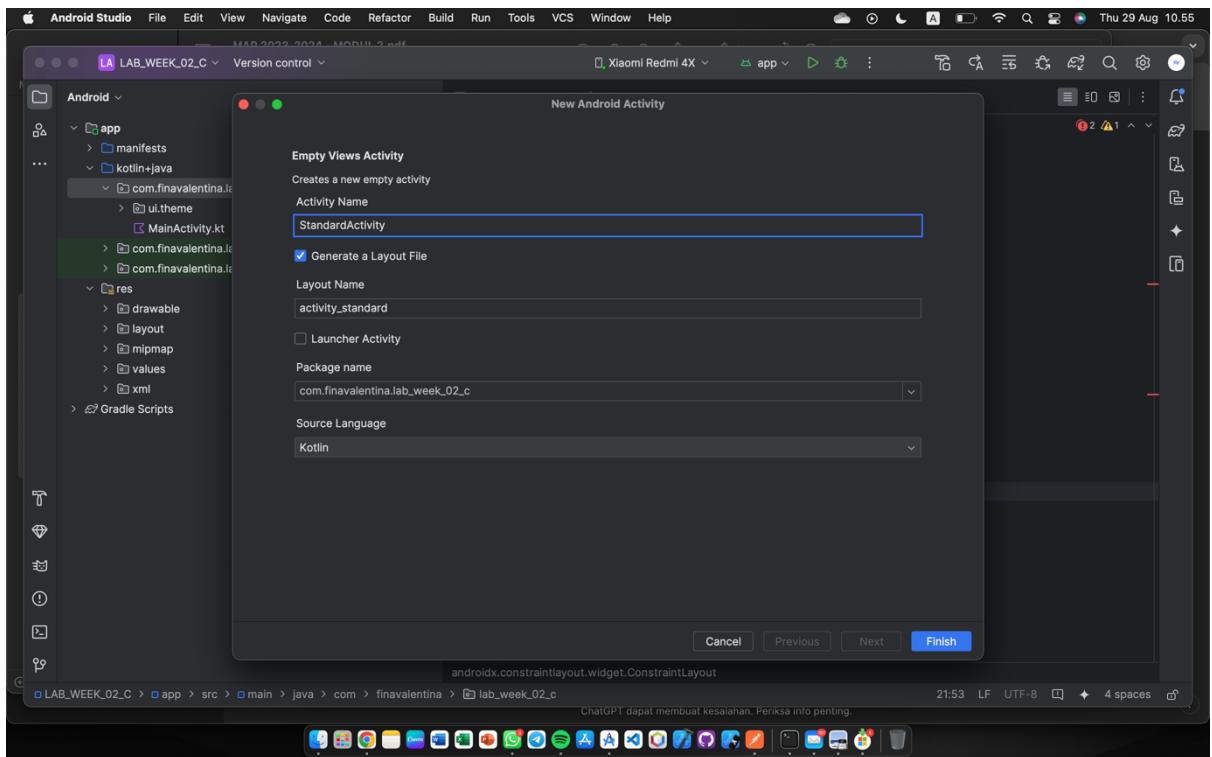
3. Sesuaikan activity_main.xml menjadi seperti ini, untuk pertama kali Add dependency terlebih dahulu pada <androidx.constraintLayout....>



4. Buat Activity pada folder "kotlin+java" dan pilih folder pertama com. **packagename** lalu klik kanan pada folder tersebut dan pilih Empty Activity



5. Masukan nama StandartActivity dan nama layout activity_standard lalu klik "Finish"



6. Ulangi langkah pembuatan Activity, dan beri nama Activity berikutnya SingleTopActivity dan layout activity_single_top

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `SingleTopActivity.kt` file, which contains the following Kotlin code:

```
1 package com.finavalentina.lab_week_02_c
2
3 import ...
4
5 class SingleTopActivity : AppCompatActivity() {
6     override fun onCreate(savedInstanceState: Bundle?) {
7         super.onCreate(savedInstanceState)
8         enableEdgeToEdge()
9         setContentView(R.layout.activity_single_top)
10        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
11            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
12            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
13        }
14    }
15
16
17
18
19
20 }
```

7. Ubah file AndroidManifest.xml menjadi seperti ini

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `AndroidManifest.xml` file, which contains the following XML code:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.LAB_WEEK_02_C"
        tools:targetRipple="3dp">
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.LAB_WEEK_02_C"
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        <activity
            android:name=".StandardActivity"
            android:launchMode="standard">
        </activity>
        <activity
            android:name=".SingleTopActivity"
            android:launchMode="singleTop">
        </activity>
    </application>
</manifest>
```

8. Ubah file activity_standart.xml menjadi seperti ini

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the XML file `activity_standard.xml`. The XML code defines a `ConstraintLayout` with a single button. The button has an ID of `@+id/button_standard`, a width of `200dp`, and a height of `wrap_content`. It is positioned with constraints relative to its parent layout.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button_standard"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:text="@string/standard_button"
        android:padding="20dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

9. Ubah file `activity_single_top.xml` menjadi seperti ini

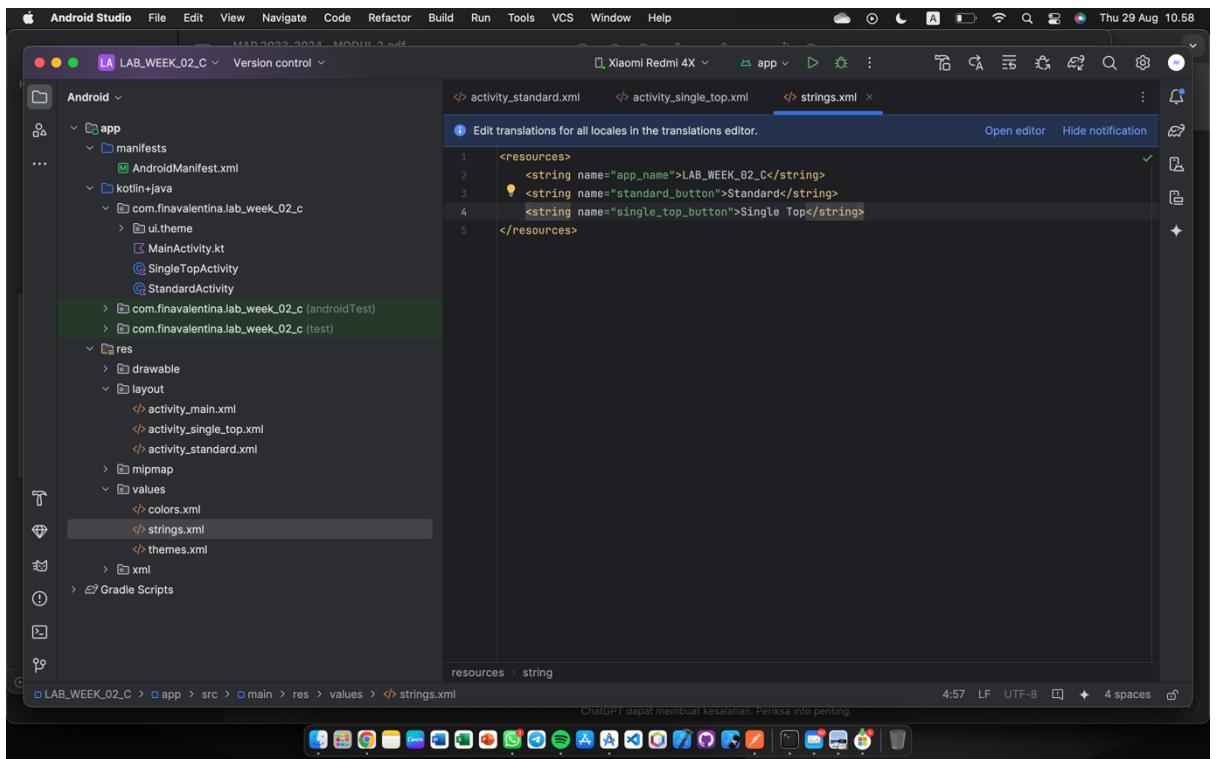
The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the XML file `activity_single_top.xml`. This XML file is identical to `activity_standard.xml`, both defining a `ConstraintLayout` with a single button. The button has an ID of `@+id/button_single_top`, a width of `200dp`, and a height of `wrap_content`. It is positioned with constraints relative to its parent layout.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

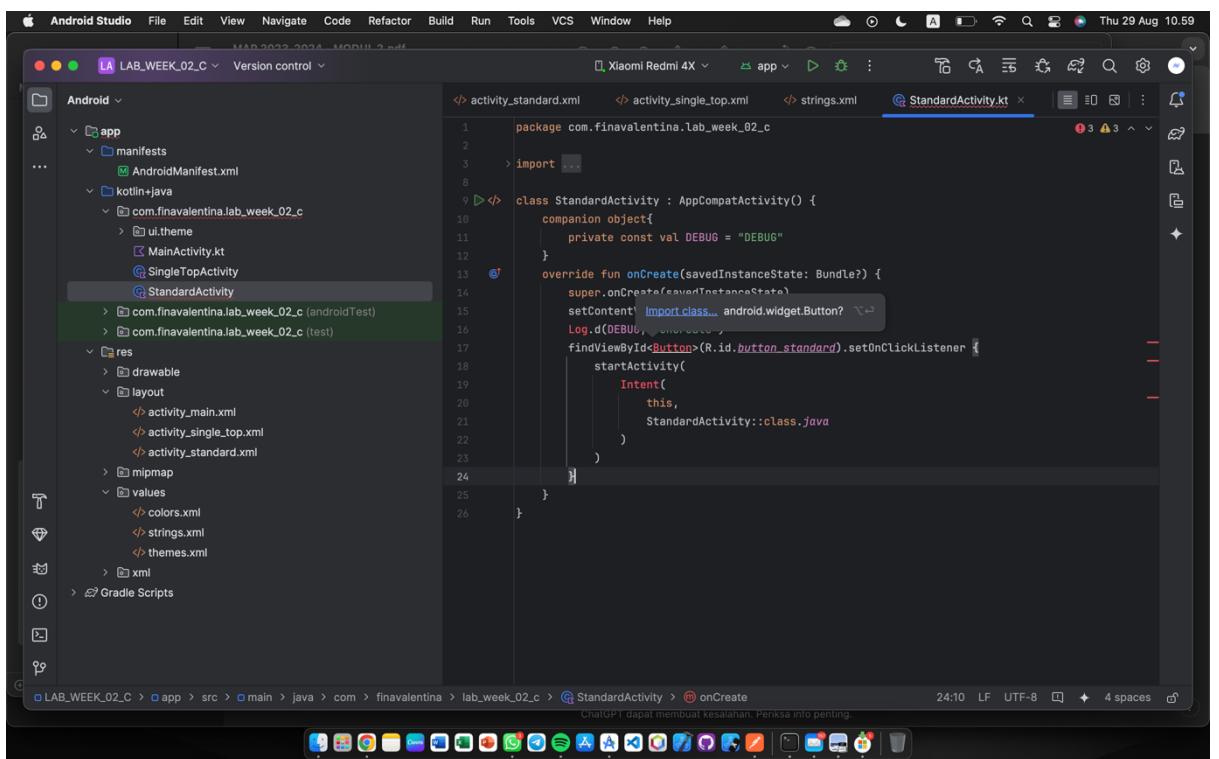
    <Button
        android:id="@+id/button_single_top"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:text="@string/single_top_button"
        android:padding="20dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

10. Ubah file `strings.xml` menjadi seperti ini



11. Buka file StandartActivity dan sesuaikan menjadi seperti ini lalu lakukan import terhadap baris yang berwarna merah



12. Lengkapi file Standart Activity menjadi seperti ini

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `StandardActivity.kt` file. The code defines a `StandardActivity` class that extends `AppCompatActivity`. It overrides the `onCreate` method to set the content view to `activity_standard`, log "onCreate", and start a new activity. It also overrides the `onNewIntent` method to log "onNewIntent". The code editor has syntax highlighting and code completion features.

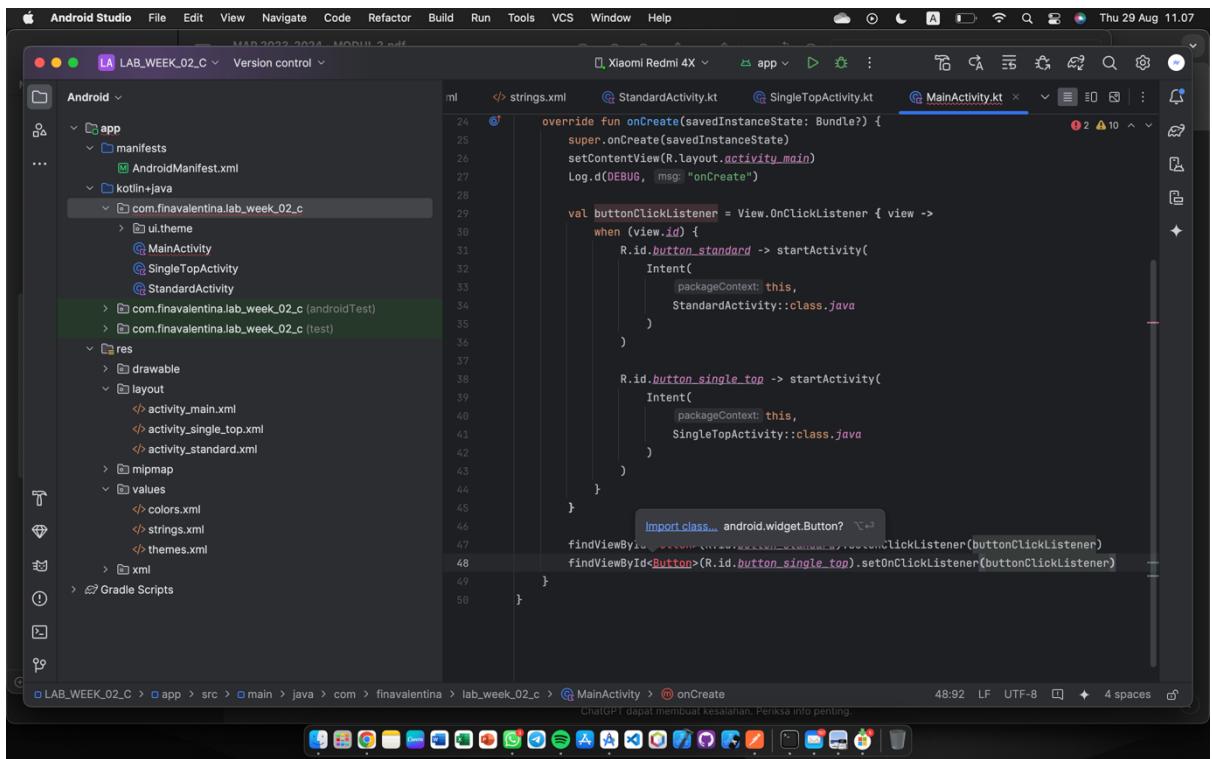
```
1 package com.finavalentina.lab_week_02_c
2
3 > import ...
4
5 <> class StandardActivity : AppCompatActivity() {
6     companion object{
7         private const val DEBUG = "DEBUG"
8     }
9
10    override fun onCreate(savedInstanceState: Bundle?) {
11        super.onCreate(savedInstanceState)
12        setContentView(R.layout.activity_standard)
13        Log.d(DEBUG, msg: "onCreate")
14        findViewById<Button>(R.id.button_standard).setOnClickListener { it: View! ->
15            startActivity(
16                Intent(
17                    packageContext: this,
18                    StandardActivity::class.java
19                )
20            )
21        }
22    }
23
24    override fun onNewIntent(intent: Intent) {
25        super.onNewIntent(intent)
26        Log.d(DEBUG, msg: "onNewIntent")
27    }
28
29 }
30
31
32
33
34
35 }
```

13. Ubah file SingleTopActivity menjadi seperti ini

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `SingleTopActivity.kt` file. The code defines a `SingleTopActivity` class that extends `AppCompatActivity`. It overrides the `onCreate` method to set the content view to `activity_single_top`, log "onCreate", and start a new activity. It also overrides the `onNewIntent` method to log "onNewIntent". The code editor has syntax highlighting and code completion features.

```
1 package com.finavalentina.lab_week_02_c
2
3 > import ...
4
5 <> class SingleTopActivity : AppCompatActivity() {
6     companion object{
7         private const val DEBUG = "DEBUG"
8     }
9
10    override fun onCreate(savedInstanceState: Bundle?) {
11        super.onCreate(savedInstanceState)
12        setContentView(R.layout.activity_single_top)
13        Log.d(DEBUG, msg: "onCreate")
14        findViewById<Button>(R.id.button_single_top).setOnClickListener { it: View! ->
15            startActivity(
16                Intent(
17                    packageContext: this,
18                    StandardActivity::class.java
19                )
20            )
21        }
22    }
23
24    override fun onNewIntent(intent: Intent) {
25        super.onNewIntent(intent)
26        Log.d(DEBUG, msg: "onNewIntent")
27    }
28
29 }
30
31
32
33
34
35 }
```

14. Ubah file MainActivity menjadi seperti ini mengikuti modul, dan lakukan import

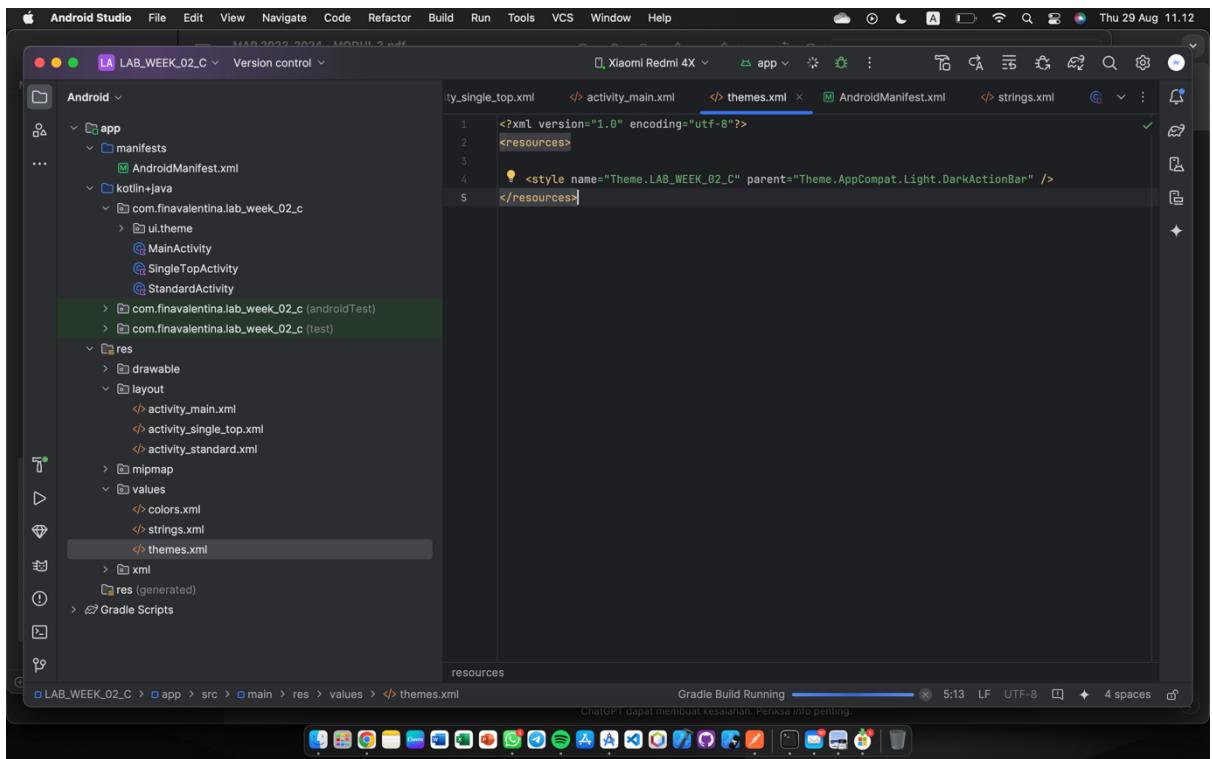


```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    Log.d("DEBUG", "onCreate")

    val buttonClickListener = View.OnClickListener { view ->
        when (view.id) {
            R.id.button_standard -> startActivity(
                Intent(
                    packageContext: this,
                    StandardActivity::class.java
                )
            )

            R.id.button_single_top -> startActivity(
                Intent(
                    packageContext: this,
                    SingleTopActivity::class.java
                )
            )
        }
    }
    findViewById(R.id.button_standard).setOnClickListener(buttonClickListener)
    findViewById(R.id.button_single_top).setOnClickListener(buttonClickListener)
}
```

15. Buka file themes.xml dan sesuaikan isi file nya seperti ini



```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="Theme.LAB_WEEK_02_C" parent="Theme.AppCompat.Light.DarkActionBar" />
</resources>
```

16. Jalankan aplikasinya dan lihat Logcat

The screenshot shows the Android Studio interface. The top navigation bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, and Help. The title bar indicates the project is 'LAB_WEEK_02_C' and the build variant is 'Module 2'. The bottom status bar shows the date as 'Thu 29 Aug 11.13'.

The main area consists of several panes:

- Code Editor:** Displays the file 'themes.xml' with the following content:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="Theme.LAB_WEEK_02_C" parent="Theme.AppCompat.Light.DarkActionBar" />
</resources>
```
- Logcat:** Shows log entries from the device 'Xiaomi Redmi 4X (326aec1a7d14)'. The log includes messages from OpenGLRenderer, BoostFramework, and com.finaventina.lab_week_02_c, indicating activity launches, timeline events, and partial code cache collections.
- Timeline:** A floating window showing performance metrics like mPerf and timeline events such as Activity_launch_request and Activity_launched.
- Bottom Bar:** Includes icons for various tools and the Android navigation bar.

17. Selesai