Kotlin

MOBILE PROGRAMMING

LAZYCOLUMN AND LAZYROW

ROBBY TAN

LazyColumn & LazyRow

LazyColumn and LazyRow

- Mengimplementasi ***lazy loading***.
  - **Lazy**: elemen (content) akan di-*load* atau ditambahkan secara bertahap saat proses *scroll*.
  - Kunci utama yang membedakan dengan *vertical| horizontal scrolling*.
- *Class* pada Android View: **RecyclerView**.

```
1.  @Composable
2.  fun LazyColumnExample() {
3.    LazyColumn(
4.      modifier = Modifier.padding(8.dp)
5.    ) {
6.      items(20) { index ->
7.        Box(
8.          modifier = Modifier
9.            .fillMaxWidth()
10.           .background(Color(0xFFF6D758))
11.       ) {
12.         Text(
13.           text = "Item ke-$index",
14.           modifier = Modifier
15.             .fillMaxWidth()
16.             .padding(16.dp)
17.         )
18.       }
19.       Spacer(modifier = Modifier.height(5.dp))
20.     }
21.   }
22. }
```

```
1.  @Composable
2.  fun LazyRowExample() {
3.    LazyRow {
4.      items(10) { index ->
5.        Box(
6.          modifier = Modifier
7.            .background(Color(0xFF6FFDF1))
8.            .padding(8.dp)
9.        ) {
10.         Text(
11.           text = "Category $index",
12.           modifier = Modifier
13.             .padding(8.dp)
14.         )
15.       }
16.       Spacer(
17.         modifier = Modifier.width(5.dp)
18.       )
19.     }
20.   }
21. }
```

LazyColumn and Card

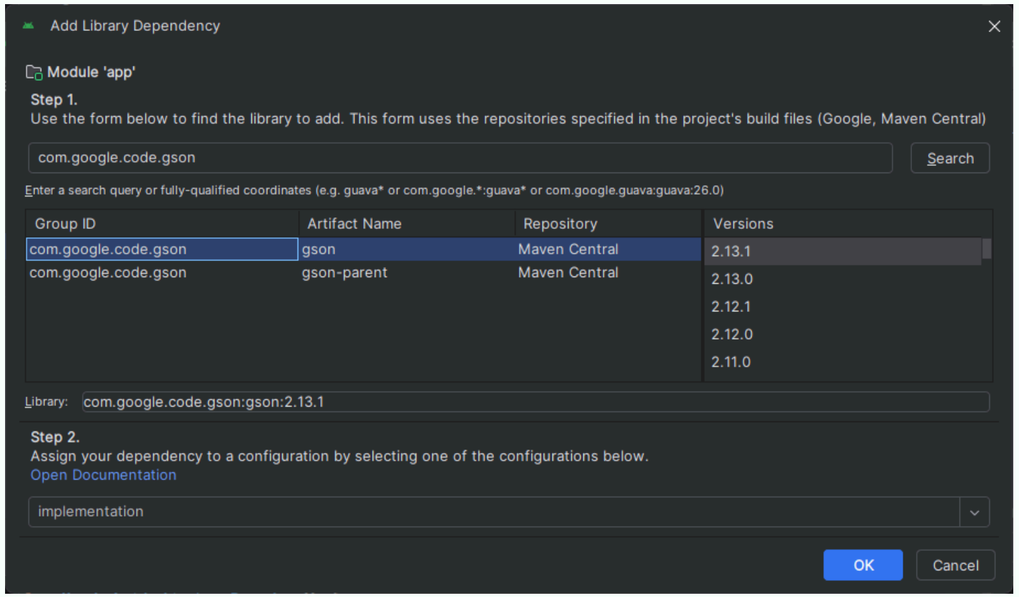
```
1.  data class Movie(var cover: String, var title: String, var description: String)
```

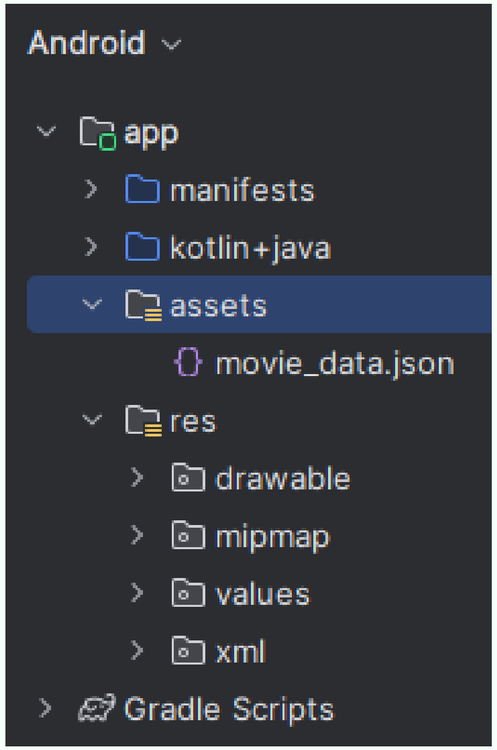
- Implementasi konsep OOP menggunakan data class.
- Penggunaan **Composable Card** yang memodelkan 1 item dalam LazyColumn atau LazyRow.

```
1.  @Composable
2.  fun MovieList(movies: List<Movie>, modifier: Modifier = Modifier) {
3.    LazyColumn(modifier = modifier) {
4.      items(movies) { movie ->
5.        MovieItem(movie)
6.      }
7.    }
8.  }
9.
10. @Composable
11. fun MovieItem(movie: Movie) {
12.   Card(
13.     modifier = Modifier
14.       .fillMaxWidth()
15.       .padding(8.dp),
16.     elevation = CardDefaults.cardElevation(4.dp)
17.   ) {
18.     Row(modifier = Modifier.padding(16.dp),
19.       verticalAlignment = Alignment.CenterVertically) {
20.       Image(
21.         painter = painterResource(R.drawable.ic_launcher_background),
22.         contentDescription = movie.title,
23.         modifier = Modifier
24.           .size(64.dp)
25.           .clip(RoundedCornerShape(8.dp))
26.       )
27.
28.       Spacer(modifier = Modifier.width(16.dp))
29.
30.       Column(modifier = Modifier.weight(1f)) {
31.         Text(text = movie.title, fontSize = 18.sp)
32.         Text(text = movie.description, fontSize = 14.sp)
33.       }
34.     }
35.   }
36. }
```

LazyColumn and JSON

- Tipe data yang digunakan dalam pertukaran data:
  - JSON
  - XML
- Penambahan dependencies: **com.google.code.gson**.
- File JSON diletakkan pada *folder* **assets**.





Read Local JSON File

```
fun loadJsonData(context: Context): List<Movie> {
  val json = context.assets.open("movie_data.json").bufferedReader().use { it.readText() }
  val type = object : TypeToken<List<Movie>>() {}.type
  return Gson().fromJson(json, type)
}
```

```
@Composable
fun MovieListPage(movies: List<Movie>, modifier: Modifier = Modifier) {
  LazyColumn(modifier = modifier) {
    items(movies) { movie ->
      Card {
        Column(modifier = Modifier.padding(8.dp)) {
          Text(
            movie.title,
            fontWeight = FontWeight.Bold,
            style = MaterialTheme.typography.titleMedium
          )
          Text(movie.description, style = MaterialTheme.typography.labelSmall)
        }
      }
      Spacer(modifier = Modifier.height(5.dp))
    }
  }
}
```

Inception
A thief who steals corporate secrets through dream-sharing technology is given the inverse task of planting an idea.

The Matrix
A computer hacker learns the true nature of his reality and his role in the war against its controllers.

Interstellar
A team of explorers travel through a wormhole in space in an attempt to ensure humanity's survival.

The Avengers
Earth's mightiest heroes must come together to stop Loki and his alien army from enslaving humanity.

Avengers: Age of Ultron
Tony Stark and Bruce Banner create Ultron, a peacekeeping AI that goes rogue and tries to destroy the world.

Avengers: Infinity War
The Avengers and their allies must stop Thanos before he collects all six Infinity Stones and wipes out half the universe.

Avengers: Endgame
After the devastating events of Infinity War, the Avengers assemble once more in a final stand to undo Thanos' actions.

Iron Man

Thank You

ROBBY.TAN@IT.MARANATHA.EDU