

LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK



JUDUL :

RETROFIT - GET API

Disusun oleh:

Kopriyanto (20102193)

TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
BANYUMAS, JAWA TENGAH
2023

PPB-20102193-09

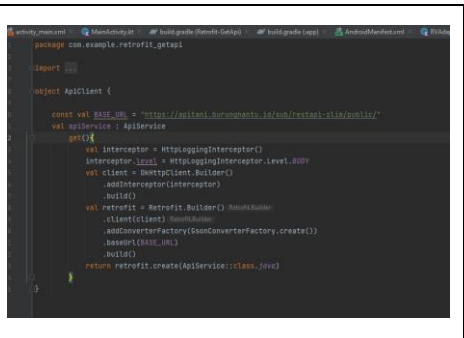
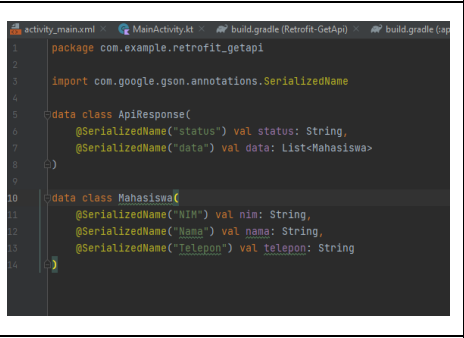
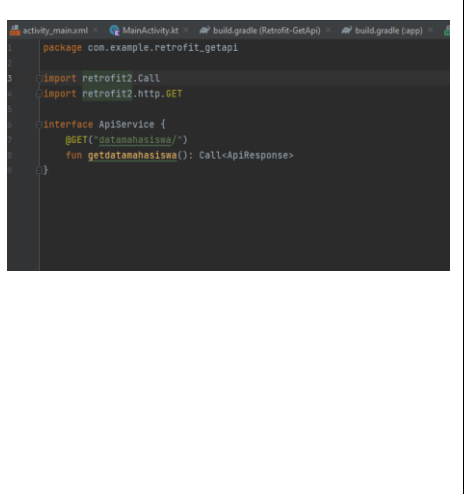
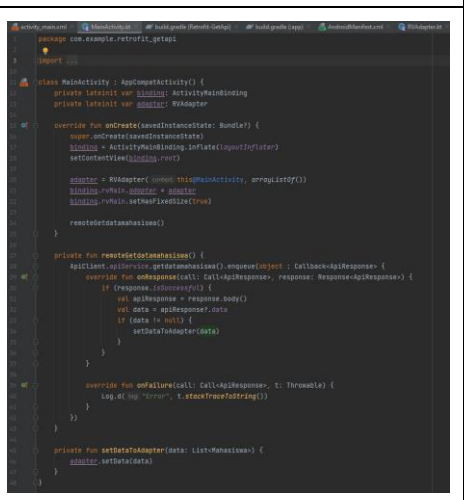
Pembahasan

Pertama pengertian dari API sendiri yaitu apa? API atau kependekan dari Application Programming Interface yang merupakan sebuah protokol yang memungkinkan berbagai software saling terhubung dan berkomunikasi. Dengan API ini akan menyediakan berbagai metode, fungsi dan instruksi yang dapat digunakan oleh developer untuk mengakses layanan tertentu. Bentuk dari API ini bisa berupa antarmuka yang disediakan oleh sistem operasi atau layanan web. Jadi ketika developer mendapatkan API ini developer dapat memiliki akses layanan yang disediakan oleh penyedia API. Ini berguna ketika aplikasi yang dikembangkan perlu adanya integrasi atau fungsionalitas tertentu yang disediakan oleh pihak tertentu. Kemudian apa itu Retrofit retrofit merupakan library yang digunakan dalam pengembangan suatu aplikasi berbasis android dimana dengan library ini aplikasi akan dapat berkomunikasi dengan server. Dengan retrofit ini request akan dilakukan penyederhanaan untuk mengakses endpoint yang disediakan oleh API. Sebagai contoh developer dapat membuat request GET, POST, Query dan masih banyak lainnya. Selain itu retrofit juga menyediakan error handling.

Dengan Retrofit ini akan mudah untuk melakukan komunikasi atau menghubungkan aplikasi dengan API server. Dengan demikian pembuatan aplikasi akan sangat dimudahkan. Maka daripada itu adanya praktikum ini untuk mencoba bagaimana mendapatkan akses API dengan retrofit.

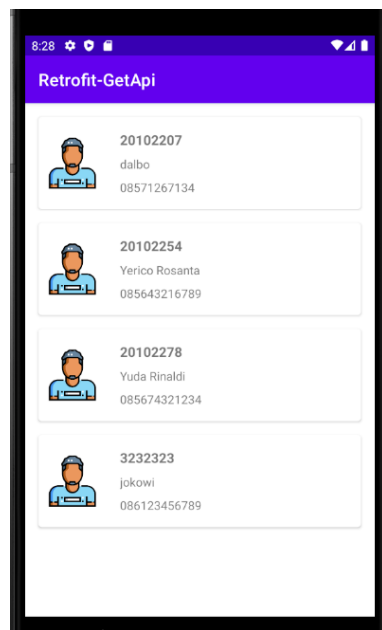
Langkah-Langkah Praktikum

Langkah Praktikum	Pembahasan
<pre>dependencies { implementation 'androidx.core:core-ktx:1.7.0' implementation 'androidx.appcompat:appcompat:1.6.1' implementation 'com.google.android.material:material:1.9.0' implementation 'androidx.constraintlayout:constraintlayout:2.1.4' testImplementation 'junit:junit:4.13.2' androidTestImplementation 'androidx.test.ext:junit:1.1.5' androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1' implementation 'com.squareup.retrofit2:retrofit:2.9.0' implementation 'com.squareup.retrofit2:converter-gson:2.9.0' implementation 'androidx.recyclerview:recyclerview:1.3.0' implementation 'androidx.cardview:cardview:1.0.0' implementation 'com.squareup.okhttp3:logging-interceptor:4.9.3' }</pre>	Buat project baru dengan nama retrofit api kemudian pada file gradle project masukan dependency baru seperti kode disamping.

 <pre> package com.example.retrofit_getapi import retrofit2.Call import retrofit2.http.GET interface ApiService { @GET("datamahasiswa") fun getdatamahasiswa(): Call<ApiResponse> } object ApiClient { const val BASE_URL = "http://pastel.bumiputera.id/sub/restapi/api/public/" val ApiService : ApiService fun get() { val interceptor = HttpLoggingInterceptor() interceptor.setLevel(HttpLoggingInterceptor.Level.BODY) val client = OkHttpClient.Builder() .addInterceptor(interceptor) .build() val retrofit = Retrofit.Builder().baseUrl(BASE_URL) .addConverterFactory(GsonConverterFactory.create()) .build() return retrofit.create(ApiService::class.java) } } </pre>	<p>Pada file APIClient masukan baris code berikut. File ini akan menampung URL dari endpoint API yang sudah disediakan. URL ini merupakan base radi semua API yang akan digunakan.</p>
 <pre> package com.example.retrofit_getapi import com.google.gson.annotations.SerializedName data class ApiResponse(@SerializedName("status") val status: String, @SerializedName("data") val data: List<Mahasiswa>) data class Mahasiswa(@SerializedName("NIM") val nim: String, @SerializedName("Nama") val nama: String, @SerializedName("Telepon") val telepon: String) </pre>	<p>Kemudian pada file response masukan baris code berikut. File ini akan digunakan menampung data yang diambil dari API yang tadi dimasukan.</p>
 <pre> package com.example.retrofit_getapi import retrofit2.Call import retrofit2.http.GET interface ApiService { @GET("datamahasiswa") fun getdatamahasiswa(): Call<ApiResponse> } </pre>	<p>Kemudian pada file service masukan code berikut yang code berikut digunakan untuk menampung request berupa get pada endpoint datamahasiswa dengan base merupakan URL yang sudah dimasukan tadi. Jika base dan endpoint ini dicari maka akan mengembalikan response sebuah json dari data mahasiswa.</p>
 <pre> package com.example.retrofit_getapi import androidx.appcompat.app.AppCompatActivity import android.os.Bundle import android.view.View import android.widget.Toast import androidx.recyclerview.widget.LinearLayoutManager import androidx.recyclerview.widget.RecyclerView import retrofit2.Call import retrofit2.Callback import retrofit2.Response class MainActivity : AppCompatActivity() { private lateinit var recyclerView: RecyclerView private lateinit var adapter: RecyclerView.Adapter<ViewHolder> override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState) setContentView(R.layout.activity_main) recyclerView = findViewById<RecyclerView>() adapter = RecyclerViewAdapter(this@MainActivity, arrayOf()) recyclerView.layoutManager = LinearLayoutManager(this) recyclerView.adapter = adapter recyclerView.scrollToPosition(0) onRequest() } private fun onRequest() { ApiService.getdatamahasiswa().enqueue(object : Callback<ApiResponse> { override fun onResponse(call: Call<ApiResponse>, response: Response<ApiResponse>) { if (response.isSuccessful) { val responseBody = response.body() val data = responseBody?.data if (data != null) { setDataToAdapter(data) } } } override fun onFailure(call: Call<ApiResponse>, t: Throwable) { Log.d("Error", t.stackTraceToString()) } }) } private fun setDataToAdapter(data: List<Mahasiswa>) { adapter.setData(data) } } </pre>	<p>Pada main activity tambahkan code berikut untuk memunculkan data lewat RvAdapter yang kemudian diteruskan ke main sehingga bisa ditampilkan pada halaman utama.</p>

```
package com.example.retrofit retrofit {  
    import androidx.  
  
    class MainActivity {  
        private val context: Context  
        private val dataList: ArrayList<Mahasiswa>  
        = RecyclerView.Adapter<ViewHolder, Mahasiswa>() {  
            class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView)  
            {  
                val title = itemView.findViewById<TextView>(R.id.textView1)  
                val tahun = itemView.findViewById<TextView>(R.id.textView2)  
                val tanggal = itemView.findViewById<TextView>(R.id.textView3)  
                val email = itemView.findViewById<TextView>(R.id.textView4)  
            }  
  
            override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {  
                val viewHolderInflater = LayoutInflater.from(context).inflate(R.layout.item_layout, parent, attachToParent = false)  
                return ViewHolder(viewHolderInflater.inflate(R.layout.item_layout, parent, attachToParent = false))  
            }  
  
            override fun onBindViewHolder(holder: ViewHolder, position: Int) {  
                holder.itemView.title.text = dataList[position].nama  
                holder.itemView.tahun.text = dataList[position].tahun  
                holder.itemView.tanggal.text = dataList[position].tanggal  
                holder.itemView.email.setOnClickListener {  
                    Toast.makeText(context, dataList[position].nama + "Email:" + dataList[position].email, Toast.LENGTH_SHORT).show()  
                }  
            }  
  
            override fun getItemCount(): Int = dataList.size  
  
            fun setData(dataList: ArrayList<Mahasiswa>) {  
                dataList.clear()  
                dataList.addAll(dataList)  
                notifyDataSetChanged()  
            }  
        }  
    }  
}
```

Pada file RV adapter ini file akan mengambil data dari data yang didapatkan melalui APIResponse kemudian akan dimasukan ke recycle view sehingga semua data yang didapat bisa dimuat pada halaman.



Ini merupakan hasil dari aplikasi yang dibuat dengan Retrofit untuk mengambil data dari API dengan menggunakan RecyclerView.