

Nama: Ika Novita Manurung.

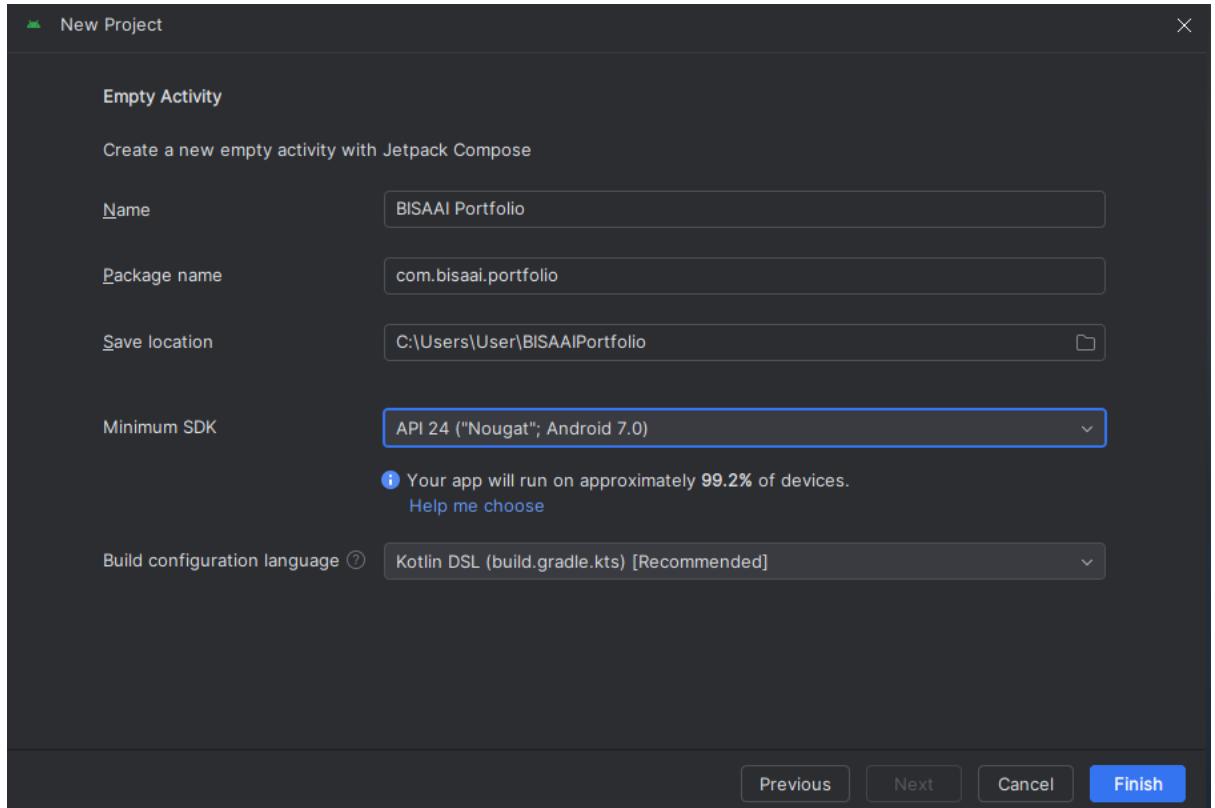
Kelas : SK231.

NIM : 202302110030.

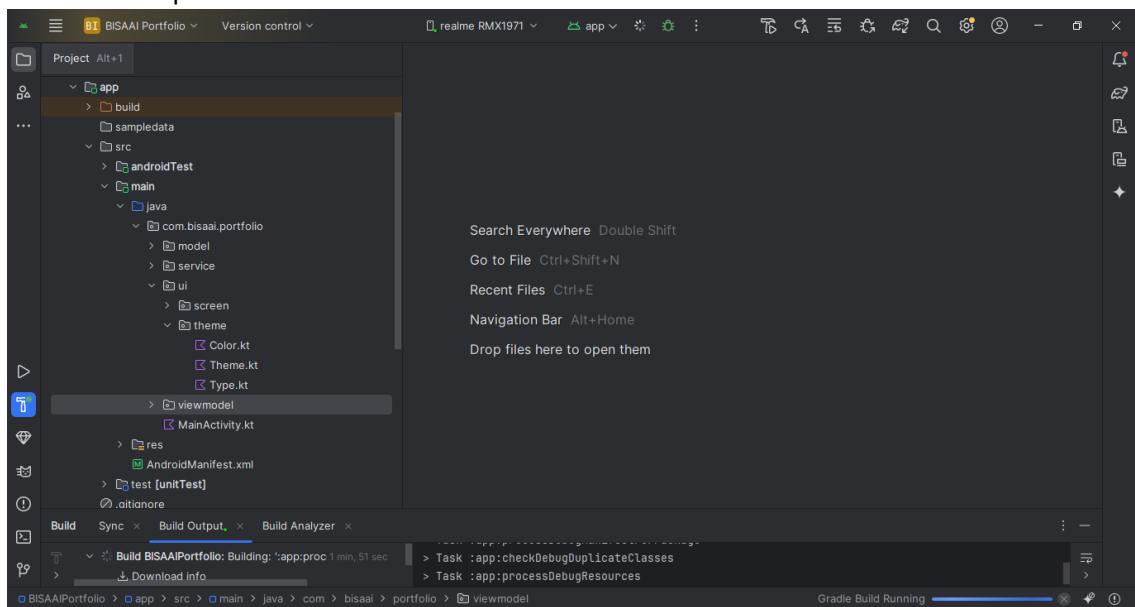
Prodi : Sistem Informasi.

Mata Kuliah: Pemrograman Mobile.

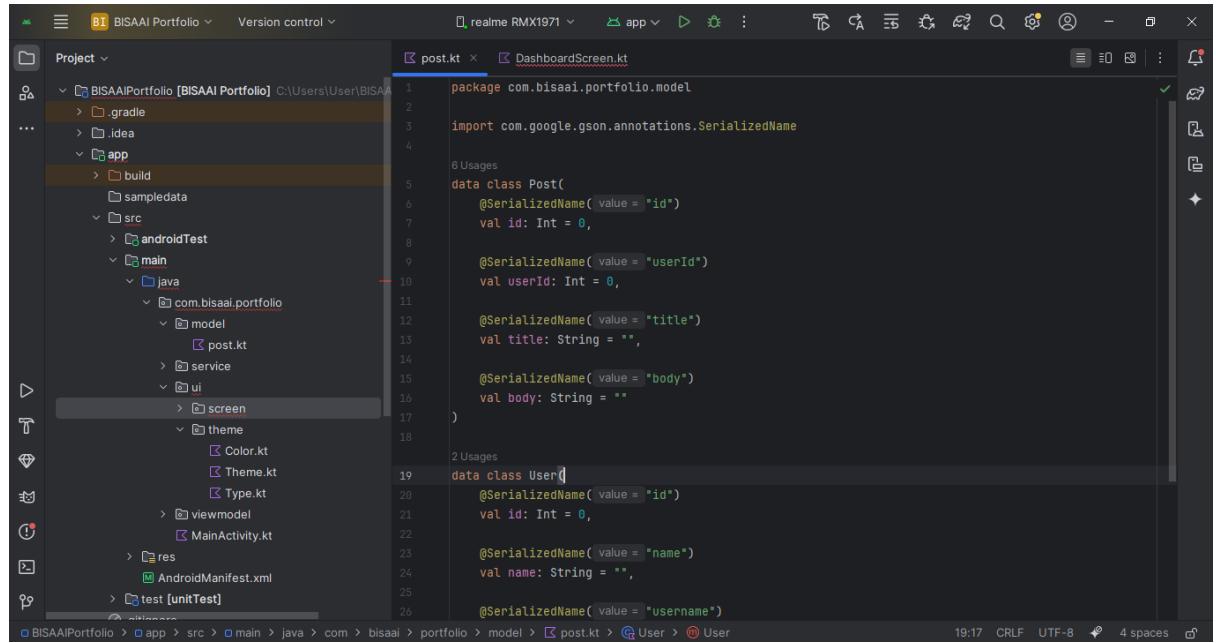
1. Lakukan setup projek seperti ini.



2. Buat folder seperti berikut

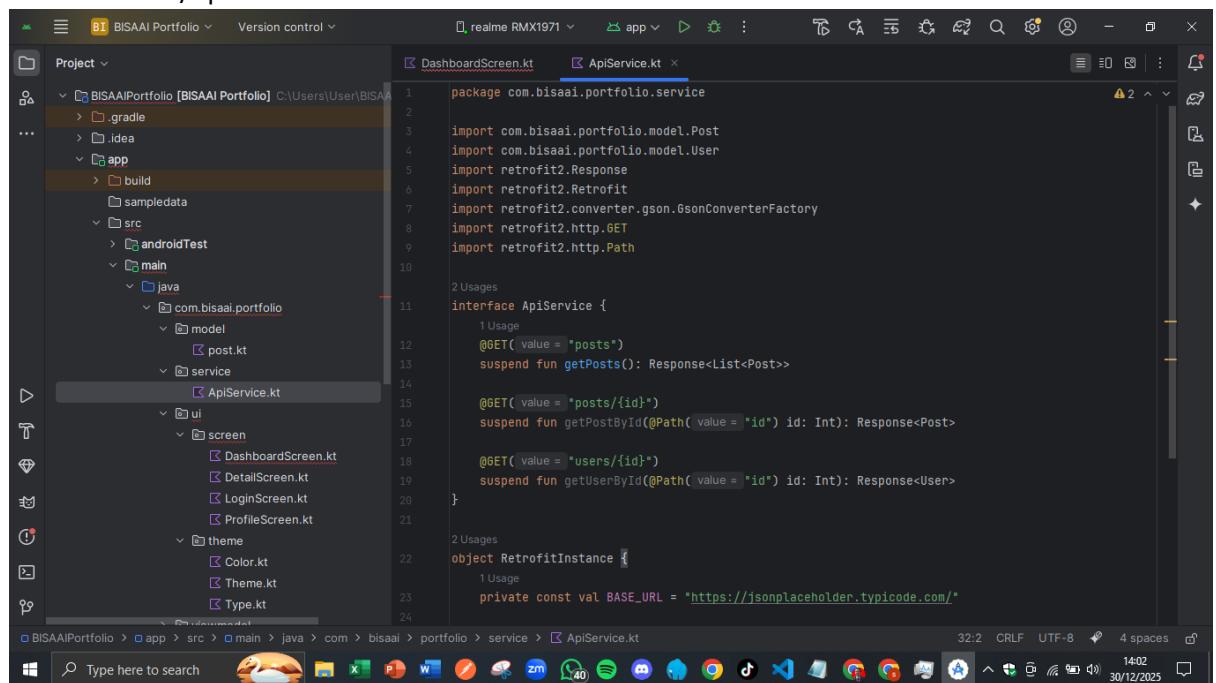


3. Isi file model/Post.kt:



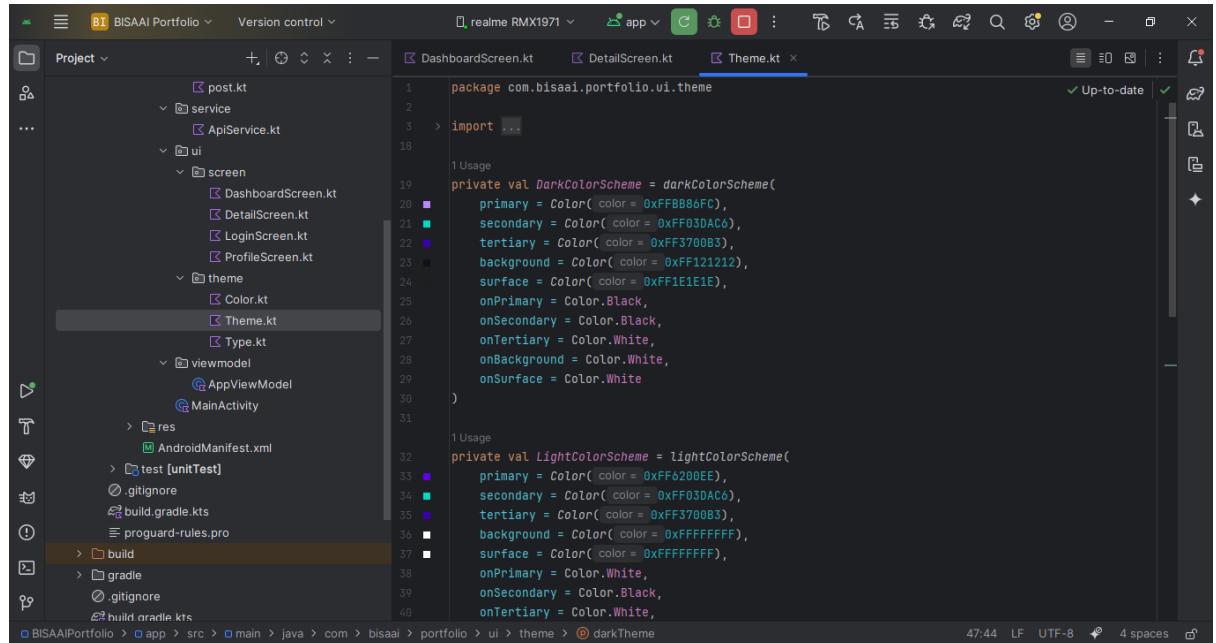
```
1 package com.bisaaai.portfolio.model
2
3 import com.google.gson.annotations.SerializedName
4
5 data class Post(
6     @SerializedName( value = "id")
7     val id: Int = 0,
8
9     @SerializedName( value = "userId")
10    val userId: Int = 0,
11
12     @SerializedName( value = "title")
13    val title: String = "",
14
15     @SerializedName( value = "body")
16    val body: String = ""
17 )
18
19 data class User(
20     @SerializedName( value = "id")
21     val id: Int = 0,
22
23     @SerializedName( value = "name")
24     val name: String = "",
25
26     @SerializedName( value = "username")
27 )
```

4. Isi file services/ApiService.kt:



```
1 package com.bisaaai.portfolio.service
2
3 import com.bisaaai.portfolio.model.Post
4 import com.bisaaai.portfolio.model.User
5 import retrofit2.Response
6 import retrofit2.Retrofit
7 import retrofit2.converter.gson.GsonConverterFactory
8 import retrofit2.http.GET
9 import retrofit2.http.Path
10
11 interface ApiService {
12     @GET("posts")
13     suspend fun getPosts(): Response<List<Post>>
14
15     @GET("posts/{id}")
16     suspend fun getPostById(@Path("id") id: Int): Response<Post>
17
18     @GET("users/{id}")
19     suspend fun getUserId(@Path("id") id: Int): Response<User>
20 }
21
22 object RetrofitInstance {
23     private const val BASE_URL = "https://jsonplaceholder.typicode.com/"
24 }
```

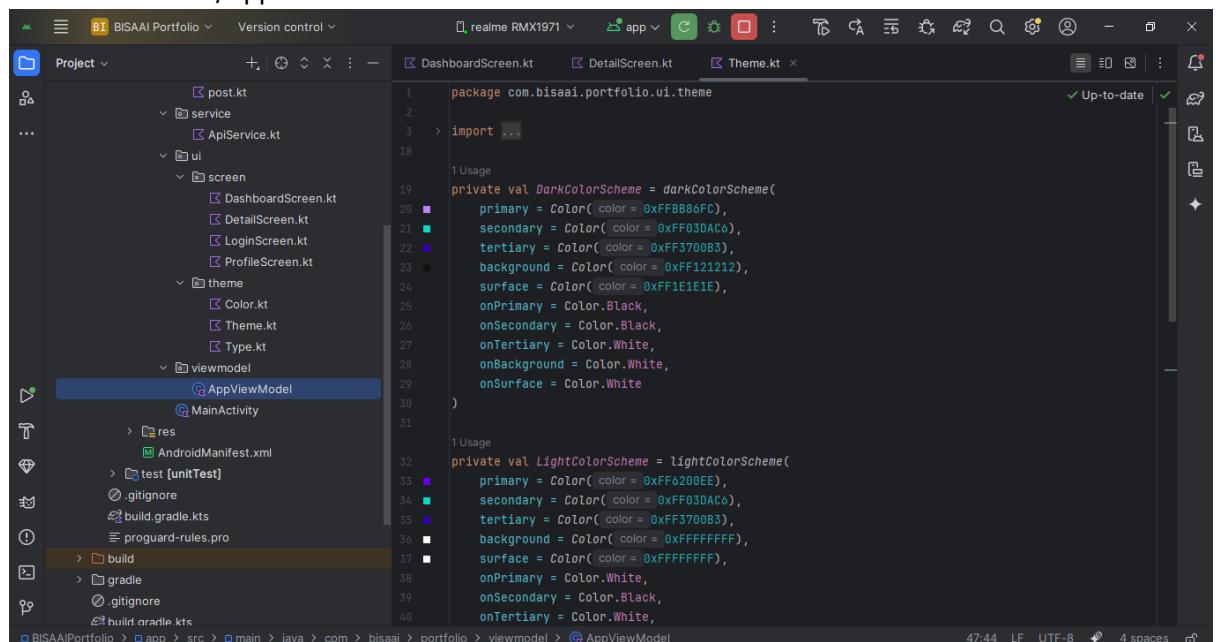
5. Isi file untuk theme/theme.kt:



The screenshot shows the Android Studio interface with the project navigation bar at the top. Below it is the file tree, which includes a 'ui' folder containing 'screen', 'theme', and 'viewmodel' subfolders. Inside 'ui/theme', there are 'Color.kt' and 'Theme.kt'. The 'Theme.kt' file is currently selected and open in the main editor window. The code in 'Theme.kt' defines two color schemes: 'DarkColorScheme' and 'LightColorScheme', each with various color definitions.

```
1 package com.bisaaai.portfolio.ui.theme
2
3 > import ...
4
5 Usage
6 private val DarkColorScheme = darkColorScheme(
7     primary = Color( color = 0xFFBB80FC),
8     secondary = Color( color = 0xFF03DAC0),
9     tertiary = Color( color = 0xFF3700B3),
10    background = Color( color = 0xFF121212),
11    surface = Color( color = 0xFF1E1E1E),
12    onPrimary = Color.Black,
13    onSecondary = Color.Black,
14    onTertiary = Color.White,
15    onBackground = Color.White,
16    onSurface = Color.White
17 )
18
19 Usage
20 private val LightColorScheme = lightColorScheme(
21     primary = Color( color = 0xFF6200EE),
22     secondary = Color( color = 0xFF03DAC0),
23     tertiary = Color( color = 0xFF3700B3),
24     background = Color( color = 0xFFFFFFFF),
25     surface = Color( color = 0xFFFFFFFF),
26     onPrimary = Color.White,
27     onSecondary = Color.Black,
28     onTertiary = Color.White,
29     onBackground = Color.White,
30     onSurface = Color.White
31 )
32
33 Usage
34 private val AppViewMode...
35
36 Usage
37 private val AppViewMode...
38
39 Usage
40
```

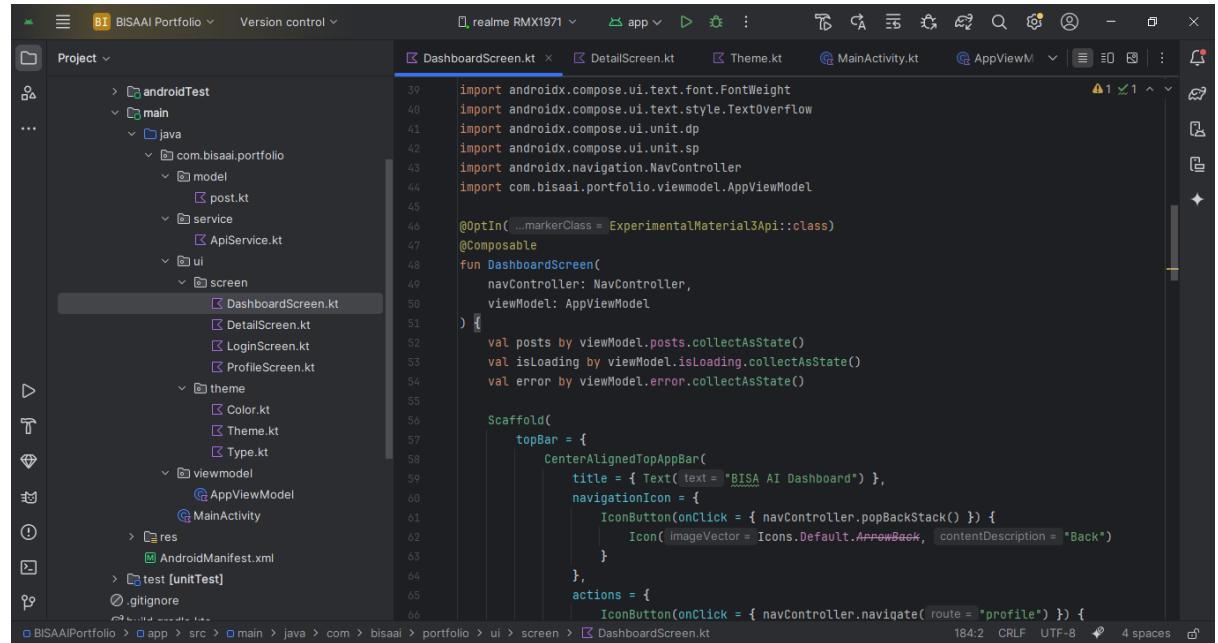
Isi file viewmodel/AppViewModel.kt:



The screenshot shows the Android Studio interface with the project navigation bar at the top. Below it is the file tree, which includes a 'ui' folder containing 'screen', 'theme', and 'viewmodel' subfolders. Inside 'ui/viewmodel', there is an 'AppViewModel' file. The 'AppViewModel' file is currently selected and open in the main editor window. The code in 'AppViewModel' imports 'Theme.kt' and uses its 'DarkColorScheme' and 'LightColorScheme'.

```
1 package com.bisaaai.portfolio.ui.theme
2
3 > import ...
4
5 Usage
6 private val DarkColorScheme = darkColorScheme(
7     primary = Color( color = 0xFFBB80FC),
8     secondary = Color( color = 0xFF03DAC0),
9     tertiary = Color( color = 0xFF3700B3),
10    background = Color( color = 0xFF121212),
11    surface = Color( color = 0xFF1E1E1E),
12    onPrimary = Color.Black,
13    onSecondary = Color.Black,
14    onTertiary = Color.White,
15    onBackground = Color.White,
16    onSurface = Color.White
17 )
18
19 Usage
20 private val LightColorScheme = lightColorScheme(
21     primary = Color( color = 0xFF6200EE),
22     secondary = Color( color = 0xFF03DAC0),
23     tertiary = Color( color = 0xFF3700B3),
24     background = Color( color = 0xFFFFFFFF),
25     surface = Color( color = 0xFFFFFFFF),
26     onPrimary = Color.White,
27     onSecondary = Color.Black,
28     onTertiary = Color.White,
29     onBackground = Color.White,
30     onSurface = Color.White
31 )
32
33 Usage
34 private val AppViewMode...
35
36 Usage
37 private val AppViewMode...
38
39 Usage
40
```

6. Halaman screen/DashboardScreen.kt:



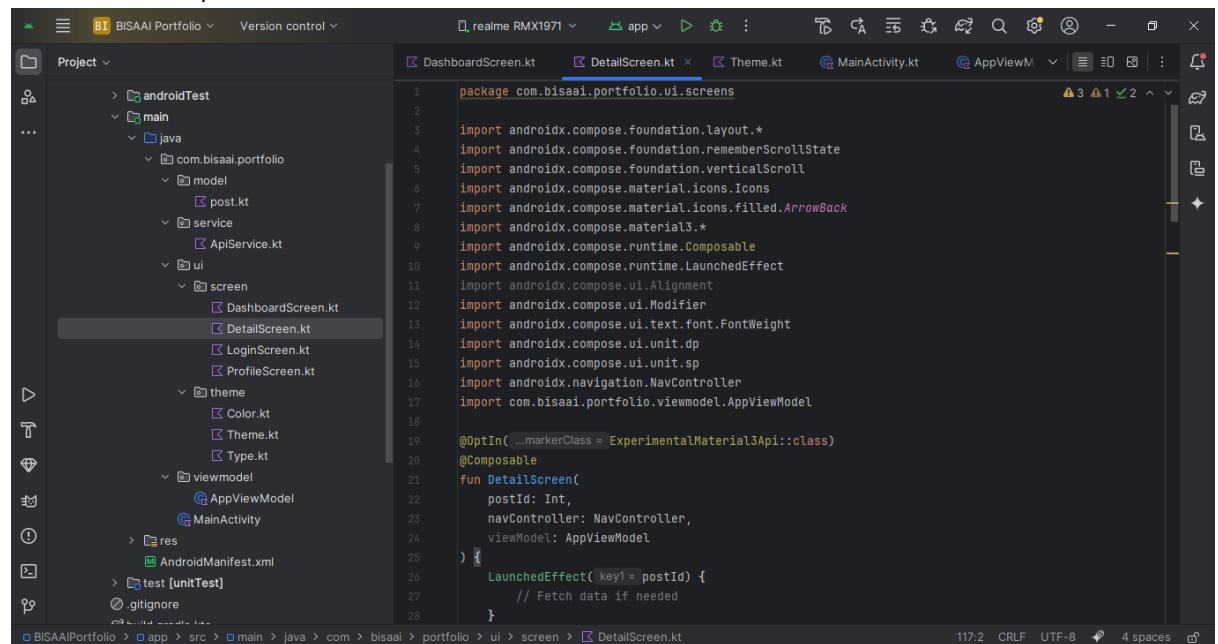
The screenshot shows the Android Studio interface with the project navigation bar at the top. The left sidebar displays the project structure under 'BISAAI Portfolio'. The main editor window shows the code for `DashboardScreen.kt`. The code uses Jetpack Compose to create a Scaffold with a top bar containing a title and a back button. It also includes logic to handle posts, loading state, and errors from the view model.

```
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextOverflow
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController
import com.bisaaai.portfolio.viewmodel.AppViewModel

@OptIn(markerClass = ExperimentalMaterial3Api::class)
@Composable
fun DashboardScreen(
    navController: NavController,
    viewModel: AppViewModel
) {
    val posts by viewModel.posts.collectAsState()
    val isLoading by viewModel.isLoading.collectAsState()
    val error by viewModel.error.collectAsState()

    Scaffold(
        topBar = {
            CenterAlignedTopAppBar(
                title = { Text(text = "BISA AI Dashboard") },
                navigationIcon = {
                    IconButton(onClick = { navController.popBackStack() }) {
                        Icon(imageVector = Icons.Default.ArrowBack, contentDescription = "Back")
                    }
                },
                actions = {
                    IconButton(onClick = { navController.navigate(route = "profile") })
                }
            )
        }
    )
}
```

Halaman screen/DetailScreen.kt:



The screenshot shows the Android Studio interface with the project navigation bar at the top. The left sidebar displays the project structure under 'BISAAI Portfolio'. The main editor window shows the code for `DetailScreen.kt`. The code uses Jetpack Compose to create a Scaffold with a top bar containing a back button. It imports various Compose components and defines a `DetailScreen` function that takes a post ID, a navigation controller, and a view model. It includes a `LaunchedEffect` block to fetch data if needed.

```
package com.bisaaai.portfolio.ui.screens

import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.icons(Icons)
import androidx.compose.material.material3.ArrowBack
import androidx.compose.runtime.Composable
import androidx.compose.runtime.LaunchedEffect
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController
import com.bisaaai.portfolio.viewmodel.AppViewModel

@OptIn(markerClass = ExperimentalMaterial3Api::class)
@Composable
fun DetailScreen(
    postId: Int,
    navController: NavController,
    viewModel: AppViewModel
) {
    LaunchedEffect(key1 = postId) {
        // Fetch data if needed
    }
}
```

Halaman screen/Loginscreen.kt:

The screenshot shows the Android Studio interface with the project navigation bar at the top. Below it is the file tree, which includes 'Project', 'src' (containing 'main', 'java', 'res', 'test [unitTest]', and '.gitignore'), and 'app' (containing 'src' with 'main' and 'java' subfolders). Under 'main/java/com.bisaii.portfolio/ui/screen', the 'Loginscreen.kt' file is selected. The code editor displays the 'Loginscreen.kt' file, which contains code for a Composable function named 'Loginscreen'. The code uses Material 3 components like 'Column', 'Text', and 'Icon'. It also imports various androidx.compose.* and kotlinx.* libraries.

```
import androidx.compose.ui.text.input.KeyboardType
import androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController
import kotlinx.coroutines.delay
import kotlinx.coroutines.MainScope
import kotlinx.coroutines.launch

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Loginscreen(
    navController: NavController,
    onLoginSuccess: () -> Unit
) {
    var email by remember { mutableStateOf("") }
    var password by remember { mutableStateOf("") }
    var isLoading by remember { mutableStateOf(false) }

    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(all = 24.dp),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        // Header
    }
}
```

Halaman Screen/ProfileScreen.kt:

The screenshot shows the Android Studio interface with the project navigation bar at the top. Below it is the file tree, which includes 'Project', 'src' (containing 'main', 'java', 'res', 'test [unitTest]', and '.gitignore'), and 'app' (containing 'src' with 'main' and 'java' subfolders). Under 'main/java/com.bisaii.portfolio/ui/screen', the 'ProfileScreen.kt' file is selected. The code editor displays the 'ProfileScreen.kt' file, which contains code for a Composable function named 'ProfileScreen'. The code uses Material 3 components like 'Scaffold', 'TopBar', and 'Icon'. It also imports various androidx.compose.* and kotlinx.* libraries.

```
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.clip
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun ProfileScreen(navController: NavController) {
    Scaffold(
        topBar = {
            CenterAlignedTopAppBar(
                title = { Text(text = "Profile") },
                navigationIcon = {
                    IconButton(onClick = { navController.popBackStack() }) {
                        Icon(imageVector = Icons.Default.ArrowBack, contentDescription = "Back")
                    }
                }
            )
        }
    ) { paddingValues ->
        Column(
            modifier = ...
        )
    }
}
```

7. Isi AndroidManifest.xml:

The screenshot shows the Android Studio interface with the project 'BISAAI Portfolio' selected. The left sidebar displays the project structure, including files like ginScreen.kt, Theme.kt, MainActivity.kt, and the current file, AndroidManifest.xml. The right side shows the XML content of the manifest file.

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

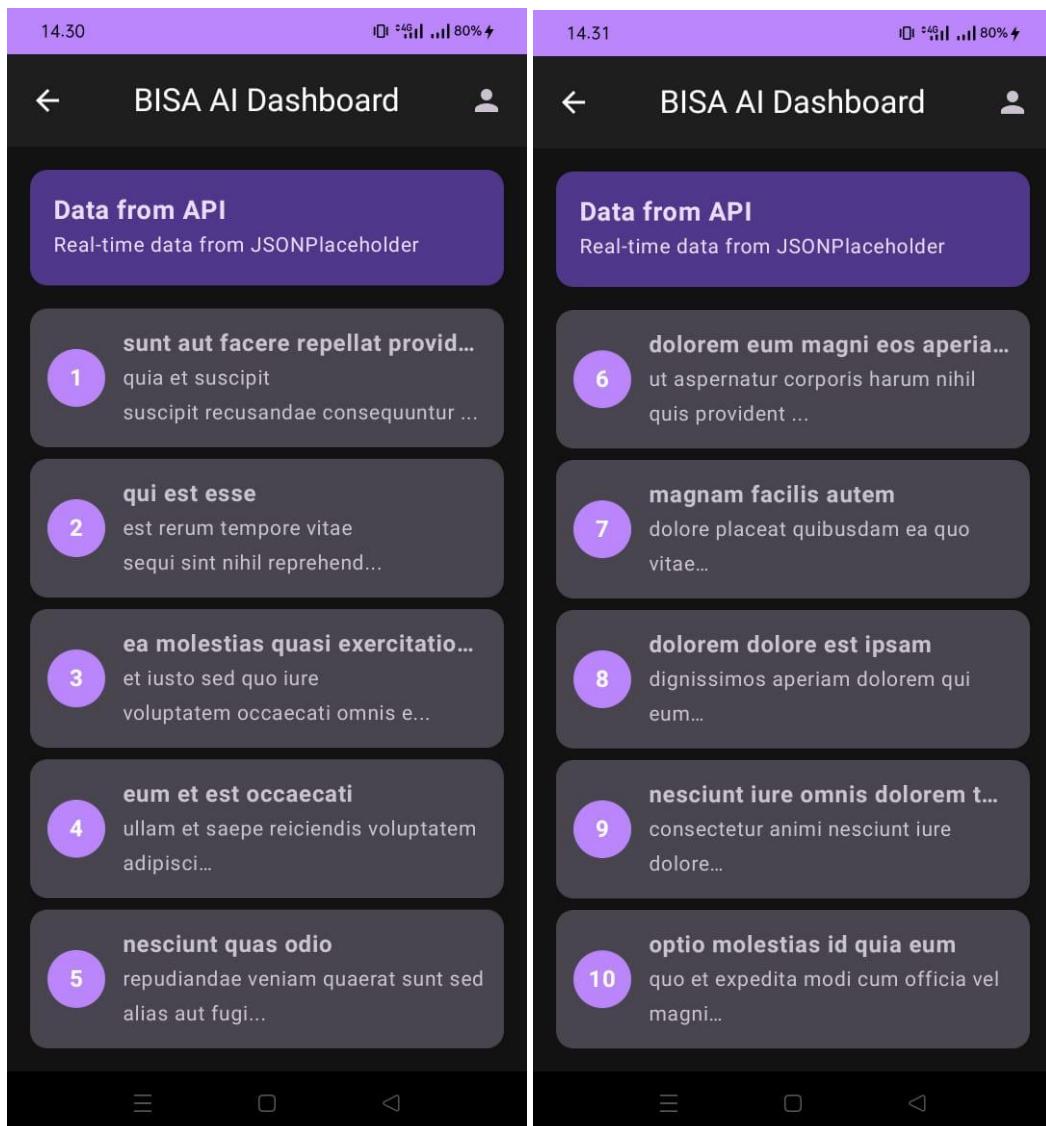
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

    <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.BISAAIPortfolio"
        android:usesClearTextTraffic="true"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="BISAAI Portfolio"
            android:theme="@style/Theme.BISAAIPortfolio">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        
    

```

The code editor shows the XML structure of the manifest, with the 'Text' tab selected at the bottom. The file path at the bottom of the screen is BISAAIPortfolio > app > src > main > AndroidManifest.xml.

Hasil test:



The screenshot shows the BISA AI Dashboard Profile screen. At the top, there are three status bars indicating battery level at 80% and signal strength. The main header "BISA AI Dashboard" is on the left, and "Profile" is on the right. Below the header, a purple box labeled "Data from API" contains five numbered items (6, 7, 8, 9, 10) with placeholder text. To the right, a section for "BISA AI Student" displays "Mobile Developer" status, "Student Information" (John Doe, student@bisa.ai, AI For Everyone), and a "Portfolio Project" section listing four bullet points: "Android App with Jetpack Compose", "API Integration with JSONPlaceholder", "Real-time data fetching", and "Material Design 3 UI".

14.31 14.42 80%

BISA AI Dashboard Profile

Data from API
Real-time data from JSONPlaceholder

BA

6 dolorem eum magni eos aperiam...
ut aspernatur corporis harum nihil
quis provident ...

7 magnam facilis autem
dolore placeat quibusdam ea quo
vitiae...

8 dolorem dolore est ipsam
dignissimos aperiam dolorem qui
eum...

9 nesciunt iure omnis dolorem t...
consectetur animi nesciunt iure
dolore...

10 optio molestias id quia eum
quo et expedita modi cum officia vel
magni...

BISA AI Student
Mobile Developer

Student Information

John Doe

student@bisa.ai

AI For Everyone

Portfolio Project

- Android App with Jetpack Compose
- API Integration with JSONPlaceholder
- Real-time data fetching
- Material Design 3 UI