|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Android Studio Minggu-6 | 80 |

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

1. Bisa membuat dan memahami fungsi layout untuk SQLDatabase yang telah dibuatkan tabelnya terlebih dahulu (sudah dibuat sebelumnya) sesuai dengan yang ada di video.

2. Menyetting (mengatur) agar ukuran dari margin awal (margin startnya) bila diberikan suatu komponen akan langsung berada di ukuran margin tersebut (seperti 16dp pada video). Mengaturnya pada logo bertuliskan angka dengan dp dan ada garis bawah (mirip kotakan seperti wadah (angkanya diwadahi)) dan bertempat di dekat logo bergambar mata (untuk mengatur tampilan design dari xml nya).

3. Handle,untuk mengatur pegangan yang terdapat 4 (empat) sisi bagian (atas, kanan, bawah, dan kiri). Mengaturnya bisa dengan menarikannya (menautkan) ke sisi yang akan diberikan pegangan atau bisa juga dengan di klik tombol berlogo tanda panah dengan border bulat biru pada Layout (Constraint Widget) di bagian Attributes. Komponen diharuskan diberi handle, karena bila tidak, maka komponen akan menuju ujung pojok kiri atas bila saat di run (dijalankan) atau bisa juga di cek sudah terakitkan atau belum dengan megeceknya di toggle (logo seperti kunci inggris).

4. Bisa mengatur margin (untuk memberikan jarak pada komponen yang ada di dalam layer (baik pada layout\_margin, pada jarak handle, maupun pada komponennya) pada komponen apapun sesuai dengan yang ada pada video.

5. Plain text,suatu komponen kosong yang bisa diisikan kata, kalimat, angka, dan yang lainnya (tidak seperti Number (Decimal) yang hanya dapat digunakan untuk mengisikan angka di dalamnya).

6. Number (Decimal),komponen yang berupa sebuah kotakan yang mengisikan suatu angka (bisa angka biasa (bulat, misal seperti 1) maupun desimal (seperti 1.5)). Pengisiannya hanya dikhususkan untuk angka. (Memakai ini dikarenakan pada tabel stok dan harga bertipekan real).

7. Wrap Content, ukuran dari suatu komponen yang mengikuti isi yang ada di dalamnya (misalkan bertambah panjang tulisannya, maka kotakan komponennya akan ikut melebar sesuai dengan isinya). Menggunakannya dengan cara tidak perlu merubah garis apapun yang ada di tengah pengaturan handle yang terdapat di dalam Constraint Widget.

8. Fixed, ukuran dari suatu komponen yang tidak mengikuti isi yang ada di dalamnya (misalkan bertambah panjang tulisannya, maka kotakan komponennya tidak akan ikut melebar (akan menetap) berapapun isi yang ada di dalamnya). Menggunakannya dengan cara mengklik sekali garis yang ada di tengah pengaturan handle yang terdapat di dalam Constraint Widget.

9. Match Constraint, ukuran dari suatu komponen yang tidak mengikuti isi yang ada di dalamnya, namun kotakannya mengikuti lebar dari tampilan layarnya, berapapun isi yang di dalamnya tidak akan memengaruhi lebar kotakan. Menggunakannya dengan cara mengklik dua kali garis yang ada di tengah pengaturan handle yang terdapat di dalam Constraint Widget.

10. Button, suatu komponen berbentuk kotakan (lebih ke arah persegi panjang). (Bilamana ingin button ini dapat menjalankan sesuatu pada saat di klik, menambahkan onClick di dalamnya (pada attributes)).

11. Memberikan id pada komponen yang dapat digunakan untuk pemanggilan saat membuat suatu project atau program nantinya (sesuai dengan yang ada pada video).

12. Text Alignment, untuk mengatur tata letak dari suatu text (seperti center pada video).

13. Hint, (pada attributes), untuk membuat tulisan seperti dibelakang layar dari kotakannya (jadi pada saat kotakannya diisi, maka tulisan tersebut akan hilang dan tergantikan oleh tulisan yang diisikan di kotakan yang telah diberi hint tadi. Bila tidak ada tulisannya, maka yang muncul adalah tulisan yang sudah dibuat di hint tadi pada kotakannya.

14. ScrollView, untuk menjadikan layar (layer) bisa di scroll ke atas dan ke bawah. Di dalam dari komponen ScrollView tersebut sudah otomatis berisikan LinearLayout di dalamnya (jadi komponen yang dimasukkan akan teratur dalam LinearLayout dari ScrollView tersebut). Ukuran layar dari ScrollView sendiri dapat diatur ingin menjadi berapa. Jadi bila ada beberapa isi di dalamnya dan ukuran layer dari ScrollViewnya tidak memuat semua isinya, maka bisa diatur sesuai dengan keinginan, mau yang dimunculkan awalnya ada berapa dan isi selanjutnya (yang lainnya) dapat dilihat dan diisikan (bilamana komponennya suatu teks) bila di scroll ke bawah. Dan bila ingin melihat komponen yang sudah hilang atau tertutup karena sudah di scroll ke bawah, maka bisa di scroll ke atas kembali untuk menampilkan dan melihatnya lagi. (Pada project kali ini menghapuskan LinearLayout yang ada di dalamnya (bawaannya) sesuai dengan yang ada di video).

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

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

17. Bisa membuat xml baru dengan cara klik folder res, kemudian pilih folder layout, lalu klik kanan>New>Layout Resource File. Setelah itu akan muncul tampilan untuk mengatur res baru tersebut (seperti file xml tersebut akan diberi nama apa (misal seperti item\_barang pada video) dan yang lainnya), kemudian klik ok.

18. Menambahkan tools:listitem=”@layout/item\_barang” pada activity\_main.xml agar dapat terkaitkan dari design yang sudah dibuat di dalam item\_barang sebelumnya (nanti masuknya pada RecyclerView yang dibagian bawah (bawahnya button)).

19. ConstraintLayout, seperti tempat atau wadah untuk komponen yang akan dimasukkan setelahnya. Ukurannya bisa diubah-ubah dan dapat diatur sesuai dengan yang diinginkan. Bila diukur lebih kecil (dikeataskan), maka otomatis komponen yang ada di dalamnya juga akan langsung mengikuti ukuran dari ConstraintLayout tersebut (akan tetapi harus dikaitkan handlenya terlebih dahulu (komponennya) agar dapat mengikuti ukuran dari ConstraintLayoutnya).

20. &#8942;, merupakan kode yang akan menghasilkan titik tiga seperti pada menu option pada umunya. Menuliskannya di komponen text view dan mengubah textnya menjadi kode tersebut. Serta menambahkan setOnClickListener juga.

21. Layout\_width, digunakan untuk mengubah lebar suatu komponen pada Android Studio (bagian design dari layernya, misal digunakan seperti pada ukuran lebar dari komponen ScrollView). Belakangnya ditambahkan dp untuk satuannya.

22. Layout\_height, digunakan untuk mengubah tinggi suatu komponen pada Android Studio (bagian design dari layernya, misal digunakan seperti pada ukuran tinggi dari komponen ScrollView). Belakangnya ditambahkan dp untuk satuannya.

23. Bisa menampilkan design yang sudah dibuat sebelumnya di dalam item\_barang.xml dan muncul di RecyclerView (bagian bawah setelah button) dari activity\_main.xml sesuai dengan yang ada pada video (terdapat tiga (3) text view disebelah kanan dan satu (1) text view berada di sebelah kiri (untuk menu optionnya (yang diubah menjadi titik tiga), serta pada saat tampil di RecyclerView pada activity\_main.xml sudah menjadi beberapa bagian ke bawah (menjadi banyak dan sama persis (seperti di copy paste)))).

24. 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)).

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

26. Bilamana membuat suatu function untuk menjalankan sesuatu, tidak lupa untuk mendeklarasikannya terlebih dahulu, termasuk variabelnya.

27. FindViewById untuk menempatkan suatu komponen yang sudah memiliki id masing-masing (dipanggilnya menggunakan ini). (Penggunaannya seperti findViewById(R.id.etBarang); (R untuk res (resource), id untuk menunjukkan bahwa akan memanggil id, dan etBarang merupakan id dari suatu komponen yang telah diubah sebelumnya (id dari kotakan))).

28. Load(), bila setelah membuat code, maka hasil yang akan ditampilkannya di lewatkan function load() dan agar dapat digunakan, 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). Jika tidak diamsukkan onCreate maka tidak akan muncul design yang telah dibuat di item\_siswa.xml ke Recycler Viewnya.

29. OnClick, akan menampilkan suatu kejadian dengan di klik terlebih dahulu (komponennya di klik terlebih dahulu, lalu akan menampilkan apa). Pada awal setelah menambahkannya, maka akan terjadi error, membenarkannya dengan mengklik logo seperti bohlam lampu namun berwarna merah di samping kiri dari baris yang error (onClick), lalu kemudian klik Create ‘simpan(View)’ in ‘MainActivity’ (pada error onClick di button simpan, bila error pada button lain atau attribute dari komponen yang lain, maka namanya akan mengikuti dari id (yang dituliskan di onClicknya, jadi bisa berbeda dengan id dari komponen tersebut) komponen tersebut (bukan lagi simpan)). Setelah itu, maka akan terbuat void atau method (public void btnsimpan (View view) {}) di main activity. Bisa juga mengetikkannya manual di classnya terlebih dahulu secara manual (public void btnsimpan (View v) {}). Hasilnya akan sama dan dapat digunakan (tidak error).

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

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

32. Get Text, untuk mendapatkan hasil atau sesuatu yang telah diisikan ke dalamnya dan menampilkannya.

33. Menggunakan if dan else, serta mengisikan beberapa code di dalamnya agar pada saat kotakan dari salah satunya ada yang kosong (baik dari barang, stok, maupun harga). Bila ada yang kosong salah satunya, maka pada saat di klik (button simpan) akan menghilangkan seluruh isi dari yang sudah diisikan (seperti bila pada saat nama barang sudah diisikan, namun pada kotakan stok dan harga masih kosong, maka pada saat button simpan di klik, nama barang yang sudah diisikan sebelumnya akan menghilang (akan kembali kosong seperti sedia kala) dan akan bertuliskan Data Kosong).

34. Sebaliknya, bila seleuruh kotakan sudah diisikan sesuai dengan isinya (baik dari kotakan barang, stok, maupun harga) maka akan bertuliskan insert berhasil (serta pada code selanjutnya akan dapat memasukkan data yang sudah diisikan ini akan tampil ke dalam SQLite Reader (aplikasi di android (bisa di jalankan atau diunduh lewat emulator juga) dan akan dapat muncul di RecyclerView pada bagian bawah buttonnya)).

35. Toast, untuk menampilkan suatu pesan saat berada di layar tampilan (menuliskan kata atau kalimat di toastnya diletakkan di bagian text (seperti pada video yang menampilkan “Data Kosong” atau “insert berhasil”, setelah button yang sudah diberikan ini di klik)). (Bisa mengisikan toast dengan otomatis (saat setelah di klik tabnya maka akan mengeluarkan isi dari toastnya) dengan mengklik tab pada keyboard).

36. Pada saat pembuatan toast di public void pesan(), dibagian petik dua dari toastnya (yang biasanya diisikan nama dari text untuk toastnya) dihapuskan dan diganti dengan kata isi. Maka nanti pada saat pemanggilan (toastnya diperlukan untuk memunculkan dialog pesan dibagian bawah), maka akan bertuliskan isi: “….” Bukan text:”….” (titik-titik tersebut berisikan suatu dialog pesan yang akan muncul nantinya sesuai dengan keinginan (seperti Data Kosong dan Insert pada video)).

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

38. IsEmpty(), untuk menunjukkan bahwa suatu hal telah kosong (dan dapat diberikan kelanjutan code seperti misalkan yang diberikan ini masih kosong akan menjalankan apa dan bila sudah terisi akan menjalankan apa).

39. Or (||), untuk menggandeng suatu hasl seperti variabel yang kan diberikan suatu keterangan bersama-sama nantinya (hanya yang ditautkan dan diberikan or (||) tersebut).

40. Insert, untuk menambahkan atau memasukkan suatu data ke dalam kolom yang telah dibuat sebelumnya (isiannya).

41. Diusahakan untuk mencoba sebuah code atau syntaxnya terlebih dahulu (dari execute) yang nantinya akan digunakan di android studio (bila berkaitan dengan SQLite), diharuskan untuk mencoba codenya terlebih dahulu apa sudah berhasil maupun error agar sukses berjalan nanti pada saat dimasukkan dan digunakan di android studionya.

42. Memasukkan barang, stok, maupun harga pada saat di androis studio harus mengganti codenya agar tidak statis (menetap itu saja dan agar dapat berubah-ubah sesuai dengan apa yang akan diisikan di dalam kotakan dari komponen yang telah dibuat sebelumnya) dengan ‘”+barang+”’ (untuk nama barang), “+stok+” (untuk jumlah stoknya), dan “+harga+” (untuk jumlah harga per barangnya). Diharuskan untuk nama barang harus menggunakan (ditambahkan) petik satu (‘ ’) karena nama barang betipekan char atau text (untuk stok dan harga tidak perlu dikarenakan mereka bertipe real).

43. Menambahkan db.runSQL(sql); agar apa yang sudah dibuat (insert) pada saat menjalankan (run) di android studionya (menambahkan isi di dalam kotakan nama barang, stok, dan harga) dapat masuk dan tampil di aplikasi dari SQLite Readernya.

44. Membuat if dan else kembali untuk pengecekan apabila ada insert (memasukkan data dari kotakan nama barang, stok, dan harga) yang gagal, maka akan diberitahukan dan bisa diulang kembali untuk membuatnya agar bisa sampai berhasil.

45. Bisa menampilkan code execute (seperti INSERT INTO tblbarang (barang,stok,harga) VALUES (‘Flashdisk’,4,7000)) dengan code pesan(sql);. (Untuk nama barang, stok, dan harga tersebut dapat berubah-ubah sesuai dengan apa yang sudah diisikan sebelumnya di dalam kotakan nama barang, stok, maupun harga (mengikuti apa yang sudah diisikan di dalam android studio yang sudah di jalankan app nya (run))).

46. Select, untuk melihat hasil dari isian yang telah dibuat sebelumnya dengan menggunakan Insert. (SELECT \* from tblbarang), maka akan memunculkan tabel yang sudah dibuatkan kolom sebelumnya beserta isian yang telah dibuat sebelumnya dengan menggunakan insert (isiannya sudah masuk sesuai dengan kolomnya masing-masing (untuk barang maka ada di bawah kolom barang, dan seterusnya)). Bisa menampilkannya hanya salah satu atau beberapa saja dari kolom yang ada (misal ingin menampilkan barang dan stoknya saja, maka menuliskan SELECT barang, stok from tblbarang). Karena bila menggunakan bintang (\*) maka akan menampilkan keseluruhan dari kolom pada tabelnya (bila ingin menampilkan semuanya menggunakan \*, tidak perlu disebutkan satu persatu).

47. Mengetikkan code boolean runSQL (String sql) di Database.java (android studionya) beserta mengisikan try, catch, dan return di dalamnya agar mencegah tidak terjadinya force close secara tiba-tiba (sesuai dengan yang ada di video).

48. Execute SQL (pada SQLite DB Browser), untuk membuat code syntax baik dari insert (memasukkan), select (menampilkan), delete (menghapus), dan lainnya yang dapat dipergunakan. Dan menjalankan code yang telah dibuat di dalamnya ini dengan mengklik seperti tombol run app di android studio (hasilnya ada di bawahnya dan langsung otomatis berbentuk tabel).

49. Browse Data (pada SQLite DB Browser), untuk menampilkan hasil dari apa yang sudah dibuatkan (syntax codenya dari execute sql) dan data yang sudah dibuat akan diisikan ke dalam tabel (tampilannya berupa tabel sesuai dengan kolom dari yang sudah dibuat sebelumnya).

50. Order by, digunakan untuk mengurutkan suatu data berdasarkan sesuatu yang sudah ditentukan nantinya (misal ORDER BY barang (maka nanti hasilnya akan mengurutkan data yang ada di dalam tabel yang sudah diisikan beberapa data akan terurut berdasarkan alfabetnya (mulai dari a sampai dengan z) dikarenakan berdasarkan barangnya (nama barangnya))).

51. Asc, untuk mengurutkan suatu data mulai dari yang terkecil hingga yang terbesar.

52. Desc, untuk mengurutkan suatu data mulai dari yang terbesar hingga yang terkecil.

53. Bisa membuat MainActivity.java (class java baru) dengan mengklik folder com.yapi.recyclerviewcardview (yang bagian java nya (dalam folder java)) lalu klik kanan, pilih New dan klik Java Class (logo dengan huruf beralfabet C dengan lingkaran biru). Setelah itu mengisikan nama dari class baru tersebut beserta mengecek class yang akan dibuat tadi apa sudah berbahasa Java (dibagian bawah pengisian nama (Name) terdapat Kind untuk memilih bahasa dari class yang akan dibuat tersebut)).

54. Constructor, untuk mengambilkan data dari variabel yang telah dibuat sebelumnya (data dari Getter dan Setter, pada saat pembuatan suatu objek atau program). Membuatnya dengan cara klik kanan, kemudian pilih Generate, lalu pilih Constructor dibagian paling atas, serta pada saat sebelum mengklik tombol ok, pastikan sudah mengklik dan memilih semua String yang dibutuhkan (pada video yaitu string untuk nama dan alamat).

55. Getter and Setter, untuk mengambil dan mendapatkan suatu data sekaligus pada saat suatu class akan memanggil sesuatu yang sudah diisikan ini dari class yang lain. (Pada video, masing-masing variabel mendapatkannya (baik dari alamat maupun namanya)). Membuatnya dengan cara klik kanan, kemudian pilih Generate, lalu pilih Getter and Setter tepat di bawah Setter, serta pada saat sebelum mengklik tombol ok, pastikan sudah mengklik dan memilih semua String yang dibutuhkan dengan mengklik salah satunya dan menahan tombol shift yang kemudian mengklik juga salah satu dari String yang belum terpilih tadi, baru kemudian dapat mengklik ok (pada video yaitu string untuk nama dan alamat).

56. Bisa membuat dan memahami kegunaan adapter (BarangAdapter.java) untuk program atau project dari SQLite Database sesuai dengan yang ada di video.

57. Extends, untuk menautkan suatu class pada sesuatu hal atau class yang lainnya (yang bermaksud bahwa class ini terkaitkan dengan sesuatu yang lain agar dapat dijalankan).

58. Menambahkan code RecyclerView.Adapter<BarangAdapter.ViewHolder> pada public class bagian paling atas yang ada di dalam adapter (setelah extends) agar dapat tersambung ke dalam recycler viewnya (dapat tampil di dalamnya juga (dari barang.java serta item\_barang.xml)). Serta pada saat masih error, mengklikkan logo seperti bohlam lampu namun berwarna merah dengan tanda seru di dalamnya dan memilih Implement methods, lalu pada saat ada pilihan dari apa saja yang akan diikut sertakan, maka memilih semua (ketiganya) dengan mengklik salah satu data yang berlogokan pink dengan tulisan m di dalamnya, kemudian menahan tombol shift serta mengklik data yang lainnya (hingga ketiganya terpilih), baru kemudian klik ok. Errornya pun akan hilang, serta akan otomatis langsung dibuatkan beberapa code sesuai dengan apa yang nantinya akan digunakan (seharusnya sudah ada).

59. Menambahkan, mengganti (mengubah), bahkan menghapus beberapa code yang sudah dibuatkan sebelumnya dengan implement methods sesuai dengan yang ada pada video agar dapat dijalankan code dari adapter tersebut.

60. OnCreateViewHolder, untuk menjadikan semua design yang ada di dalam item\_siswa.xml bisa masuk ke dalam recycler view yang telah dibuat sebelumnya (bisa menyatu (mengisikan) dengan list yang banyak dari recycler view). Menjadikan keseluruhan dari isi yang ada di dalam item\_siswa.xml masuk semua ke dalam recycler view dibarengi dengan inflate.

61. OnBindViewHolder, untuk mengisikkan suatu data satu persatu (dari banyaknya data yang ada) dan nantinya dapat mengambil datanya dari sini (pada saat di klik dan memunculkan nama serta alamatnya pada bagian bawah (karena menggunakan toast)).

62. GetItemCount, untuk mengambil sejumlah data yang dibutuhkan. (Bersamaan dengan return yang diketikkan code dengan size(). Size tersebut untuk mengetahui berapa banyak jumlah data yang dapat ia ambil nantinya).

63. Context, merupakan pengganti nama dari suatu activity, untuk menjadikan baik dari form maupun activity dapat menggunakan sesuatu code dari adapter yang telah dibuat (jadi tidak dikhususkan, karena tidak bisa tahu hal apa yang akan membutuhkan suatu code tersebut nantinya).

64. List, untuk menandakan suatu data yang jumlahnya banyak.

65. LayoutInflater, agar code yang telah dibuat di dalam adapter serta barang dapat terkaitkan dan tampil di dalam RecyclerView dari MainActivity.java (di design activity\_main.xml).

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

67. Array List, untuk data yang banyak namun sudah beurutan sesuai dengan urutan pembuatannya.

68. SetLayoutManager, (yang selanjutnya menambahkan (new LinearLayoutManager)), agar data yang diisikan nantinya bisa langsung tertata rapi (karena adanya LinearLayout).

69. SetHasFixedSize, untuk menetapkan recycler viewnya agar menetap (fix).

70. GetCount, untuk menentukan suatu hal harus mengambil sesuatu bila angka yang sudah di berikan (menuliskan angkanya setelah ini) apakah lebih besar atau lebih kecil (bilamana lebih besar akan mendapatkan apa dan juga bila lebih kecil akan mendapatkan apa (akan ada yang dihasilkan nantinya)).

71. Menggunakan if dan else serta beberapa code lainnya di dalam public void selectData() yang nantinya digunakan untuk menampilkan suatu data dari yang sudah dibuat sebelumnya (dengan insert) di dalam recycler view di dalam design dari activity\_main.xml (di bawah buttin simpan) sesuai dengan yang ada pada video.

72. While, digunakan untuk melooping suatu data yang banyak (diletakkan di dalam if dan else pada selectData() (lebih tepatnya pada if nya)).

73. MoveToNext, untuk memasukkan satu persatu dari banyaknya data yang sudah dibuat sebelumnya.

74. Databarang.add(), untuk memasukkan data sebelumnya yang telah dibuat di MainActivity.java ke dalam modelnya.

75. NotifyDataSetChanged, untuk otomatis langsung merefresh sesuatu bila sudah terdapat perubahan di dalamnya.

76. Bisa menampilkan data di dalam recycler view dari design activity\_main.xml, baik dari nama barang, stok, dan harga nya sesuai dengan design yang sudah dibuat sebelumnya (yang terdapat titik tiga di sebelah kanannya). (Datanya sudah sama seperti yang ada di dalam tabel SQLite Reader (aplikasi)).

77. Bisa membuat dan menggunakan PopUp Menu sesuai dengan yang ada pada video.

78. Popup Menu, digunakan pada kode titik tiga agar dapat berinteraksi dengan titik tiga tersebut serta dibarengi dengan onClick untuk membuatnya dapat menjalankan atau mengeluarkan sesuatu (misal seperti UBAH dan HAPUS seperti pilihan dari menu option yang ada pada video).

79. Bisa membuat folder sendiri di dalam res (seperti folder bawaan dari android studio (seperti layout)) dengan cara mengklik folder res yang kemudian mengklik kanan, lalu memilih New > dan klik Android Resource Directory. Setelah di klik, maka akan mengisikan nama dari foldernya serta memilih tipe resource dari folder tersebut (mirip seperti pengisian saat pembuatan class), lalu klik ok. (Pada video diberi nama menu\_item untuk foldernya dan tipe dari resourcenya yang awalnya values diubah menjadi menu).

80. Menu Item, merupakan sebuah komponen yang mirip seperti text view namun ukurannya sudah paten tidak terlalu besar (pada video diletakkan di pojok bagian kanan atas).

81. Title (mirip seperti text pada attributes), untuk mengganti judul atau nama text sesuai dengan yang diinginkan di dalam menu item (seperti ubah dan hapus pada video).

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

83. Show Display UI, untuk menampilkan tampilan layer pada saat pembuatan dengan design jadi lebih mirip dengan aslinya (sama dengan pada saat di run (dijalankan) di emulatornya). Logonya seperti mata dan letaknya ada dibawah dari pemilihan layer design dan blueprint (yang logonya seperti lapisan kertas yang bertumpuk).

84. Show(), untuk menampilkan sesuatu hal (misal isi pilihan yang terdapat di dalam menu option (titik tiga)).

85. Bisa membuat popup menu dengan menu itemnya dapat dijalankan dan dipergunakan sesuai dengn kegunaannya. Seperti titik tiga yang dapat di klik dan memunculkan menu item ubah dan hapus (sudah di setting sebelumnya). Serta pada tiap menu item dapat berjalan sesuai dengan kegunaannya (namun untuk ubah belum bisa berjalan dengan semestinya (sesuai dengan kegunaannya), (baru hanya mengeluarkan toast bertuliskan ubah bila di klik) dikarenakan belum dibuat untuk codenya). Untuk menu item hapus dapat digunakan untuk menghapus data yang tampil di dalam recycler view pada design activity\_main.xml dari yang sudah dibuat sebelumnya (yang dibuat dimasukkan melalui kotakan nama barang, stok, dan harga).

86. SetOnMenuItemClickListener, untuk membuat pilihan dari isi yang ada di dalam menu option (titik tiga) bisa menjalankan sesuatu bila salah satu pilihannya di klik, yang akan diisikan ke dalamnya (misal diberikan toast, diberikan perintah apa, dan yang lainnya. Seperti pada saat di klik ubah baru hanya mengeluarkan toast berisikan tulisan ubah dan pada saat mengklik hapus, maka akan menghapus data yang ada di dalam recycler view dari activity\_main.xml (yang sudah dibuat sebelumnya di kotakan atasnya (nama barang, stok, dan harga)).

87. 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. Seperti misalkan switch case untuk ubah yang hanya menampilkan toast berisikan tulisan ubah dan pada switch case hapus akan menghapus data yang tampil di recyclew view pada design activity\_main.xml dan memunculkan tulisan Data Sudah dihapus.

88. 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.

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

90. Bisa membuat, menggunakan, dan memahami SQL Delete sesuai dengan yang ada di video.

91. Delete, untuk menghapus suatu id dari isi yang ada di dalam tabel kolom yang sudah dibuat sebelumnya.

92. Mengetikkan DELETE from tblbarang WHERE idbarang = 1, maka id yang dituliskan akan hilang terhapus (hilang dari isi kolom maupun di recycler view di android studio (sudah dibuatkan codenya pada menu item)). Pada android studio, mengganti dan menambahkan codenya dengan DELETE FROM tblbarang WHERE idbarang = “+idbarang+” (hanya diberikan petik dua double (“ ”) dikarenakan idbarang bertipekan integer (real juga hanya diberikan petik dua double saja (“ ”) tanpa petik satu mengapit diantaranya (seperti ‘”+barang+”’))). Dituliskan idbarang dan bukan langsung id dari suatu barangnya agar dinamis mengikuti dari idbarang dari setiap barang yang sudah dihapuskan.

93. Where, untuk menunjukkan apa yang akan diubah, dihapus, ataupun yang lainnya bila ada.

94. Bisa mengopykan code yang ada di website untuk belajar (misal seperti w3schools.com) dan diletakkan di dalam kotakan untuk membuat code (pada SQLite DB Browser). Setelah di paste (disalin), maka tinggal mengganti beberapa halnya untuk diisikan ke dalam tabel kolom yang sudah dibuat sebelumnya (sama seperti pada saat membuatnya dengan manual (mengetikkannya langsung tanpa mengopy atau mengambil dari referensi (seperti web yang dibuat untuk belajar tersebut)). Seperti code yang awalnya INSERT INTO table\_name (column1, column2, column3, …), yang tinggal mengganti table\_name menjadi nama tabel (dari tabel yang telah dibuat sebelumnya (tblbarang) serta pada column1, cloumn2, column3, … diganti menjadi nama kolom yang ada di dalam tabel barang yang telah dibuat sebelumnya (seperti barang, stok, harga) dan kemudian menambahkan VALUES untuk isi dari masing-masing kolom tersebut (disesuaikan tempatnya dengan penempatan dari kolomnya))).

95. Menambahkan if dan else lagi pada saat menunjukkan apakah datanya sudah bisa terhapus atau belum. Bila sudah bisa terhapus, maka akan memunculkan tulisan Data Sudah dihapus. Dan bila tidak bisa terhapus (gagal), maka akan memunculkan tulisan Data tidak bisa dihapus.

96. Menambahkan code ((MainActivity) context).deleteData (barangList.get(i).getIdbarang()); untuk menghapus data yang tampil di recycler view pada activity\_main.xml. MainActivity merupakan nama dari classnya dan deleteData merupakan public void yang telah dibuat sebelumnya di dalam MainActivty.java (deleteData tersebut sudah diatur agar digunakan untuk menghapus data yang ada di dalam recycler view itu juga, jadi tinggal memanggil deleteData nya (mempermudah agar tidak terlalu panjang code yang ada di dalam switch case nya)).

**Saya Belum Mengerti**

1. Masih sedikit bingung pada saat penggunaan Cursor pada video 32.

2.

3.