|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Android Studio Minggu Ke-5SMST5 (Video 21-28 Android Studio Untuk Siswa SMK) | 95 |

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

1. Membuat sebuah Model yang berfungsi untuk menghubungkan database dengan adapter atau data dengan adapter.

2. Android menyediakan Context sebagai pengganti dari nama activity yang tidak diketahui. Contohnya private Context context yang berarti apabila kita tidak tahu activity mana yang tadi diberi nama “context”.

3. Di dalam Adapter terdapat

public ViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i)

yang berfungsi untuk menampilkan kembali sebuah function dari constructor yang telah dibuat sebelumnya, yaitu

return new ViewHolder(v);

4. Function selanjutnya yang masih ada di dalam adapter yaitu

public void onBindViewHolder(@NonNull ViewHolder viewHolder, int i)

yang berfungsi sebagai tempat inputan data. Di dalam function ini terdapat

Siswa siswa = siswaList.get(i);

Fungsinya untuk mengambil nilai komponen dari sebuah variable yang ada di dalam sebuah function yang sudah dibuat sebelumnya. Lalu ada

viewHolder.tvNama.setText(siswa.getNama());  
viewHolder.tvAlamat.setText(siswa.getAlamat());

yang berfungsi untuk mengisi variable tvNama ataupun tvAlamat yang ada di dalam function viewHolder dengan text yang diambil dari komponen id sebelumnya.

5. Ada sebuah tag yang fungsinya meload semua isi dari sebuah variable yang memiliki elemen dan sudah dibuat sebelumnya, tag tersebut adalah

return siswaList.size();

6. Penggunaan TextView tvNama, tvAlamat; untuk mengambil komponen yang telah dibuat sebelumnya.

7. tvNama = itemView.findViewById(R.id.yvNama); berfungsi untuk mengambil sebuah komponen berdasarkan id yang ada pada komponen tersebut.

8. Membuat statement baru untuk menampilkan sebuah function yang di dalamnya terdapat proses untuk menampilkan sebuah data dari recycleview dan adapter.

9. Untuk mengatur tampilan dari isi element sebuah function yang telah dibuat sebelumnya dengan cara

10. Karena sebelumnya model sudah dibuat sehingga adapter dan data bisa terhubung, selanjutnya adalah proses untuk menampilkan data yang dibuat. Pada main activity.xml terdapat komponen recycleview yang diberi id rcvSiswa, recycleview ini memiliki banyak item yang bisa diisi dengan komponen cardview dari xml lain. Di dalam mainactivity.java nya ada

RecyclerView recyclerView;

public void load(){  
 recyclerView = findViewById(R.id.*rcvSiswa*);  
 recyclerView.setLayoutManager(new LinearLayoutManager(this));  
}

yang berarti dibuatkan sebuah variable dengan type recycleview dimana variable ini disambungan dengan komponen yang memiliki id rcvSiswa, lalu variable ini diberi sebuah layout berupa LinearLayout agar tampilannya lebih rapi. Selanjutnya ada

SiswaAdapter adapter;  
List<Siswa> siswaList;

public void isiData(){  
 siswaList = new ArrayList<Siswa>();  
 siswaList.add(new Siswa("Joni", "Surabaya"));  
 siswaList.add(new Siswa("Eko", "Surabaya"));  
 siswaList.add(new Siswa("Tejo", "Surabaya"));

adapter = new SiswaAdapter(this, siswaList);  
 recyclerView.setAdapter(adapter);  
}

yang berarti dibuatkan variable untuk list sesuai dengan model Siswa bernama siswaList, siswaList ini kemudian diisi dengan ArrayList agar dapat menyimpan banyak data dengan indeks. Untuk menambahkan data ke dalam array tersebut caranya dengan mengetikkan nama variablenya tadi yaitu siswaList.add(new siswa()) di dalam siswa() ketika diketikkan “” akan otomatis tersambung bahwa elemen yang pertama dimasukkan adalah nama dan setelah koma adalah alamat, sesuai dengan yang disetting dalam modelnya. Kemudian dibuatkan variable bernama adapter sebagai adapter untuk diisi dengan siswaList tadi ke dalam adapternya. Setelah itu data yang disimpan dalam adapter akan disambungkan dengan recycleView dengan id rcvSiswa tadi agar data yang dimasukkan bisa ditampilkan.

11. Untuk mengambil sebuah data yang ada di dalam recycleview caranya dengan mengaturnya pada function onBindViewHolder pada file java adapternya. Di dalam function ini sebagai contoh mengambil datanya menggunakan onClick, jadi ketika item yang dipilih diklik akan muncul rincian data atau data yang dipilih sesuai yang disetting. Codenya

viewHolder.itemView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Toast.*makeText*(context, "Nama: "+siswa.getNama()+" Alamat: "+siswa.getAlamat(), Toast.*LENGTH\_SHORT*).show();  
 }  
});

Jadi ketika itemview yang ada di dalam recycleview diklik maka dia akan menjalankan function onClicknya yaitu menampilkan sebuah Toast atau pesan singkat yang berisi string Nama: lalu + siswa.getNama yang berarti dia akan mengambil data nama dari item yang dipilih begitu juga dengan alamatnya. Jadi untuk mengambil sebuah data recycleview menggunakan nama list itemnya yaitu siswa lalu “.” get dan nama data yang akan diambil misalnya data nama berarti getNama.

12. Menambahkan sebuah menu di dalam setiap itemnya, caranya dengan membuat menu\_option.xml di dalam folder menu yang ada di folder res. Folder menu ini ditambahkan dengan menekan new->Android Resource Directory->type menu->oke. Lalu di dalam file menu xml nya diisi dengan 2 macam menu item dengan title Simpan (id menu\_simpan) dan title Hapus (id menu\_hapus). Setelah itu di dalam item\_siswa.xml atau halaman tempat setting tampilan item ditambahkan sebuah textview dengan id tvMenu yang diisi dengan text &#8942; agar tampil seperti button titik 3. Setelah itu di dalam file java adapter pada bagian function onBindViewHolder ditambahkan code untuk setting apabila tvMenu diklik seperti ini

viewHolder.tvMenu.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {

kemudian diisi dengan

PopupMenu popupMenu = new PopupMenu(context, viewHolder.tvMenu);  
popupMenu.inflate(R.menu.*menu\_option*);

Yang berarti dibuatkan variable popupMenu ketika tvMenu diklik dari salah satu item, dan popupMenu ini diisi dengan data yang diambil dari menu\_option, jadi ketika titik 3 diklik akan muncul popupMenu yang isinya sesuai dengan menu\_option. Karena dari menu\_option tadi ada 2 macam menu yaitu simpan dan hapus, maka kedua menu itu disetting agar mengambil Langkah ketika salah satunya diklik. (sebelumnya sambungkan komponen tvMenu dengan variable tvMenu yang ada di file java menggunakan itemView.findViewById dalam function ViewHolder)

popupMenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {  
 @Override  
 public boolean onMenuItemClick(MenuItem item) {  
 switch (item.getItemId()){  
 case R.id.*menu\_simpan*:  
 Toast.*makeText*(context, "Simpan Data "+siswa.getNama(), Toast.*LENGTH\_SHORT*).show();  
 break;  
 case R.id.*menu\_hapus*:  
 siswaList.remove(i);  
 notifyDataSetChanged();  
 Toast.*makeText*(context, siswa.getNama()+" Sudah di Hapus", Toast.*LENGTH\_SHORT*).show();  
 break;  
 }  
 return false;  
 }  
});

berarti ketika salah satu menu dari popupMenu diklik maka dia akan menjalankan switch dimana switch ini memiliki 2 case, case yang pertama apabila yang diklik adalah menu simpan maka dia akan menjalankan Toast yang berisi pesan “Simpan Data :”+(nama item yang dipilih) sedangkan case yang kedua yaitu apabila yang diklik adalah menu hapus maka dia akan meremove atau menghapus item yang dipilih tadi dengan tetap memberikan sebuah notif apabila hapus berhasil, notifnya berupa Toast yang berisi (nama item yang dipilih)+“ Sudah di hapus”. Jika switch sudah jadi, terakhir ditambahkan

popupMenu.show();

agar popupMenu bisa ditampilkan dan dijalankan.

13. Untuk melihat perubahan sebuah data apabila ada penambahan maupun perubahan data dengan cara merefresh recycleview. Untuk merefresh atau memuat ulang data menggunakan

adapter.notifyDataSetChanged();

14. Download dan Install SQLite melalui web sqlitebrowser.org dengan memilih yang DB Browser for SQLite – Standar installer for 64-bit Windows.

15. Membuat table baru dalam database yang ada di SQLite dengan menekan Create Table, menambahkan kolom dengan cara menekan Add field, penggunakan SQLite sama halnya dengan Sql yang pernah dipelajari sebelumnya dalam php.

16. Untuk menambahkan isi dari kolom table yang ada di dalam database SQLite menggunakan code di Execute SQL caranya dengan mengetikkan INSERT INTO tblbarang (barang,stok,harga) VALUES ("Buku",5,3000) yang berarti tambahkan data ke dalam tblbarang dengan urutan untuk kolom barang, stok, dan harga. Lalu 3 kolom tersebut diisi dengan urut yaitu buku, 5, dan 3000. Dengan begitu kolom barang akan terisi dengan buku, stok dengan 5, dan harga dengan 3000.

17. Untuk menampilkan table melalui Execute SQL caranya dengan mengetikkan SELECT\*from lalu nama table nya yaitu tblbarang, jadinya seperti ini SELECT\*from tblbarang lalu ketika dijalankan dia akan menampilkan table yang dipilih tadi.

18. Jika sebelumnya menampilkan semua kolom pada sebuah table menggunakan SELECT\*from saja, maka untuk menampilkan salah satu kolom pada sebuah table caranya dengan mengetikkan SELECT namakolom from tblbarang.

19. Untuk mengubah atau mengupdate isi data dari sebuah kolom caranya dengan mengetikkan UPDATE namatable SET namakolom = data WHERE id = idnya. Contoh UPDATE tblbarang SET harga = 5000 WHERE idbarang = 1 yang berarti update pada tablebarang pada bagian kolom harga diganti dengan 5000, kolom yang dipilih yaitu yang ber-id 1.

20. Untuk menghapus data dari salah satu baris di sebuah table caranya dengan mengetikkan DELETE from namatable WHERE id = idnya. Contoh DELETE from tblbarang WHERE idbarang = 1 yang berarti hapus di dalam tblbarang pada baris yang ber-id 1.

21. Download dan Installasi MemuPlay untuk emulator yang baru (sudah di root), lalu menginstall sqlite reader dari playstore yang ada di emulator memu.

22. Menyambungkan Android Studio dengan emulator yang baru yaitu memu, caranya menggunakan usb debugging dari memu dan akan otomatis terdeteksi dalam Android Studionya.

23. Membuat java class untuk database di dalam folder yang sama dengan MainActivity.java, kemudian function Database nya ditambahkan extends SQLiteOpenHelper.

24. Di dalam function Database extends SQLiteOpenHelper diisi dengan

private static final String *DATABASE\_NAME* = "dbtoko";  
private static final int *VERSION* = 1;

Untuk membuat database dengan nama dbtoko dan version 1. Lalu ada

public Database(Context context) {  
 super(context, *DATABASE\_NAME*, null, *VERSION*);  
 db = this.getWritableDatabase();  
}

yang berfungsi untuk menulis atau membuat sebuah database baru yang akan dipakai dalam projek android studio.

25. Karena di dalam projek yang baru saja dibuat sudah memiliki database, maka ketika dijalankan di dalam emulator, aplikasi sqlite reader akan membaca adanya sqlite di dalam aplikasi dari projek tersebut. Apabila di dalam aplikasi sqlite reader pada menu private from apps tidak ada sama sekali, maka aktifkan root mode terlebih dahulu dari memunya. Setelah itu aplikasi yang memiliki database private akan muncul dalam menu private from apps dan bisa dicek aplikasi projek ada di dalamnya.

26. Untuk membuat sebuah table dan dimasukkan ke dalam database yang sudah dibuat di android studionya caranya dengan membuat function runSQL seperti ini

boolean runSQL(String sql){  
 try {  
  
 db.execSQL(sql);  
 return true;  
  
 }catch (Exception e){  
 return false;  
 }  
}

yang fungsinya untuk menjalankan sqlnya. Karena create table termasuk ke dalam sql maka function tersebut dibutuhkan dalam menjalankan sql. Lalu buat function seperti ini

public void buatTable(){  
 String tblbarang = "CREATE TABLE \"tblbarang\" (\n" +  
 "\t\"idbarang\"\tINTEGER,\n" +  
 "\t\"barang\"\tTEXT,\n" +  
 "\t\"stok\"\tREAL,\n" +  
 "\t\"harga\"\tREAL,\n" +  
 "\tPRIMARY KEY(\"idbarang\" AUTOINCREMENT)";  
  
 runSQL(tblbarang);  
}

yang berarti dibuatkan sebuah variable dengan type String bernama tblbarang yang diisi dengan code create table. Code ini didapatkan dari copas code untuk membuat table yang ada di DB Browser SQLite bagian modify table dari tblbarang. Kemudian tblbarang dikirimkan ke dalam database menggunakan runSQL. Dengan begitu table akan masuk ke dalam database dan bisa dicek melalui SQLite Reader pada bagian private from apps.

**Saya Belum Mengerti**

1.