|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Tugas coding Android Studio minggu - 8 Movie Catalog Retrofit + API | 94 |

**Saya Sudah Belajar dan Mengerti dan Saya BISA**

1. Membuat sebuah layout catalog movie di dalam android studio dengan menggunakan statement detail dan search
2. Mmebuat sebuah API yang di gunakan di dalam project android studio movie catalog di dalam website tmdb dengan membuat API terlebih dahulu lalu akan otomatis mmuncul sebuah api dengan beberapa pendukungnya
3. Penggunaan API Key di dalam pembuatan API mmelalui tmdb dan merubahnya untuk mendapatkan key atau sebuah codingan API dari API Key tersebut yang bisa di dapatkan melalui websitenya langsung atau melalui website postman dan merubahnya di dalam website jsonschema2pojo dengan menggunakan format java dan source type menggunakan json dan menggunakan anatation GSON
4. Penggunaan tag public static final String URL\_BASE = "https://api.themoviedb.org"; yang di gunakan untuk mengambil url base atau url asal yang akan di gunakan untuk di ambil data secara keseluruhan untuk di dalam project yang akan di buat
5. Memasukkan coding bagian exam yang di gunbaple java ke dalam file class Response yang telah di buat sebelumnya dengan melakukan beberapa impor tag di dalam file tersebut
6. Melakukan gradle sync terhadap beberapa implement pendukung seperti rettrifator,glid dan sebagainya
7. Memasukkan coding pada bagian result yang telah di rubah sebelumnya dan di masukkan ke dalam file result yang telah di buat sebelumnya
8. Penggunaan tag @GET("/3/movie/{category}") untuk mengambil data dari bagian dari sebuah link yang telah di dapatkkan sebelumnya dengan melihat beberapa category data tersebut
9. Penggunaan tag Call<Response> getMovie( yang di gunakan untuk memanggil data atau sebuah function atau sebuah file Response yang telah di buat sebelumnya untuk di isikan beberapa element yang di butuhkan data untuk di tampilkan
10. Penggunaan tag @Path("category") String category, yang di gunakan untuk mengambil atau menggunakan statement path dengan menggunakan atau membuat sebuah string dengan variable category yang telah di sesuaikan dengan data yang akan di ambil
11. Penggunaan tag @GET("/3/search/movie") yang di gunakan untuk mengambil data secara keseluruhan yang di gunakan untuk di beberapa statement dan mengambil beberapa data dengan menggunakan statement getQuery yang berisi beberapa variable pendukung agar data tersebut dapat di ambil dan di tampilkan kembali
12. Penggunaan tag private static final String BASE\_URL = ServerConfig.API\_ENDPOINT; yang di gunakan untuk mengatur final sebuah variable BASE\_URL dengan file serverconfig yang telah di buat sebelumnya
13. Penggunaan tag private static Retrofit retrofit = null; yang menandakan bahwa sebuah variable tersebut bernilai null
14. Penggunaan tag public static Retrofit getClient(){

if (retrofit == null ){ yang di gunakan untuk apabila variable tersebut bernilai null,maka akan menjalankan sebuah statement dengan tag retrofit = new Retrofit.Builder().baseUrl(BASE\_URL).addConverterFactory(GsonConverterFactory.create()).build(); yang di gunakan untuk membaut sebuah baseurl dengan menggunakan BASE\_URL yang telah di buat sebelumnya dan mengconvertnya ke dalam bentuk Gson dan membuatnya ulang

1. Penggunaan tag public static final String EXTRA\_MOVIE = "string\_extra"; yang di gunakan untuk membuat sebuah String extra yang di gunakan dalam beberapa statement yang akan di buat
2. Penggunaan tag tvTitle = findViewById(R.id.tvJudulDeskripsi); yang di gunakan untuk menemukan komponen dari id komponen tersebut dan di isikan ke dalam sebuah variable yang akan di gunakan di beberapa statement
3. Penggunaan tag result = getIntent().getParcelableExtra(EXTRA\_MOVIE); yang di gunakan untuk mengirim data antar activity secara keseluruhan dan di tampilkan kembali secara keseluruhan data tersebut
4. Penggunaan tag title = result.getO riginalTitle(); yang di gunakan untuk mengambil data result dengan mengambil data Title atau judul yang telah di sediakan oleh Response sebelumnya dan melalui retrofit dan di tampilkan kembali ke dalam project tersebut
5. Penggunaan tag overview = result.getOverview(); untuk mengambil data secara keseluruhan dan di tampilkan kembali
6. Penggunaan tag image = result.getPosterPath(); untuk mengambil data gamnbar yang telah di buat sebelumnya melalui API yang telah di sediakan dan telah di buat sebelumnya dan di masukkan ke dalam file result
7. Penggunaan tag tvTitle.setText(title); yang di gunakan untuk mengatur judul Dari variable yang berisi data yang telah di buat sebelumnya
8. Penggunaan tag Glide.with(getApplicationContext()) yang di gunakan untuk membuat sebuah icon atau gambar yang telah di sediakan oleh website tmdb tersebut
9. Penggunaan tag if(getSupportActionBar() != null){ yang di gunakan apabila sebuah statement tidak sama dengan null,maka akan menjalankan sebuah statement di dalamnya
10. Penggunaan tag getSupportActionBar().setDisplayHomeAsUpEnabled(true); yang di gunakan untuk membuat sebuah function atau statement untuk arah panah balik atau kembali kehalaman utama atau sebelumnya
11. Penggunaan tag private MovieAdapter adapter; yang di gunakan untuk membaut sebuah variable baru dengan type private
12. Penggunaan tag String API\_KEY = "a2945948a3697cef4f68307a28f9a5f5"; yang di gunakan untuk mmengambil data API Key yang telah di buat sebelumnya untuk mengambil data secara keseluruhan
13. Penggunaan tag String LANGUAGE = "en-US"; yang di gunakan untuk mengambil data bahasa dari data keseluruhan yang akan di ambil datanya
14. Penggunaan tag String CATEGORY = "popular"; yang di gunakan untuk mengambil data dengan category popular sesuai dengan pilihan atau data yang ingin di ambil saat pembuatan API tersebut
15. Penggunaan tag ApiInterface apiInterface = ApiClient.getClient().create(ApiInterface.class); yang di gunakan untuk mengambil data dari sebuah file Interface yang di dalamnya terdapat sebuah statement GET API yang telah di bu yang di gunakan utat sebelumnya
16. Penggunaan tag Call<Response> call = apiInterface.getMovie(CATEGORY,API\_KEY,LANGUAGE,PAGE); yang di gunakan untuk mengambil data dari sebuah file Response yang berisi API dengan menggunakan atau memanggil beberapa data sesuai dengan beberapa variable data yang telah di baut sebelumnya dan memasukkannya ke dalam sebuah variable
17. Penggunaan tag call.enqueue(new Callback<Response>() { yang di gunakan untuk menggunakan statement callback atau memanggil kembali sebuah file Response dengan menggunakan statement onResponse
18. Penggunaan tag List<Result> mList = response.body().getResults(); yang di gunakan untuk mengisi sebuah variable tersebut dengan statement function body() dengan mengambil data hasil atau getResults yang terdapat di dalam file response yang di dalamnya terdapat coding API bagian results atau hasil API tersebut
19. Pengggunaan tag adapter = new MovieAdapter(MainActivity.this, mList); yang di gunakan untuk membuat sebuah objek baru dengan mengambil data dari sebuah MainActivity
20. Penggunaan tag ecyclerView.setAdapter(adapter); yang di gunakan untuk memanggil kembali variable adapter yang telah di buat sebelumnya
21. Penggunaan tag public boolean onCreateOptionsMenu(Menu menu) { yang di gunakan untuk menjalankan variable menu yang telah di buat sebelumnya
22. Penggnaan tag searchView = (SearchView) menu.findItem(R.id.search).getActionView(); yang di gunakan untuk membuat statement baru dari sebuah objek atau variable yang telah di buat sebelumnya dengan menggunakan sebuah id komponen tersebut yang di gunakan untuk membuat sebuah statement pencarian dari nama atau judul dari data yang di ambil dari API yang telah di buat sebelumnya di website tmdb
23. Penggunaan tag view = inflater.inflate(R.layout.item\_movie, parent, false); yang di gunakan untuk atau mengisi pada bagian beberapa komponen dengan mengambil id komponen tersebut dengan parent
24. Penggunaan tag viewHolder.relativeLayout.setOnClickListener(view1 -> { yang di gunakan untuk membuat sebuah event OnClickListener atau saat user meng klik sebuah opsi tersebut maka akan menajalankan sebuah statement di dalamnya
25. Penggunaan tag Intent i = new Intent(parent.getContext(), DetailMovieActivity.class); yang di gunakan untuk membuat sebuah intent baru atau proses mengirim data dengan menggunakan Context dari file Detail yang telah di baut sebelumnya
26. Penggunaan tag Result result = new Result(); yang di gunakan untuk membuat sebuah objek baru
27. Penggunaan tag result.setOriginalTitle(resultList.get(viewHolder.getAdapterPosition()).getOriginalTitle()); yang di gunakan untuk men set atau mengatur sebuah variable atau komponen dengan mengambil data Tittle API yang telah di buat sebelumnya di dalam file Result
28. Penggunaan tag yang di gunakan untuk data String yang telah di buat sebelumnya dengan mengambil sebuah data variable atau komponen yang telah di buat sebelumnya
29. Penggunaan tag holder.tvTittle.setText(resultList.get(position).getTitle()); yang di gunakan untuk menset atau mengatur sebuah judul yang di dapatkan dari data API yang telah di buat sebelumnya
30. Penggunaan tag Glide.with(context) yang di gunakan untuk membuat sebauh statement Glid dengan menggunakan variable atau komponen context yang telah di buat sebelumnya
31. Penggunaan tag ImageView imgPoster; yang di gunakan untuk mengambil data atau sebuah komponen yang akan di jalankan di beberapa statement di dalam project tersebut
32. Penggunaan tag imgPoster = itemView.findViewById(R.id.imgMovie); yang di gunakan untuk mengambil data dari id komponen tersebut melalui statement itemView yang telah di buat sebelumnya
33. Penggunaan tag <uses-permission android:name="android.permission.INTERNET"/> yang di gunakan untuk agar project tersebut dapat terhubung ke internet secara public dengan melalui beberapa perizinan dalam mengambil data API secara keseluruhan dengan API yang telah di buat sebelumnya
34. Penggunaan tag .load("https://image.tmdb.org/t/p/w185" + resultList.get(position).getPosterPath()) yang di gunakan untuk mengambil data sebuah gambar dari website tmdb yang telah di sediakan dengan menggunakan statement getPosterPath yang akan di gunakan di beberapa statement