|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Android Studio Minggu-7 | 91 |

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

1. Bisa memahami, membuat, dan menggunakan select where sesuai dengan yang ada di video.

2. 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))).

3. Browse Data (pada SQLite DB Browser), untuk melihat hasil tabel yang telah dibuat sebelumnya (sudah terbentuk dalam bentuk tabel sesuai dengan kolom dan baris yang sudah dibuat sebelumnya (bila nanti ada penambahan (insert), penggantian (update), dan penghapusan (delete) maka data yang ada di dalam tabel tersebut akan terganti dan dapat dilihat dengan select pada execute nya (dilihat di execute dalam bentuk tabel kotakan))).

4. Execute SQL (pada SQLite DB Browser), untuk membuat code penambahan (insert), penggatian (update), penghapusan (delete), dan melihat hasil dari apa yang sudah dibuat (seperti tampilan pada browser data) dengan menggunakan select. Akan menunjukkan hasil atau syntax bila code yang sudah dibuat berhasil dan tidak error.

5. 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).

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

7. Pada saat di execute SQL (pada SQLite DB Browser), pada saat mengisikan isiannya (isian dari tabelnya), bilamana bertipekan integer, maka tidak memerlukan petik dua satu maupun double. Dan untuk yang bertipekan char, maka harus diberi petik dua double diantara kata-katanya. (Kalau tidak, maka akan terjadi error dan code tidak bida dijalankan atau digunakan).

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

9. Pada android studio, mengganti dan menambahkan code dari SELECT \* FROM tblbarang WHERE idbarang = “+id+” (hanya diberikan petik dua double (“ ”) dikarenakan id bertipekan integer (int) (real juga hanya diberikan petik dua double saja (“ ”) tanpa petik satu mengapit diantaranya (seperti ‘”+barang+”’))). Dituliskan id dan bukan langsung id dari suatu barangnya agar dinamis mengikuti dari id dari setiap barang yang sudah dipilihkan (begitu pula dengan code lainnya yang hasilnya agar menjadi dinamis dan dapat langsung terubah-ubah sesuai dengan apa yang sudah diinputkan maupun digantikan (misal seperti pada update)).

10. Bisa menampilkan code execute (seperti SELECT \* FROM tblbarang WHERE idbarang = “+id+” 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))).

11. Menambahkan code ((MainActivity) context).selectUpdate (barangList.get(i).getIdbarang()); untuk mengganti 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 (selectUpdate tersebut sudah diatur sebelumnya di dalam MainActivity.java agar digunakan untuk mengganti data yang ada di dalam recycler view itu juga, jadi tinggal memanggil selectUpdate nya (mempermudah agar tidak terlalu panjang code yang ada di dalam switch case nya)).

12. 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.

13. 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.

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

15. Bisa membuat menu option (yang ada titik tiga) dari ubah yang sebelumnya belum bisa digunakan sesuai dengan kegunaannya yang kemudian ditambahkan code untuk membuat menu option ubah tersebut berfungsi sesuai dengan kegunaannya (seperti dengan yang ada di video (seperti menambahkan public void selectUpdate dan mengganti yang awalnya terdapat toast di dalam ubah (di switchnya) lalu menggantinya dengan code yang mirip dengan code yang digunakan di dalam switch hapus (hanya saja mengganti functionnya (public void)))). Yang setelah itu juga dapat memasukkan data dari yang dipilih atau diklik kan menu option ubah (bila nama barangnya pulpen, maka akna mengisikan pulpen beserta stok dan harganya. Namun bila kertas, maka yang tampil di dalam kotakan teks tiap komponennya ialah data dari nama barang kertas beserta stok dan harganya) yang ada di dalam recycler view sebelumnya (sudah di insertkan terlebih dahulu) di dalam kotakan teks tiap komponennya (misal dari data nama barang maka akan masuk kembali ke teks komponen untuk nama barang (seperti pada saat awal membuat). Pada saat semua teks sudah masuk kembali ke tiap teks dari komponennya masing-masing, maka tulisan yang awalnya insert di bagian atasnya akan terganti secara otomatis (sudah diatur sebelumnya di dalam selectUpdate) menjadi update. Dan bila telah mengganti apa yang ingin diganti dari data yang sebelumnya (bila ada kesalahan atau yang lainnya), maka akan mengganti kembali tulisan update diatasnya tersebut menjadi insert kembali (setelah di klik simpan)).

16. Cursor, digunakan pada saat menggunakan syntax atau code dari select.

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

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

19. Ctrl + D, untuk menjadikan code yang diinginkan dapat disalin dan ditempelkan dengan cepat dengan menekan tombol ke bawah juga. Mirip dengan ctrl+c dan paste, namun ctrl+d tersebut lebih cepat (langsung muncul dari apa yang disalinkan dan persis sama).

20. Bisa memahami, membuat, dan menggunakan update sesuai dengan yang ada di video.

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

22. Mengganti isian yang ada di dalam tabel kolom dengan UPDATE tblbarang SET barang = “KERTAS KARTON”, stok = 20, harga = 5000 WHERE idbarang = 2. (Untuk mengganti nama barang dari barang yang memiliki id 2 (dari idbarang = 2) yang awalnya bernamakan KERTAS menjadi KERTAS KARTON). Dan bila di tampilkan kembali (menggunakan SELECT maupun di Browse Data) maka sudah berubah menjadi nama barang yang sudah diubah dari (KERTAS KARTON yang awalnya hanya KERTAS saja). Untuk yang tidak diubah atau diganti seperti stok dan harganya, maka pada saat ditampilkan akan tetap sama seperti awalnya (namun bila nanti ikut diubah maka akn tergantikan juga sesuai dengan apa yang sudah diubahkan (misal stoknya diganti menjadi 5 seperti yang ada pada video)).

23. 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.

24. Bisa mengganti data (update) pada saat program nya dijalankan (run app). Data yang ingin diubah di klik pada bagian menu optionnya dan memilih ubah. Setelah itu data yang sudah dipilih tersebut akan masuk kembali di dalam kotakan tiap teks komponennya masing-masing (nama barang di kotakan nama barang, stok di kotakan stok, dan harga di kotakan harga). Dan setelah mengubah apa yang diinginkan bisa langsung mengklik button simpan dan data yang sudah diubah tersebut akan tampil kembali di dalam recycler view yang ada di bawahnya dan langsung otomatis memunculkan data dari yang sudah diubah sebelumnya (bukan data pada awalnya yang sebelum diubah).

25. Menggunakan if dan else untuk code update serta memunculkan pesannya bilamana data sudah berhasil di ubah (“data sudah di ubah”) dan memunculkan tulisan data tidak bisa di ubah (“data tidak bisa diubah”) bilamana data yang ingin diubah tersebut gagal atau tidak berhasil untuk diubah (serta menambahkan selectData() pada bagian if di bawahnya String untuk update agar langsung merefesh). (if dan else yang digunakan ikut terpaut dengan if dan else dari data yang sebelumnya sudah dibuatkan (berawal dari if yang terdapat isEmpty() di dalamnya (barang.isEmpty() || stok.isEmpty() || harga.isEmpty() ))).

26. 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)).

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

28. Bisa memahami, membuat, dan menggunakan dialog (untuk delete) sesuai dengan yang ada di video.

29. AlertDialog.Builder, digunakan untuk membuat kotakan konfirmasi sebelum menghapus. (Diperlukan ini agar berjaga-jaga bilamana ada kesalahan tiba-tiba pada saat tidak sengaja mengklik tombol hapus (agara ada dialog konfirmasinya terlebih dahulu dan tidak langsung hilang terhapus)).

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

31. Al.show(), untuk dapat menampilkan alert dialognya nanti pada saat di klik (bila tidak ditambahkan ini maka nanti alert dialognya tidak akan dapat muncul).

32. Set Title, untuk menyetting judul dari kotakan dialog sebelum konfirmasi penghapusan nantinya (misal seperti “PERINGATAN !” pada video).

33. Set Message, berisikan isi dari kotakan dialog konfirmasinya sebelum pilihan iya dan tidak (misal seperti “Yakin akan menghapus ?” pada video).

34. SetPositiveButton, untuk pilihan positif dari alert dialog button (positif yang dimaksud yakni seperti jawabannya YA yang merajuk ke arah setuju (positif)). (Dan setelah di klik atau di setujui maka akan menjalankan tulisan “Data Sudah dihapus” (seperti pada saat di video)).

35. SetNegativeButton, untuk pilihan negatif dari alert dialog button (negatif yang dimaksud yakni seperti jawabannya TIDAK yang merajuk ke arah tidak setuju (negatif)). (Dan setelah di klik atau di setujui (untuk memilih tidak) maka data tersebut tidak akan terhapus (tidak akan hilang dan tetap di recycler view tempat asalnya) seperti pada saat di video).

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

37. Dialog.cancel(), untuk membatalkan suatu pilihan (digunakan di dalam setNegativeButton).

38. Memindahkan String dari sql deletenya menjadi di dalam setPositiveButton agar dapat digunakan sesuai dengan urutannya (berurutan). Serta menghapus String pada bagian id nya (dibagian public void deleteData) karena stringnya sudah ikut dari string yang di deklarasikan di awalnya (di bagian atas) sesuai dengan yang ada di video.

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

40. Bisa memahami, membuat, dan menggunaan sharedpreferences sesuai dengan yang ada di video.

41. SharedPreferences, digunakan untuk menyimpan pengaturan (setting) ataupun suatu file dan hal lainnya (sesuai apa yang dimasukkan untuk disimpan) dengan tipe data apapun pada aplikasi. Filenya diletakkan di dalam code android studio dan pada saat di run akan bisa tetap menyimpan (ditampilannya). Pada video kali ini dipergunakan untuk menyimpan suatu tulisan nama barang beserta jumlah stoknya. (Saat di insertkan (ditambahkan di kotakan teks tiap komponennya) dan ditampilkan (dengan button tampil di bawah button simpan) maka akan muncul teks nya sesuai dengan tiap kotakan komponennya. Lalu pada saat aplikasinya ditutup (aplikasi untuk sharedpreferences) lalu dibuka kembali dan mengklik tampil, maka apa yang sebelumnya di insertkan dan di klik simpan akan dapat langsung muncul kembali (memunculkannya persis dengan apa yang sebelumnya dimasukkan)). (Misal seperti menyimpan data (ditambahkan (insert)) nama barang pensil yang jumlah stoknya 5, maka pada saat di klik button tampilnya akan memunculkan nama barang dan stoknya tersebut sama persis dengan apa yang ditambahkan (setelah mengklik button simpan)).

42. Bisa membuat project baru (untuk sharedpreferences) sesuai dengan yang ada di video.

43. Design (di atas pojok kanan), untuk menampilkan layar untuk mengatur tampilannya (namun bukan dengan code, melainkan di klik dan di tarik).

44. + (plus), di bagian kanan bawah (dibawahnya tombol bergambar tangan), untuk memperbesar layar tampilan dari design.

45. - (min), di bagian kanan bawah (dibawahnya tombol + (plus)), untuk memperkecil layar tampilan dari design.

46. 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).

47. 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).

48. 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).

49. 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 pada saat di run (dijalankan) atau bisa juga di cek sudah terkaitkan atau belum dengan megeceknya di toggle (logo seperti kunci inggris).

50. 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.

51. 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.

52. 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.

53. 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.

54. 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.

55. 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)).

56. Bisa mengganti maupun menghilangkan suatu teks dari suatu komponen sesuai dengan yang diinginkan (sesuai juga dengan yang ada di video).

57. Bisa mengganti atau menambahkan suatu id pada tiap komponen sesuai dengan yang dibutuhkan (sesuai dengan yang ada di video). Menambahkan id agar mempermudah dan dapat menjalankan suatu code pada program bila diperlukan dan dipergunakan nantinya.

58. 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.

59. Bisa mencari suatu attributes secara cepat dengan mengklik tombol search di bagian atas (berlogokan seperti kaca pembesar) dan menuliskan attributes apa yang diperlukan (misalkan seperti mencari hint pada saat mencari attributes hint).

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

61. 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 simpan (View view) {}) di main activity. Bisa juga mengetikkannya manual di classnya terlebih dahulu secara manual (public void simpan (View v) {}). Hasilnya akan sama dan dapat digunakan (tidak error).

62. Code (di atas pojok kanan), untuk menampilkan codenya (menambahkan elemen yang diketik).

63. 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 dimasukkan onCreate maka apa yang sudah dibuat sebelumnya tidak akan muncul di designnya.

64. 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))).

65. Get, untuk mendapatkan hasil dari suatu hasil yang telah diambil atau diisi oleh set sebelumnya. (digunakan untuk get pada getSharedPreferences).

66. Mode\_Privat, digunakan agar hanya aplikasi yang sudah dibuat dan diberikan ini saja yang nantinya dapat menggunakan file dari sharedpreferencesnya.

67. Float (sama seperti double), digunakan untuk menjadikan suatu angka menjadi berbentuk decimal (seperti 1.5) dan tidak bulat (seperti 1). (Pada saat di video huruf depan dari floatnya (f nya) menggunkaan huruf kecil, karena nanti bilamana ada ketidak samaan dari huruf depannya akan dapat mempengaruhi code tersebut nantinya).

68. SharedPreferences.Editor, untuk mengedit dan menambahkan beberapa code atau hal yang nantinya digunakan dan dapat dipanggil lewat button simpan dan tampil. Serta menambahkan editor.putString (untuk memasukkan data dari nama barangnya (barang)), editor.putFloat (untuk memasukkan data dari stoknya), dan editor.apply() (untuk menjalankan dan memasukkan data dari editor.put…. (titik-titik tersebut berisikan string atau float)) di dalamnya.

69. Bisa membuat button simpan (untuk menyimpan data yang telah diinsertkan (dimasukkan)) dan tampil (menampilkan datanya baik pada saat setelah di run app maupun setelah aplikasinya ditutup dan dibuka kembali (tanpa run app terlebih dahulu di dalam android studionya)) dapat digunakan sesuai dengan kegunaannya (sesuai dengan yang ada di video).

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

71. ParseFloat (float.parseFloat), digunakan untuk mengkonversi (mengubah) data yang awalnya bertipekan float menjadi string (namun tetap dalam tipe float yang hanya saja dapat ditampilkan dalam bentuk tulisan (string)).

72. Menggunakan if dan else (serta memasukkan beberapa code) untuk menjadikan bilamana pada kotakan teks barang maupun stok tidak ada isinya, maka datanya tidak dapat tersimpan (menggunakan isEmpty() pada keduanya (baik pada barang maupun stok)). Serta menambahkan toast bertuliskan pesan “Data Kosong” bilamana data pada kotakan tersebut kosong dan tidak terisikan. Bila data tersebut sudah terisi (diberikan isiSharedPreferences juga pada elsenya), maka akan memunculkan pesan dari toast yang bertuliskan “Data Sudah di simpan”.

73. 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).

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

75. Sama dengan dua kali (==), digunakan untuk apa yang diberikan ini hasilnya dapat sama persis dengan yang sudah ditautkan atau diisikan (lebih akurat bila memakai sama dengan dua kali, seperti stok pada video (stok == 0.0 (yang berartikan bahwa stok tersebut bila ditampilkan awalnya selalu diawali dengan 0.0))).

76. Mengosongkan kembali kotakan setelah dapat diisikan suatu data (diinsertkan) dari barang maupun stok dengan cara menambahkan set text yang di dalamnya kosongan (misalkan seperti etBarang.setText(“”);. Begitu pula dengan stoknya (hanya saja mengganti etBarang tersebut menjadi etStok (etStok.setText(“”)) (dituliskannya berada di bawah dari etBarang dan etStok)))).

77. Menambahkan getString di dalam string dari public void tampil, agar dapat mendapatkan teks dari simpan yang sebelumnya sudah diisikan di dalamnya.

78. DefValue, untuk menampilkan sesuatu pada saat apa yang didapatkan tidak atau belum ada isinya (jadi pada video diisikan kosong (defValue:””), agar pada saata belum terisi berarti nantinya isi dari kotakan teks dari komponennya akan kosong pada saat di klik button tampilnya).

79. Bisa memahami dan menemukan error atau bug sesuai dengan yang ada di video.

80. Debugging, dari kata kutu yang berarti kecil, mengarah pada suatu error yang terjadi dari banyaknya data yang dapat membuat suatu aplikasi tidak dapat berjalan dan digunakan dengan semestinya.

81. Mencari suatu bug atau error dapat dicari pada bagian paling bawah dari android studio. Mencarinya dengan melihat run terlebih dahulu (selihat saya, run tersebut baru akan muncul apabila suatu program atau aplikasinya sudah saya jalankan (run app) terlebih dahulu). Kemudian baru melihat lebih detail dari error maupun bugnya di bagian logcat. Error ataupun bug yang muncul disini (di bagian bawah dari android studio) biasanya ditimbulkan oleh kesalahan pada saat membuat suatu code, string, atau variabel sendiri (bukan dari javanya). Bila error atau bug tersebut terdapat pada javanya, maka akan langsung memunculkan warna merah pada suatu code yang salah tersebut, serta bisa langsung dibetulkan dengan mengklik logo seperti bohlam yang warna merah (seperti onClick salah satunya).

82. Run (dibagian bawah dari android studio), satu baris dengan logcat, TODO, dan yang lainnya. Memunculkan segala data dari suatu program yang telah dijalankan (run app), mulai dari awal startnya hingga akhir (finish launchnya), maupun bila ada kesalahan atau error (bug) yang ada (akan ditampilkan disini). Serta dapat digunakan untuk mencari permasalahan dari suatu error atau bug yang terjadi dari kesalahan penulisan ataupun code yang dibuat sendiri. (Bila ada suatu error ataupun bug, dicarinya dari run ini terlebih dahulu (diutamakannya)).

83. Logcat, untuk memunculkan segala data dari sebuah project yang sedang dijalankan (di run app), bilamana ada error atau bug juga bisa dilihat dan ditemukan solusinya lewat sini. Berada dibagian bawah sendiri berjejeran dengan beragam hal salah satunya seperti TODO, Problems, Run dan lainnya. Pada Logcat bisa juga di search ingin mencari data yang di load yang mana. Pada saat kali ini digunakan untuk mencari error atau bug yang muncul (menggunakan logcat ini diharuskan untuk membuka dan mencari error atau bugnya lewat run terlebih dahulu (baru di logcat ini)).

84. Bisa mencari beberapa error ataupun bug yang terdapat pada contoh (yang ada pada video, seperti pada saat kesalahan di tabel barang yang kurang huruf g (tblbaran)). Di run maupun logcat pada saat pencarian bugnya, mencari kalimat atau baris yang memiliki warna merah (dari sekian banyak baris dan kalimat yang ada). Error ataupun bugnya juga akan menunjukkan tempatnya di sebelah mana (meskipun tidak langsung tertuju pada permasalahannya). Tempat tersebut akan memiliki warna merah (berbeda dengan baris yang lain, karena diberikan tanda warna merah untuk bagian errornya). Sebelum menuju tempat atau baris dari bugnya, pertama-pertama di klik terlebih dahulu baris kalimat yang berwarna merah (yang terdapat di dalam baris run dan logcat, memunculkan baris ke berapa dari kesalahannya (misal di MainActivity.java:98 seperti pada video)). Setelah di klik yang berwarna merah (juga akan memunculkan terdapat di baris mana letak dari error atau bugnya), maka akan bisa menuju baris dari code yang terdapat error atau bug tersebut (misal di baris ke 98, maka nantinya bila di klik atau di cari secara manual, di baris tersebutlah yang terdapat error atau bugnya (meskipun terkadang salahnya ada di bagian atasnya, tidak selalu menentu langsung dari baris kesalahan tersebut (tidak pasti di baris 98 tersebut))). Jadi harus tetap membaca dengan teliti dan perlahan (dari error atau bugnya) apa yang diinginkan dari suatu code tersebut memberitahu apa dan mana yang menjadikan sesuatu menjadi tidak bisa berjalan dengan semestinya ataupun menjadikan suatu aplikasi menjadi tidak bisa dijalankan atau dibuka (menutup dengan sendirinya (force close))).

85. Pernah mencari beberapa bug di logcat pada saat mencoba membuat beberapa project kecil, meskipun belum bisa langsung menyelesaikannya dan mencari penyelesainnya lewat web di internet.

86. Toast yang memiliki pesan “data tidak bisa diubah” dan beberapa pesan yang memiliki kegagalan dalam penambahan, penampilan, pengubahan, maupun penghapusan atau pada saat apapun yang sudah ditambahkan codenya dapat muncul bilamana ada yang error dari code execute di sql nya. (Misal seperti SELECT \* FROM, dan yang lainnya).

**Saya Belum Mengerti**

1.

2.

3.