|  |  |
| --- | --- |
| Materi | Nilai |
| Penjelasan tentang pembuatan dan proses kerja layout di dalam AndroidStudio  (UjiKompetensi-4 #16) | 95 |

**Deskripsi Tentang Layout Di Dalam Android Studio**

1. Recyclerview adalah komponen dasar yang digunakan untuk menampilkan data berupa list dan dapat digunakan untuk menampung ribuan bahkan ratusan ribu data tanpa mengalami lag.
2. Recyclerview memiliki 3 komponen utama, yaitu layout, viewholder, dan adapter.
3. Jadi alur untuk membuat dan menampilkan recycler view adalah, mengisi data > data dimasukkan ke dalam adapter > membuat adapter > isi dari adapter akan dimasukkan ke dalam viewholder > ditampilkan ke dalam layout.
4. Viewholder berisi objek dimana setiap item yang ditampilkan pada recyclerview adalah objek ViewHolder. Kita perlu menyediakan layout XML untuk setiap item yang akan ditampilkan dimana semua komponen yang ada di layout XML ini akan digunakan di objek ViewHolder.
5. Adapter digunakan sebagai perantara untuk menghubungkan data dengan tampilan. Adapter menerima atau mengambil data, melakukan pekerjaan yang diperlukan agar data dapat ditampilkan.
6. Setiap viewholder menampung satu rangkaian data dan adapter menambahkan data ke viewholder untuk ditampilkan oleh pengelola layout.
7. Model pada android studio digunakan untuk menghubungkan database dan adapter. Untuk masuk ke adapter harus membuat model dulu agar bisa terhubung.
8. Constructor adalah method / function khusus yang akan dijalankan ketika pembuatan objek dan digunakan untuk mempersiapkan data untuk objek.
9. Set() adalah fungsi yang digunakan untuk mengatur / mengisi data yang akan dikirimkan.
10. Get() adalah fungsi yang digunakan untuk mengambil nilai data.
11. Nama dari method setter dan getter pada android studio harus diawali dengan set untuk setter dan diawali dengan get untuk getter.
12. Cara untuk membuat method setter dan getter adalah, klik kanan > generate > setter and getter > pilih variabel apa yang akan ditambahkan method setter and getter > finish.
13. itemView adalah tag yang digunakan ketika pemanggilan item menggunakan RecyclerView.
14. Mendeklarasikan variabel activity yang tidak diketahui yang mana yang akan digunakan dapat menggunakan variabel context yang sudah disediakan oleh android studio. Jadi ketika activity dipanggil, maka akan otomatis ditambahkan.
15. Function onCreateVIewHolder digunakan untuk membuat objek viewholder dan akan menentukan layout XML yang akan digunakan untuk setiap item.
16. Function onBindViewHolder digunakan untuk menghubungkan data yang ada dengan objek ViewHolder.
17. Function getItemCount digunakan untuk melihat / menghitung ada berapa banyak data pada list di dalam barangList. Dan untuk menampilkan jumlah barang di dalam getItemCount diberi fungsi size() setelah barangList dipanggil.
18. LayoutInflater adalah library yang digunakan untuk memasukkan dan mengconvert semua data file xml yang ada ke dalam recyclerview.
19. getContext() digunakan untuk mengambil nilai dari context yang akan dipanggil saat proses pemanggilan layout.
20. Inflate() adalah fungsi dari LayoutInflater yang digunakan untuk mengisi view baru dari spesifik xml resource. Parameter dari inflate() ini berisi resource, root, attached to root.
21. Resource berisi id dari xml layout resource yang akan di load.
22. Root breisi view parent yang akan dipanggil.
23. Attachtoroot berisi jika bernilai salah, root hanya digunakan untuk membuat subclass yang sesuai dengan LayoutParams untuk root di dalam view pada xml.
24. setLayoutManager() adalah perintah yang digunakan untuk mengatur layoutmanager dari recyclerview yang akan digunakan.
25. LinearLayoutManager() adalah perintah di dalam LayoutManager yang digunakan untuk membuat vertical linearlayoutmanager.
26. setAdapter() adalah perintah yang digunakan untuk mengatur adapter apa yang dijalankan untuk menyimpan data yang diambil.
27. Scrollview adalah container pada tampilan UI dalam android yang digunakan agar user bisa melakukan scroll secara vertikal pada layout yang sudah dibuat jika layout terlalu besar.
28. Scroll view hanya bisa memiliki satu elemen anak, jadi jika ingin menambahkan beberapa elemen sekaligus agar dapat di scroll, kita harus membungkus elemen – elemen tersebut menggunakan viewgroup (relative layout, linear layout, dan yang lain).
29. Membuat bagian history untuk admin. Bagian history pada admin ini digunakan untuk melihat siapa saja yang sudah pernah untuk melakukan penyewaan mobil dan menampilkan detail data mobil yang disewa apa saja, tanggal penyewaan dan tanggal pengembalian mobil, serta harga total selama penyewaan itu berapa. Sehingga admin tau untung yang akan dibayarkan pada developer.
30. Pertama, menambahkan icon history ke dalam folder drawable dengan ukuran 27dp x 27dp.
31. Kemudian, membuat file menu resource file dengan nama history, dimana di dalamnya memanggil icon history yang ada pada drawable yang kemudian ditampilkan ke dalam file menu tersebut, menambahkan id yaitu iconHistory, menambahkan title yaitu History, dan menambahkan fungsi app:showAsAction always, yang berarti dimana icon ini akan selalu ditampilkan.

<menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
  
 <item  
 android:id="@+id/iconHistory"  
 android:icon="@drawable/ic\_history"  
 android:title="History"  
 app:showAsAction="always" />  
  
</menu>

Hasil dari code yang sudah dibuat adalah :

Graphical user interface, text, application

Description automatically generated

1. App:showAsAction adalah perintah yang digunakan untuk menampilkan icon. Apabila showAsAction dipilih always, maka tampilan icon akan selalu ditampilkan.
2. Membuat aksi Ketika icon history di klik, maka akan diarahkan ke dalam tampilan history admin.

@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 if (id == R.id.*iconProfil*) {  
  
 Intent intent = new Intent(HomeAdmin.this, ProfilAdmin.class);  
 startActivity(intent);  
 }  
  
 if (id == R.id.*iconHistory*){  
 Intent intent = new Intent(HomeAdmin.this, HistoryAdmin.class);  
 startActivity(intent);  
 }  
  
 return super.onOptionsItemSelected(item);  
}

Karena di dalam action bar terdapat 2 icon, yaitu icon profil dan icon history, maka terdapat 2 pengujian di dalam function onOptionsItemSelected. Yang pertama adalah apabila icon yang diklik adalah icon profil dengan mengambil id iconProfil, maka halaman akan diarahkan ke dalam halaman profil.

Pengujian yang kedua adalah apabila icon yang diklik adalah icon history dengan mengambil nilai id adalah iconHistory, maka halaman akan diarahkan ke dalam halaman history admin.

1. Karena nilai dari history akan bertambah seiring dengan banyaknya penyewa mobil, maka dari itu, untuk membuat tampilan history diperlukan recyclerview di dalamnya. Jadi pertama kita membuat file xml dengan nama item\_history\_admin dulu, dimana file tersebut digunakan sebagai template untuk mengisi data yang akan ditampilkan.

Graphical user interface, text, application

Description automatically generated

Di dalam isi dari item\_history, kita harus melakukan pengaturan pada tampilan untuk jarak dan margin agar ketika aplikasi dijalankan terlihat rapi. Terdapat beberapa textview yang digunakan sebagai isi dari data. Pertama terdapat nama penyewa, alamat penyewa, tanggal menyewa, tanggal Kembali, jenis mobil, merk mobil, dan total pembayaran oleh penyewa sebanyak berapa. Setiap textview diberi id masing – masing, karena nanti, setiap nilai dari textview akan diisi dan untuk pengisian itu panggil menggunakan id yang sudah diberi.

1. Membuat model HistoryAdminModel yang digunakan untuk memasukkan data ke dalam adapter.

Karena di dalam item\_history\_admin, semua data yang ada dalam textview akan diambil, jadi pertama kita mendeklarasikan nama – nama dari komponen yang dibutuhkan.

Kedua, membuat constructor dari komponen – komponen yang sudah dideklarasikan.

Ketiga, membuat function get dan set untuk setiap komponen yang dibutuhkan.

Nama – nama dari komponen yang dideklarasikan :

private String namapenyewa;  
private String alamatpenyewa;  
private String tglsewa;  
private String tglkembali;  
private String jenismobil;  
private String merkmobil;  
private String hargatotal;

Constructor yang dibuat :

public HistoryAdminModel(String namapenyewa, String alamatpenyewa, String tglsewa, String tglkembali, String jenismobil, String merkmobil, String hargatotal) {  
 this.namapenyewa = namapenyewa;  
 this.alamatpenyewa = alamatpenyewa;  
 this.tglsewa = tglsewa;  
 this.tglkembali = tglkembali;  
 this.jenismobil = jenismobil;  
 this.merkmobil = merkmobil;  
 this.hargatotal = hargatotal;  
}

Function setter dan getter yang dibuat :

public String getNamapenyewa() {  
 return namapenyewa;  
}  
  
public void setNamapenyewa(String namapenyewa) {  
 this.namapenyewa = namapenyewa;  
}  
  
public String getAlamatpenyewa() {  
 return alamatpenyewa;  
}  
  
public void setAlamatpenyewa(String alamatpenyewa) {  
 this.alamatpenyewa = alamatpenyewa;  
}  
  
public String getTglsewa() {  
 return tglsewa;  
}  
  
public void setTglsewa(String tglsewa) {  
 this.tglsewa = tglsewa;  
}  
  
public String getTglkembali() {  
 return tglkembali;  
}  
  
public void setTglkembali(String tglkembali) {  
 this.tglkembali = tglkembali;  
}  
  
public String getJenismobil() {  
 return jenismobil;  
}  
  
public void setJenismobil(String jenismobil) {  
 this.jenismobil = jenismobil;  
}  
  
public String getMerkmobil() {  
 return merkmobil;  
}  
  
public void setMerkmobil(String merkmobil) {  
 this.merkmobil = merkmobil;  
}  
  
public String getHargatotal() {  
 return hargatotal;  
}  
  
public void setHargatotal(String hargatotal) {  
 this.hargatotal = hargatotal;  
}

1. Setelah membuat model, selanjutnya membuat adapter dari HistoryAdmin. Dimana digunakan untuk megambil data yang masuk dan mengirimkan data ke dalam viewholder untuk ditampilkan.

Pertama, membuat class ViewHolder yang isinya adalah mengenalkan komponen – komponen ke dalam viewholder yang mengambil komponen dari file item\_list\_mobil.

public class ViewHolder extends RecyclerView.ViewHolder{  
  
 TextView tvNamaPenyewa, tvAlamatPenyewa, tvTglSewa, tvTglKembali, tvJenisMobil, tvMerkMobil, tvHargaTotal;  
  
 public ViewHolder(@NonNull View itemView) {  
 super(itemView);  
  
 tvNamaPenyewa = itemView.findViewById(R.id.*tvNamaPenyewa*);  
 tvAlamatPenyewa = itemView.findViewById(R.id.*tvAlamatPenyewa*);  
 tvTglSewa = itemView.findViewById(R.id.*tvTanggalMenyewa*);  
 tvTglKembali = itemView.findViewById(R.id.*tvTanggalKembali*);  
 tvJenisMobil = itemView.findViewById(R.id.*tvJenisMobil*);  
 tvMerkMobil = itemView.findViewById(R.id.*tvHistoryMerk*);  
 tvHargaTotal = itemView.findViewById(R.id.*tvHargaTotal*);  
  
 }  
}

class ViewHolder yang disambungkan ke dalam recyclerview memanggil beberapa komponen yang dibutuhkan sesuai dengan id dari komponen yang ada pada item\_history\_admin. Yaitu memanggil komponen tvNamaPenyewa, tvAlamatPenyewa, tvTanggalMenyewa, tvTanggalKembali, tvJenisMobil, tvHistoryMerk, dan tvHargaTotal.

1. Membuat private variable untuk activity apa yang dipanggil dan data apa saja yang dimasukkan, serta membuat constructor untuk kedua variable. Jadi, ketila HistoryAdminAdapter dijalankan, maka dia akan mencari siapa yang menggunakan data dan data apa yang dimasukkan, dimana data – data yang dibutuhkan sudah disimpan ke dalam constructor.

private Context context;  
private List<HistoryAdminModel> historyAdminList;  
  
public HistoryAdminAdapter(Context context, List<HistoryAdminModel> historyAdminList) {  
 this.context = context;  
 this.historyAdminList = historyAdminList;  
}

1. Function onCreateViewHolder digunakan untuk membuat objek viewHolder dan memanggil layout xml mana yang akan digunakan untuk setiap data yang akan dimasukkan.

@NonNull  
@Override  
public ViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {  
 View v = LayoutInflater.*from*(viewGroup.getContext()).inflate(R.layout.*item\_history\_admin*, viewGroup, false);  
  
 return new ViewHolder(v);  
}

Function onCreateViewHolder mempunyai parameter yang berisi viewGroup dan viewType. Di dalam function terdapat variable v dimana LayoutInflater digunakan untuk membuat objek ViewHolder yang memperoleh data context dari perintah from dengan fungsi getContext untuk memperoleh data dari viewGroup. Kemudian inflate yang mengisi view dari r.layout.item\_history\_admin, viewGroup, dan attachtoroot yang bernilai false.

Selanjutnya nilai direturn dengan memanggil function ViewHolder yang di dalamnya dipanggil variable v.

1. Selanjutnya mengisi function onBindViewHolder, dimana function ini digunakan untuk menghubungkan data yang ada dengan objek ViewHolder.

@Override  
public void onBindViewHolder(@NonNull ViewHolder viewHolder, int i) {  
HistoryAdminModel historyAdminModel = historyAdminList.get(i);  
  
 viewHolder.tvNamaPenyewa.setText(historyAdminModel.getNamapenyewa());  
 viewHolder.tvAlamatPenyewa.setText(historyAdminModel.getAlamatpenyewa());  
 viewHolder.tvTglSewa.setText(historyAdminModel.getTglsewa());  
 viewHolder.tvTglKembali.setText(historyAdminModel.getTglkembali());  
 viewHolder.tvJenisMobil.setText(historyAdminModel.getJenismobil());  
 viewHolder.tvMerkMobil.setText(historyAdminModel.getMerkmobil());  
 viewHolder.tvHargaTotal.setText(historyAdminModel.getHargatotal());  
}

Pada function onBindViewHolder memanggil class HistoryAdminModel dan membuat variable Bernama historyAdminModel yang mengambil nilai dari historyAdminList dengan menggunakan perintah get().

Menyambungkan isi dari text ke dalam viewholder dengan mengambil data tvNamaPenyewa, tvAlamatPenyewa, tvTglSewa, tvTglKembali, tvJenisMobil, tvMerkmobil, dann tvHargaTotal dari historyAdminModel lalu dikirimkan ke dalam viewHolder.

1. Function getItemCount digunakan untuk menghitung banyaknya data pada historyAdminList menggunakan perintah size().

@Override  
public int getItemCount() {  
 return historyAdminList.size();  
}

1. Membuat activity baru dengan nama historyadmin. Di dalam emptyactivity akan terdapat 2 file yang akan dibuat, yaitu file java dan file xml. Selanjutnya pada file xml dari history admin, membuat recycler view dimana file item\_history\_admin akan dimasukkan ke dalam recyclerview dari historyadmin.

Di dalam file xml historyadmin, terdapat scrollview dimana awalnya isi dari scroll view adalah linear layout, kemudian komponen linearlayout dihapus dan diganti dengan recyclerview. Recyclerview membutuhkan scrollview karena isi data yang akan disimpan di recyclerview akan bertambah, jadi untuk dapat menyimpan dan melihat semua data, maka recyclerview disimpan ke dalam scrollview.

Setelah memasukkan komponen – komponen yang dibutuhkan, kemudian kita mengatur jarak dan margin agar setiap komponen terlihat rapi dan gampang untuk dilihat.

Memberikan id untuk recyclerview yaitu rcvHistoryAdmin. Karena id dari recyclerview ini dibutuhkan ketika mengenalkan dan memanggil recyclerview yang akan dimasuki oleh data.

A picture containing graphical user interface

Description automatically generated

1. Selanjutnya, pada file java HistoryAdmin, pertama adalah mendeklarasikan komponen – komponen yang digunakan. Yaitu RecyclerView dengan nama recyclerView, HistoryAdminAdapter dengan nama adapter, dan memanggil List<HistoryAdminModel> dengan nama historyAdminModels.

RecyclerView recyclerView;  
HistoryAdminAdapter adapter;  
List<HistoryAdminModel> historyAdminModels;

1. Memanggil recyclerview yang akan diisi ke dalam HistoryAdmin menggunakan function load().

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

Memanggil recyclerview dengan menggunakan id yaitu rcvHistoryAdmin. Kemudian mengatur tampilan recyclerview dengan perintah setLayoutManager dimana di dalamnya terdapat perintah LinearLayourManager untuk membuat vertical linearlayoutmanager dengan data yang diambil adalah data dalam function (this).

1. Mengisi data dan memasukkannya ke dalam model kemudian model disambungkan ke dalam adapter agar data dapat ditampilkan.

public void isiData(){  
 historyAdminModels = new ArrayList<HistoryAdminModel>();  
 historyAdminModels.add(new HistoryAdminModel("Penyewa 1", "Jl.Mawar", "28 - 02 -2021", "02 - 03 - 2021", "Toyota", "Avanza", "750000"));  
 historyAdminModels.add(new HistoryAdminModel("Penyewa 2", "Jl.Kamboja", "12 - 03 -2021", "14 - 03 - 2021", "Toyota", "Avanza", "750000"));  
 historyAdminModels.add(new HistoryAdminModel("Penyewa 3", "Jl.Melati", "17 - 03 -2021", "19 - 03 - 2021", "Toyota", "Avanza", "750000"));  
 historyAdminModels.add(new HistoryAdminModel("Penyewa 4", "Jl.Anggrek", "25 - 03 -2021", "27 - 03 - 2021", "Toyota", "Avanza", "750000"));  
  
  
 adapter = new HistoryAdminAdapter(this, historyAdminModels);  
 recyclerView.setAdapter(adapter);  
  
}

Mendeklarasikan historyAdminModels kemudian membbuat ArrayList baru dengan memasukkannya ke dalam model HistoryAdminModel. Sehingga isi dari data yang akan dimasukkan akan sesuai dengan data yang ada di dalam model.

Menambahkan data ke dalam historyAdminModels dengan menggunakan perintah add().

Mengisi variable adapter dengan memanggil HistoryAdminModel dan mengisi parameternya dengan this dan memanggil historyAdminModels.

Mengatur adapter apa yang akan dijalankan ke dalam recyclerview dengan menggunakan perintah setAdapter dan menjalankan variable adapter.

1. Hasil dari history admin yang sudah dibuat menggunakan adapter dan recyclerview.

Table

Description automatically generated

1. Menambahkan button back yang ketika button back di klik, maka akan kembali ke halaman sebelumnya.

@Override  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_history\_admin*);  
  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
 getSupportActionBar().setDisplayShowHomeEnabled(true);  
  
 load();  
 isiData();  
}

@Override  
public boolean onSupportNavigateUp() {  
 onBackPressed();  
 return true;  
}

1. Selanjutnya, membuat history user dengan Langkah – langkah yang sama dengan pembuatan history pada admin.
2. Melakukan presentasi dan mendapatkan beberapa hal yang perlu direvisi, yaitu :
3. Menambahkan filter pada halaman user agar user dapat mencari beberapa data mobil sesuai yang diinginkan.
4. Menghapus button jenis mobil dan merk mobil, dan menggantinya dengan list mobil, yang di klik akan menampilkan detail sewa dari mobil.
5. Menghapus button tambah mobil dan update pada halaman homeadmin, karena mempunyai fungsi yang sama dengan list mobil ketika mobil tersebut diklik.
6. Menampilkan history pada admin sesuai dengan tanggal sewa yang terbaru.