|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Android Studio vid 36 s/d 41 | 90 |

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

1. **Menggunakan perintah SELECT dalam SQLite.** Penggunaan perintah SELECT digunakan untuk memilih sebuah data yang ingin di tampilkan atau yang ingin ditunjuk untuk dihapus, maupun diupdate. Cara penulisannya adalah “SELECT \*(menunjuk semua data yang ada di dalam sebuah database) FROM tblbarang(nama tabel yang ingin diambil datanya) WHERE idbarang(kolom id/lokasi datanya)=2(nama idnya/urutan idnya)”.
2. **Menampilkan isi dari salah satu data yang berasal dari database.** Dengan menggunakan perintah Select. Pada Main Activitynya membuat function baru yaitu selectUpdate dengan menggunakan value String id, Kemudian deklarasikan untuk idbarangnya di luar function dengan menggunakan String idbarang agar mudah dikenali oleh java, Lalu memasukkan idbarang yang di deklarasikan tadi dengan menggantinya dengan menggunakan variable id atau idbarang = id. Kemudian, memasukkan perintah select pada sqlnya dengan String sql = “SELECT \*(menunjuk semua data yang ada di dalam sebuah database) FROM tblbarang(nama tabel yang ingin diambil datanya) WHERE idbarang(kolom id/lokasi datanya)=”+id+”;(agar semua id yang ada di dalam kolom tersebut terbaca)”. Kemudian, masukkan perintah sql tadi dengan menggunakan cursor dengan memberi nama variabel Cursor dengan cursor jadi Cursor cursor = db.select(sql). Lalu, perintah cursor agar dapat mengisi data sesuai dengan urutannya dengan menggunakan perintah moveToNext() jadi cursor.moveToNext(). Kemudian, masukkan penempatan yang digunakan untuk data yang digunakan dengan menggunakan perintah misal etBarang(nama id dari lokasi untuk penempatan data yang diambil).setText(cursor.getString(cursor.getColumnIndex(“barang(nama kolom yang mau diambil)”))), begitu juga untuk data yang lainnya/yang diperlukan. Kemudian, atur agar perintah dijalankan pada saat update dengan tvPilihan.settext(“update”). Kemudian, pada adapternya untuk bagian switch case untuk menu ubah, perintahnya di ubah menjadi ((MainActivity)context).selectUpdate(barangList.get(i).getIdbarang()). Dengan begitu ketika di jalankan jika di klik menu ubah maka data yang di klik akan muncul ke dalam edit Text sesuai dengan id yang ditunjuk dan data yang ada.
3. **Menggunakan perintah UPDATE dalam SQLite.** Penggunaan perintah UPDATE digunakan untuk mengubah data yang ada di dalam database. Cara penulisan perintahnya adalah dengan “UPDATE tblbarang(menunjuk semua data yang ada di dalam tabel) SET barang(nama kolom) = "Buku"(memasukkan data yang ingin diubah di dalamnya), stok(nama kolom) = 3(memasukkan data yang ingin diubah di dalamnya), harga(nama kolom) = 3500(memasukkan data yang ingin diubah di dalamnya) WHERE (kolom id/lokasi datanya)=1(nama idnya/urutan idnya)”.
4. **Menjalankan Menu Ubah/Update.** Memasukkan perintahnya ke dalam function onClick pada button yang digunakan untuk menyimpan data yang diubah. Karena sudah membuat perintah untuk insert maka sekarang membuat perintah untuk ubah menggunakan if dan else, pada menu ubah di letakkan pada else setelah perintah dari insert. Maka, memasukkan perintah String sql = “UPDATE tblbarang SET barang=’ ”+barang+” ’(menggunakan tanda petik satu dan petik dua dikarenakan datanya berupa string atau text), stok = “+stok+”(menggunakan tanda petik dua karena datanya berupa real atau angka), harga=”+harga+”(menggunakan tanda petik dua karena datanya berupa real atau angka) WHERE idbarang = “+idbarang+(sesuai dengan yang di deklarasikan untuk kolom idnya)”;”. Kemudian, menggunakan perintah id dan else untuk menguji keberhasilan dan kegagalan dari proses update data yang dilakukan, dengan mengisi perintah if untuk menjalankan perintah sqlnya yang sudah di tuliskan tadi dan memasukkan pesan, maka if(db.runSQL(sql)){pesan(“Data sudah diubah”) lalu, menggunakan function selectData() agar data yang masuk agar terefresh atau memperbarui dengan sendirinya tanpa adanya error}, Lalu, perintah else digunakan untuk memasukkan pesan jika data yang diperbarui tadi tidak berhasil dengan else{pesan(”Data tidak bisa diubah”)}.
5. **Menggunakan Alert Dialog.** digunakan untuk memberi peringatan terhadap suatu tindakan yang dibuat untuk menghindari tidak kesengajaan dalam suatu kejadian, jadi dengan membuat adanya Alert Dialog maka akan meminta persetujuan dahulu kepada user apakah setuju atau tidak untuk melakukan tindakan tersebut. Untuk membuatnya dengan menggunakan AlertDialog.Builder, untuk mengatur tittle atau judulnya dengan menggunakan setTittle(“memasukkan judul yang diinginkan”), untuk mengatur pesannya dengan menggunakan setMessage(“masukkan pesan yang diinginkan”), untuk mengatur buttonnya dengan menggunakan setPossitiveButton(“kata yang ingin digunakan”) untuk tombol persetujuan dan setNegativeButton(“kata yang ingin digunakan”) untuk tombol tidak setuju dalam suatu perintah yang diinginkan.
6. **Menggunakan Shared Preferences.** Digunakan sebagai file - file yang bertujuan untuk mengatur sebuah aplikasi yang di buat yang nantinya bisa digunakan di file mana saja, tipe datanya bisa berupa string, float atau integer. Untuk penggunaannya dengan mendeklarasikan dulu dan mengubah Shared Preferences menjadi objek seperti SharedPreferences sharedPreferences. Kemudian untuk mengambil Shared Preferencenya dengan menggunakan etSharedPreferences(MODE\_PRIVATE(penggunaan mode private bertujuan untuk file yang berada di shared preferences ini hanya bisa digunakan oleh aplikasi yang kita buat saja)). Untuk mengisi SHared Preferencenya dengan menggunakan perintah SharedPreference.Editor dan menjadikannya objek menjadi editor = sharedPreferences.edit(). Untuk tipe data String menggunakan perintah putString() dan jika float menggunakan perintah putFloat(). Ketika sudah, menggunakan perintah apply() agar bisa digunakan. Untuk menyimpan data ke dalam Shared Preferencenya adalah jika tipe datanya string maka menggunakan getText().toString, jika tipe datanya menggunakan float maka menggunakan Float.parseFloat(getText().toString()).Untuk menampilkan data yang diisikan ke dalam Shared Preferences adalah dengan menggunakan perintah getString(key,defValue:””) dan jika float diisi dengan getFLoat(Key,defValue:0). Data yng sudah tersimpan ke dalam SHared Preferences akan selalu ada walaupun sudah simatikan programnya.
7. **Debugging atau Pencarian error.** Karena setiap penulisan program sangat banyak jadi memungkinkan terjadi kesalahan dalam menulis program dan menyebabkan error dan hal tersebutlah yang disebut dengan bug karena proram tidak berjalan seperti semestinya. Dalam pencarian bug membutuhkan kesabaran dan ketelitian dalam melakukannya. Jika di dalam java jika terjadi kesalahan dalam penulisan maka akan muncul tanda warna merah yang menunjukkan kesalahan. Jika ada kesalahan yang tidak ditunjukkan oleh java maka program aplikasi yang dibuat akan terjadi force closed atau menutup dengan sendirinya. Dan kita bisa mencari sendiri bug yang terjadi di dalam logcat karena disitu biasannya akan ditunjukkan oleh java penyebab dan dimana terjadinya bug atau error.

**Saya Belum Mengerti**

1.

2.

3.