|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Android Studio Minggu ke 8 | 90 |

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

1. **Fragment.** Fragment adalah sebuah bagian dari sebuah dari user interface atau perilaku sebuah aplikasi yang biasanya tempatnya berada di activity yang memungkinkan untuk mendesain activity yang modular, Sebuah fragment tidak dapat berjalan tanpa adanya activity.
2. **Penggunaan Material Design untuk penerapan fragment yang menggunakan bottom navigation.** Caranya, pada File pilih Project Structure, kemudian klik depenency, kemudian pilih library depedency, search untuk material design, lalu pilih com.google.android.material kemudian pilih versi yang terbarunya atau di sesuaikan dengan yang diinginkan, lalu klik ok, Material design yang dibuat tadi akan muncul di dalam Folder Grandle Scripts.
3. **Pembuatan Fragment.** Pada layout ActivityMain pada designnya pilih Frame Layout, kemudian menambahkan icon untuk fragmentnya dengan cara mengeklik new pada folder drawable kemudian pilih gambar icon yang diinginkan untuk fragmentnya nanti, kemudian menambahkan Android Resource Directory dengan resource typenya menu, Kemudian menambahkan di dalam directory tadi dengan Android Directory File dengan nama yang diinginkan. Pada file menu untuk android directory tadi menambahkannya dengan item list, kemudian memasukkannya dengan semua icon yang dipilih tadi. Pada Activity Main menambahkan BottomNatigation.BottomNavigationView, kemudian isikan dengan menu yang sudah dibuat dengan icon - icon yang dipilih tadi. Lalu, otomatis icon - icon yang di buat pada menu tadi akan berbentuk seperti navigasi di bagian bawah layar layout.
4. **Mengisi perintah pada menu - menu fragment agar bekerja sesuai dengan yang diperintahkan.** Pada folder package yang berada di dalam folder java klik kanan, kemudian pilih new fragment, lalu pilih blank fragment, kemudian tambahkan sesuai dengan jumlah menu fragment yang digunakan. Pada masing - masing layout fragment ubah layoutnya menjadi Relative Layout, kemudian menambahkan perintah dengan menambahkan text view agar mengetahui apakah perintah pada fragment tersebut berjalan. Pada Main Activitynya tambahkan implements pada classnya untuk BottomNavigationView.OnNavigationSelectedListener kemudian implement methodnya, pada function OnCreate menambahkan function loadFragment(new HomeFragment()) agar ketika aplikasinya terbuka maka yang pertama tampil adalah HomeFragment atau menu fragment pertama yang ditampilkan, kemudian membuat objek untuk BottomNavigationView navigationView = findViewbyId(R.id.navigation) dengan mencari idnya, kemudian objek dari BottomNavigationView tadi atau navigationView perintahkan untuk men-set atau menjalankan perintah dari menu yang di click atau dipilih, jadi navigationView.SetOnNavigationSelectedListener(this). Kemudian, hasil dari implement method untuk BottomNavigationView tadi isikan dengan objek untuk fragment = null, dan isikan dengan switch case untuk masing - masing dari menu fragment, kemudian return dengan loadFragment(fragment). Kemudian, untuk hasil method dari loadFragment tadi isikan dengan if yang diisikan dengan jika fragmentnya jika tidak di klik/null maka dia menjalankan getSupportFragmentManager().beginTransaction().replace(R.id.fragment\_container, fragment).commit() kemudian return true dan untuk bagian di liar ifnya tambahkan return false agar jika fragmentnya sudah di pilih salah satu maka perintah dari if tadi tidak dijalankan.
5. **Membuat layout catalog untuk retrofit** yang nantinya diisi dengan data yang diambil dari API.
6. **Retrofit.** Sebuah library khusus pemrograman android yang berfungsi untu menghandle yang berkaitan dari android ke internet. Contohnya, merequest data ke server kemudian menghandle apa yang terjadi saat koneksi, bisa juga mengintersect data sebelum di proses android itu sendiri.
7. **Membuat retrofit dengan menghubungkan library yang dibutuhkan ke project yang di buat.** Pada bagian gradlenya tambahkan dengan library yang dibutuhkan untuk membuat retrofit yaitu com.squareup.retrofit:retrofit:2.4.0, com.squareup.retrofit:retrofit2:converter-gson:2.4.0, com.google.code.gson:gson:2.8.5. Dan juga menambahkan library untuk recycler viewnya yaitu androidx.recyclerview:recyclerview:1.1.0.
8. **Penggunaan toRightOf(kanan dari), toLeftOf(kiri dari), toBottomOf(bawah dari), toTopOf(atas dari).** Untuk mengatur layout suatu objek pada program yang digunakan agar bergeser atau berpindah sesuai dengan yang diperintahkan.
9. **Penggunaan maxLines.** Untuk membatasi jumlah dari baris yang digunakan untuk sebuah paragraf.
10. **Menjalankan menu yang ada di dalam folder menu ke dalam Main Activity.** Dengan menambahkan function onCreateOptionsMenu kemudian di dalamnya perintahkan dengan getMenuInflater().inflate(R(lokasi folder menu).menu(nama folder menu).main\_menu(nama file menu yang digunakan), menu), kemudian return super.onCreateOptionsMenu(menu).
11. **Agar icon di dalam menu tertampil.** Dengan menggunakan showAction kemudian pilih collapseActionView dan ifRoom.
12. **Menkonfigurasi data API melalui web themoviedb.org.** Kemudian melakukan pendaftaran untuk memasukkan akun ke dalam themoviedb, lalu menuju ke menu API untuk mengambil API Keynya. Kemudian, menuju documentationnya untuk mengambil data movienya, lalu menuju ke Try it out kemudian pastekan API key yang sudah di copy tadi didalam tabel yang sudah di sediakan, kemudian isikan pathnya dengan kategori film di dalam themoviedb yang mau diambil misalnya popular, lalu send request, lalu copykan responsenya. Karena milik saya error jadi saya harus melalui postman untuk mengambil data filmnya dengan cara yang sama pada themoviedb. Kemudian, karena bentuk dari response datanya adalah JSON maka harus merubahnya menuju Java dengan melalui web jsonschema2pojo.org, kemudian copykan hasil responsenya tadi.
13. **Membuat model untuk data APInya.** Pada android, membuat package baru untuk hasil copyannya tadi dengan nama model dan isikan dengan JavaClass dengan nama Response kemudian pastekan hasilnya tadi yang untuk class response lalu tambahkan lagi untuk JavaClassnya dengan nama Result untuk mempaste hasil responsenya tadi untuk class result.
14. **Membuat folder untuk konfigurasinya.** Pada Android, membuat folder baru dengan nama cofig, kemudian isikan dengan JavaClass dengan nama ServerConfig agar tersambung dengan server APInya. Pada ServerConfignya, isikan dengan URL dari web themoviedb maka, mengisikannya dengan public static final String URL\_BASE = “https://www.themoviedb.org/”. Tetapi, karena saya menggunakan apinya maka urlnya diganti dengan “https://api.themoviedb.org” supaya tidak terjadi error. Kemudian mengubah URL\_BASE menjadi API\_ENDPOINT maka, public static final String API\_ENDPOINT = URL\_BASE.
15. **Membuat folder untuk clientnya.** Pada Android, membuat folder baru dengan nama rest, lalu menambahkannya dengan JavaClass dengan nama ApiClient untuk memproses konfigurasinya. Pada ApiClient, isikan dengan BASE\_URL yang sudah diisikan pada confignya tadi maka, private static final String BASE\_URL = ServerConfig.API\_ENDPOINT. Kemudian isikan retrofitnya dengan null maka, private state Retrofit retrofit = null. Kemudian membuat function getClient() untuk mengisikan jika retrofitnya dalam keadaan null maka menjalankan apa saja maka, public static Retrofit getClient(){if(retrofit == null){retrofit = new Retrofit.Builder().baseUrl(BASE\_URL).addConverterFactory(GsonConverterFactory.create()).build()} return retrofit}. Lalu membuat file lagi untuk interfacenya, kemudian mengisikan dengan data - data yang ada di dalam web themoviedbnya yang mau diambil maka, @GET(“/3/movie{category}/”) Call<Response> getMovie(@Path(“Category”) String category, @Query(“api\_key”) String api\_key, @Query(“language”) String language, @Query(“page”) int page).
16. **Membuat folder untuk adapternya.** Pada Android, membuat folder baru dengan nama adapter, lalu menambahkannya dengan JavaClass dengan nama MovieAdapter. Pada class MovieAdapternya extends dengan RecyclerView.Adapter<MovieAdapter.MyViewHolder> kemudian jika merah maka klik alt+enter kemudian implement method agar dibuatkan function - functionnya. Kemudian di dalam class tersebut melakukan pendeklarasian agar dikenali oleh java dengan membuat objeknya maka, private Context context, lalu private List<Result> resultList, kemudian membuat constructornya untuk context dan list tadi dengan cara klik kanan kemudian Generate lalu pilih context dan list tadi lalu klik ok. Kemudian pada method untuk onCreateViewHolder yang dibuat tadi untuk MovieAdapter.MyViewHolder isikan dengan View view, LayoutInflater inflater = layoutInflater.from(parent.getContext()), view = inflater.inflate(R.layout.item\_movie, arent,false), kemudian dengan Movie.MyViewHolder viewHolder = new MoviewAdapter.MyViewHolder(view) kemudian return viewHolder. Kemudian pada method MyViewHolder extends RecyclerView.ViewHolder diisikan dengan objek pada layout yang akan diisikan dengan data Api maka ubah ke dalam objek jadi Imageview imgPoster dan TextView tvTittle, tvDescription. Kemudian pada function untuk MyViewHoldernya saja diisikan dengan mencari id dari masing masing objek tadi maka, imgPoster = itemView.findViewById(R.id.imgMovie) bagitu juga yang lainnya. Kemudian, pada function getItemCount diisikan dengan return resutList.seize(). Lalu, pada function onBindViewHolder diisikan dengan holder.tvTittle.setText(resultList.get(position).getTittle()), holder.tvDescription.setText((resultList.get(position).getTittle()) dan Glide.with(context).load(“https://image.tmdp.org/t/p/w185”+resultList.get(position).getPosterPath()).into(holder.imgPoster) diisikan sesuai dengan objek yang ditambahkan untuk Glide ditambahkan terlebih dahulu pada gradlenya dengan implementasinya yaitu implementation ‘com.github.bumptech.glide:glide:4.11.0’.
17. **Mendeklarasikan semua ke dalam MainActivitynya.** Pada MainActivitynya, tambahkan private MovieAdapter adabpter, private SearchView searchView, String API\_KEY = “(mengisikan dengan api key yang kita punya)”, String LANGUAGE = “(bahasa yang digunakan)”, String CATEGORY = “(kategori yang diambil)”, int page = (jumlah halamannya). Kemudian pada function onCreatenya tambahkan dengan recyclerView = findViewById(R.id.rvMovie(id dari recycler layout main activity)), recyclerView.setHasFixedSize(true), recyclerView.setLayoutManager(new LinearLayoutManager(this)) kemudian CallRetrofit(). Kemudian buat function untuk CallRetrofitnya dan isikan dengan ApiInterface apiInterface = ApiClient.getClient.create(ApiInterface.class), Call<Response(mengambil yang ada di model yang sudah dibuat tadi)> call=apiInterface.getMoview(CATEGORI,API\_KEY,LANGUAGE,PAGE), call.enqueue(new Callback<Response>(){}) Maka otomatis akan terbuat function didalamnya dengan nama onResponse dan onFailure, pada function onResponsenya diisikan dengan List<Result> mList=response.body().getResults(), adapter=new MovieAdapter(MainActivity.this,mList), recyclerView.setAdapter(adapter).
18. **Membuat fitur search** dalam aplikasi agar jika kita mengetik suatu kata maka film yang berkaitan dengan kata tersebut akan muncul. Pada ApiInterfacenya, membuat query baru dengan mengambil data yang ada di dalam web themoviedbnya pada bagian documentation pilih yang search kemudian klik search movie, kemudian pilih try it out kemudian ambil bagian urlnya yaitu 3/search/movie, kemudian letakkan di dalam quesrynya maka, @GET(“/3/search/movie”) lalu, Call<Response> getQuery, lallu di dalamnya diisikan dengan @Query(“api\_key”) String api\_key, @Query(“language”) String language, @Query(“query”) String query, @Query(“page”) int page. Kemudian, pada Main Activitynya pada function onCreateOptionsMenu, tambahkan dengan searchView = (SerachView(memilih yang berada di androidx)) menu.findItem(R.id.search).getActionView(), kemdian tambahkan searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener) maka, otomatis akan dibuatkan dua function oleh anroid yaitu onQueryTextSubmit dan onQueryTextChanges. Kemudian pada function onQueryTextChanges tambahkan dengan if yang menjalankan perintah jika textnya panjangnya lebih dari satu(newText.length()>1) maka akan menjalankan ApiInterface apiInterface = ApiClient.getClient().create(ApiInterface.class), Call<Response> call = apiInterface.getQuery(API\_KEY,LANGUAGE,newtext,PAGE), kemudian call.enqueue(new Callback<Response>()) akan otomatis dibuatkan dua function oleh android yaitu function onResponse dan onFailure kemudian pada function onResponse diisi dengan List<Result> mList=response.body().getResults(), adapter=new MovieAdapter(MainActivity.this,mList), recyclerView.setAdapter(adapter). Kemudian return true.
19. **Memasukan data ke halaman detail film.** Pada Activity untuk halaman detail movienya atau DetailMovieActivitynya, menambahakannya dengan mendeklarasikan hal hal yang perlu diketahui oleh java nantinya, untuk extra movienya yaitu dengan public static final String EXTRA\_MOVIE = “extra\_movie”, kemudian String title, overview, image, lalu deklarasikan objek layout yang digunakan ImageView imgDetail, TextView tvTitle, tvDetail, kemudian Result result. Kemudian pada function onCreate menambahkan dengan mencari id untuk setiap objek yang digunakan yaitu dengan tvTitle = findViewById(R.id.tvDetailTitle) begitu juga untuk objek yang lainnya, kemudian result=getIntent().getParselableExtra(EXTRA\_MOVIE), lalu isikan untuk String yang sudah di deklarasikan tadi dengan title=result.getOroginalTitle(), overview=result.getOverview(), image=result.getPosterPath(), lalu tambahkan untuk objek yang digunakan dengan string yang sudah diisi tadi maka, tvTitle.setText(title), tvDetail.setText(overview) lalu untuk bagian gambar menambahkannya dengan menggunakan Glide.with(getApplicationContext()).load(“https://image.tmdb.org/t/p/w185”(mengcopy dari movie adapter) + image).into(imgDetail). Pada Movie Adapternya, bagian function untuk MyViewHoldernya menambahkannya dengan mencari id untuk relative layoutnya yaitu relativeLayout = itemview.findViewById(R.id.layoutMovie). Kemudian pada bagian function untuk Movie Adapternya menambahkannya dengan viewHolder.relativeLayout.setOnClickListener(new view.OnClickListener) maka otomatis akan muncul function untuk onClicknya, pada function onClicknya menambahkan degan Intent intent = new Intent(parent,getCOntext(), DetailMovieActivity(activiy untuk halaman detailnya).class), Result result = new Result(), result.setOriginalTittle(resultList.get(viewHolder.getAdapterPosition.getOriginalTitle())) begitu juga untuk bagian overview dan untuk image, lalu intent.putExtra(DetailMovieActivity.EXTRA\_MOVIE, result), parent.getContext().startActivity(intent). Kemudian implementasikan Resultnya ke dalam Parcelable.
20. **Membuat tombol back untuk kembali kehalaman utama jika sedang menjalankan halaman lain.** Pada DetailMovieActivitynya membuat if untuk menjalankan support action bar ketika keadaannya adalah null maka if(getSupportActionBar()!=null){getSupportActionBar().setTitle(title) getSuportActionBar().setDisplayHomeAsUpEnabled(true)}. Kemudian membuat function untuk onSupportNavigation dan mengisikannya dengan perintah finish().

**Saya Belum Mengerti**

1.

2.

3.