

---

## Worksheet Praktikum 10: Networking dengan Retrofit & Glide

Mata Kuliah	Bahasa Pemrograman II (Praktikum)
Kode MK	S1085
Topik	Networking (REST API), JSON Parsing, & Image Loading
Pertemuan	10 (Sepuluh)
Dosen Pengampu	Afrig Aminuddin, S.Kom., M.Eng., Ph.D.
Alokasi Waktu	170 Menit (60 Menit Praktikum, 10 Menit Diskusi, 30 Menit Tugas, 60 Menit Mandiri)
Nama Mahasiswa	Revardi Maulaya Labib
NIM	24.12.3102

---

### A. Tujuan Pembelajaran

Setelah menyelesaikan sesi praktikum ini, mahasiswa diharapkan:

- Memahami konsep dasar **REST API** dan format data **JSON**.
  - Mampu mengimplementasikan library **Retrofit** untuk mengambil data dari internet.
  - Mampu mengubah (parsing) data JSON menjadi Objek Kotlin menggunakan **GSON**.
  - Mampu menampilkan gambar dari URL internet menggunakan library **Glide**.
  - Mampu menangani izin akses internet pada Android Manifest.
- 

### B. Dasar Teori Singkat

1. **Retrofit**: Library HTTP client type-safe untuk Android yang dibuat oleh Square. Ini adalah standar industri untuk melakukan request jaringan (GET, POST, PUT, DELETE).
  2. **JSON (JavaScript Object Notation)**: Format pertukaran data yang ringan. Retrofit akan otomatis mengkonversi JSON dari server menjadi Data Class Kotlin.
  3. **Glide**: Library pemuat gambar yang cepat dan efisien. Glide menangani *caching* dan *decoding* gambar dari URL secara otomatis.
- 

### C. Alat dan Bahan

1. PC/Laptop dengan Android Studio.
2. **Koneksi Internet** (Wajib, karena kita akan mengakses server publik).

3. API Publik untuk latihan: <https://reqres.in/> (Kita akan menggunakan endpoint List Users).

---

#### D. Langkah-Langkah Praktikum (Guided Practice)

**Skenario:** Membuat aplikasi "User List" yang mengambil data (Nama, Email, dan Foto Avatar) dari server reqres.in dan menampilkannya di RecyclerView.

##### Langkah 1: Konfigurasi Dependency & Izin

1. Buka build.gradle (Module: app). Tambahkan library Retrofit dan Glide: Gradle

```
// Retrofit & Gson  
implementation 'com.squareup.retrofit2:retrofit:2.9.0'  
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'  
// Glide (Image Loader)  
implementation 'com.github.bumptech.glide:glide:4.16.0'
```

Klik Sync Now.

2. Buka AndroidManifest.xml. Tambahkan izin internet di atas tag <application>: XML

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

##### Langkah 2: Siapkan Model Data (Data Class)

Kita perlu menyesuaikan Data Class dengan struktur JSON dari <https://reqres.in/api/users>.

1. Buat file UserResponse.kt:  
Kotlin

```
data class UserResponse(  
    val data: List<User>  
)  
  
data class User(  
    val id: Int,  
    val email: String,  
    val first_name: String,  
    val last_name: String,  
    val avatar: String // Ini berisi URL gambar  
)
```

##### Langkah 3: Buat Interface API

Buat file interface baru ApiService.kt. Di sini kita mendefinisikan endpoint URL.

Kotlin

```
import retrofit2.Call  
import retrofit2.http.GET  
  
interface ApiService {  
    @GET("api/users?page=1") // Endpoint untuk mengambil list user halaman 1  
    fun getListUsers(): Call<UserResponse>  
}
```

##### Langkah 4: Konfigurasi Retrofit Client

Buat file object ApiConfig.kt untuk membuat koneksi.

## Kotlin

```
import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory

object ApiConfig {
    private const val BASE_URL = "https://reqres.in/"

    fun getApiService(): ApiService {
        val retrofit = Retrofit.Builder()
            .baseUrl(BASE_URL)
            .addConverterFactory(GsonConverterFactory.create())
            .build()
        return retrofit.create(ApiService::class.java)
    }
}
```

## Langkah 5: Siapkan Adapter dengan Glide

1. Buat layout item\_user.xml: CardView berisi ImageView (avatar) dan TextView (nama & email).
2. Buat UserAdapter.kt.
  - o Di dalam onBindViewHolder, gunakan **Glide** untuk memuat gambar:

### Kotlin

```
// Di dalam ViewHolder / onBind
Glide.with(holder.itemView.context)
    .load(user.avatar) // URL gambar
    .circleCrop() // Opsional: membuat gambar bulat
    .into(holder.imgAvatar) // Target ImageView

holder.tvName.text = "${user.first_name} ${user.last_name}"
holder.tvEmail.text = user.email
```

## Langkah 6: Panggil API di MainActivity

Hubungkan semuanya di MainActivity.kt.

## Kotlin

```
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response

class MainActivity : AppCompatActivity() {
    // ... inisialisasi RecyclerView & Adapter ...

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

        // Setup RecyclerView...

        // Panggil Fungsi Ambil Data
        getDataFromApi()
    }

    private fun getDataFromApi() {
        val client = ApiConfig.getApiService().getListUsers()

        // Enqueue berjalan secara Asynchronous (Background Thread)
```

```

client.enqueue(object : Callback<UserResponse> {
    override fun onResponse(call: Call<UserResponse>, response:
Response<UserResponse>) {
        if (response.isSuccessful) {
            val responseBody = response.body()
            if (responseBody != null) {
                // Masukkan data ke Adapter
                adapter.setData(responseBody.data)
            }
        } else {
            Log.e("Main", "onResponse: ${response.message()}")
        }
    }

    override fun onFailure(call: Call<UserResponse>, t: Throwable) {
        Log.e("Main", "onFailure: ${t.message}")
        Toast.makeText(this@MainActivity, "Gagal koneksi: ${t.message}",
Toast.LENGTH_SHORT).show()
    }
}
}

```

## E. Latihan Mandiri (Tugas)

Sempurnakan aplikasi di atas dengan fitur berikut:

### 1. Loading Indicator (ProgressBar):

- Saat aplikasi baru dibuka dan data sedang diambil, tampilkan ProgressBar berputar di tengah layar.
- Saat data berhasil didapat (`onResponse`) atau gagal (`onFailure`), sembunyikan ProgressBar tersebut.

### 2. Detail Page:

- Saat salah satu user di list diklik, pindah ke DetailActivity.
- Kirim data (Nama, Email, URL Avatar) menggunakan Intent.
- Tampilkan foto avatar dengan ukuran besar di halaman detail menggunakan Glide.

---

## F. Kriteria Penilaian

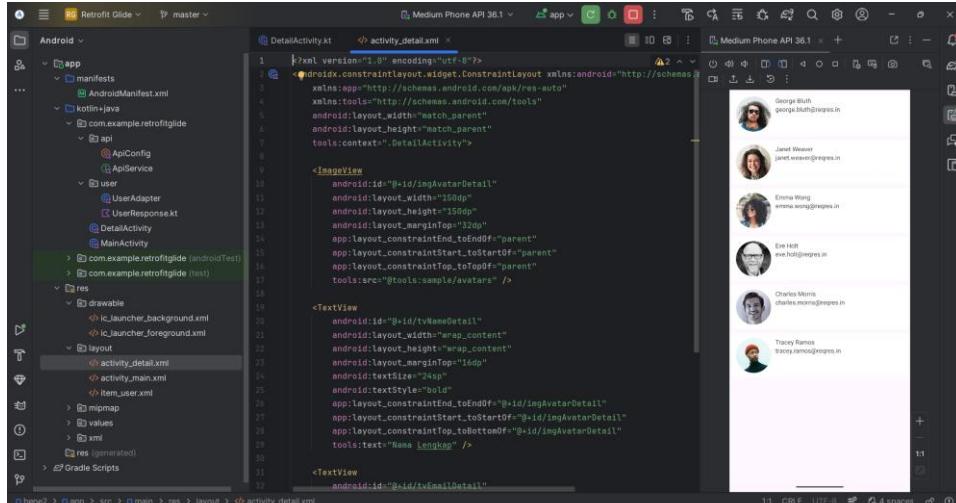
- **Koneksi API:** Aplikasi berhasil menampilkan daftar user dari `reqres.in` (bukan data dummy).
- **Image Loading:** Gambar avatar setiap user berhasil muncul.
- **UX:** ProgressBar berfungsi dengan benar (muncul saat loading, hilang saat selesai).
- **Kode:** Struktur kode rapi (pemisahan Interface, Model, dan Activity).

---

## G. Instruksi Pengumpulan

- *Commit* dan *Push* project ke GitHub.
- Beri pesan commit: `feat: implement retrofit and glide`.
- Lengkapi identitas dalam worksheet ini, kemudian kumpulkan ke Google Form yang disediakan (<https://forms.gle/DrU3UcnjsGksghYh8>).

## Lembar Jawab: Screenshot:



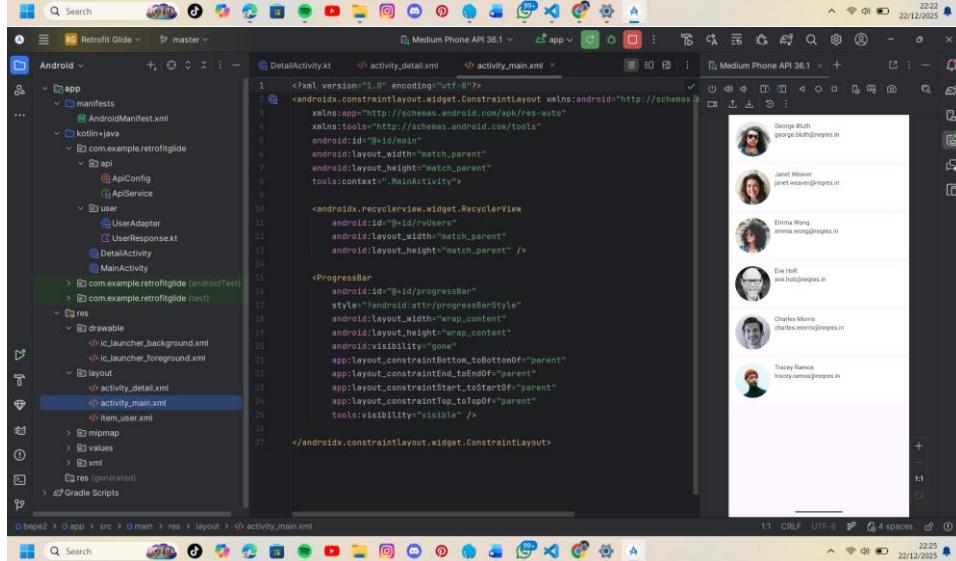
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
   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

    <ImageView
        android:id="@+id/imgAvatarDetail"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:layout_marginTop="12dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:src="@tools:sample/avatars" />

    <TextView
        android:id="@+id/tvNameDetail"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:textSize="24sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="@+id/imgAvatarDetail"
        app:layout_constraintStart_toStartOf="@+id/imgAvatarDetail"
        app:layout_constraintTop_toBottomOf="@+id/imgAvatarDetail"
        tools:text="Nama Lengkap" />

    <TextView
        android:id="@+id/tvEmailDetail"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:textColor="#888888"
        android:textSize="16sp"
        android:textStyle="italic"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvNameDetail" />

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



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

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rvUsers"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <ProgressBar
        android:id="@+id/progressBar"
        style="android:attr/progressBarStyle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:visibility="gone"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:visibility="visible" />

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

**ApiConfig.kt**

```

package com.example.retrofitglide.api

import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory
import java.util.concurrent.TimeUnit

object ApiConfig {
    private const val BASE_URL = "https://raw.githubusercontent.com/girfa/testapi/master"

    fun getApiClient(): ApiService {
        val retrofit = Retrofit.Builder()
            .baseUrl(BASE_URL)
            .addConverterFactory(GsonConverterFactory.create())
            .build()

        return retrofit.create(ApiService::class.java)
    }
}

```

**ApiService.kt**

```

package com.example.retrofitglide.api

import com.example.retrofitglide.user.UserResponse
import retrofit2.Call
import retrofit2.http.GET

interface ApiService {
    @GET("value = \"user.json\") // Endpoint untuk mengambil list user halaman 1
    fun getListUsers(): Call<UserResponse>
}

```

Link GitHub: [https://github.com/myNameRev/Retrofitglide\\_3102.git](https://github.com/myNameRev/Retrofitglide_3102.git)