|  |  |
| --- | --- |
| Materi | Nilai |
| Penjelasan tentang pembuatan dan proses kerja layout di dalam AndroidStudio  (UjiKompetensi-2 #14) | 95 |

**Deskripsi Tentang Layout Di Dalam Android Studio**

1. Mengatur warna dari tampilan pada aplikasi searchcar, yaitu dengan memasukkan kode warna ke dalam file colors.xml yang ada pada folder res > values > colors.xml.
2. Mengatur tampilan dari statusbar menjadi lebih gelap dengan mengisi android:statusBarColor dengan nama dari warna colorPrimarySurface.
3. Android:statusBarColor adalah atribut yang digunakan untuk mengatur tampilan warna dari status bar pada aplikasi.
4. Mengubah tema aplikasi yang awalnya dark (gelap) menjadi berwarna terang (light) dengan memasukkan atribut Theme.MaterialComponents.Light.DarkActionBar
5. Android:drawable adalah atribut yang digunakan untuk membuat tampilan yang mengambil gambar atau memasukkan warna latar.
6. Android:gravity adalah atribut yang digunakan untuk mengatur posisi di dalam view.
7. Android:width adalah atribut yang digunakan untuk mengatur panjang dari gambar.
8. Android:height adalah perintah yang digunakan untuk mengatur lebar dari gambar.
9. Android:src adalah atribut yang digunakan untuk memanggil posisi / letak dari komponen bergambar yang dipanggil.
10. <layer-list> adalah tag yang berisi elemen akar boleh berisi satu atau beberapa elemen item.
11. <item> adalah tag yang mendefiniskan komponen – komponen yang dapat dimasukkan adalah komponen yang bergambar. Letak dari <item> ini adalah di dalam <layer-list>.
12. <bitmap> adalah tag yang digunakan untuk mendefinisikan sumber bitmap dan propertinya.
13. Membuat splashscreen untuk aplikasi, pertama membuat file splashscreen\_bg.xml ke dalam folder /res/drawable dan mengisikan code :

<?xml version="1.0" encoding="utf-8"?>  
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">  
  
 <item android:drawable="@color/purple\_200"/>  
  
 <item android:gravity="center" android:width="150dp" android:height="150dp">  
 <bitmap  
 android:gravity="fill\_horizontal|fill\_vertical"  
 android:src="@drawable/logoapk"/>  
 </item>  
  
</layer-list>

Di dalam file xml, berisi memasukkan warna latar dari splashscreen, yaitu dengan atribut android:drawable. Lalu mengatur posisi di dalam view menggunakan atribut android:gravity dimana letak dari icon yang dimasukkan adalah di tengah (center) serta mengisi posisi secara horizontal dan vertikal sesuai dengan ukuran panjang dan lebar yang dibuat, yaitu 150dp. Dan memanggil path/posisi dari logo, yaitu dengan nama logoapk di dalam folder drawable.

1. Android:windowBackground adalah atribut yang digunakan untuk mengatur tampilan latar belakang background dengan satu warna tertentu.
2. Membuat file style.xml di dalam folder res > values > style.xml yang berisi memanggil isi dari file splashscreen\_bg pada folder drawable menggunakan atribut android:windowBackground yang memabaca file splashscreen\_bg dan mengisikan code :

<resources>  
 <style name="SplashTheme" parent="Theme.AppCompat.Light.NoActionBar">  
 <item name="android:windowBackground">@drawable/splashscreen\_bg</item>  
 </style>  
</resources>

1. Membuat activity baru dengan nama SplashActivity, dimana karena file SplashActivity ini hanya digunakan untuk splashscreen, maka tidak perlu memberikan layout, jadi setContentView dihapus kemudian menuliskan code :

protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
 // langsung pindah ke MainActivity atau activity lain  
 // begitu memasuki splash screen ini  
 Intent intent = new Intent(this, MainActivity.class);  
 startActivity(intent);  
 finish();  
}

Setelah menjalankan halaman splashactivity, maka halaman akan otomatis pindah ke dalam halaman dari main acitivity.

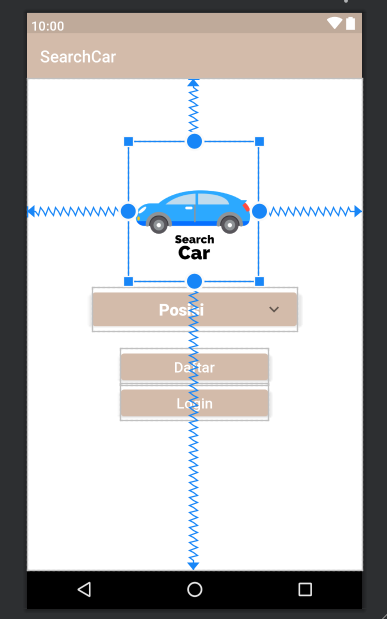
1. Intent adalah perintah yang digunakan untuk berpindah halaman dari satu activity ke activity yang lain.
2. <intent-filter> adalah atribut yang di dalamnya berisi lebih dari satu tindakan, lalu android akan mencocokkan kegiatan yang disebutkan di dalamnya.
3. Memasukkan activity ke dalam androidmanifest dan mengatur SplashActivity sebagai activity yang dipanggil pertama kali setelah aplikasi dibuka.

<activity android:name=".SplashActivity" android:theme="@style/SplashTheme">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
</activity>  
  
<activity android:name=".MainActivity">  
  
 <!-- Hapus Intent Filter MainActivity -->  
</activity>

Mendaftarkan activity dengan android:name adalah .SplashActivity dan android:theme yang memanggil SplashTheme pada file style. Di dalam intent-filter terdapat 2 tindakan, yaitu <action> dan <category>. <action> berisi main, dimana terdapat action MAIN yaitu sebagai titik masuk utama. Dan category LAUNCHER yaitu sebagai activity yang ditampilkan pertama kali.

Karena activity pertama yang akan dilakukan adalah splashscreen, jadi intent filter yang ada di dalam mainactivity harus dihapus.

1. Membuat tampilan awal setelah splashscreen yaitu tampilan login, dimana pada tampilan ini user diminta untuk memilih posisi dirinya sebagai admin atau sebagai user (penyewa). Pertama, menampilkan logo dengan memasukkan logo ke dalam folder drawable. Kemudian membuat button yang apabila diklik maka akan menampilkan 2 opsi, sebagai admin atau user. Ketiga setelah memilih posisi, apabila user belom mempunyai akun, maka mengklik button daftar, tetapi jika user sudah mempunyai akun, maka memilih button login.



1. Membuat dropdown menu untuk menampilkan opsi pada posisi.

Di dalam file dropdown\_menu.xml membuat 2 item yang menjadi opsi, yaitu item admin dan user.

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
  
 <item android:id="@+id/admin"  
 android:title="Admin"/>  
  
 <item android:id="@+id/user"  
 android:title="User"/>  
</menu>

1. Di dalam file LoginUser.java membuat code ketika jika button dropdown di klik, maka akan memilih item sesuai dengan item yang di klik. Pertama, mendeksripsikan nama dari Popupmenu yang akan dibuat yaitu dropDownMenu. Kemudian, perintah inflate yang memanggil xml dropdown\_menu untuk mengisi view dari dropdownmenu. Selanjutnya, membuat action untuk setiap item yang di klik di dalam function setOnMenuItemClickListener yang di dalamnya mendapatkan id dari setiap menu yang di klik. terdapat switch dengan case, mendapatkan id dari setiap item, jika id nya adalah admin, maka mengganti tulisan button dengan Admin. Dan jika id nya adalah user, maka mengganti tulisan button dengan user. Setelah itu dropdown ditampilkan dengan perintah show().

button.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 PopupMenu dropDownMenu = new PopupMenu(getApplicationContext(), button);  
 dropDownMenu.getMenuInflater().inflate(R.menu.*dropdown\_menu*, dropDownMenu.getMenu());  
 dropDownMenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {  
  
 @Override  
 public boolean onMenuItemClick(MenuItem item) {  
 switch (item.getItemId()){  
 case R.id.*admin* :  
 button.setText("Admin");  
 break;  
  
 case R.id.*user* :  
 button.setText("User");  
 break;  
 }  
 return false;  
 }  
 });  
 dropDownMenu.show();  
 }  
});

1. Melakukan pengujian dengan mengambil posisi yang sudah dimasukkan. Apabila btnDaftar diklik, maka akan menjalankan pengujian yang pertama, yaitu dengan mengambil text dari button, apabila text sama dengan admin, maka akan diarahkan ke halaman DaftarAdmin.

Pengujian kedua, apabila text yang diambil adalah user, maka hanya menampilkan pesan notifikasi saja.

Juga melakukan pengujian pada btnLogin, apabila btnLogin diklik, maka

public void btnDaftar(View view) {  
 if (button.getText()=="Admin"){  
 Intent intent = new Intent(this, DaftarAdmin.class);  
 startActivity(intent);  
 }  
  
 if (button.getText()=="User"){  
 Toast.*makeText*(this, "USER", Toast.*LENGTH\_LONG*).show();  
 }  
}

public void btnLogin(View view) {  
 if (button.getText()=="Admin"){  
 Intent intent = new Intent(this, HomeAdmin.class);  
 startActivity(intent);  
 }  
  
 if (button.getText()=="User"){  
 Toast.*makeText*(this, "USER", Toast.*LENGTH\_LONG*).show();  
 }  
  
}

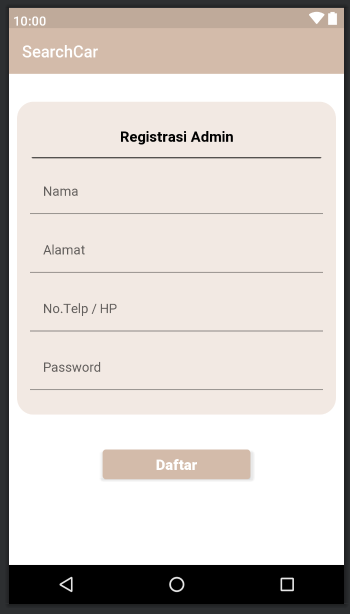
1. Constraintlayout adalah salah satu komponen viewgroup yang dapat digunakan untuk menyusun tampilan aplikasi yang kompleks tanpa nestedlayout. Di dalam constraintlayout menggunakan drag and drop pada tab design dan menambahkan button yang ada pada palette, lalu di drag dan drop ke dalam constraintlayout pada component tree. Selanjutnya kita harus mengatur posisi dari view agar view terhubung dengan parent layout lain.

Pada setiap view setidaknya harus mempunyai minimal satu vertikal dan horizontal constraint.

1. Pengaturan posisi dari setiap komponen pada constraint layout digunakan agar ketika tampilan layar membesar, maka posisi dari komponen tidak berubah, karena setiap komponen sudah diatur masing – masing berapa ukuran jarak dan posisi pada parentlayout nya.
2. Wrapcontent artinya lebar dari komponen mengikuti isi di dalam komponen.
3. Fixed artinya isi komponen tidak memengaruhi lebar yang sudah ditentukan.
4. Matchconstrains artinya lebar komponen akan menyesuaikan lebar layar ditambah dengan jarak yang sudah ditambahkan.
5. Layout\_width adalah attribute yang digunakan untuk mengatur lebar dari layout.
6. Layout\_height adalah attribute yang digunakan untuk mengatur panjang dari layout.
7. Inputtype adalah atribut yang digunakan untuk mengatur tipe dari teks yang dimasukkan.
8. Intent activity adalah sebuah jembatan yang menghubungkan interaksi antar Activity di aplikasi android dan juga dapat digunakan untuk membawa dan mengirimkan data ke Activity lain atau ke aplikasi lain.
9. startActivity() adalah function yang digunakan untuk memulai activity.
10. Toast adalah pesan teks yang ditampilkan pada android yang muncul secafa pop-up dan berfungsi untuk memberikan informasi dalam bentuk teks terkait aktivitas yang sedang dilakukan yang hilang dalam batas waktu tertentu.
11. Toast.makeText() adalah method yang terdapat dalam Toast dan digunakan untuk membuat / menambahkan text pada Toast yang akan ditampilkan.
12. show() pada Toast adalah method yang digunakan untuk menampilkan Toast yang sudah dibuat.
13. LENGTH\_LONG adalah perintah yang berfungsi untuk menampilkan pesan toast dalam waktu yang panjang yaitu sekitar 3 – 5 detik.
14. LENGTH\_SHORT adalah perintah yang berfungsi untuk menampilkan pesan toast dalam waktu yang singkat yaitu sekitar 2 detik.
15. Pada pembuatan toast, dibutuhkan 3 parameter, yaitu context application, text (berisi pesan teks untuk toast), dan durasi (LENGTH\_LONG atau LENGTH\_SHORT).
16. LayoutInflater adalah library yang digunakan untuk memasukkan dan mengconvert semua data file xml yang ada ke dalam recyclerview.
17. Inflate() adalah fungsi dari LayoutInflater yang digunakan untuk mengisi view baru dari spesifik xml resource yang sudah dibuat sebelumnya. Parameter dari inflate() ini berisi menuRes (file yang akan dipanggil) dan menu (menu apa yang akan di inflate ke dalamnya, biasanya menjalankan perintah getMenu()).
18. Membuat file drawable dengan nama bg\_form yang berisi background dengan bentuk rectangle yang berwarna coklat susu, jadi ketika ingin membuat form, hanya tinggal memanggil file bg\_form dan tinggal mengatur ukuran yang sesuai dengan isi dari form nya.

item>  
 <shape  
 android:shape="rectangle">  
 <solid  
 android:color="#F2E9E3"/>  
 <corners  
 android:radius="20dp"/>  
 </shape>  
</item>

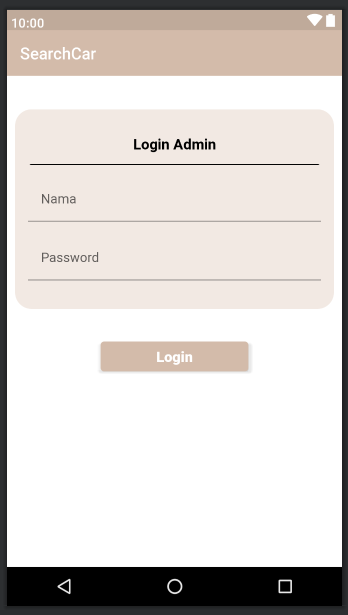
1. Membuat form untuk daftar jika user belom mempunyai akun sebagai admin. Untuk pembuatan formnya, pertama, menambahkan constraint layout di dalam constraint layout, lalu memanggil background xml yang sudah dibuat. Lalu membuat form isian dengan menggunakan atttribut textinputlayout agar ketika form teks di klik, maka judulnya bisa slash ke atas.



1. Apabila button btndaftar diklik, maka akan diarahkan ke tampilan LoginAdmin.

public void btnRegister(View view) {  
 Intent intent = new Intent(this, LoginAdmin.class);  
 startActivity(intent);  
}

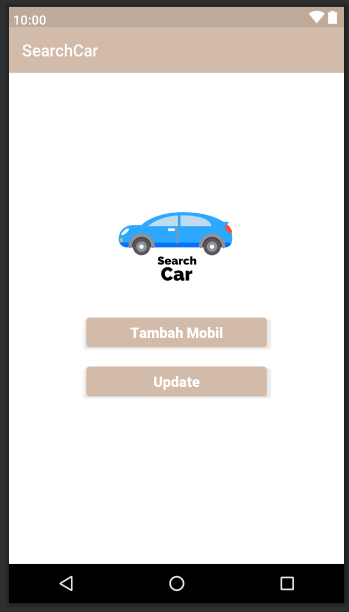
1. Di dalam LoginAdmin juga terdapat form isian, jadi sistem pengisian sama dengan sistem dari DaftarAdmin, yaitu pertama membuat constraint layout di dalam constraint layout, selanjutnya memanggil bg\_form untuk background. Lalu membuat isian form dengan menggunakan atribut teksinputlayout.



1. Mengarahkan btnLogin ke dalam tampilan HomeAdmin apabila btnLogin diklik.

public void btnLogin(View view) {  
 Intent intent = new Intent(this, HomeAdmin.class);  
 startActivity(intent);  
}

1. Setelah berhasil login, tampilan yang dijalankan adalah tampilan homeadmin, dimana tampilan ini berisi icon profil di pojok kanan, button tambah mobil dan button update.



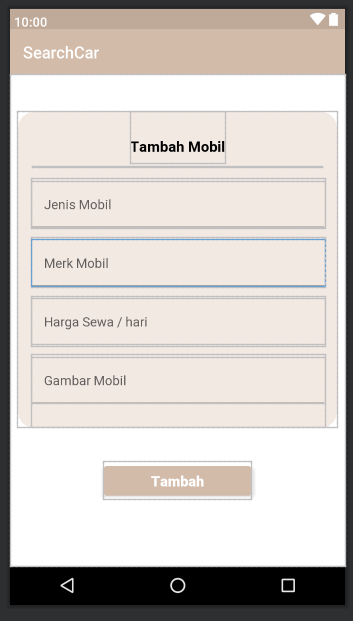
1. Menambahkan icon profil dengan menggunakan vector asset.
2. Karena di dalam HomeAdmin terdapat icon yang ditambahkan menggunakan vector asset. Maka untuk memanggil dan melakukan action untuk icon profil, pertama – tama harus membuat file menu profil\_menu yang berisi memanggil icon profil yang sudah dibuat.

<item  
 android:id="@+id/iconProfil"  
 android:icon="@drawable/ic\_profil"  
 android:title="Profil"  
 app:showAsAction="always" />

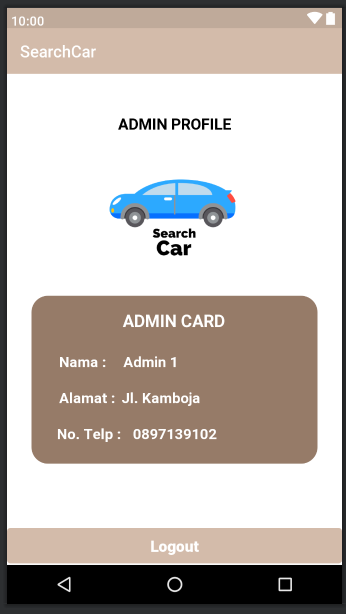
1. Lalu memasukkan item profil ke dalam HomeAdmin dengan menggunakan perintah inflate. Yaitu pertama memanggil xml profil\_menu dan mengisi menu. Kemudian melakukan aksi ketika item diklik, yaitu variabel id berisi perintah untuk mengambil id dari item. Selanjutnya terdapat pengujian, apabila nilai id sama dengan iconProfil, maka akan mengarahkan halaman ke dalam halaman ProfilAdmin. Jadi, ketika icom profil di klik, maka admin dapat melihat info dari icon profil akun mereka.

@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*profil\_menu*, menu);  
 return true;  
}  
  
@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);  
 return true;  
 }  
  
 return super.onOptionsItemSelected(item);  
}

1. Ketika button tambah mobil diklik, maka akan diarahkan ke dalam form untuk menambahkan data mobil. Untuk pembuatan formnya, pertama, menambahkan constraint layout di dalam constraint layout, lalu memanggil background xml yang sudah dibuat. Lalu membuat form isian dengan menggunakan atttribut textinputlayout agar ketika form teks di klik, maka judulnya bisa slash ke atas.



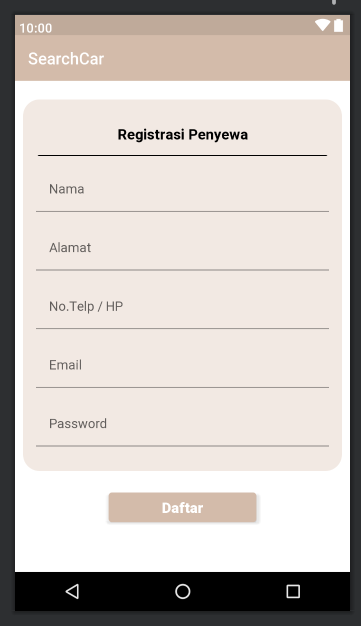
1. Ketika button update di klik, maka tampilan akan diarahkan ke dalam tampilan update mobil. dimana di dalam form akan diisi sesuai dengan id yang dipilih, lalu melakukan perubahan pada data.
2. Alert Dialog adalah sebuah pop up yang muncul pada saat-saat tertentu dan berfungsi untuk memberi info pada pengguna, berbentuk kotak.
3. AlertDialog.builder digunakan untuk membuat interface dari kotak dialog, seperti title, pesan yang ingin ditampilkan, tombol, dan icon. Setelah menambahkan alertbuilder, kita dapat membuat beberapa konfigurasi atau atribut, yaitu, setTitle(), setMessage(), setPositiveButton() dan setNegativeButton(), setCancelable(), dan show().
4. setTitle() adalah perintah yang digunakan untuk membuat title / judul pada pesan dialog yang muncul.
5. SetIcon() adalah perintah yang digunakan untuk menampilkan icon apa yang akan ditampilkan ketika alert dialog berjalan.
6. setMessage() adalah perintah yang digunakan untuk menampilkan pesan.
7. setPositiveButton() dan setNegativeButton() adalah perintah yang digunakan untuk menampilkan opsi pilihan ya / tidak.
8. setCancelable() adalah perintah yang digunakan untuk menentukan apakah dialog yang muncul bisa dibatalkan atau tidak.
9. Show() adalah perintah yang digunakan untuk menampilkan alert dialog.
10. Tampilan profil admin ketika icon pada profil di klik, jadi di dalam profil ini terdapat admin card yang berisi informasi tentang admin. Dan terdapat button logout apabila user ingin keluar dari aplikasi.



1. Ketika button logout di klik, akan terdapat jendela window yang terdapat 2 pilihan, yaitu OK atau Cancel. Apabila memilih iya, maka akan diarahkan ke tampilan login awal aplikasi. Icon yang digunakan adalah icon logoapk yang diambil dari folder drawable. Kemudian judulnya adalah Logout Aplikasi dengan pesan yang ada adalah apakah anda yakin ingin logout dari posisi admin?

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();  
}

1. Mulai membuat tampilan untuk user (penyewa). Jadi, pertama ketika button daftar diklik, maka akan diarahkan ke tampilan untuk pembuatan akun user ketika user belom mempunyai akun. Untuk pembuatan formnya, pertama, menambahkan constraint layout di dalam constraint layout, lalu memanggil background xml yang sudah dibuat. Lalu membuat form isian dengan menggunakan atttribut textinputlayout agar ketika form teks di klik, maka judulnya bisa slash ke atas.



1. Tetapi, jika user sudah mempunyai akun, maka user hanya tinggal mengklik button login. Jadi sistem pengisian sama dengan sistem dari DaftarUser, yaitu pertama membuat constraint layout di dalam constraint layout, selanjutnya memanggil bg\_form untuk background. Lalu membuat isian form dengan menggunakan atribut teksinputlayout.

