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

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

1. Fragment, merupakan sub bagian (suatu bagian yang terpisah) yang menempel dengan MainActivity (semacam sub-activity). Ditempatkan di bagian activity, serta digunakan untuk membuat sesuatu yang seperti activity, namun dapat berubah-ubah sesuai dengan tempat da nisi dari masing-masing fragmentnya.

2. Bisa membuat project baru sesuai dengan yang ada di video (beserta pengaturan dari activity yang mana (empty activity)) dengan menggunakan (mengklik) create new project.

3. Library material, digunakan untuk memanggil button navigation (karena dibutuhkan nantinya dalam pembuatan project fragment). Menggantinya dengan cara mengklik File > Project Structure > Dependencies > lalu pada All Dependencies memilih tombol dengan logo plus (+) > kemudian klik dan pilih Library Dependency > setelah itu mencari kata kunci material, yang nantinya akan memunculkan group ID dengan berisikan nama com.google.android.material > lalu memilih versi (pada versions) yang paling stabil atau paling atas (terbaru) dari android studionya (saya memilih 1.5.0-alpha02) > setelah itu klik ok > kemudian ok kembali.

4. Untuk mengecek material dari yang sudah ditetapkan di dalam Project Structure, bisa di cek di dalam gradlenya (pada bagian build.gradle (Module Fragment.app di bagian bawah sendiri (dari kelompok dependencies))).

5. Bisa membuat dan mengganti suatu komponen beserta dengan margin, padding, id, dan attributes lainnya langsung di dalam codenya.

6. Frame Layout, mirip seperti constraint layout, namun lebih langsung ada di dalamnya (terkotak layaknya frame).

7. Match\_parent, digunakan untuk menjadikan ukuran lebar dan tinggi suatu komponen menjadi full (penuh) satu layer (layar).

8. Wrap\_content, digunakan untuk menjadikan ukuran lebar dan tinggi suatu komponen mengikuti isi dari yang ada di dalamnya.

9. Constraint left to left of, untuk menjadikan suatu komponen memiliki margin ke sebelah kiri (lebih kuat dari margin dan dapat langung membuat suatu komponen tersebut mengarah ke sebelah kiri).

10. Constraint top to top of, untuk menjadikan suatu komponen memiliki margin ke bagian atas (lebih kuat dari margin dan dapat langung membuat suatu komponen tersebut mengarah pada bagian atas).

11. Pada saat pemberian satuan, pada ukuran bagian (seperti margin) diberikan satuan dp. Tetapi bila pada saat penggantian ukuran dari suatu tulisan atau textnya, menggunakan satuan sp.

12. Vector Asset, berisi berbagai macam logo yang dapat digunakan untuk menunjang kebutuhan pada saat pembuatan project (seperti gambar kepala dari logo android (bawaan dari android studionya (yang bernamakan ic\_launcher\_background.xml) dan gambar-gambar lainnya (seperti gambar home pada video). Diharuskan memberi nama untuk Vector Assetnya dengan huruf kecil dan tanpa spasi (jarak).

13. Bisa mengganti warna dari Vector Asset dengan cara menambahkan warna pada file dari Vector Asset yang ingin diubah yang ada pada folder drawable (bagian code android:fillColor=”….” (titik-titik tersebut berisikan nama maupun code dari warnanya)). Warna bisa dipilih dari file colors, menambahkan warnanya sendiri (langsung memilih), maupun dengan memasukkan heksadesimal dari warna tersebut. (Namun pada Android Studio saya, saya perlu menghilangkan (menghapus) code android:tint=”?attr/colorControlNormal” agar warna yang telah di setting untuk perubahan warna dapat berubah sesuai dengan warna yang sudah dipilihkan tersebut. Bila tidak maka warnanya akan tetap hitam seperti awal mula bawaan dari saat pertama kali dipilih (Vector Assetnya)). Warna tersebut juga bisa diganti langsung pada saat awal pemilihan vector asset beserta pemberian namanya (seperti yang dilakukan pada video).

14. Bisa membuat Android Resource Directory (folder untuk xml) baru dengan cara klik kanan dari folder layout > pilih Android Resource Directory > lalu mengubah Resource type (tipe untuk foldernya) nya sesuai dengan yang akan digunakan (menggunakan menu seperti yang ada di video) > setelah itu klik ok.

15. Bisa membuat file dengan tipe xml baru dengan cara mengklik kanan foldernya peletakan file xml nya (pada video seperti menu) > pilih New > …. Resource File (titik-titik tersebut bernamakan sesuai dengan tipe dari suatu foldernya tersebut (pada video bertipen menu, maka menjadi Menu Resource File)) > setelah itu memberikan nama dari file baru dari xml nya tersebut > klik ok.

16. Item (pada saat pembuatan komponen langsung lewat codenya), mirip seperti menu option (isi pilihan dari titik tiga, karena bentuknya berupa kotakan persegi panjang yang panjang dari ukurannya setengah lebar dari layar dan berkotakan kecil (namun juga tidak terlalu kecil).

17. Bisa menggunakan vector asset untuk logo-logo kecil yang bisa di klik dan menuju ke setiap file nya masing-masing (jadi isi di dalamnya berbeda-beda) pada pembuatan navigasi yang diletakkan di bagian bawah (mirip seperti navigasi dari youtube).

18. Mengklik alt+enter pada bagian code yang dirasa ada yang kurang lengkap maupun error (ada tanda garis berwarna merah di bawahnya). Pada saat di klik akan memunculkan logo seperti bohlam (yang keluar pada saat biasanya di onClick bila baris pada onClick nya di klik). Tidak hanya pada saat error saja logo bohlam tersebut akan keluar, bila digunakan untuk penambahan sesuatu (yang bersifat warning, seperti penambahan @string pada video fragment) maka bohlam tersebut akan berwarna kuning seperti lampu. Namun bila digunakan pada saat ada yang error, maka bohlam tersebut akan berwarna merah.

19. Logo seperti bohlam (keluar pada saat di suatu dari codenya di klik alt+enter), akan memberikan solusi dari penyelesaian masalah bilama ada errornya dan ada kurangnya, yang bila setelah sesuatu dari pilihan bohlam tersebut di klik, maka akan otomatis membuatkan sesuatu yang kurang tersebut.

20. Com.google.android.material, digunakan untuk memunculkan navigasi dari bottom\_menu (yang mirip seperti navigasi youtube). Mengeluarkannya dengan menambahkan app:menu=”@menu/bottom\_menu” di dalamnya.

21. Constraint bottom to bottom of, untuk menjadikan suatu komponen memiliki margin ke bagian bawah (lebih kuat dari margin dan dapat langung membuat suatu komponen tersebut mengarah di bagian bawah (seperti bottom\_menu untuk navigasi)).

22. Constraint right to right of, untuk menjadikan suatu komponen memiliki margin ke sebelah kanan (lebih kuat dari margin dan dapat langung membuat suatu komponen tersebut mengarah ke sebelah kanan). Seperti pada saat pembuatan text JUDUL MOVIE diberikan ini, karena text tersebut menumpuki komponen yang lain (perlu di geser ke sebelah kanan agar tidak menumpuki komponen yang lainnya).

23. Menambahkan (membuat) fragment dengan mengklik kanan folder package yang pertama (com.yapi.belajarfragment)> New > Fragment > lalu pilih Fragment(Blank) > kemudian setelah mengisikan nama dari file fragmentnya klik finish. Di dalam fragment tersebut sudah terdapat beberapa code yang sudah otomatis langsung dibuatkan oleh android studionya. Fragment ini nantinya terletak di dalam folder com yang sama dengan MainActivity.java. (Pada video membuat 4 (empat) fragment sesuai dengan jumlah vectornya).

24. Relative Layout, layout yang bebas dan tidak terdapat batasan di dalamnya (misal langsung tertata dengan rapi, dan lain sebagainya).

25. Text Size, untuk mengubah ukuran dari suatu teks sesuai dengan yang diinginkan (belakang dari angkanya harus diberi sp (satuannya) untuk mengubah ukuran teksnya, misal seperti 20sp).

26. Android:gravity, untuk membuat letak sebuah komponen menjadi di mana (misal di tengah (center) seperti yang ada pada video).

27. Extends, menjadikan suatu file bisa tergabung dengan file yang lainnya. Dengan extends tersebut menjadikan satu file dengan file lainnya dapat terhubung dan memanggil apa yang dibutuhkan file tersebut yang ada pada file lainnya.

28. LoadFragment(), sama seperti load() yang biasanya digunakan, hanya saja berbeda nama. Digunakan untuk menampilkan suatu code yang telah dibuat agar dapat tampil atau muncul, harus dimasukkan ke dalam function onCreate bawaan dari MainActivity.java dari android studionya (di bawahnya, bukan langsung di dalamnya, namun masih satu tutup kurung dengan onCreate).

29. ButtomNavigationView, untuk digunakan meletakkan tempat dari buttom navigation tersebut akan di tampilkan di mana.

30. FindViewById, untuk menempatkan suatu komponen yang sudah memiliki id masing-masing (dipanggilnya menggunakan ini). (Penggunaannya seperti findViewById(R.id.tvTitle); (R untuk res (resource), id untuk menunjukkan bahwa akan memanggil id, dan tvTitle merupakan id dari suatu komponen yang telah diubah sebelumnya (id dari text view judul dari movienya (yang depan)))).

31. This, untuk menunjukkan bahwa suatu code akan digunakan atau diletakkan disini (dipanggil di tempat ini).

32. Set, untuk meletakkan atau mengisi suatu hasil.

33. Get, untuk mendapatkan hasil dari suatu hasil yang telah diambil atau diisi oleh set sebelumnya.

34. Switch, untuk memberikan beberapa atau hingga banyaknya pilihan yang dapat dijalankan. Yang dijalankan hanya salah satu dari yang dipanggil dari pilihannya (yang sama juga pilihannya dari pemanggilan ditiap case dari switchnya). Meskipun pilihan di dalamnya banyak (casenya banyak), namun bila tidak dipanggil, maka tidak akan dijalankan. Serta pada setiap akhir dari case diberi break untuk jedanya.

35. Case, merupakan tiap-tiap pilihan yang ada di dalam switch (isi dari switch), bisa menghasilkan sesuatu yang berbeda pada satu case dan case yang lainnya.

36. Break, merupakan jeda pada tiap akhir dari case (harus diberikan ini), untuk menghentikkan masing-masing isi dari tiap case yang ada.

37. Menambahkan if dibagian loadFragment pada saat suatu fragment tidak sama dengan null, maka akan mengisikan atau memunculkan sesuatu.

38. Pada saat penulisan tanda seru dan sama dengan yang setelahnya terdapat kata null, maka tanda seru dan sama dengan tersebut tidak boleh diberikan jarak (spasi), harus digabung dikarenakan akan membuat project tersebut tidak dapat berjalan dengan semestinya. ( if(fragment ! = null \***tidak boleh**), if(fragment != null **\*boleh dan dianjurkan (harus)**)).

39. Bisa membuat fragment sesuai dengan yang ada di video, serta membetulkan beberapa code agar project tersebut dapat dijalankan.

40. Bisa membuat movie catalog dengan retrofit dan API sesuai dengan yang ada di video, serta membenarkan beberapa code yang membuat project atau aplikasi tersebut tidak dapat dijalankan.

41. Retrofit, merupakan library di pemrograman android. Dapat digunakan untuk menghandle semua hal yang berkaitan dengan koneksi data dari android ke internet.

42. Untuk menambahkan retrofit, perlu menambahkan beberapa implementation di dalam module dari gradlenya (gradle nomer 2 setelah yang pertama (project)). (Saya sudah menambahkannya sesuai dengan yang ada pada video. Namun pada implementation ‘com.google.code.gson:gson:2.8.5’, saya perlu menghilangkan angka :2.8.5 di dalamnya, karena pada android studio saya akan terjadi error dan tidak dapat di jalankan (run app) bilamana menuliskannya sama dengan yang ada pada video (bagian :2.8.5 nya)). Setelah menambahkan atau membuat beberapa perubahan (sekecil apapun) di dalam gradle, perlu mengklik Sync now (harus) dibagian atas yang nantinya baru akan muncul bilamana sudah membuat suatu perubahan.

43. ShowAsAction (dengan ifRoom), untuk menyembunyikan item (yang seperti pilihan dari menu option) agar tidak terlihat di tampilan depannya (namun sebenarnya masih ada (tidak benar-benar hilang)).

44. Menggunakan androidx (pilihan dari suatu codenya) untuk xml dan bukan android saja (android.widget). Karena nanti akan berbeda kegunaannya, yang bilamana seharusnya menggunakan androidx namun bila diberi yang pilihan android, maka nanti akan terjadi error dan aplikasi tersebut tidak dapat dijalankan (seperti pada searchview dan android:actionViewClass).

45. Recycler View, untuk mengisikan banyak data di dalamnya (langsung terdapat tulisan-tulisan dari bawaan Android Studionya (seperti text 0, text 1, dan masih banyak lagi lainnya)).

46. ImageView, digunakan untuk menampilkan suatu gambar (nantinya, seperti gambar poster film pada video).

47. Text Color, digunakan untuk memberikan warna pada suatu text atau tulisan.

48. Android:layout\_below, digunakan untuk menjadikan suatu komponen berada di bawah suatu komponen lainnya. Seperti pada saat text view (komponen) INI DESKRIPSI MOVIE.

49. Android:maxLines, digunakan untuk membatasi sesuatu dari isi komponen yang dapat diatur menjadi berapa saja yang tampil untuk maksimalnya (tampil di layar depan pada saat aplikasi di jalankan). Seperti deskripsi dari suatu movie yang hanya bertuliskan empat (4) baris saja pada awalnya, yang bila setelah di klik, maka baru akan memunculkan keseluruhan isi dari deskripsi movie tersebut.

50. Tools:listitem, digunakan untuk memanggil item yang sudah dibuat sebelumnya dari file xml lain. Seperti isi yang sudah dibuat sebelumnya di dalam item\_movie.xml dapat tampil setelah dituliskan menggunakan code ini di dalam xml dari activity\_main.xml.

51. Bisa membuat activity baru dengan cara mengklik kanan folder dari com.yapi.catalogmovie > pilih New > Activity > lalu pilih yang Empty Activity > setelah itu mengatur dari mulai nama activity tersebut apa, layoutnya (sudah otomatis tidak perlu diganti karena mengikuti dari nama activity nya), package namenya, serta source language nya > setelah selesai mengaturnya, klik finish.

52. Bisa juga membuat suatu komponen dari designya (menariknya (drag) dan menautkan handlenya satu persatu).

53. Text Style, untuk menjadikan suatu teks menjadi lebih tebal (bold), miring (italic), maupun normal (pada kotakan disebelah dari tulisan pilihannya ini harus di klik, untuk memakai style tersebut).

54. OnCreateOptionMenu, salah satu code yang berfungsi untuk dapat menjalankan dari pencarian suatu flim (search).

55. Run app (berlogo segitiga warna hijau menghadap ke kanan di bagian atas (di sebelah tampilan sambungan device)), untuk menjalankan/memulai code yang sudah dibuat sebelumnya (memutar dan menunjukkan hasilnya di device yang sudah ditujukan (misal di nox, maka akan langsung muncul di layar nox tersebut (bila success/berhasil) namun bila gagal/error, maka tidak akan muncul di tampilan layar nox dan menampilkan bagian errornya yang mana dan apa)).

56. Stop (berlogo warna merah persegi/kotak di bagian atas (dekat dengan run app)), untuk memberhentikan aplikasi/project yang sedang dijalankan.

57. Bisa membuat akun di themoviedb.org yang kegunaannya untuk mendapatkan API Key (v3 auth), serta mengambil beberapa link yang digunakan agar dapat menjalankan aplikasi dari moviecatalog yang dibuat.

58. Pada saat send request untuk mendapatkan data film dari json di themoviedb.org nya, saya menggunakan postman web untuk mendapatkan data-data tersebut (datanya sama dan tetap menggunakan api key yang sudah didapatkan dari web themoviedb.org). Dikarenakan pada saat saya coba mendapatkan data jsonnya langsung di dalam web themoviedb.org, terjadi suatu error yang membuat data-data tersebut tidak dapat muncul (mungkin dikarenakan adblocker dari web browser saya atau hal lainnya yang menyebabkan data tersebut tidak dapat muncul).

59. Jsonschema2pojo (dari web browser, bukan suatu aplikasi), digunakan untuk mengubah suatu code yang awalnya bertipekan json lalu dapat diubah menjadi tipe Gson. Pada tipe Gson tersebut codenya menjadi sudah lebih rapi (seperti beberapa code ada yang sudah tersedia public void). Tidak lupa untuk mengubah source type agar bertipen json (sesuai dengan tipe code yang dimiliki). (Code yang sudah tipenya tersebutlah yang nantinya akan digunakan dan di copy kan di dalam android studio (dimasukkan di dalam file java Response dan Result)).

60. Bisa membuat package baru dengan cara mengklik kanan folder dari com.yapi.catalogmovie > pilih New > Package > kemudian memberikan nama dari package tersebut > klik enter.

61. Bisa memindahkan (cut) file java (activity) dari satu file ke file lainnya (seperti dengan yang ada di video).

62. Bisa membuat config berserta isinya sesuai dengan yang ada pada video. Config tersebut digunakan untuk meletakkan url dari data-data yang akan digunakan.

63. Bisa membuat interface (tipenya bukan yang java namun interface (namun masih java nantinya dari ekstensinya) pada saat pembuatan barunya) beserta isinya sesuai dengan yang ada pada video. Interface tersebut digunakan untuk memasukkan dan memanggilkan data dari tiap-tiap hal yang dibutuhkan untuk katalog dari filmnya (seperti category, api\_key, language, newText dan page).

64. Bisa membuat adapter beserta isinya sesuai dengan yang ada pada video. Adapter ini digunakan untuk menyambungkan satu file activity dengan file activity lainnya (mirip seperti penggerak dari file-file tersebut, karena beberpa file ber-extendkan adapter tersebut).

65. ViewHolder, digunakan untuk memunculkan sesuatu dan diketikkan di dalam adapter.

66. Tidak lupa untuk mendeklarasikan suatu variabel dari apa yang nanti digunakan di dalam function. Tipe dari variabel tersebut juga harus diperhatikan, karena apabila nantinya berbeda dengan yang akan digunakan di dalam code functionnya, maka nanti akan terjadi error dan aplikasi tersebut tidak dapat dijalankan.

67. Menambahkan glide di dalam dependencies dari gradle (letaknya sama dengan implenetasi dari retorift, pada gradle nomer dua (2) dari atas yang bagian module). Ditambahkan karena beberapa code di dalam adapter movie nya menggunakan glide, bila tidak ditambahkan di dalam gradlenya, maka code tersebut tidak akan dapat digunakan.

68. Glide, digunakan untuk memindahkan dan meletakkan dari suatu hal diarahkan ke mana (seperti poster untuk filmnya).

69. CallRetrofit(), digunakan untuk memanggil retrofit dari yang sudah dibuat sebelumnya.

70. Create, digunakan untuk membuat sesuatu seperti (ApiInterface.class).

71. Pada saat pembuatan public void dari CallRetrofit(), response yang ada di dalam ApiInterface.java harus disamakan agar pada saat pembuatan di MainActivitynya dapat terhubung dan code call.enqueue(new Callback<Response>()) dapat menyala dan langsung otomatis membuatkan @Overide dan public void dari onResponse dan onFailure. Karena bila tidak terhubung dan codenya tidak dapat menyala, meskipun dengan membuat codenya sendiri pun maka code nya tidak akan dapat dijalankan (harus dari android studionya sudah dibuatkan otomatis, karena sudah dianggap terhubung).

72. Mengubah link pada String URL\_BASE yang ada di dalam ServerConfig.java (“https://www.themovie.org”) menjadi (“https://api.themovie.org”). Dikarenakan mengambil data dari movie (film) nya berada di dalam API nya, bukan dari webnya (meskipun linknya berasal dari web, namun di dalam android studio datanya sudah diletakkan di dalam API yang sudah dibuat sebelumnya).

73. List<Result>, digunakan untuk mendapatkan hasil list dari mList.

74. GetResults(), digunakan untuk mendapatkan hasil isi dari file result yang sebelumnya sudah diberikan isi data dari filmnya (di dalam file result bagian mainactivity nya (class java)).

75. Bisa membuat fitur search sesuai dengan ada pada video (sudah bisa digunakan untuk mencari judul film yang diinginkan juga). Search tersebut digunakan untuk mencari nama (judul) film yang sedang dicari dengan menuliskan huruf inisial dari nama (judul) maupun langsung berupa nama (judul) dari film tersebut. Setelah dicari, maka akan menemukan beberapa (judul) film yang mirip, hampir mirip, maupun (judul) film yang memang sedang dicari tersebut dengan nama (judul) film yang tadinya sedang ingin dicari.

76. String, untuk code yang berisikan teks atau kalimat menggunakan ini dan bukannya integer (int) maupun double.

77. Pada saat pembuatan query (di dalam ApiInterface.java), bila di bawahnya masih ada penambahan query lagi, maka perlu diberikan tanda koma (,) di bagian akhir dari code per barisnya. Bilamana di bawahnya sudah tidak ada lagi penambahan untuk querynya, maka tidak perlu diberikan tanda apapun, baik dari tanda koma (,) maupun petik koma (;). Karena bila diberikan tanda apada bagian akhir dari penambahan query tersebut, maka akan terjadi error. Untuk tanda petik koma (;) nya diberikan pada tutup kurung bagian akhir setelah penambahan query yang terakhir (tutup kurung dari Call<Response> nya).

78. Membuat fitur dari search, codenya (searchView) diletakkan di dalam MainActivity.java pada bagian function dari public Boolean onCreateOptionMenu (Menu menu).

79. Menggunakan if pada saat menentukan harus diberi minimal berapa huruf di dalam pencarian (search) dari nama (judul) film yang nantinya akan dicari.

80. Length(), digunakan untuk membatasi harus berapa jumlah dari sesuatu hal bila ingin melakukan sesuatu hal (seperti pada saat fitur search, menggunakan batasan jumlah dari huruf yang digunakan untuk mencari nama (judul) dari film tersebut ialah harus lebih dari satu (1), (if(newText.length() > 1))).

81. Mengopy code dari public void onResponse yang ada di dalam public void dari CallRetrofit() untuk digunakan di dalam public void onResponse yang ada di dalam public Boolean onQueryTextChange (di dalam public void dari public Boolean onCreateOptionMenu). Dicopykan karena memerlukan isi yang sama seperti onResponse yang sebelumnya telah dibuat (pada CallRetrofit).

82. Mengganti return false yang terdapat di dalam public Boolean onQueryTextChange menjadi return true (tidak sama seperti yang ada pada CallRetrofit).

83. Bisa memasukkan poster film, nama (judul) film, serta deskripsi dari filmnya ke dalam recycler view yang telah dibuat sebelumnya di dalam xml activity\_detail\_movie.xml (sesuai dengan yang ada di video).

84. GetParcelableExtra, digunakan pada saat membutuhkan hasil dari EXTRA\_MOVIE.

85. Set Text, untuk mendapatkan teks atau sesuatu yang telah diisikan ke dalamnya dan menampilkannya.

86. OnCreate (function), digunakan untuk membuat sesuatu yang dapat diisikan banyak code yang sesuai dengan apa yang akan dibuat.

87. GetOriginalTitle, digunakan untuk mendapatkan nama dari judul film masing-masing (judulnya akan berubah-ubah sesuai dengan judul film apa saja yang terdapat di dalamnya).

88. GetOverview, digunakan untuk (mendapatkan) menampilkan suatu teks yang dimaksud disini ialah deskripsi dari film tersebut (deskripsinya akan berubah-ubah sesuai dengan deskripsi dari masing-masing setiap film).

89. GetPosterPath, digunakan untuk mengambil link tempat dari poster itu berada (yang nantinya akan dimunculkan di dalam ImageView yang sudah dibuatkan sebelumnya). (Posternya (gambar filmnya) akan berubah-ubah sesuai dengan filmnya).

90. SetOnClickListener, menjadikan sesuatu (misalkan button) pada saat di klik akan memunculkan sesuatu (hampir mirip dengan onClick).

91. OnClick, akan menampilkan suatu kejadian dengan di klik terlebih dahulu (komponennya di klik terlebih dahulu, lalu baru akan menampilkan apa).

92. Parent.getContext.startActivity(intent);, digunakan pada saat pergantian code yang awalnya sudh banyak intent dan akan (menuju) di mulai (aplikasi tersebut).

93. Membenarkan beberapa ukuran, baik dari ukuran poster maupun besar teks dari deskripsinya di dalam activity\_detail\_movie.xml setelah sudah di coba run (dijalankan) aplikasinya.

94. Menambahkan button back (kembali) yang digunakan untuk kembali bila pada saat tampilan dari layar depan sudah mengklik suatu film (yang jadi lebih mendetail). Menambahkan codenya di dalam DetailMovieActivity.java (getSupportActionBar().setDisplayHomeAsUpEnabled(true), yang lalu setelahnya menambahkan onSupportNavigateUp() (otomatis akan membuat public void beserta @Overide sendiri bila memilih codenya yang benar) dan memberikan finish() diantara kedua code ini (yang bawahnya ada returnya dari super.onSupportNavigateUp). Serta sebelumnya memberikan if dan diisikan code, bahwa tombol (button) dari back (kembali) tersebut berada di bagian atas dan ada di sebelah dari title atau judul filmnya (lebih tepatnya sebelah kirinya))). (setDisplayHomeAsUpEnabled(true) nya lah yang membuat button dari back (kembali) tersebut).

**Saya Belum Mengerti**

1. Masih sedikit bingung dengan fungsi dari code parent bagian pada xml maupun javanya.

2. Masih sedikit bingung dengan relative layout.

3. Sedikit bingung bila membuat suatu codenya langsung banyak (pada video kali ini mohon maaf beberapa penjelasannya bagi saya kurang dapat saya pahami).