

Praktikum #13

Program Aplikasi Mobile & Pemrograman Seluler

Dosen Pengampu: Arif Laksito, M. Kom

Update dan Delete data

Pada fitur update dan delete ini yang perlu kita perhatikan adalah saat menjalankan query pastikan telah menambahkan kondisi data yang akan dihapus. Misalnya kita akan menghapus data dengan kondisi id = ?

Kondisi tersebut bisa kita terapkan sebagai parameter dalam fungsi di class DBHelper

```
1. public void deleteUser(int id){  
2.     //kode untuk menghapus data  
3.  
4. }  
5.  
6. public int updateUser(int id, String nim, String name){  
7.     //kode untuk update data  
8.  
9. }
```

Codelab : Aplikasi Students

Pada Codelab ini kita akan melengkapi kode sebelumnya dengan menambahkan fitur update dan delete untuk data yang kita pilih di RecyclerView.

Konsep kerja aplikasi:

Melakukan input data → Melakukan update data → Menghapus data.

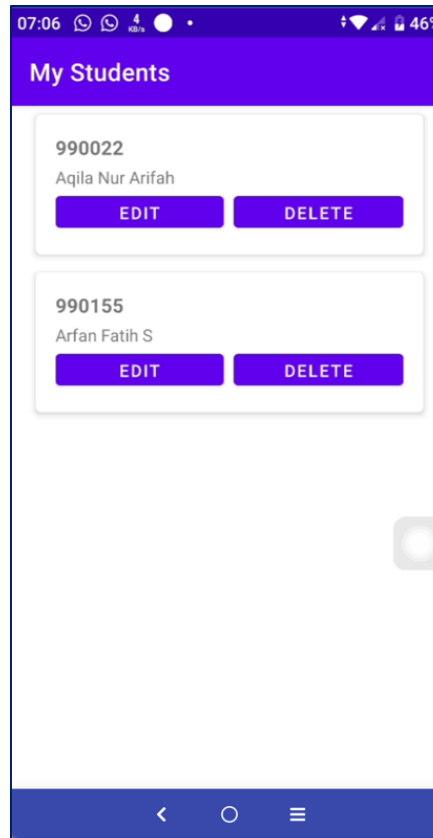
1. Buka file **DbHelper.java** kemudian tambahkan method atau fungsi berikut ini:

```
1. public void deleteUser(int id){
2.     SQLiteDatabase db = this.getWritableDatabase();
3.     db.delete(TABLE_STD, KEY_ID + " = ?",
4.         new String[]{String.valueOf(id)});
5. }
6.
7. public int updateUser(int id, String nim, String name){
8.     SQLiteDatabase db = this.getWritableDatabase();
9.
10.    ContentValues values = new ContentValues();
11.    values.put(KEY_NAME, name);
12.    values.put(KEY_NIM, nim);
13.
14.    return db.update(TABLE_STD, values, KEY_ID + " = ?",
15.        new String[]{String.valueOf(id)});
16.
17. }
```

2. Edit file layout **item_student.xml** dan tambahkan 2 tombol untuk delete dan update dengan kode dibawah ini:

```
1. <LinearLayout
2.     android:layout_width="match_parent"
3.     android:layout_height="match_parent"
4.     android:orientation="horizontal">
5.
6.     <Button
7.         android:id="@+id/btn_edit"
8.         android:layout_width="wrap_content"
9.         android:layout_height="38dp"
10.        android:layout_weight="1"
11.        android:text="Edit" />
12.
13.     <Button
14.         android:id="@+id/btn_delete"
15.         android:layout_marginStart="8dp"
16.         android:layout_width="wrap_content"
17.         android:layout_height="38dp"
18.         android:layout_weight="1"
19.         android:text="Delete" />
20. </LinearLayout>
```

Setelah anda tambahkan kode tersebut, maka tampilan dari list student akan seperti berikut:



3. Update file **StudentAdapter.java** dengan menambahkan atribut pada class **StudentViewHolder** dan memberi nilai sesuai id pada layout xml (baris 7 dan baris 11-12)

```
1. public class StudentViewHolder extends RecyclerView.ViewHolder {  
2.  
3.     final TextView tvNim, tvName;  
4.     final Button btnEdit, btnDelete;  
5.  
6.     public StudentViewHolder(@NonNull View itemView) {  
7.         super(itemView);  
8.         btnEdit = itemView.findViewById(R.id.btn_edit);  
9.         btnDelete = itemView.findViewById(R.id.btn_delete);  
10.        tvNim = itemView.findViewById(R.id.tv_item_nim);  
11.        tvName = itemView.findViewById(R.id.tv_item_name);  
12.  
13.    }  
14. }
```

4. Selanjutnya masih di file **StudentAdapter.java** pada method **onBindViewHolder** tambahkan action untuk menghapus data dan intent untuk menuju ke activity UpdateActivity

```

1. @Override
2. public void onBindViewHolder(@NonNull StudentViewHolder holder, int position) {
3.     holder.tvNim.setText(listStudents.get(position).getNim());
4.     holder.tvName.setText(listStudents.get(position).getName());
5.     holder.btnEdit.setOnClickListener((View v) -> {
6.         Intent intent = new Intent(activity, UpdateActivity.class);
7.         intent.putExtra("user", (Serializable) listStudents.get(position));
8.         activity.startActivity(intent);
9.     });
10.
11.     holder.btnDelete.setOnClickListener((View v) -> {
12.         AlertDialog.Builder builder = new AlertDialog.Builder(activity);
13.
14.         builder.setTitle("Konfirmasi hapus");
15.         builder.setMessage("Apakah yakin akan dihapus?");
16.
17.         builder.setPositiveButton("YA", new DialogInterface.OnClickListener() {
18.             @Override
19.             public void onClick(DialogInterface dialog, int which) {
20.                 dbHelper.deleteUser(listStudents.get(position).getId());
21.                 Toast.makeText(activity, "Hapus berhasil!", Toast.LENGTH_SHORT).show();
22.                 Intent myIntent = new Intent(activity, ListStudentsActivity.class);
23.                 activity.startActivity(myIntent);
24.                 activity.finish();
25.             }
26.         });
27.
28.         builder.setNegativeButton("TIDAK", new DialogInterface.OnClickListener() {
29.             @Override
30.             public void onClick(DialogInterface dialog, int which) {
31.                 dialog.dismiss();
32.             }
33.         });
34.
35.         AlertDialog alert = builder.create();
36.         alert.show();
37.
38.     });
39. }

```

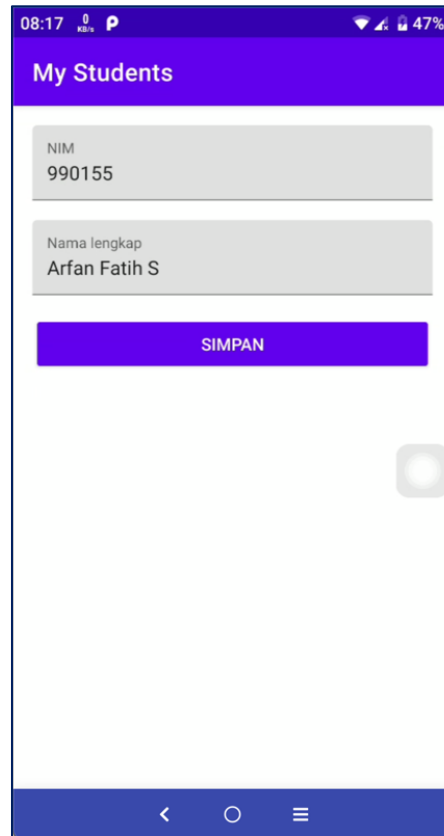
5. Pada baris ke-7 terdapat intent yang mengirimkan data berupa **object Student**, supaya object tersebut dapat dikirim melalui intent maka class Student perlu di implementasikan ke **Serializable**. Buka file class **Student.java** kemudia tambahkan implement berikut:

```

1. public class Student implements Serializable {
2.     //kode sebelumnya
3.
4. }

```

6. Create activity baru dengan nama **UpdateActivity**, kemudian layout bisa di **copy** dari **activity_main.xml** dengan **menghilangkan** komponen button daftar student. Tampilan untuk layout seperti berikut:



7. Selanjutnya lengkapi kode pada activity UpdateActivity dengan fungsi update seperti ini:

```

1. @Override
2. protected void onCreate(Bundle savedInstanceState) {
3.     super.onCreate(savedInstanceState);
4.     setContentView(R.layout.activity_update);
5.
6.     dbHelper = new DbHelper(this);
7.
8.     etName = findViewById(R.id.edt_name);
9.     etNim = findViewById(R.id.edt_nim);
10.    btnSave = findViewById(R.id.btn_submit);
11.
12.    Intent intent = getIntent();
13.    student = (Student)intent.getSerializableExtra("user");
14.
15.    etName.setText(student.getName());
16.    etNim.setText(student.getNim());
17.
18.    btnSave.setOnClickListener((View v) -> {
19.        dbHelper.updateUser(student.getId(),etNim.getText().toString(),etName.getText().toString());
20.        Toast.makeText(UpdateActivity.this, "Updated berhasil!", Toast.LENGTH_SHORT).show();
21.        finish();
22.    });
23.
24. }

```

8. Setelah proses edit berhasil maka akan ditampilkan list daftar student, supaya **RecyclerView** dapat update data terakhir maka perlu ditambahkan override method di activity **ListStudentsActivity** seperti berikut:

```
1. @Override
2. protected void onResume() {
3.     super.onResume();
4.     studentsArrayList = dbHelper.getAllUsers();
5.     adapter.setListStudents(studentsArrayList);
6.     adapter.notifyDataSetChanged();
7. }
```

9. Silahkan di run aplikasi tersebut dan bisa diujicoba untuk dilakukan penambahan, perubahan dan penghapusan data.

Referensi

Dicoding Acedemy - Belajar Fundamental Aplikasi Android, Codelab Aplikasi Catatan,
<https://www.dicoding.com/academies/14/tutorials/527>

Android SQLite Tutorial | CRUD Operation Example, <https://demonuts.com/android-sqlite/#simple>