|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Android Studio Minggu Ke-7 SMST5 (Video 36-41 Android Studio Untuk Siswa SMK) | 95 |

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

1. Menggunakan SQL untuk mengambil sebuah data dari database atau Select data dengan mengetikkan SELECT \* FROM tblbarang WHERE idbarang = 2; yang berarti ambil data dari tblbarang dengan mencocokkan kolom idbarangnya yang berisi 2.

2. Menggunakan Select data untuk proses update data sesuai dengan data yang dipilih. Caranya dengan mengambil id nya terlebih dahulu, jadi dalam recycleview terdapat beberapa item yang memiliki popup menu update dan delete yang setting kedua popup ini diletakkan di dalam case. Pada bagian case update diisi dengan

((MainActivity)context).selectUpdate(barangList.get(i).getIdbarang());

Yang berarti ketika popup menu update diklik dia akan menjalankan context yang ada di mainactivty atau file lain yaitu mainactivity, dia akan menjalankan function bagian selectUpdate dan mengirimkan id yang diambil dari data idbarang pada database milik data yang dipilih.

3. Setelah function selectUpdate dijalankan dan mendapat kiriman id, id ini akan dimasukkan ke dalam variable idbarang dan id nya juga digunakan di dalam code sqlnya seperti ini

String sql = "SELECT \* FROM tblbarang WHERE idbarang = "+id+";";

Jadi dapat diartikan sql = ambil data dari tblbarang yang idbarangnya adalah id (id yang didapat sesuai dengan data yang dipilih atau item yang dipilih).

4. Setelah mengambil data dari database berdasarkan id nya, data pada kolom barang, stok, dan harga ditampilkan di dalam komponen etBarang, etStok, dan etHarga dengan cara

Cursor cursor = db.select(sql);  
cursor.moveToNext();

Yang berarti cursor untuk menjalankan sql selectnya kemudian menyimpan datanya, setelah itu data yang ada di dalam cursor dimove-to atau dipindahkan ke dalam sini

etBarang.setText(cursor.getString(cursor.getColumnIndex("barang")));  
etStok.setText(cursor.getString(cursor.getColumnIndex("stok")));  
etHarga.setText(cursor.getString(cursor.getColumnIndex("harga")));

komponen etBarang di-set-text atau diisi dengan data yang disimpan cursor tadi untuk diambil datanya pada bagian kolom barang, begitu juga dengan etStok yang diisi dengan data dari kolom stok, dan harga yang diisi dengan data dari kolom harga. Dengan begitu beberapa komponen tersebut akan muncul data yang diambil dari kolom yang ditentukan. Dan untuk tvPilihan atau textview yang ditampilkan diatas disetting seperti ini

tvPilihan.setText("update");

jadi ketika proses update, tvPilihan akan berubah menjadi update agar memudahkan button simpan ketika menjalankan update bukannya insert.

5. Kembali ke function simpan atau function milik onClick button simpan pada bagian penentuan apabila tvPilihan insert maka dia akan menjalankan insert dan else nya dia akan menjalankan update. Di dalam else ini diisi seperti ini

String sql = "UPDATE tblbarang SET barang = '"+barang+"', stok = "+stok+", harga = "+harga+" WHERE idbarang = "+idbarang+";";  
if (db.runSQL(sql)){  
 pesan("data sudah diubah");  
 selectData();  
}else {  
 pesan("data gagal diubah");  
}

Yang berarti dibuatkan variable sql untuk menyimpan code update, jadi dia akan mengupdate data yang ada di tblbarang dengan ketentuan kolom barang diisi dengan variable barang, stok diisi dengan variable stok, begitu juga harga dengan ketentuan idbarang adalah variable idbarang(id dari item yang dipilih untuk diupdate). Variable barang, stok, dan harga ini didapat dari

String barang = etBarang.getText().toString();  
String stok = etStok.getText().toString();  
String harga = etHarga.getText().toString();

Yang berarti dia mengambil text yang ada pada komponen tersebut yaitu etBarang, etStok, dan etHarga yang sebelumnya sudah diisi dengan data dari item yang mau diupdate. Kemudian terdapat if else yang artinya jika sql berhasil dijalankan maka dia akan muncul pesan data sudah diubah lalu menjalankan function selectData() agar tampilan recycleview nya terrefresh, jika sql tidak berhasil dijalankan maka dia akan menjalankan elsenya. Kemudian setelah proses update, tvPilihan akan berubah menjadi insert dan komponen etBarang, etStok, etHarga akan kosong sesuai dengan code yang ada dipaling bawah dalam function onClick button ini.

etBarang.setText("");  
etStok.setText("");  
etHarga.setText("");  
tvPilihan.setText("Insert");

6. Mengatur dialog konfirmasi atau alert dialog untuk popup menu hapus, karena sebelumnya menu hapus ini ditekan langsung menghapus data maka ditambahkan alert dialog sebagai konfirmasi terlebih dahulu apakah yakin mau menghapus datanya. Di dalam function deleteData dibuatkan sebuah objek alert dialog bernama al seperti ini

AlertDialog.Builder al = new AlertDialog.Builder(this);

Jadi al dapat digunakan untuk mengatur alertDialognya akan dibuat seperti apa, codenya seperti ini

al.setTitle("PERINGATAN!");  
al.setMessage("Yakin akan menghapus?");

Jadi dalam alert dialog ini judulnya adalah PERINGATAN! Dan pesannya adalah Yakin akan menghapus? Kemudian ditambahkan 2 button untuk Ya dan Tidak dengan mengetikkan

al.setPositiveButton("YA", new DialogInterface.OnClickListener()

yang nantinya setelah mengetikkan DialogInterface.OnClickListener dan mengklik enter akan muncul code

{  
 @Override  
 public void onClick(DialogInterface dialog, int which) {

}  
});

Di dalam situ dapat diberikan code untuk mengatur apabila button positive YA diklik maka dia akan menjalankan code yang ada di dalam onClick ini, karena YA maka proses hapus dijalankan, codenya seperti ini

String sql = "DELETE FROM tblbarang WHERE idbarang = "+idbarang+";";  
if (db.runSQL(sql)){  
 pesan("Data Sudah Dihapus");  
 selectData();  
} else {  
 pesan("Data Gagal Dihapus");  
}

Jadi yang sebelumnya code tersebut ada di luar onClick dipindahkan di dalamnya. Kemudian untuk al.setNegativeButton sama seperti PositiveButton yang diberi onClick, bedanya di dalam Negative button ini diisi dengan Tidak dan isi onClicknya adalah dialog.cancel(); jadi ketika diklik Tidak dia akan otomatis menutup dialognya.

7. SharedPreferences adalah file yang digunakan untuk menyimpan setting-setting dari aplikasi. File ini bisa diisi dengan string, float, integer, dan lainnya. SharedPrefrences ini bisa menyimpan data dan data akan tetap ada meskipun aplikasi sudah ditutup. Contohnya dibuatkan 4 komponen untuk editText Barang (string), editText Stok (Number Decimal/Float), Button Simpan, dan Button Tampil. Jadi cara kerjanya etBarang dan etStok diisi data lalu klik simpan dan sharedPreferences akan menyimpannya, kemudian setelah disimpan komponen etBarang dan etStok akan kosong kembali, tetapi ketika diklik button tampil maka dia akan menampilkan data yang disimpan tadi ke dalam 2 komponen editText tadi. Code pada bagian onClick simpannya seperti ini

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("");

Jadi variable barang untuk menyimpan data yang diambil dari komponen etBarang untuk dijadikan string, dan variable stok untuk menyimpan data yang diambil dari etStok berupa string diubah menjadi float. Jika barang kosong atau stok 0.0 maka dia akan menampilkan data kosong, tetapi jika tidak kosong maka dia akan menjalankan function isiSharedPreferences untuk dikirimkan data dari variable barang dan stok juga menampilkan pesan Data Sudah Disimpan. Dan terakhir komponen etBarang dan etStok akan dikosongi. Karena tadi ada yang menjalankan function isiSharedPreferences, sebelumnya tambahkan code

sharedPreferences = getSharedPreferences("barang", *MODE\_PRIVATE*);

di dalam function load terlebih dahulu untuk membuat sharedpreferences dengan nama barang. kemudian function isisharedpreferencesnya

SharedPreferences.Editor editor = sharedPreferences.edit();  
  
editor.putString("barang", barang);  
editor.putFloat("stok", stok);  
editor.apply();

Class SharedPreferences.Editor dengan objek editor diisi dengan sharedPreferences yang sudah dibuat tadi. sharedPreferences tersebut sudah masuk ke dalam objek editor sehingga dengan objek editor dapat mengedit sharedPreferencesnya, terdapat editor.putString(“barang”, barang) yang berarti disimpan ke dalam data bentuk string dengan key barang dan value barang begitu juga yang float stok, kemudian editor ini diapply atau digunakan. Karena tadi function ini mendapat kiriman variable barang dan stok maka value barang diisi dengan var barang, dan value stok diisi dengan var stok. Jadi SharedPreferences ini sama halnya dengan table database yang memiliki key/index atau kolomnya dan valuenya diisi sesuai keynya atau kolomnya.

8. Pada bagian function onClick Tampil dari sharedPreferences diisi seperti ini

String barang = sharedPreferences.getString("barang", "");  
float stok = sharedPreferences.getFloat("stok", 0);  
  
etBarang.setText(barang);  
etStok.setText(stok+"");

Yang berarti String barang menyimpan data yang diambil dari sharedPreferences pada bagian key barang, kemudian float stok diisi dengan data yang diambil dari sharedPreferences pada bagian key stok. Karena sharedPreferences key barang dan stok tadi sebelumnya menyimpan data, maka var barang dan stok dapat diisi. Kemudian data tadi dimasukkan ke dalam komponen, var barang dimasukkan ke etBarang dan var stok (yang sebelumnya tersimpan float agar berubah menjadi string lagi) ke etStok. Jadi ketika button tampil diklik, maka komponen etBarang akan terisi dengan data dari sharedPreferences dan komponen etStok akan terisi dengan data dari sharedPreferences juga. Meskipun aplikasi sudah ditutup namun sharedPreferences ini masih menyimpan data yang sebelumnya disimpan.

9. Debug diambil dari kata bug yang berarti kutu atau yang menyebabkan error, karena kutu biasanya kecil begitu juga dengan error yang harus teliti diperhatikan. Error biasanya akan memunculkan sebuah tanda merah, misalkan ada salah penulisan code pada file MainActivity maka tulisan MainActivitynya dibawahnya terdapat garis bergerigik warna merah yang menandakan adanya error, atau ada tanda error di HighLight sebelah kanan atas dari android studio. Selain itu tanda error yang tidak terlihat misalkan pada proses sql yang biasanya menggunakan string seperti ini “SELECT \* FROM tblbarang“ pada bagian itu kurang huruf b nya jadi seperti ini “SELECT \* FROM tblarang” tidak ada pertanda error dalam android studionya namun ketika dibuka aplikasinya langsung tertutup sendiri, kesalahannya dapat dilihat di menu navbar bawah android studio bagian run, karena semua proses run dapat dilihat pada menu tersebut. Bisa juga dilihat di menu logcat, jadi bisa dicek semua sampai paham errornya bagian mana.

**Saya Belum Mengerti**

1.