

## Mendapatkan Data dari Internet

Berikut adalah contoh aplikasi untuk mendapatkan data dari internet menggunakan Retrofit dengan **converter Scalar** di aplikasi Android Jetpack Compose. Converter ini digunakan untuk menangani data API yang hanya berupa teks atau string tanpa struktur JSON. URL yang diakses adalah:

<https://jsonplaceholder.typicode.com/posts>

URL ini jika dijalankan di browser akan menghasilkan:

```
[
  {
    "userId": 1,
    "id": 1,
    "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
    "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
  },
  {
    "userId": 1,
    "id": 2,
    "title": "qui est esse",
    "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi nulla"
  },
  . . .
  . . .
]
```

### 1. Setup Proyek Android Studio

#### 1. Buat proyek baru:

- Pilih **Empty Compose Activity**.
- Beri nama proyek, misalnya `RetrofitScalarComposeApp`.
- Pilih minimum SDK, misalnya API 21 (Android 5.0).

#### 2. Tambahkan dependensi yang diperlukan: Buka file `build.gradle (Module)` dan tambahkan:

```
dependencies {

    // ViewModel
    implementation ("androidx.lifecycle:lifecycle-viewmodel-compose:2.8.7")

    // Retrofit
    implementation ("com.squareup.retrofit2:retrofit:2.9.0")
    // Retrofit with Scalar Converter
    implementation ("com.squareup.retrofit2:converter-scalars:2.9.0")

}
```

Klik **Sync Now** setelah menambahkan dependensi.

## 2. Buat API Service

1. Buat paket bernama `network`
2. Buat file Kotlin baru di paket `network` dengan nama `ApiService.kt`:

```
import retrofit2.Retrofit
import retrofit2.converter.scalars.ScalarsConverterFactory
import retrofit2.http.GET

private const val BASE_URL = "https://jsonplaceholder.typicode.com"

private val retrofit = Retrofit.Builder()
    .addConverterFactory(ScalarsConverterFactory.create())
    .baseUrl(BASE_URL)
    .build()

// tentukan antarmuka yang disebut ApiService yang menentukan
// cara Retrofit berkomunikasi dengan server web
// Tambahkan fungsi yang bernama getPosts()
// untuk mendapatkan respon string dari layanan web.
// Saat metode getPosts() dipanggil, Retrofit memanggil:
// https://jsonplaceholder.typicode.com/posts
interface ApiService {
    @GET("posts")
    suspend fun getPosts(): String
}

// membuat objek singleton
// inisialisasi variabel retrofitService menggunakan metode
retrofit.create()
object RetrofitInstance {
    val retrofitService : ApiService by lazy {
        retrofit.create(ApiService::class.java)
    }
}
```

## 3. Buat ViewModel

1. Buat paket baru bernama `viewmodel`
2. Buat kelas baru bernama `PostViewModel.kt` di paket `viewmodel`:

```
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import com.example.internetskalar.network.RetrofitInstance
import kotlinx.coroutines.flow.MutableStateFlow
import kotlinx.coroutines.flow.StateFlow
```

```

import kotlinx.coroutines.launch

class PostViewModel : ViewModel() {
    private val _posts = MutableStateFlow<String>("")
    val posts: StateFlow<String> = _posts

    init {
        fetchPosts()
    }

    private fun fetchPosts() {
        viewModelScope.launch {
            try {
                val response = RetrofitInstance.retrofitService.getPosts()
                _posts.value = response
            } catch (e: Exception) {
                _posts.value = "Error: ${e.message}"
            }
        }
    }
}

```

## 4. Buat UI

1. Buat file `PostScreen.kt`

```

import android.text.Layout
import androidx.compose.runtime.Composable
import androidx.compose.runtime.collectAsState
import androidx.compose.runtime.getValue
import com.example.internetskalar.viewmodel.PostViewModel
import androidx.lifecycle.viewmodel.compose.viewModel
import androidx.compose.foundation.layout.Box
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.CircularProgressIndicator
import androidx.compose.material3.Text
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.unit.dp

@Composable
fun PostScreen(modifier: Modifier = Modifier) {
    val viewModel: PostViewModel = viewModel()
    val posts by viewModel.posts.collectAsState()

    Box(
        contentAlignment = Alignment.Center,
        modifier = modifier.fillMaxSize()
    ) {

```

```

        if (posts.isEmpty()) {
            CircularProgressIndicator(
                modifier = Modifier.align(Alignment.Center)
            )
        } else {
            Text(
                text = posts,
                modifier = Modifier.padding(16.dp)
            )
        }
    }
}

```

## 5. Hubungkan ke MainActivity

1. Ubah MainActivity.kt:

```

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            RetrofitScalarComposeAppTheme {
                Scaffold(modifier = Modifier.fillMaxSize()) { innerPadding
->
                    PostScreen(
                        modifier = Modifier.padding(innerPadding)
                    )
                }
            }
        }
    }
}

```

## 6. Tambahkan Izin Internet

Buka file `manifests/AndroidManifest.xml`. Tambahkan baris ini sebelum tag `<application>`:

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

## 7. Jalankan Aplikasi

- Jalankan aplikasi di emulator atau perangkat fisik.
- Anda akan melihat data string mentah yang diambil dari API ditampilkan di layar.

## Tampilan UI:

- Jika data berhasil diambil, string akan ditampilkan di layar.
- Jika ada masalah (seperti tidak ada koneksi), error message akan muncul.

