|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Android Studio 7 | 95 |

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

1. Where untuk hanya mengekstrak records yang memenuhi spesifikasi dari kondisi tertentu.

2. Mencoba sql command SELECT \* FROM namaTabel WHERE Kondisi di dalam dbbrowser. Jika commandnya sudah benar saat pengujian baru dipindahkan ke dalam android studio.

3. Membuat deklarasi terlebih dahulu yaitu String idbarang;.

4. Idbarang = id; String sql = "SELECT \* FROM tblbarang WHERE idbarang="+id+";"; Setelah deklarasi lalu dimasukkan kedalam function untuk digunakan. Digunakan untuk menselect data dari tabel lalu mencari kondisinya yaitu idbarang.

5. Saat menggunakan select diusahakan menggunakan Cursor.

6. Untuk menampilkannya menggunakan cursor.movetonext.

7. EtBarang.setText(cursor.getString(cursor.getColumnIndex("barang")));

etStok.setText(cursor.getString(cursor.getColumnIndex("stok")));

etHarga.setText(cursor.getString(cursor.getColumnIndex("harga"))); Untuk menampilkan data yang akan diubah.

8. TvPilihan.setText("update"); untuk ketika dalam proses ubah maka tv nya akan berubah menjadi update.

9. Update merupakan sql command untuk merubah data yang sudah ada.

10. String sql = "UPDATE tblbarang \n" +

"SET barang = \'"+barang+"', stok = "+stok+", harga = "+harga+"\n" +

"WHERE idbarang = "+idbarang+";"; Untuk menambahkan sql command update ke dalam android studio.

11. RunSql digunakan untuk menjalan sql command yang sudah dibuat. Dan membuat pengujian jika true maka akan mengupdate data dan menampilkan pesan data sudah di ubah. Dan jika hasilnya false maka tidak akan merubah data dan menampilkan pesan data tidak bisa di ubah.

12. Membuat alert dialog untuk jika tidak sengaja mengklik popup hapus maka datanya tidak akan langsung terhapus.

13. AlertDialog.Builder al = new AlertDialog.Builder(this); Untuk membuat alert dialog.

14. Lalu memindahkan command dan coding untuk sql delete ke dalam setpositivebutton, agar ketika ditekan setpositivebuttonnya akan menjalankan delete data.

15. Dialog.Cancel untuk mengcancel atau membatalkan command.

16. Sharedpreferences adalah file yang digunakan untuk menyimpan setting-setting dari aplikasi.

17. Membuat activity baru lalu menambahkan komponen dan menambahkan atribut onclick di dalam komponen button setelah itu mendeklarasikan komponen ke dalam .java, setelah itu membuat function load dan mendeklarasikan sharedpreferences.

18. Getsharedpreferences untuk mengambil dan menahan file preferensi, lalu mengembalikan sharedpreferences yang dimana kira dapat mengambil dan mengubah valuenya.

19. *MODE\_PRIVATE* berarti file ini hanya bisa digunakan pada aplikasi yang sedang dibuat saja.

20. SharedPreferences.Editor adalah interface yang digunakan untuk memodifikasi value yang ada di dalam objek sharedpreferences.

21. sharedPreferences.edit() untuk membuat editor baru untuk preferences.

22. Putstring untuk menset atau menetapkan value string di dalam editor preferences.

23. Putfloat untuk menset atau menetapkan value float di dalam editor preferences.

24. Apply untuk melakukan perubahan preferensi kembali dari editor ke dalam objek sharedpreferences yang sedang di edit.

25. Public void isiSharedPreferences (String barang, float stok) {

SharedPreferences.Editor editor = sharedPreferences.edit ();

editor.putString("barang", barang);

editor.putFloat("stok", stok);

editor.apply ();

} Untuk memasukkan value/isi kedalam shared preferences.

26. Public void simpan(View view) {

String barang = etBarang.getText().toString();

float stok = Float.parseFloat(etStok.getText().toString());

if (barang.isEmpty() || stok == 0.0) {

Toast.makeText(this, "Data Kosong", Toast.LENGTH\_SHORT).show();

} else {

isiSharedPreferences(barang, stok);

Toast.makeText(this, "Data sudah disimpan", Toast.LENGTH\_SHORT).show();

}

etBarang.setText("");

etStok.setText("");

} Untuk ketika sudah diisi, value yang di dapat akan di kirim ke dalam sharedpreferences.

27. GetString untuk mengambil sebuah nilai string dari preferences.

28. GetFloat untuk mengambil sebuah nilai float dari preferences.

29. public void tampil(View view) {

String barang = sharedPreferences.getString("barang", "");

float stok = sharedPreferences.getFloat("stok", 0);

etBarang.setText(barang);

etStok.setText(stok + "");

} Untuk menampilkan data barang yang ada di dalam preferences.

30. Di dalam java ketika terdapat kesalahan penulisan maka android studio akan memunculkan tanda merah sebagai peringatan jika ada yang salah/error.

31. Di bagian kanan dekat scroll akan terdapat warna yang jika berwarna merah maka menandakan error.

32. Jika sudah betul maka tanda berwarna merah akan hilang.

33. Jika yang salah ada di dalam string seperti sql command maka java dalam android studio tidak akan menunjukkan tanda merah yang menandakan error.

34. Dan ketika dijalankan maka android studio akan melakukan force close aplikasi ketika sudah dijalankan.

35. Dan jika dilihat di run maka akan ditunjukkan ketika ada error akan ditunjukkan dengan warna merah.

36. Ketika dalam kesalahan tulisnya tidak diberi tahu maka kita harus melihatnya dari yang menyebabkan aplikasi itu tidak bekerja/yang alurnya tersendat.

37. Di dalam logcat juga akan diberi tahu bagian mana yang error. Seperti halnya : No such table: tblbaran.

38. Melihat ada nya error adalah dari run terlebih dahulu setelah itu baru melihatnya di dalam logcat.

39. Ketika insert dan update data aplikasi nya akan tetap jalan tetapi ketika melakukan proses insert dan update akan menampilkan pesan gagal yang sudah dibuat. Baru kelihatan errornya ketika proses insert dan update dilakukan.

**Saya Belum Mengerti**

1.