|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Android Studio Minggu-5 | 80 |

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

1. Model, untuk menghubungkan antara database dan adapter (antara data dan adapternya).

2. Bisa membuat dan memahami fungsi dari model untuk RecyclerViewCardView sesuai dengan yang ada di video.

3. Adapter, untuk memasukkan pilihan isi yang sudah dibuat sebelumnya.

4. Bisa membuat dan memahami fungsi dari adapter sesuai dengan yang ada pada video.

5. 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)).

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

7. 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).

8. 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).

9. ViewHolder, untuk dapat menautkan item\_siswa.xml ke dalam adapter agar dapat dikenali oleh javanya.

10. 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).

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

12. 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).

13. Mengisikan file adapter (<>) yang berada di atas (setelah extends) untuk dapat langsung mengenali menggunakan adapter apa beserta menambahkan ViewHolder nya juga. (Pada saat error (masih berwarna merah dengan tanda seru dibagian pojok kanan atas), bisa langsung mengklik logo yang seperti bohlam namun berwarna merah dengan tanda seru, lalu memilih Implemet methods dan mengkliknya. Error tersebut akan hilang, serta setelah di klik akan langsung (otomatis) memunculkan beberapa code yang seharusnya ada dan sudah langsung tersedia (sudah dibuatkan). Namun nantinya dapat diganti sesuai dengan kebutuhan code yang digunakan).

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

15. Private, diperuntukkan untuk suatu hal (misal suatu variabel) yang tidak diperkenankan untuk sembarang code (jadi nanti harus dipanggil terlebih dahulu untuk yang membutuhkan ini).

16. Public, untuk suatu hal yang bersifat umum dan dapat langsung dikases segala macam code tanpa dipanggil terlebih dahulu seperti private.

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

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

19. 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)).

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

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

22. 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.

23. Memberikan id untuk setiap komponen yang membutuhkan untuk digunakan pada saat pemanggilan komponen di pembuatan code.

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

25. 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). Begitu juga untuk isidata(), kalau tidak diamsukkan onCreate maka tidak akan muncul design yang telah dibuat di item\_siswa.xml ke Recycler Viewnya.

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

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

28. ArrayList, untuk data yang banyak namun sudah beurutan sesuai dengan urutan pembuatannya.

29. Membuat isian dari komponen card view yang nantinya ditampilkan di Recycler View dengan code siswaList.add (new Siswa (nama : “….”, alamat : “….”));. Titik-titik pada bagian nama diisikan dengan nama orang yang akan dimasukkan ke dalam card view (misal Joni seperti pada video) dan titik-titik pada bagian alamat diisikan dengan alamat dari orang tersebut berasal dari mana (misal Surabaya seperti pada video).

30. Ctrl + D, digunakan untuk menjadikan data pada saat pengisian nama dan alamat dari card view menjadi lebih banyak sesuai dengan yang diinginkan dan sama persis isiannya. (Sama seperti copy dan paste, namun akan langsung keluar cepat dengan otomatis bilamana ctrl+d tersebut ditekan dan diarahkan ke bawah (dengan tombol ke bawah). Mirip juga dengan shift+alt pada saat di visual studio code). Nama dan alamat yang sudah dilebih banyakkkan tadi juga bisa diganti di tiap masing-masing barisnya.

31. SetAdapter, untuk menempatkan adapternya pada suatu bagian (seperti pada bagian recyclerView.setAdapter(adapter) agar dapat tersambung dan menyatu dengan card view, serta dapat tampil di recycler viewnya).

32. 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 “Nama : Joni Alamat : Surabaya”, 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).

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

34. Pada saat menambahkan (new View.OnClickListener()), maka akan otomatis langsung dibuatkan @overide beserta public voidnya oleh android studio tersebut.

35. Isi dari toast yang awalnya hanya langsung memunculkan nama dari isian card view sesuai dengan yang sudah di klik dengan code siswa.getNama(). Bisa dimodifikasi sesuai selera dan sesuai kebutuhan (bisa digunakan untuk menjalankan apa misalkan card view tersebut di klik). (Pada video isi dari toast nya dimodifikasi menjadi “Nama : ”+siswa.getNama()+“Alamat : ”+siswa.getAlamat() yang menampilkan Nama : …. Alamat : …. pada toastnya (Titik-titik tersebut berisikan nama dan alamat dari masing-masing isi card view sesuai dengan card view yang di klik (misal mengklik card view dari Joni yang beralamatkan di Surabaya, maka isi tampilan dari toast tersebut ialah Nama : Joni Alamat : Surabaya)). Besar dan kecil dibagian nama dan alamat tersebut sesuai dengan isi dari card viewnya, bila besar (berkapital) maka akan juga sesuai dan ikut besar pada saat di klik dan ditampilkan di toastnya.

36. Bisa menambahkan dan menggunakan Menu Option sesuai dengan yang ada di video.

37. &#8942;, merupakan kode yang akan menghasilkan titik tiga seperti pada menu option biasanya (seperti yang ada di whatsapp bagian pojok atas kanan). Menuliskannya di komponen text view dan mengubah textnya menjadi kode tersebut. Serta menambahkan setOnClickListener juga.

38. 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 untuk foldernya dan tipe dari resourcenya yang awalnya values diubah menjadi menu).

39. Setelah membuat folder di dalam res (folder menu), maka akan memuat file .xml di dalam folder tersebut dengan cara klik kanan, lalu memilih New > Menu resource file dan akan mengisikan nama dari filenya akan diberi nama apa, settingan sourcenya, dan lokasi dari tempatnya (folder menu), lalu kik ok. Setelah di klik ok akan langsung memunculkan tampilan (design) dari xml yang telah dibuat tadi, serta tampilannya langsung beserta show display ui nya. (Pada video diberi nama menu\_option yang source set nya main dan diletakkan di directory name menu).

40. 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).

41. 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)).

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

43. 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).

44. Bisa mengganti nama (text) dari tampilan komponennya (namun pada menu item, text tersebut bernamakan title (untuk mengganti nama dari tampilan di komponennya)).

45. Bisa mengatur margin dari tiap komponen sesuai dengan yang ada pada video.

46. Margin, untuk memberikan jarak pada komponen yang ada di dalam layer (baik pada layout\_margin, pada jarak handle, maupun pada komponennya).

47. Bisa mengatur besar dan kecil dari suatu komponen sesuai dengan yang diinginkan (serta sesuai seperti yang ada di video).

48. Menambahkan komponen baru (menu option yang titik tiga) di dalam public ViewHolder pada bagian SiswaAdapter.java (adapternya). Diletakkan di bawah dari komponen text view nama dan alamat.

49. 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 simpan dan hapus seperti pilihan dari menu option yang ada pada video).

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

51. 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. Namun pada video hanya diberikan tulisan (toast)).

52. 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. (Misalkan pada video dipilihan dari komponen menu option (titik tiga) Simpan diberikan case yang diisikan toast dan berisikan tulisan Simpan data beserta nama dari card view yang di klik pilihan simpan tersebut (dengan code toast yang pada textnya diisikan “Simpan Data ”+siswa.getNama() (misal, Simpan Data Joni). Berbeda lagi nantinya dengan pilihan option dari Hapus (bila hapus di klik, maka akan mengeluarkan hasil yang telah dibuat dari case hapus itu sendiri (karena berbeda case dengan Simpan, yaitu akan menghapus atau menghilangkan card view dari yang sudah dipilih hapus tersebut, dan bila suatu card view sudah di hapus, maka akan memunculkan tulisan …. Sudah di Hapus (titik-titik tersebut berisikan nama dari card view yang sudah dipilih dan akan dihapuskan tersebut (mengatur tulisannya di dalam toast)))))).

53. 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.

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

55. Remove, untuk menghapus card view yang sudah dipilih. Menuliskannya ada di dalam code switch dari case hapus.

56. NotifyDataSetChanged, untuk otomatis langsung merefresh sesuatu bila sudah terdapat perubahan di dalamnya (seperti pada saat penghapusan salah satu card view yang sudah dipilih).

57. 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 ‘btnTambah(View)’ in ‘MainActivity’ (pada error onClick di button btnTambah, bila error pada button lain atau attribute dari komponen yang lain, maka namanya akan mengikuti dari id komponen tersebut (bukan lagi btnTambah)). Setelah itu, maka akan terbuat void atau method (public void btnTambah (View view) {}) di main activity.

58. Bisa menambahkan data dengan onClick yang di letakkan pada button dibagian atas dari Recycler View. Pada saat button TAMBAH tersebut di klik (yang sudah diberi onClick), maka akan menambahkan data dari nama dan alamat yang sebelumnya sudah ditambahkan di dalam code MainActivity.java dibagian public void btnTambah (View view). Nama dan alamat yang sudah ditambahkan di dalamnya akan bisa muncul dan ikut bergabung menjadi card view dan berada di dalam Recycler View dibagian bawah (diberi notifyDataSetChanged untuk merefresh otomatis agar data dari nama dan alamat yang baru saja ditambahkan lewat button TAMBAH bisa langsung muncul dan terlihat (bisa tampil)). Berapa banyak pun mengklik buttonnya, maka akan tetap muncul dan keluar data dari nama dan alamat yang telah dimasukkan.

59. Bisa memahami SQLite beserta konsep dan fungsinya.

60. Bisa membuat SQLite beserta isi di dalamnya sesuai dengan yang ada di dalam video.

61. Bisa mendownload SQLite DB Browser (aplikasinya) dan mengisi beberapa tabel di dalamnya sesuai dengan yang ada di video.

62. Membuat database baru dengan klik File, kemudian beri nama dari database nya (misal dbtoko pada video), beserta bisa diatur ingin diletakkan di mana (tinggal memilih tempat atau folder untuk menyimpan database nya), lalu klik Save.

63. Setelah di save, maka akan memunculkan kotakan yang harus diisi untuk membuat tabelnya beserta mengisi kolom-kolomnya (seperti tblbarang untuk tabelnya dan idbarang untuk kolomnya).

64. Dibawah dari pengisian kolom, terdapat code CREATE beserta isian dari kolom-kolom yang sudah dibuat atau baru saja dibuat (akan langsung masuk ke dalam code yang ada di bawahnya (CREATE)). Bila sudah selesai membuat kolomnya sesuai dengan yang diperlukan, maka klik ok. Code tersebut bisa digunakan untuk pembuatan project atau program di android studio nantinya.

65. Integer, merupakan type yang ada di dalam kolom SQLite DB Browser yang bertipekan untuk angka.

66. Auto Increment (AI), untuk menjadikan suatu kolom (isiannya) menjadi langsung terdapat id dan beberapa hal yang bila dibutuhkan akan langsung dimasukkan (karena diberikan AI ini).

67. Text, merupakan type yang ada di dalam kolom SQLite DB Browser yang bertipekan untuk teks, bisa berupa kata maupun kalimat.

68. Real, merupakan type yang ada di dalam kolom SQLite DB Browser yang bertipekan untuk angka yang sebenarnya (sesuai dengan keadaan yang sedang terjadi).

69. Menambahkan isian di dalam kolom databasenya menggunakan (code yang diketikkan di dalam Execute SQL). (INSERT INTO tblbarang (barang,stok,harga) VALUES (“Buku”,5,2000)). Diharuskan menggunakan petik dua yang double (“ ”), karena bila menggunakan petik dua saja (‘ ’), maka akan merah dan tidak bisa dibuat atau dijalankan (error (syntax error)). Menuliskan urutan pada valuesnya harus sesuai dengan tabel yang ada (agar sesuai), misalkan tblbarang (barang,stok,harga), maka penulisan untuk valuesnya harus dimulai dengan nama barangnya terlebih dahulu, kemudian stok barangnya, dan yang terakhir harga dari barangnya (bila tertukar dan tidak sesuai urutannya, maka tidak sesuai dengan tabel di kolomnya nantinya).

70. Bilamana ada error dan tidak bisanya suatu code yang telah dibuat di dalam dari code tadi (isian yang ada di dalam kolom), maka akan memunculkan syntax error di bawahnya (code dengan warna merah di bawahnya).

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

72. 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).

73. Update, untuk mengubah (mengupdate) suatu data dari isian yang ada di dalam kolom sesuai dengan kolom yang isiannya ingin diganti.

74. Mengganti isian yang ada di dalam tabel kolom dengan UPDATE tblbarang SET harga = 5000 WHERE idbarang = 1. (Untuk mengganti harga dari barang yang memiliki id 1 (dari idbarang = 1) yang awalnya senilai 2000 menjadi 5000). Dan bila di tampilkan kembali (menggunakan SELECT) maka sudah berubah menjadi harga yang sudah diubah dari (5000 yang awalnya 2000).

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

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

77. Mengetikkan DELETE from tblbarang WHERE idbarang = 1, maka id yang dituliskan akan hilang terhapus (hilang dari isi kolom).

78. 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))).

79. Bisa mendownload memu play untuk digunakan melihat database yang telah dibuat dengan SQLite DB Browser.

80. Memu play, emulator yang sama seperti nox player kegunaannya. Namun ganti menggunakan memu play ini dikarenakan ada fitur root untuk androidnya (karena memerlukan android atau hp yang telah di root agar bisa melihat database yang telah dibuat sebelumnya).

81. Memasang atau menginstall SQLite Reader di dalam emulator memu play untuk bisa membuka dan melihat hasil dari database yang telah dibuat dengan menggunakan SQLite DB Browser.

82. Membuka pilihan pengembang (opsi developer) pada emulator memu play (sama seperti pada saat membuka pilihan pengembang di emulator nox player).

83. Bisa menyambungkan database yang sebelumnya telah dibuat di SQLite DB Browser ke dalam android studio dengan membuat file MainActivity.java baru yang bernamakan Database. Di sebelah public class ditambahkan extends SQLiteOpenHelper dan kemudian bila di klik (pilihan SQLiteOpenHelper tadi) maka akan langsung otomatis dibuatkan beberapa code yang dibutuhkan. Serta nantinya menambahkan, mengganti, serta menghapus beberapa codenya agar tidak terjadi error dan dapat dijalankan. Setelah membuat di activity Adapternya, kemudian disambungkan ke MainActivity.java dengan function load (public void load()) dan mendeklarasikan databasenya diatas, namun tetap di dalam classnya (Database db;) (sesuai dengan yang ada pada video).

84. Private static final, digunakan untuk menuliskan nama dari database yang akan digunakan (yang sudah dibuat sebelumnya) di sini. (private static final String DATABASE\_NAME = “….” (titik-titik tersebut berisikan nama dari database yang sebelumnya telah dibuat di dalam SQLite DB Browser (misalkan tblbarang))).

85. Untuk mengecek apakah database yang sudah dibuat sebelumnya telah masuk ke dalam android aatau belum, dengan mengklik SQLite Readernya (aplikasinya) kemudian mengklik garis tiga di pojok kiri atas yang kemudian memilih Private form apps. Setelah itu, bila aplikasinya terlalu banyak bisa langsung mencarikan aplikasi android yang telah dibuat sebelumnya di tombol search (dengan logo seperti kaca pembesar) dan mengetikkan nama dari aplikasi androidnya tadi (SQLite Database). Dan bila aplikasinya tidak terlalu banyak, bisa langsung dicari ke bawah dengan mengescrollnya ke bawah hinga menemukan aplikasi android yang sudah dibuat sebelumnya (SQLite Database). Kemudian bila SQLite Database ini di klik akan memunculkan database yang telah dibuat tadi (dbtoko) maka databasenya sudah bisa dikaitkan ke dalam android studionya.

86. Bisa membuat tabel yang awalnya ada di dalam SQLite DB Browser dan dimasukkan di dalam android studio agar bisa dilihat (tampil) serta bisa membuat method dari SQL.

87. Mengopy syntax dari code pada saat pembuatan tabel pertama kali (CREATE da nisi lainnya) untuk bisa membuat tabel dan menampilkannya nanti di aplikasi SQLite Reader. Syntax code dari CREATE tadi bisa langsung di copy pada saat pembuatan pertama kali tabelnya, maupun mengklik kanan dari tabel yang telah dibuat sebelumnya, lalu memilih Copy Create Statement. Untuk nantinya diletakkan di android studio agar bisa dijalankan untuk menampilkan hasil dari tabelnya di aplikasi SQLite Reader.

88. 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).

89. Bila sudah menambahkan boolean runSQL, maka harus menambahkan public void buatTabel() (sesuai dengan yang ada di video) karena sudah merupakan satu kesatuan untuk membuat suatu tabel (karena membuat tabel tersebut juga merupakan SQL). Pada saat menambahkan code CREATE dan lainnya (isi yang sudah dibuat di dalamnya), tidak lupa untuk diberikan petik dua double (“ ”). Jadi code syntax dari CREATE nya ada di dalam dari petik dua double (“ ”) tersebut.

90. Bila sudah membuat boolean runSQL dan public void buatTabel(), maka mengetikkan runSQL(tblbarang); di bawah dari public void buatTabel() (tetapi masih satu buka tutup kurung ({ })). Menambahkan code tersebut untuk dapat menjalankan dua (2) code yang sebelumnya telah dibuat (boolean runSQL dan public void buatTabel()).

91. Setelah membuat code yang diperlukan untuk menambahkan tabel (agar muncul dan dapat dilihat nantinya di aplikasi SQLite Reader), maka menambahkan db.buatTabel(); agar dapat muncul dan terlihat. buatTabel() tersebut berasal dari public void yang isinya code CREATE beserta isi yang lainnya yang diambil dari code syntax pada saat pembuatan tabelnya pertama kali di SQLite DB Browser (codenya juga bisa diambil setelah membuat tabelnya dengan mengklik kanan tabel tersebut (yang sudah dibuat sebelumnya) lalu mengklik (memilih) Copy Create Statement). Kemudian menjalankan (run) project android databasenya untuk melihat hasil penambahan tabel yang telah dibuat sebelumnya.

92. Pada saat melihat hasil tabel SQLite yang telah dibuat sebelumnya di android studio dan menjalankannya (run), diharuskan untuk menutup aplikasi SQLite Readernya dan membukanya kembali, dikarenakan agar hasil tabel yang baru saja dibuat bisa terlihat (sudah terrefresh oleh aplikasinya karena ditutup terlebih dahulu). (Cara melihat hasil tabelnya sama seperti pada saat pengecekan apakah databasenya (dbtoko) sudah masuk (ada) ataukah belum (di aplikasi SQLite Reader > garis tiga dibagian pojok kiri atas > Private from apps > Mencari aplikasi SQLite Database > mengklik aplikasi ini dan kemudian memunculkan databasenya dbtoko. Setelah databasenya (dbtoko) di klik, maka akan memunculkan hasil dari apa saja isi yang terdapat di dalam database tersebut (terdapat tblbarang, yang berarti sudah berhasil memunculkan atau menampilkan tabel tersebut yang awalnya dibuat di SQLite dan kemudian dibuat juga di android studio (yang hasilnya bisa tampil disini (namun yang muncul dan dapat dilihat hanya tampilan luar dari tabelnya, bilamana di klik, maka di dalamnya belum terdapat isi apapun (karena belum ditambahkan di android studionya)))))).

**Saya Belum Mengerti**

1. Masih sedikit bingung pada saat pembuatan adapter pada video 21.

2.

3.