|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| * Melengkapi coding project konversi suhu pada java * Membuat function untuk project konversi suhu pada java * Melengkapi pembuatan opsi pada konversi suhu (Tugas) * Mengganti Icon Launcher pada app project Konversi suhu * Memahami fungsi activity lifecycle * Memebuat function activity lifecycle * Membuat Toast * Membuat Alert dialog * Membuat Alert dialog dengan button * Membuat Activity Baru * Mengganti Label Action Bar * Memanggil Activity kedua dan seterusnya * Menampilkan Kalender * Mengambil Tanggal * Menggunakan Vector Asset * Merubah Warna Vector Asset * Menggunakan CardView * Membuat Layout di CardView * Membuat RecycleView Layout * Membuat Item list | 90 |

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

**1.** Memahami konsep konversi suhu

**2.** Dapat mengatur back end coding java untuk app konversi suhu serta memahami proses nya, yaitu sebagai berikut :

* Menambahkan terlebih dahulu value array untuk mengisi opsi pada spinner di dalam folder res→values→strings.xml
* Pada spinner di bagian UI Display pada atribut entries tambahkan **@array/pilihan** untuk memasukkan value array yang sudah di input sebelumnya untuk opsi konversi nantinya
* Function isiSpinner di off kan karna sekarang kondisinya tidak perlu menggunakan function manual tersebut untuk menyusun app konversi suhu
* Mendeklarasikan atribut **EditText etNilai;** karena akan digunakan untuk input nilai yang akan dikonversi oleh user
* Deklarasikan juga atribut **TextView tvHasil;** untuk display tampilan hasil dari nilai yang telah dikonversikan nantinya
* Pada function load() panggil nilai dari etNilai dan tvHasil menggunakan **findViewById(R.id.*value*);**
* Membuat function untuk menghitung nilai konversi lalu mengisinya dengan variable berbentuk double guna memanggil value etNilai karena form nya akan digunakan diisi lalu diambil nilainya **double suhu = Double.parseDouble(etNilai.getText().toString());** untuk dihitung sesuai rumus yang ditentukan **double hasil = (9.0/5.0) \* suhu + 32;** [rumus nya disesuaikan]
* Lalu menggunakan tvHasil untuk mengeset hasilnya **tvHasil.setText(hasil+"");** [Tidak lupa tanda “” karena nilainya berbentuk string jadi simbol tersebut berfungsi untuk konversi bentuk nilai sesuai dengan yg ditentukan, disini berbentuk double]
* Pada function untuk button konversi yaitu btnKonversi() tambahkan if kondisi apabila pada form kosong dan tidak terisi nilai maka akan mengeluarkan warning message dan proses konversi otomatis tidak akan berjalan apabila tidak ada nilai yang di isi if (etNilai.getText().toString().equals("")){

Toast.makeText(this, "Nilai Tidak Boleh Kosong", Toast.LENGTH\_SHORT).show();

* Lalu untuk else nya isi dengan berbagai if untuk menyesuaikan kondisi dengan @array/pilihan pada konversi jenis apa yang akan digunakan untuk proses konversi sehingga program akan dapat memproses harus menggunakan function mana yang if (pilihan.equals("Celcius To Reamur")){

cToR();

} [buat berkali-kali untuk diisi dengan function yang berbeda sesuai dengan kondisi]

**3.** Memahami bahwa script **Toast.makeText(this, "", Toast.LENGTH\_SHORT).show();** berfungsi untuk menampilkan jendela/pop up sebagai message pencegahan karena kondisi yg diharuskan dalam sebuah program tidak terpenuhi

**4.** Memahami cara mengganti icon launcher dari aplikasi yang dibuat menggunakan android studio yaitu :

* Copy file image kedalam folder res→drawable→ copy image
* Pastikan file gambar name nya huruf kecil semua
* Kemudian buka folder app→manifest→AndroidManifest.xml pada andoroid:icon dan :roundIcon ganti setelah slash / menjadi /file gambar

**5.** Memahami bahwa Activity lifecycle merupakan proses dimana sebuah aplikasi android berjalan mulai dari aplikasi tersebut menyala hingga menutup

**6.** Memahami bahwa ketika sebuah aplikasi dijalankan maka function yang terpanggil/digunakan diantaranya onCreate(), onStart() dan onPauseResume()/onResume()

**7.** Memahami bahwa ketika aplikasi tersebut berhenti digunakan/di tutup maka function yang terpanggil/digunakan adalah onPause(), onStop() dan onDestroy()

**8.** Setiap jenis function/method berjalan berurutan sesuai dengan fungsi penggunaan

**9.** Dapat memahami basic proses Activity lifecycle melalui sebuah project yaitu :

* Membuat project baru
* Memanggil function onCreate() pada lapak yang sudah tersedia karena method onCreate sudah otomatis ada pada java andorid studio
* Memanggil method onStart() dan onResume melalui klik kanan→generate→overide method→search method nya lalu choose maka akan otomatis dibuatkan tinggal di sesuaikan
* Hal tersebut juga berlaku pada pemanggilan method untuk onPause() onStop() dan onDestroy()

**10.** Memahami bahwa Message Dialog/Dialog Alert berfungsi untuk menampilkan pesan atau untuk berinteraksi dengan user sebagai opsi dalam sebuah program

**11.** Menurut pemahaman saya terdapat 3 jenis Dialog yang digunakan diantaranya Alert(), AlertDialog() dan AlertDialogButton()

* Jenis Alert() akan memberikan tampilan berupa warning message pop up biasanya digunakan sebagai peringatan/himbauan
* Jenis AlertDialog() juga berjenis pop up namun lebih ke bentuk window dan tidak hanya mini pop up sehingga berfungsi apabila message terbut berisi lebih banyak atau biasanya bisa untuk message intruksi
* Jenis AlertDialogButton() berbentuk pop up window dengan opsi jadi sebuah warning message akan muncul lalu terdapat beberapa opsi yang akan mengarahkan kedalam suatu method tertentu yang telah dijalankan seperti Yes No Exit

**12.** Sebelum memulai membuat Alert Dialog terlebih dahulu menyiapkan layout nya yaitu sebagai berikut :

* Menyiapkan new empty layout lalu merubah jenis nya pada bagian code menjadi LinearLayour
* Tidak lupa menambahkan andorid:orientation=”vertical”
* Menyiapkan 3 buah button untuk masing-masing jenis alert
* Memberi id masing-masing button sesuai dengan alert nya karena akan digunakan daam proses coding nantinya
* Menambahkan onClick untuk create function yang akan digunakan sebagai pemicu button dialog alert nantinya

**13.** Dapat mempraktikkan pembuatan Dialog berjenis Alert() yaitu :

* Membuat function terlebih dahulu untuk menampilkan message nya dengan memanfaatkan script Toast **public void showToast(String pesan){**

**Toast.makeText(this, pesan, Toast.LENGTH\_SHORT).show(); }**

* Mengatur button nya agar apabila button alert yang di makshud berjalan apabila button Alert di tekan **public void btnToast(View view) {**

**showToast("Belajar Membuat Pesan"); }**

**14.** Dapat mempraktikkan pembuatan Dialog berjenis AlertDialog() yaitu :

* Membuat function untuk show alertDialog dengan memanfaatkan AlertDialog.Builder **AlertDialog.Builder buatAlert = new AlertDialog.Builder(this);**
* Lalu set judul/isi dari yang akan ditampilkan **buatAlert.setTitle("PERHATIAN");**
* Kemudian setMessage nya **buatAlert.setMessage(pesan);** dan gunakan function nya buatAlert.show();
* Tampilkan dengan menggunakan pemicu event onClick dari yang sudah dibuat pada button yang di tentukan **public void btnAlert(View view) {**

**showAlert("Silahkan Dicoba"); }**

**15.** Dapat mempraktikkan pembuatan AlertDialogButton() yaitu :

* Membuat function untuk alert dialog terlebih dahulu dengan memanfaatkan **AlertDialog.Builder showAlert = new AlertDialog.Builder(this);**
* Lalu mengisi title/isi pesannya **showAlert.setTitle("PERINGATAN!");**
* Kemudian mengeset pesannya **showAlert.setMessage(pesan);**
* Untuk menggunakan opsi maka perlu menggunakan script setPositiveButton atau setNegativeButton
* Untuk opsi benar maka menggunakan setPositiveButton ↓ **showAlert.setPositiveButton("YA", new DialogInterface.OnClickListener() {**

**@Override**

**public void onClick(DialogInterface dialog, int which) {**

**showToast("Data sudah dihapus");**

**} });**

* Lalu untuk opsi tidak maka menggunakan setNegativeButton ↓ **showAlert.setNegativeButton("TIDAK", new DialogInterface.OnClickListener() {**

**@Override**

**public void onClick(DialogInterface dialog, int which) {**

**showToast("Data tidak akan dihapus");**

**} });**

* Kemudian panggil function nya showAlert.show();
* Seperti sebelumnya atur button agar ketika di klik maka akan menggunakan method ini **public void btnAlertDialogButton(View view) {**

**showAlertButton("Yakin Akan Menghapus?"); }**

**16.** Memahami bahwa mengapa dalam menampilkan method nya menggunakan variable pesan bukan menulis pada pesan nya langsung. Hal tersebut dikarenakan agar isi dari pesan lebih flexible untuk dirubah karena sudah disimpan dalam bentuk variable.

**17.** Memahami bahwa this pada tiap jenis alert menandakan bahwa alert nya dibuat di dalam class tersebut

**18.** Memahami bahwa Penggunaan DialogInterface.onClickListener adalah sebagai build opsi yes/no pada jendela alert dialog button

**19.** Memahami bahwa intent merupakan sebuah jembatan yang menghubungkan interaksi antar Activity di aplikasi Android. Intent juga dapat membawa dan mengirimkan data ke Activity lain, bahkan ke aplikasi lain (Gmail, Google Map dsb). Intinya, Intent merupakan mekanisme untuk melakukan sebuah action dan komunikasi antar komponen aplikasi.

Contohnya : sebuah halaman aplikasi memiliki button di dalamnya, apabila button tersebut di klik maka akan menuju ke halaman lain. Misalnya saat membuka aplikasi pada HP.

**20.** Memahami bahwa terdapat 2 macam intent yaitu intent implicit dan intent explicit.

* Intent Implicit : berfungsi melakukan perpindahan activity (halaman) menuju ke internal smartphone, contohnya saat membuka galeri di HP.
* Intent Explicit : berfungsi melakukan perpindahan activity (halaman) ke halaman lainnya dan bekerja menggunakan nama class yang dituju.

**21.** Dapat menerapkan contoh pemakaian intent dalam pembuatan aplikasi, yaitu :

* Membuat new activity yang akan digunakan sebagai base utama/halaman utama
* Kemudian menambahkan komponen EditText dan 3 button
* EditText diatur menjadi match constraint dan diatur juga margin nya lalu diberi id tvHasil
* Untuk button nya diberi nama Barang, Penjualan dan Pembelian lalu diatur juga menjadi match constraint dan diberi margin untuk space
* Menambahkan id untuk masing-masing button
* Untuk mengubah agar label halaman sesuai maka di atur dalama AndroidManifest. Tepatnya pada Activity Android Name

<activity android:name=".Barang"

android:label="Barang"

android:parentActivityName=".MainActivity"> </activity>

* Konsep disini adalah apabila selesai menginputkan nama barang lalu di pilih button mana sesuai dengan tujuan barang tersebut akan diapakan apakah disimpan saja sebagai barang atau dijual/dibeli. Jadi setelah diklik button nya maka nama barang nya akan tersimpan pada halaman tiap button, lalu button nya juga dapat di klik untuk melihat barang apa saja yg ada di dalam halaman button tersebut
* BARANG, untuk mengatur halaman pertama button barang maka perlu membuat new activity dan diberi nama barang.
* Kemudian menambahkan textView pada halaman tersbeut guna menampilkan nama barang yang sudah di input
* Pada MainActivity.java membuat method load() untuk memanggil idKomponen. Terlebih dahulu di deklarasikan EditText etBarang; ; lalu di dalam method panggil menggunakan etBarang = findViewById(R.id.etBarang); seperti biasa
* Pada button Barang tambahkan onClick lalu create method btnBarang yang akan otomatis dibuat di MainActivity.java lalu di dalam method ini ambil nilai dari dalam form yang sudah di inputkan tadi String barang = etBarang.getText().toString();
* Kemudian menggunakan Intent sebagai transisi ke halaman khusus barang dengan memanfaatkan class Barang Intent intent = new Intent(this, Barang.class);
* Nilai yang sudah di ambil tadi disimpan terlebih dahulu karena akan di kirim untuk ditampilkan pada halaman barang intent.putExtra("ISI", barang); disini nama variable nya ISI
* Jalankan proses inten nya startActivity(intent);
* Pada halaman barang.java untuk menampilkan nilai yang dikirim terlebih dahulu deklarasikan form untuk penampilnya yaitu TextView tvHasil;
* Membuat method load() untuk memanggil nilai dari TextView dengan menggunakan tvBarang = findViewById(R.id.tvBarang);
* Membuat function untuk mengambil data yang telah dikirim oleh halaman utama dengan menggunakan variable tadi sebagai perantara nya String ambil = getIntent().getStringExtra("ISI");
* Lalu di set nilai nya tvBarang.setText(ambil);
* Tidak lupa function load() dan yang lain di tampilkan pada method utama onCreate() kalau tidak maka program tidak akan jalan.
* Berlaku juga untuk pembuatan inten Penjualan dan pembelian

**22.** Memahami bahwa Scroll view berfungsi untuk membantu tampilan sebuah aplikasi tetap dalam satu halaman apabila screen device nya tidak mampu menampung semuanya maka peran scrollview nya akan menjadi seperti tombol gulir sehingga aplikasinya dapat di scroll/gulir ke bawah untuk menemukan elemen lainnya.

**23.** Dapat menerapkan penggunaan scrollview dalam pembuatan aplikasi android

**24.** Memahami bahwa Date Picker Dialog berfungsi untuk menampilkan kolom tanggal

**25.** Memahami bahwa dalam pengurutan variable untuk membuat date picker tidak boleh salah format urutan yaitu tahun-bulan-hari karena apabila terbalik maka perhitungannya iut salah

**26.** Memahami bahwa dalam pembuatan Date Picker untuk menampilkan bulan, variable nya perlu di tambah satu karena apabila tidak maka bulan yang tampil menjadi bulan sebelumnya

**27.** Dapat mempraktikkan pembuatan Date Picker yaitu :

* Membuat new activity (Pastinya)
* Mengatur layout untuk plain text jadi apabila di klik maka akan menampilkan Date Picker Dialog
* Mengatur margin dan match constraint untuk plain text nya lalu diberi id dan onClick
* Pada function dari onClick nya buat variable untuk mendapat data kalender yaitu Calendar cal = Calendar.getInstance();
* Lalu membuat variable untuk masing-masing elemen. Hari int tgl = cal.get(Calendar.DAY\_OF\_MONTH); lalu bulan int bln = cal.get(Calendar.MONTH); kemudian tahun int thn = cal.get(Calendar.YEAR);
* Deklarasikan plain text karena akan digunakan sebagai tombol penampil tanggal EditText = idPlainText→ etTanggal;
* Buat method load() untuk memanggil plain text etTanggal = findViewById(R.id.etTanggal); tidak lupa sertakan function ini kedalam onCreate
* Untuk membuat dialog nya dengan membuat objek DatePickerDialog dtp = new DatePickerDialog(this, new DatePickerDialog.OnDateSetListener()
* Kemudian mengatur variable di dalam onDateSet agar sesuai dengan yang sudah dibuat sebelumnya

{

@Override

public void onDateSet(DatePicker view, int thn, int bln, int tgl) {

etTanggal.setText(tgl+"-"+(bln+1)+"-"+thn);

}

}, thn, bln, tgl);

* Kemudian menampilkan dengan dtp.show();

**28.** Memahami bahwa vector asset merupakan fitur dalam android studio yang membantu menambahkan ikon material dan mengimpor file Scalable Vector Graphic (SVG) dan Adobe Photoshop Document (PSD) ke dalam project sebagai resource vektor drawable. Secara umum berarti membantu menambahkan icon ke dalam aplikasi yang bisa diatur baik dari segi fungsi maupun display nya.

**29.** Dapat menambahkan vector asset pada android studio dan mengatur size, color serta constraint nya.

**30.** Memahami bahwa cardView berfungsi untuk membuat tata letak layout berbasis kartu

**31.** Memahami bahwa cardView dapat berfungsi untuk list tampilan per group dan semacamnya karena bentuk nya yang berupa card cukup efisien.

**32.** Dapat menerapkan pembuatan cardView yaitu :

* Mmebuat ikon menggunakan vector asset
* Menginputkan cardView lalu mengatur constraint nya ke bawah atas dan memberi margin
* Menambahkan Linear layout horizontal untuk menambahkan gambar dari vector yaitu berbentuk tag orang dari yang dibuat sebelumnya
* Menambahkan Linear layout vertical untuk mengisi text view nya

**33.** Memahami bahwa fungsi atribut bias dapat digunakan untuk mengatur start dari constraint sebuah elemen

**34.** Memahami bahwa recycle view berfungsi untuk menampilkan banyak data dalam sebuah activity.

**35.** Dapat menerapkan pembuatan Layout untuk recycle view dengan memanfaatkan ScrollView dan CardView, yaitu :

* Membuat new layout resource pada folder layout di res
* Menambahkan cardView lalu Linear Layour dan 2 elemen TextView untuk menampilkan teks nantinya.
* Mengatur constraint dan margin dari cardView dan TextView