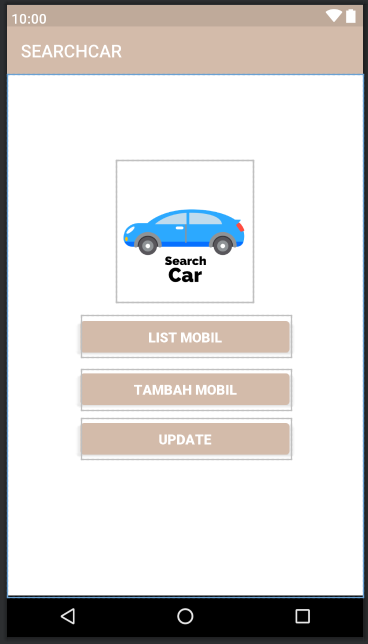
|  |  |
| --- | --- |
| Materi | Nilai |
| Penjelasan tentang pembuatan dan proses kerja layout di dalam AndroidStudio  (UjiKompetensi-3 #15) | 95 |

**Deskripsi Tentang Layout Di Dalam Android Studio**

1. Melanjutkan kembali tampilan layout untuk admin, yaitu mengerjakan tampilan homeadmin, dimana ketika admin berhasil login atau masuk ke dalam aplikasi, setelah itu admin akan diarahkan ke tampilan homeadmin. Dimana di dalam homeadmin ini berisi 3 button dan satu icon profil pada pojok kanan. 3 button tersebut yaitu, list mobil, tambah mobil, dan update.



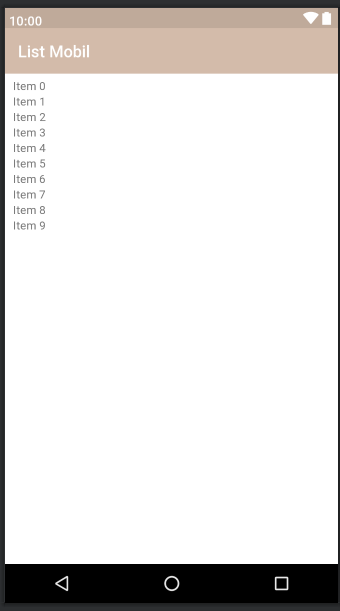
Jadi, ketika admin mengklik button list mobil, maka akan diarahkan ke dalam halaman list mobil yang berisi data mobil apa saja yang tersedia. Ketika admin mengklik tambah mobil, maka halaman yang ditampilkan adalah halaman form untuk tambah mobil. Dan ketika button yang diklik adalah button update mobil, maka halaman yang ditampilkan adalah halaman form untuk update mobil dengan mengambil nilai id dari data yang dipilih.

Admin akan mengetahui data – data yang sudah dimasukkan lewat profil, data – data yang dapat dilihat admin diantaranya, data nama, alaman, dan nomor telepon.

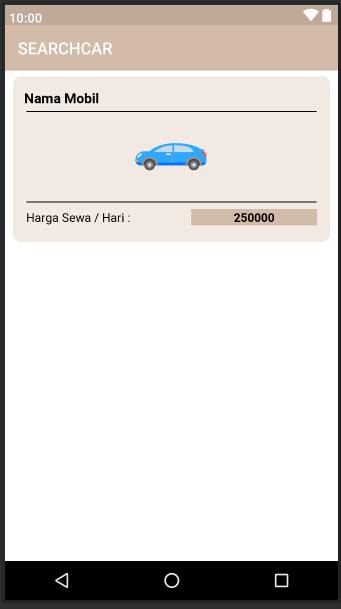
1. Recyclerview adalah komponen dasar yang digunakan untuk menampilkan data berupa list dan dapat digunakan untuk menampung ribuan bahkan ratusan ribu data tanpa mengalami lag.
2. Recyclerview memiliki 3 komponen utama, yaitu layout, viewholder, dan adapter.
3. Jadi alur untuk membuat dan menampilkan 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. Pada tampilan list mobil terdapat data mobil apa saja yang tersedia. Karena data yang ditampilkan akan bertambah atau berkurang dan isinya juga ada yang berbeda, maka dari itu tidak mungkin jika kita menambahkan letak dan tampilan secara manual atau satu per satu. Jadi untuk membuat data yang isinya banyak dan berubah sesuai data yang dimasukkan oleh admin, maka di dalam tampilan list mobil ini menggunakan recyclerview. Di dalam listmobil juga terdapat icon tambah, jadi ketika admin mengklik icon tersebut, maka akan diarahkan ke halaman tambah mobil yang berisi form. Dan juga terdapat icon back, jadi ketika icon tersebut di klik, maka halaman yang ditampilkan adalah halaman sebelumnya. Yaitu halaman homeadmin.



1. Untuk membuat tampilan seperti pada gambar di atas, pertama kita harus membuat activity baru (empty activity) yang berisi java dan xml file. Setelah itu, di dalam file xml, kita menambahkan scrollview agar ketika data yang ada lebih dari size screen layar, maka tampilan bisa di scroll untuk melihat data yang lainnya. Kemudian, karena di dalam scrollview terdapat linearlayout, kita hapus linearlayoutnya dan kita ganti dengan recycler view. Selanjutnya, kita tinggal mengatur jarak dan margin agar sesuai.



1. Membuat file xml item\_list\_mobil yang digunakan sebagai template data dimana file xml tersebut akan disambungkan ke dalam adapter.



Pertama membuat constraintlayout dan mengatur jarak dan margin agar ketika data yang ditambahkan otomatis berjarak. Menambahkan komponen – komponen yang dibutuhkan di dalamnya, seperti textview, imageview, dan cardview yang digunakan sebagai garis bawah.

Kedua, memberi id dari komponen yang akan diisi data, karena untuk memasukkan data, di dalam adapter, komponen harus dikenalkan ke dalam java agar bisa diproses.

1. Karena file xml untuk template sudah dibuat, jadi kita membuat model untuk memasukkan data dan menghubungkannya ke dalam adapter. Karena data di dalam template item\_list\_mobil yang akan berubah hanya data pada merkmobil, hargasewa, dan gambarmobil, maka kita hanya mendeklarasikan 3 komponen itu saja. Setelah itu, kita harus membuat constructor kemudian membuat function set dan get untuk setiap komponen.

public class Mobil {  
  
 private String merkmobil, hargasewa;  
 private Integer gambarmobil;  
  
 public String getMerkmobil() {  
 return merkmobil;  
 }  
  
 public void setMerkmobil(String merkmobil) {  
 this.merkmobil = merkmobil;  
 }  
  
 public String getHargasewa() {  
 return hargasewa;  
 }  
  
 public void setHargasewa(String hargasewa) {  
 this.hargasewa = hargasewa;  
 }  
  
 public Integer getGambarmobil() {  
 return gambarmobil;  
 }  
  
 public void setGambarmobil(Integer gambarmobil) {  
 this.gambarmobil = gambarmobil;  
 }  
  
 public Mobil(String merkmobil, String hargasewa, Integer gambarmobil) {  
 this.merkmobil = merkmobil;  
 this.hargasewa = hargasewa;  
 this.gambarmobil = gambarmobil;  
 }  
}

1. Setelah membuat model, kita masuk ke dalam file ListMobilAdapter dan membuat class ViewHolder yang isinya adalah mengenalkan komponen – komponen ke dalam viewholder yang akan diambil pada file xml item\_list\_mobil.

public class ViewHolder extends RecyclerView.ViewHolder{  
  
 TextView tvMerkMobil, tvHargaSewa;  
 ImageView ivGambarMobil;  
  
  
 public ViewHolder(@NonNull View itemView) {  
 super(itemView);  
  
 tvMerkMobil = itemView.findViewById(R.id.*tvMerkMobil*);  
 tvHargaSewa = itemView.findViewById(R.id.*tvHargaSewa*);  
 ivGambarMobil = itemView.findViewById(R.id.*ivMobil*);  
 }  
}

class ViewHolder yang disambungkan ke dalam recyclerview, class tersebut memanggil tvMerkMobil menggunakan id dan memasukkannya ke dalam variabel tvMerkMobil. Memanggil tvHargaSewa menggunakan id dan memasukkannya ke dalam variabel tvHargaSewa. Terakhir, memanggil ivMobil dan memasukkannya ke dalam variabel ivGambarMobil. Karena di dalam superclass menggunakan itemView, jadi setiap komponen yang diambil menggunakan id harus diberi itemView di depannya.

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

private Context context;  
private List<Mobil> mobilList;  
  
public ListMobilAdapter(Context context, List<Mobil> mobilList) {  
 this.context = context;  
 this.mobilList = mobilList;  
}

Ketika ListMobilAdapter dijalankan, maka adapter akan membutuhkan 2 data, yaitu siapa yang akan menggunakan yang disimpan ke dalam variabel private context dan data apa yang dimasukkan, dimana disimpan ke dalam variabel mobilList dimana List memanggil model Mobil.

Kemudian, variabel context dan mobilList disimpan ke dalam constructor ListMobilAdapter.

Jadi, ketika ListMobilAdapter berjalan, maka dia akan memanggil 2 data yang sudah disimpan ke dalam constructor.

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 viewType) {  
 View v = LayoutInflater.*from*(viewGroup.getContext()).inflate(R.layout.*item\_list\_mobil*, viewGroup, false);  
  
 return new ViewHolder(v);  
}

Function onCreateViewHolder mempunyai parameter yang berisi viewGroup dan viewType. Di dalam function terdapat variabel v dimana terdapat LayoutInflater yang 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 resource.layout. item\_list\_mobil, viewGroup, dan attachtoroot yang bernilai false.

Selanjutnya, nilai direturn dengan memanggil function Viewholder yang di dalamnya dipanggil variabel v.

1. Glide adalah image library yang digunakan utnuk menampilkan gambar, video, dan animasi. Dimana di dalam glide ini kita dapat melakukan pengaturan atau edit seperti untuk resize, rotate, atau imagecatching.
2. Untuk membuat glide, pertama kita harus mendaftarkan library dengan menambahkan beberapa kode di dalam build.grade(module:app).

implementation 'com.github.bumptech.glide:glide:4.9.0'  
 annotationProcessor 'com.github.bumptech.glide:compiler:4.9.0'  
**}**

1. With pada glide digunakan untuk memasukkan parameter yang berisi activity atau fragment.
2. Load pada glide digunakan untuk memuat gambar yang akan ditampilkan, dapat diambil dari local ataupun internet.
3. Placeholder pada glide digunakan untuk menampilkan gambar pengganti ketika gambar utama belum selesai diload.
4. Centercrop pada glide digunakan untuk membuat gambar di crop menengah.
5. Into pada glide digunakan utnuk komponen tujuan untuk menampilkan gambar, biasanya berupa imageview.
6. Function onBindViewHolder digunakan untuk menghubungkan data yang ada dengan objek ViewHolder.

@Override  
public void onBindViewHolder(@NonNull ViewHolder viewHolder, int i) {  
  
 Mobil mobil = mobilList.get(i);  
  
 viewHolder.tvMerkMobil.setText(mobil.getMerkmobil());  
 viewHolder.tvHargaSewa.setText(mobil.getHargasewa());  
 Glide.*with*(context)  
 .load(mobil.getGambarmobil())  
 .placeholder(R.drawable.*logo*)  
 .into(viewHolder.ivGambarMobil);  
  
}

Pada function onBindViewHolder memanggil class Mobil dan membuat variabel bernama mobil yang mengambil nilai dari mobilList.

Menyambungkan isi dari text ke dalam viewholder dengan mengambil data merkmobil menggunakan perintah getMerkMobil yang akan diisikan ke dalam tvMerkMobil lalu dikirimkan ke dalam viewholder.

Menyambungkan isi dari text ke dalam viewholder dengan mengambil data hargasewa menggunakan perintah getHargaSewa yang akan diisikan ke dalam tvHargaSewa lalu dikirimkan ke dalam viewholder.

Menggunakan library glide untuk menampilkan gambar yang diambil dari perintah getGambarMobil lalu memasukkannya ke dalam viewholder dengan komponen ivGambarMobil.

1. Function getItemCount digunakan untuk menghitung banyaknya data yang ada pada list di dalam mobilList.

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

Pada function getItemCount digunakan untuk menghitung banyaknya data yang ada pada mobilList menggunakan perintah size().

1. Memanggil recyclerview yang akan diisi ke dalam file ListMobil.

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

Memanggil recyclerview dengan menggunakan id yaitu rcvListMobil. Setelah itu mengatur tampilan recyclerview dengan perintah setLayoutManager dimana di dalamnya terdapat perintah LinearLayoutManager 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(){  
 mobilList = new ArrayList<Mobil>();  
 mobilList.add(new Mobil("Avanza", "250000", R.drawable.*avanza*));  
 mobilList.add(new Mobil("Avanza", "250000", R.drawable.*avanza*));  
 mobilList.add(new Mobil("Avanza", "250000", R.drawable.*avanza*));  
 mobilList.add(new Mobil("Avanza", "250000", R.drawable.*avanza*));  
 mobilList.add(new Mobil("Avanza", "250000", R.drawable.*avanza*));  
 mobilList.add(new Mobil("Avanza", "250000", R.drawable.*avanza*));  
 mobilList.add(new Mobil("Avanza", "250000", R.drawable.*avanza*));  
  
 adapter = new ListMobilAdapter(this, mobilList);  
 recyclerView.setAdapter(adapter);  
}

Mendeklarasikan mobilList yang membuat ArrayList baru dengan memasukkannya ke dalam model Mobil. Sehingga isi dari data yang akan dimasukkan akan sesuai dengan data yang ada di dalam model.

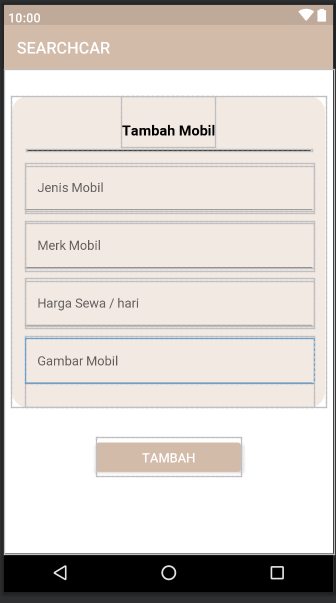
Menambahkan data ke dalam mobilList dengan menggunakan perintah add().

Mengisi variabel adapter dengan memanggil ListMobilAdapter dan mengisi parameternya dengan this dan memanggil mobilList.

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

1. Tampilan tambah mobil ketika admin mengklik button tambah mobil pada homeadmin. Di dalam tambah mobil terdapat form yang digunakan ketika admin ingin menambahkan data mobil yang tersedia. Di dalam form terdapat beberapa data yang harus diisi seperti jenis mobil, merk mobil, harga sewa / hari dan gambar mobil, data – data yang diisi tersebut menggunakan textview. Ketika semua data sudah terisi dan admin mengklik button tambah, maka data akan ditambahkan dan diberi pesan notifikasi jika data yang ditambahkan berhasil atau gagal.

Di dalam tampilan tambahmobil juga terdapat icon home di sebelah pojok kanan, yang ketika di klik, maka tampilan akan diarahkan ke tampilan home yaitu homeadmin.



1. Intent adalah perintah yang digunakan untuk berpindah halaman dari satu activity ke activity yang lain.
2. Memasukkan icon home yang ada di folder menu ke dalam tambahmobil.java.

@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*home\_menu*, menu);  
 return true;  
}  
  
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 if (id == R.id.*iconHome*) {  
  
 Intent intent = new Intent(TambahMobil.this, HomeAdmin.class);  
 startActivity(intent);  
 return true;  
 }  
  
 return super.onOptionsItemSelected(item);  
}

Memanggil file xml home\_menu kemudian menginflatenya dan variabel menu yang akan menerima inflate.

Memanggil item sesuai dengan id nya menggunakan perintah getItemId, kemudian melakukan pengujian apabila id yang diambil sama dengan iconHome maka halaman akan dipindahkan menggunakan perintah intent.

Intent akan memindahkan halaman tambahmobil ke dalam halaman homeadmin.

1. getMenuInflater() adalah perintah yang digunakan untuk membuat file xml menu menjadi objek menu dan menginflate menu yang dipanggil.
2. Memindahkan tampilan ketika btnTambah di klik dan menampilkan pesan notifikasi menggunakan toast.

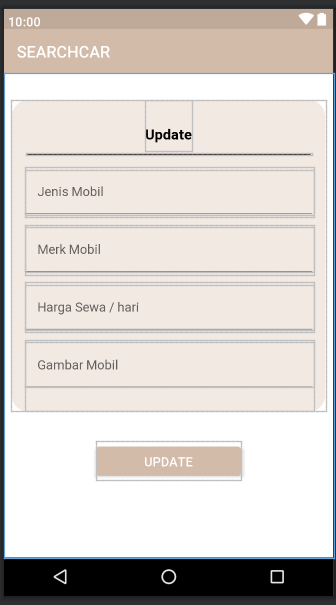
public void btnTambah(View view) {  
 Intent intent = new Intent(this, HomeAdmin.class);  
 startActivity(intent);  
  
 Toast.*makeText*(this, "Data Mobil Berhasil Ditambahkan", Toast.*LENGTH\_LONG*).show();  
}

Apabila btnTambah di klik, maka halaman akan dipindahkan menggunakan intent, yaitu halaman tambahmobil akan diarahkan ke halaman homeadmin.

Setelah itu akan muncul pesan notifikasi bahwa data mobil berhasil ditambahkan.

1. Tampilan update mobil ketika admin mengklik button update pada homeadmin. Di dalam update terdapat form yang digunakan ketika admin ingin mengubah data mobil yang tersedia. Di dalam form terdapat beberapa data yang dapat diubah seperti jenis mobil, merk mobil, harga sewa / hari dan gambar mobil, data – data yang diisi tersebut menggunakan textview. Ketika data yang ingin diubah sudah diganti dan admin mengklik button update, maka data akan diperbarui dan diberi pesan notifikasi jika data yang diupdate berhasil atau gagal.

Di dalam tampilan update juga terdapat icon home di sebelah pojok kanan, yang ketika di klik, maka tampilan akan diarahkan ke tampilan home yaitu homeadmin.



1. Memindahkan tampilan ketika btnUpdate di klik kemudian ketika berhasil update maka pesan notifikasi berhasil atau gagal update akan dikirim.

public void btnUpdate(View view) {  
 Intent intent = new Intent(this, HomeAdmin.class);  
 startActivity(intent);  
  
 Toast.*makeText*(this, "Data Mobil Berhasil Di-update", Toast.*LENGTH\_LONG*).show();  
}

Apabila btnUpdate di klik, maka halaman akan dipindahkan menggunakan intent, yaitu halaman update akan diarahkan ke halaman homeadmin.

Setelah itu akan muncul pesan notifikasi menggunakan toast bahwa data mobil berhasil diperbarui atau diupdate.

1. Ketika button profil pada homeadmin diklik, maka tampilan akan diarahkan ke dalam tampilan profil dimana admin dapat melihat data – data yang tersedia. Selanjutnya, di dalam profil juga terdapat button logout, dimana aka nada alertdialog untuk memastikan apakah admin benar – benar ingin logout dari aplikasi.

Di dalam tampilan profil admin, juga terdapat icon home, dimana ketika icon home diklik, maka akan diarahkan kembali ke tampilan homeadmin.

1. Menampilkan alertdialog ketika btnLogout diklik.

public void btnLogout(View view) {  
 new AlertDialog.Builder(this)  
 .setIcon(R.drawable.*logoapk*)  
 .setTitle("Logout Aplikasi")  
 .setMessage("Apakah anda yakin ingin logout dari posisi admin?")  
 .setPositiveButton("OK", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 Intent intent = new Intent(ProfilAdmin.this, LoginUser.class);  
 startActivity(intent);  
 }  
 })  
 .setNegativeButton("Cancel", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
  
 }  
 })  
 .show();  
}

Ketika btnLogout diklik, maka akan muncul tampilan alertdialog untuk memastikan apakah admin benar – benar ingin keluar dari aplikasi.

Alertdialog dijalankan di dalam function itu sehingga tertulis this, dengan mengatur icon yang mengambil gambar dengan nama logoapk dari folder drawable. Mengatur judul dengan isi Logout Aplikasi dan mengatur pesan yang ada di dalam alertdialog dengan isi Apakah anda yakin ingin logout dari posisi admin?.

Apabila admin mengklik OK, maka akan diarahkan kembali ke halaman login awal. Dan ketika admin mengklik cancel, maka admin akan tetap ke dalam halaman profil.

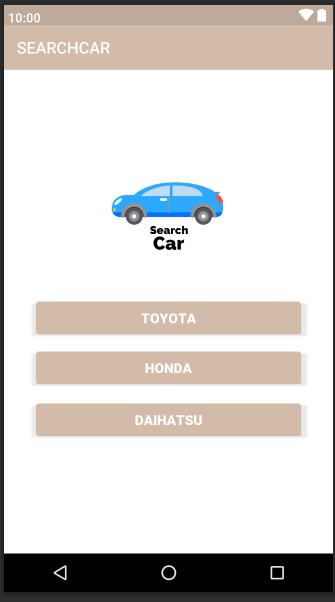
1. Tampilan homeuser atau mainactivity akan dijalankan ketika user berhasil login. Di dalam tampilan mainactivity ini terdapat 3 button yang berisi jenis mobil yang tersedia di dalam aplikasi, yaitu Toyota, Honda, dan Daihatsu.

Ketika button Toyota diklik, maka halaman akan diarahkan ke halaman list mobil dengan jenis Toyota.

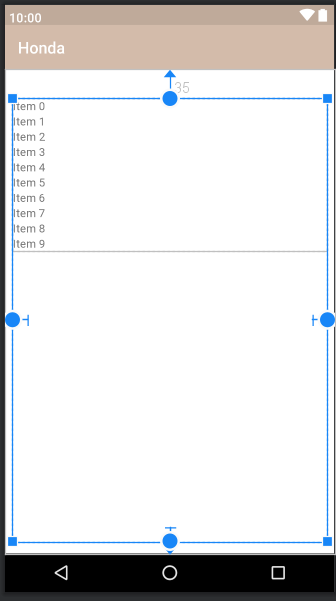
Ketika button Honda diklik, maka halaman akan diarahkan ke halaman list mobil dengan jenis Honda.

Ketika button Daihatsu diklik, maka halaman akan diarahkan ke halaman list mobil dengan jenis Daihatsu.

Di dalam mainactivity juga terdapat icon profil di sebelah kanan atas, dimana ketika icon profil diklik, maka akan diarahkan ke dalam tampilan profil user.

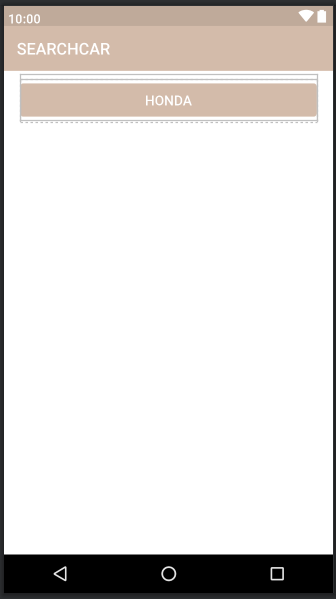


1. Karena tampilan dari list mobil untuk Toyota, Honda, dan Daihatsu akan bertambah sesuai data yang akan dimasukkan oleh admin, maka dari itu, untuk membuat list mobil – mobil tersebut, pertama saya membuat file emptyactivity dengan nama listtoyota, listhonda, dan listdaihatsu.



Isi dari file activity\_list\_honda, activity\_list\_toyota, dan activity\_list\_daihatsu sama. Yaitu di dalam constraint layout berisi scrollview dan recyclerview. Scroll view digunakan agar jika data yang tersedia memakan tampilan lebih dari ukuran layar hp, tampilan dapat discroll ke atas, sehingga user dapat melihat data yang lebih banyak. Di dalam scrollview terdapat recyclerview, setelah memasukkan recyclerview. Pengaturan margin, jarak, dan ukuran layout sangatlah penting, maka dari itu, kita harus mengatur jarak dan ukuran layout pada scrollview dan recyclerview, agar ketika aplikasi dijalankan di device yang lain, isi dari tampilan tetap dan tidak berantakan.

1. Selanjutnya, membuat file xml untuk setiap list, file xml ini akan digunakan sebagai template untuk mengisi recycllerview.



Gambar tersebut merupakan contoh tampilan ui dari file item\_list\_honda, jadi di dalam file xml itemlisthonda terdapat constraintlayout yang diisi dengan button. Pertama melakukan pengaturan jarak dan margin pada constraint layout, kemudian melakukan pengaturan pada button, yaitu ukuran teks, teksstyle, warna button, dan jarak margin agar setiap data yang masuk nanti memiliki jarak dan tidak berdempet sehingga terlihat rapi.

1. Membuat file model untuk setiap jenis mobil.

public class Honda {  
 private String btnhonda;  
  
 public Honda(String btnhonda) {  
 this.btnhonda = btnhonda;  
 }  
  
 public String getBtnhonda() {  
 return btnhonda;  
 }  
  
 public void setBtnhonda(String btnhonda) {  
 this.btnhonda = btnhonda;  
 }  
}

Pertama, mendeklarasikan jenis data apa yang akan diambil. Karena isi dari button adalah teks, maka tipe data yang digunakan untuk mendeklarasikan adalah string. Selanjutnya membuat constructor btnhonda. Kemudian, membuat function get dan set untuk mendapatkan dan mengatur nilai dari btnhonda.

1. Setelah membuat model, selanjutnya membuat file adapter untuk menghubungkan data dan tampilan yang akan diisi. untuk mengisi file ListHondaAdapter, pertama adalah dengan membuat membuat class ViewHolder yang isinya mengenalkan komponen – komponen ke dalam viewholder yang akan diambil pada file xml item\_list\_honda.

public class ViewHolder extends RecyclerView.ViewHolder{  
  
 Button btnListHonda;  
  
 public ViewHolder(@NonNull View itemView) {  
 super(itemView);  
  
 btnListHonda = itemView.findViewById(R.id.*btnListHonda*);  
 }  
}

class ViewHolder yang disambungkan ke dalam recyclerview, lalu button dideklarasikan dengan nama btnListHonda. Class tersebut memanggil btnListHonda menggunakan id dan memasukkannya ke dalam variabel btnListHonda. Karena di dalam superclass menggunakan itemView, jadi setiap komponen yang diambil menggunakan id harus diberi itemView di depannya.

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

private Context context;  
private List<Honda> hondaList;  
  
public ListHondaAdapter(Context context, List<Honda> hondaList) {  
 this.context = context;  
 this.hondaList = hondaList;  
}

Ketika ListHondaAdapter dijalankan, maka adapter akan membutuhkan 2 data, yaitu siapa yang akan menggunakan dan disimpan ke dalam variabel private context dan data apa yang dimasukkan, dimana disimpan ke dalam variabel hondaList dimana List memanggil model Honda.

Kemudian, variabel context dan hondaList disimpan ke dalam constructor ListHondaAdapter.

Jadi, ketika ListHondaAdapter berjalan, maka dia akan memanggil 2 data yang sudah disimpan ke dalam constructor.

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\_list\_honda*, viewGroup, false);  
  
 return new ViewHolder(v);  
}

Function onCreateViewHolder mempunyai parameter yang berisi viewGroup dan i. Di dalam function terdapat variabel v dimana terdapat LayoutInflater yang 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 resource.layout. item\_list\_honda, viewGroup, dan attachtoroot yang bernilai false.

Selanjutnya, nilai direturn dengan memanggil function Viewholder yang di dalamnya dipanggil variabel v.

1. Function onBindViewHolder digunakan untuk menghubungkan data yang ada dengan objek ViewHolder.

@Override  
public void onBindViewHolder(@NonNull ViewHolder viewHolder, int i) {  
  
 Honda honda = hondaList.get(i);  
  
 viewHolder.btnListHonda.setText(honda.getBtnhonda());  
  
}

Pada function onBindViewHolder memanggil class Honda dari model Honda dan membuat variabel bernama mobil yang mengambil nilai dari mobilList.

Menyambungkan isi dari text ke dalam viewholder dengan mengambil data list honda menggunakan perintah getBtnhonda yang akan diisikan ke dalam btnListHonda lalu dikirimkan ke dalam viewholder.

1. Function getItemCount digunakan untuk menghitung banyaknya data yang ada pada list di dalam mobilList.

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

Pada function getItemCount digunakan untuk menghitung banyaknya data yang ada pada hondaList menggunakan perintah size().

1. Setelah selesai membuat file model dan file adapter, selanjutnya di dalam file ListHonda kita memanggil recyclerview yang akan dimasukkan data untuk list mobil Honda. Pertama, mendeklarasikan dulu komponen komponen yang akan digunakan, yaitu recyclerview, dan model.

RecyclerView recyclerView;  
ListHondaAdapter adapter;  
List<Honda> hondaList;

Mendeklarasikan komponen yang akan digunakan, yaitu Recyclerview dengan nama recyclerView, memanggil ListHondaAdapter dan menyimpannya dengan nama adapter, dan memanggil model Honda dan memasukkan nya ke dalam <List> serta menyimpannya ke dalam hondaList.

1. Memanggil recyclerview yang akan diisi ke dalam file ListMobil.

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

Memanggil recyclerview dengan menggunakan id yaitu rcvListHonda. Setelah itu mengatur tampilan recyclerview dengan perintah setLayoutManager dimana di dalamnya terdapat perintah LinearLayoutManager 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(){  
hondaList = new ArrayList<Honda>();  
  
hondaList.add(new Honda("Jazz"));  
hondaList.add(new Honda("Mobilio"));  
hondaList.add(new Honda("Brio"));  
hondaList.add(new Honda("HR-V"));  
  
adapter = new ListHondaAdapter(this, hondaList);  
recyclerView.setAdapter(adapter);

}

Mendeklarasikan hondaList yang membuat ArrayList baru dengan memasukkannya ke dalam model Honda. Sehingga isi dari data yang akan dimasukkan akan sesuai dengan data yang ada di dalam model.

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

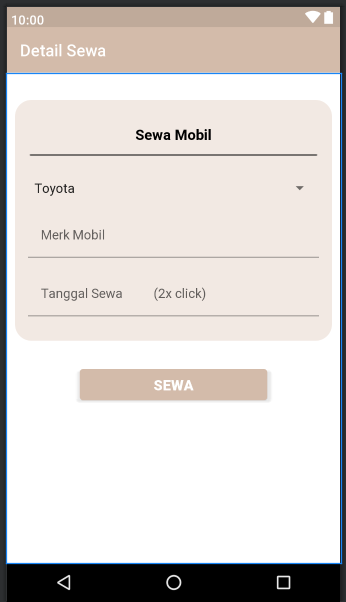
Mengisi variabel adapter dengan memanggil ListHondaAdapter dan mengisi parameternya dengan this dan memanggil hondList.

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

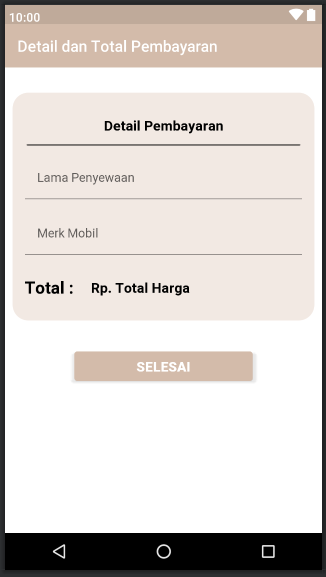
1. Membuat tampilan detail sewa untuk user apabila ingin melanjutkan transaksi sewa. Di tampilan ini, user memilih jenis mobil apa dan memasukkan merk mobil serta tanggal kapan user memulai untuk menyewa mobil.

Pembuatan menggunakan spinner untuk membuat jenis mobil, karena jenis mobil yang dapat dipilih hanya 3, yaitu Toyota, Honda, dan Daihatsu. Setelah itu untuk merk mobil menggunakan komponen teksinputlayout. Dan terakhir untuk tanggal sewa, karena basisnya adalah kalender, maka menggunakan class Calendar dan kalender akan muncul ketika form kalender di klik 2 kali.

Setelah itu, apabila semua form sudah terisi, maka user mengklik tombol sewa, selanjutnya akan diarahkan ke halaman detail\_pembayaran untuk melanjutkan transaksi dan mengetahui detail pembayaran.



1. Membuat tampilan detail dan total pembayaran yang akan ditampilkan setelah user mengklik tombol sewa pada detail sewa.



Di dalam tampilan detail pembayaran, terdapat 2 form yaitu lama penyewaan dan merk mobil dimana form tersebut menggunakan teksinputeditlayout.

Selanjutnya Total harga akan ditampilkan setelah dilakukan penghitungan untuk lamahari sewa dikalikan dengan merkmobil yang disewa.

1. Setelah mengisi detail pembayaran, maka tampilan akan diarahkan ke dalam tampilan kode pembayaran. User akan menerima tampilan kode qr mereka, yang jika discan, maka akan berisi informasi tentang data pemesanan yang dimasukkan. Sehingga setelah user sampai di tempat penyewaan, maka user tinggal menscan dan dapat menyewa mobil setelah melakukan pembayaran. Setelah user sudah menerima kode pembayaran, user harus meng screenshot kode qr yang didapatkan. Dan jika user mengklik button selesai, maka tampilan akan diarahkan kembali ke dalam tampilan mainactivity atau tampilan homeuser.

