



2023

Pemrograman Mobile

DIII SISTEM INFORMASI



TATA TERTIB PRAKTIKUM MOBILE PROGRAMMING D3 SISTEM INFORMASI

DOSEN/KOORDINATOR PRAKTIKUM

1. Dosen harus hadir saat praktikum minimal 15 menit di awal kegiatan praktikum untuk mengisi materi dan menandatangani presensi kehadiran praktikum.
2. Dosen membuat modul praktikum, soal seleksi asisten, pre-test, post-test, dan responsi dengan berkoordinasi dengan asisten dan pengampu mata praktikum.
3. Dosen berkoordinasi dengan koordinator asisten praktikum untuk evaluasi praktikum setiap minggu.
4. Dosen menandatangani surat kontrak asisten praktikum dan koordinator asisten praktikum.
5. Dosen yang tidak hadir pada slot praktikum tertentu tanpa pemberitahuan selama 2 minggu berturut-turut mendapat teguran dari Kepala Laboratorium, apabila masih berlanjut 2 minggu berikutnya maka Kepala Laboratorium berhak mengganti koordinator praktikum pada slot tersebut.

PRAKTIKAN

1. Praktikan harus hadir 15 menit sebelum kegiatan praktikum dimulai, dan dispensasi terlambat 15 menit dengan alasan yang jelas (kecuali dosen menentukan lain dan patokan jam adalah jam yang ada di Laboratorium, terlambat lebih dari 15 menit diperbolehkan mengikuti praktikum tetapi presensi dianggap tidak hadir).
2. Praktikan yang tidak mengikuti praktikum dengan alasan apapun, dan ketidakhadiran maksimal 3 kali praktikum dan jika lebih dari 3 kali maka praktikum tidak bisa mengikuti Ujian Tengah Semester atau Ujian Akhir Semester.
3. Praktikan harus berpakaian rapi sesuai dengan ketentuan Universitas, sebagai berikut:

- a) Tidak boleh memakai Kaos Oblong, termasuk bila ditutupi Jaket/Jas Almamater (Laki-laki / Perempuan) dan Topi harus Dilepas.
 - b) Tidak Boleh memakai Baju ketat, Jilbab Minim dan rambut harus tertutup jilbab secara sempurna, tidak boleh kelihatan di jidat maupun di punggung (khusus Perempuan).
 - c) Tidak boleh memakai baju minim, saat duduk pun pinggang harus tertutup rapat (Laki-laki / Perempuan).
 - d) Laki-laki tidak boleh memakai gelang, anting-anting ataupun aksesoris Perempuan.
4. Praktikan tidak boleh makan dan minum selama kegiatan praktikum berlangsung, harus menjaga kebersihan, keamanan dan ketertiban selama mengikuti kegiatan praktikum atau selama berada di dalam laboratorium (tidak boleh membuang sampah sembarangan baik kertas, potongan kertas, bungkus permen baik di lantai karpet maupun di dalam ruang CPU).
 5. Praktikan dilarang meninggalkan kegiatan praktikum tanpa seizin Dosen atau Laboran.
 6. Selama praktikum dilarang **NGENET/NGE-GAME**, kecuali mata praktikum yang membutuhkan atau menggunakan fasilitas Internet.
 7. Praktikan dilarang melepas kabel jaringan atau kabel power praktikum tanpa sepengetahuan laboran
 8. Praktikan harus memiliki **FILE** Petunjuk praktikum dan digunakan pada saat praktikum dan harus siap sebelum praktikum berlangsung.
 9. Praktikan dilarang melakukan kecurangan seperti mencontek atau menyalin pekerjaan praktikan yang lain saat praktikum berlangsung atau post-test yang menjadi tugas praktikum.
 10. Praktikan dilarang mengubah setting software/hardware komputer baik menambah atau mengurangi tanpa permintaan asisten atau laboran dan melakukan sesuatu yang dapat merugikan laboratorium atau praktikum lain.

11. Koordinator Praktikum, Kepala laboratorium dan Laboran mempunyai hak untuk menegur, memperingatkan bahkan meminta praktikan keluar ruang praktikum apabila dirasa Rekan mengganggu praktikan lain atau tidak melaksanakan kegiatan praktikum sebagaimana mestinya dan atau tidak mematuhi aturan lab yang berlaku.
12. Pelanggaran terhadap salah satu atau lebih dari aturan diatas maka Nilai praktikum pada pertemuan tersebut dianggap 0 (NOL) atau tidak hadir.

LAIN-LAIN

1. Pada Saat Ujian Harus menggunakan Baju Kemeja untuk Laki-laki dan Perempuan untuk Praktikan.
2. Ketidakhadiran praktikum dengan alasan apapun dianggap INHAL.
3. Izin praktikum mengikuti aturan izin Perkuliahan (Dengan syarat ditandatangani Dosen Ber – NIP/NIK).
4. Yang tidak berkepentingan dengan praktikum dilarang mengganggu praktikan atau membuat keributan/kegaduhan.
5. Penggunaan lab diluar jam praktikum maksimal sampai pukul 21.00 dengan menunjukkan surat ijin dari Kepala Laboratorium Prodi Informatika.

DAFTAR ISI

DAFTAR ISIii
Praktikum 11
1. Sub Pokok Pembahasan:1
2. Capaian Pembelajaran.....	.1
3. Pendahuluan.....	.1
a) Kontrak Kuliah1
b) Ruang Lingkup.....	.4
c) Contoh Aplikasi Berbasis Mobile.....	.4
4. Persiapan Software4
a) Installasi Software Android Studio4
b) Membuat Projek Android Pertama5
c) Jalankan Aplikasi di Emulator dan Smartphone.....	.8
d) Build APK.....	.13
5. Tugas Praktikum15
Praktikum 216
a) Sub Pokok Pembahasan16
b) Capaian Pembelajaran.....	.16
c) Membuat Layout Application.....	.16
d) Membuat Splash Screen Android.....	.25
e) Tugas Praktikum30
Praktikum 331
1. Sub Pokok Pembahasan31
2. Capaian Pembelajaran.....	.31
3. View & Views Group.....	.31
4. Style & Theme45
5. Tugas Praktikum55
Praktikum 456
1. Sub Pokok Pembahasan56
2. Capaian Pembelajaran.....	.56
3. Merancang Program Application56
4. Tugas Praktikum71



Praktikum 5	72
1. Sub Pokok Pembahasan	72
2. Capaian Pembelajaran.....	72
3. Intent	72
1) Logika Dasar	73
2) Eksplisit	77
3) Implisit.....	81
4. Tugas Praktikum	83
Praktikum 6	84
1. Sub Pokok Pembahasan	84
2. Capaian Pembelajaran.....	84
3. Intent Lanjutan	84
1) Intent Explicit dengan Parcelable	85
2) Intent dengan ResultActivity	95
4. Tugas Praktikum	102
Praktikum 7	104
1. Sub Pokok Pembahasan	104
2. Capaian Pembelajaran.....	104
3. Fragment.....	104
4. RecyclerView Mode List	114
5. Tugas Praktikum	151
DAFTAR REFERENSI	152



Praktikum 1

Pendahuluan dan Persiapan Software

1. Sub Pokok Pembahasan:

- a) Kontrak Kuliah
- b) Ruang Lingkup,
- c) Contoh Aplikasi Berbasis Mobile
- d) Installasi Software Android Studio
- e) Membuat Projek Sangat Sederhana (hello World)
- f) Jalankan Aplikasi di Emulator dan Smartphone
- g) Build APK

2. Capaian Pembelajaran

- a) Mahasiswa mampu menanamkan sikap jujur, bertanggung jawab, disiplin, saling menghargai dan berinovasi dengan tetap menjunjung tinggi nilai-nilai agama dan ketuhanan.
- b) Mahasiswa mampu menunjukkan sikap nasionalisme, mampu bekerja dalam tim sesuai kaidah keilmuan pada proyek pengembangan aplikasi *mobile programming*.
- c) Mahasiswa mampu berpartisipasi aktif, bertanggung jawab, dan memiliki motivasi mengembangkan diri dalam mendapatkan solusi permasalahan melalui aplikasi mobile programming

3. Pendahuluan

a) Kontrak Kuliah

Nama Mata Kuliah : Praktikum Pemrograman Mobile
Semester : 4 (Empat)
Jurusan : D3 Sistem Informasi
Nama Dosen :
- Muhammad Panji Muslim, S.Pd., M.Kom
- Radinal Setyadinsa, S.Pd., M.T.I

1) Manfaat Mata Kuliah

Praktikum Pemrograman Mobile adalah mata kuliah yang mempunyai fokus terhadap pembuatan aplikasi berbasis mobile, dalam mata kuliah ini akan secara utuh membahas pembuatan aplikasi berbasis



mobile mulai dari konsep perancangan, sampai langkah – langkah teknis dalam pembuatan atau pengkodingan, sehingga setelah mengambil mata kuliah ini mahasiswa dapat membuat prototype aplikasi mobile yang dapat dikembangkan secara mandiri untuk tugas akhir ataupun sebagai startup.

2) Deskripsi Perkuliahan

Matakuliah ini berisi tentang konsep dan praktek mengenai implementasi pengembangan aplikasi berbasis mobile. Pembahasan difokuskan kepada aplikasi berbasis mobile dengan platform android serta implementasi program skala kecil dan sedang serta kompleks.

3) Tujuan Instruksional

Di akhir perkuliahan, mahasiswa diharapkan mampu membangun aplikasi berbasis mobile secara utuh dan lengkap agar dapat digunakan dalam mengatasi permasalahan instansi dengan mengimplementasikannya ke dalam aplikasi berbasis bahkan dikembangkan secara mandiri sebagai startup.

4) Strategi Perkuliahan

Perkuliahan secara full dilakukan secara tatap muka, dengan metode perkuliahan mata kuliah system basis data diawali dengan ceramah sebagai pengantar dan pembuka wawasan, kemudian perkuliahan selanjutnya digunakan praktikum melalui modul yang diberikan. Terakhir matakuliah ini dilengkapi dengan adanya project akhir yang menghasruskan mahasiswa membangun aplikasi berbasis mobile secara berkelompok. Dengan adanya projek ini mahasiswa selain mendapat pendalaman materi secara tidak langsung juga akan mendapat wawasan dan pengalaman yang berguna sebagai bekal nantinya.

5) Tugas Perkuliahan

Dalam pelaksanaan perkuliahan tatap muka ini setiap selesai mengerjakan praktikum maka akan ada tugas yang harus dikerjakan untuk mengetahui kemampuan mahasiswa untuk praktikum hari tersebut.



Masa ujian, baik UTS maupun UAS diselenggarakan berdasarkan kalender akademik Universitas, dengan menunjukan progress project yang dikerjakan.

6) Kriteria Penilaian

Tabel 1 Kriteria Penilaian

Nilai Huruf	Nilai Angka	Angka Mutu
A	80 – 100	4
A-	78 – 79	3,7
B+	74 – 77	3,3
B	70 – 73	3
B-	65 – 69	2,7
C+	60 – 64	2,3
C	55 – 59	2
C-	50 – 54	1,7
D+	45 – 49	1,3
D	40 – 44	1
E	< 40	0,00
TL	Tidak Lengkap (Bila salah satu komponen kosong)	–

7) Jadwal Perkuliahan

Tabel 2 Jadwal Perkuliahan dengan Materi Praktikum

Minggu	Materi Praktikum
1	Pendahuluan dan Persiapan Software
2	Merancang <i>Layout Application</i> dan Membuat <i>Splash Screen</i>
3	View & View Group dan Style & Theme
4	Merancang Program Application
5	Intent
6	Intent lanjutan
7	Fragment + RecyclerView Mode List
8	UTS
9	RecyclerView Mode Grid dan CardView
10	SQL Lite
11	Webservice dengan Json
12	Webservice dengan MySQL
13	Alarm
14	Broadcast Receiver
15	Presentasi Project Akhir 1
16	Presentasi Project Akhir 2



b) Ruang Lingkup

Praktikum Pemrograman Mobile adalah mata kuliah yang memiliki hasil sebuah aplikasi berbasis mobile sehingga mahasiswa setelah mengambil mata kuliah ini akan memiliki kemampuan untuk merancang dan membuat aplikasi berbasis mobile. Dalam mata kuliah ini mahasiswa akan belajar konsep perancangan aplikasi mobile dan langkah teknis dalam membuat aplikasi berbasis mobile.

Dalam perancangan aplikasi mobile, akan digunakan software android studio sebagai editor dan compilatornya, kemudian dapat menggunakan emulator android atau dengan smartphone (berbasis Android) sebagai media running programnya, selanjutnya sebagai pelengkap kehandalan aplikasi mobile akan digunakan MySqlLite atau MySql sebagai penyimpanan data selama aplikasi tersebut digunakan.

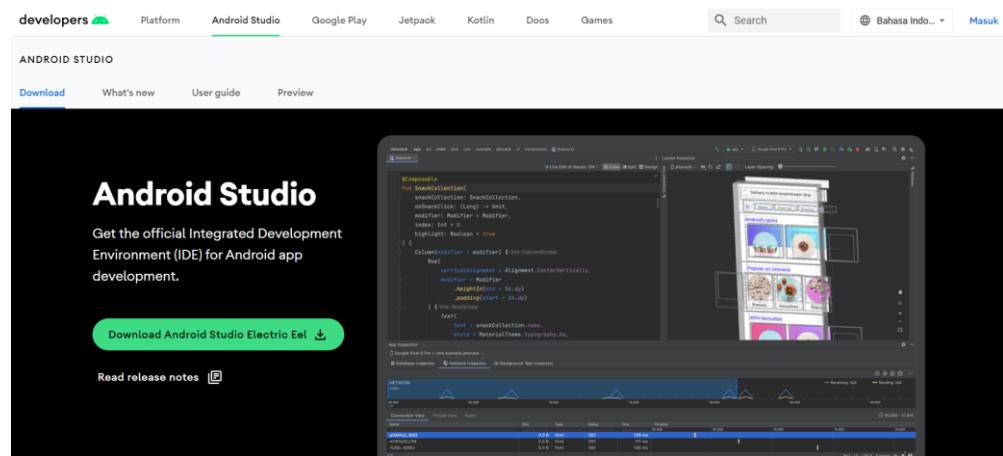
c) Contoh Aplikasi Berbasis Mobile

Gojek, Tokopedia, detik.com, dll

4. Persiapan Software

a) Installasi Software Android Studio

1) Pada mata kuliah kali ini akan menggunakan Android Studio sebagai IDE (*Integrated Development Environment*). Android Studio dirilis 16 Mei 2013 saat Google IO berlangsung. Android Studio berbasiskan JetBrains IntelliJ IDEA, dan dikhawasukan untuk pengembangan *software* berbasis Android. Android studio bisa diunduh di [link berikut ini](#)



Gambar 1 Tampilan untuk Download Android Studio



- 2) Kemudian *install* Android Studio ke device rekan-rekan dan ikuti petunjuknya sampai selesai. Saat instalasi berlangsung rekan diminta untuk menginstal SDK (Software Development Kit) yang membutuhkan koneksi internet. Maka pastikan saat menginstal device rekan-rekan terhubung dengan internet.
- 3) Setelah selesai melakukan instalasi Android Studio, akan muncul seperti gambar di bawah ini.



Gambar 2 Tampilan setelah Installasi

Gambar di atas berarti aplikasi Android Studio sudah bisa digunakan.

b) Membuat Projek Android Pertama

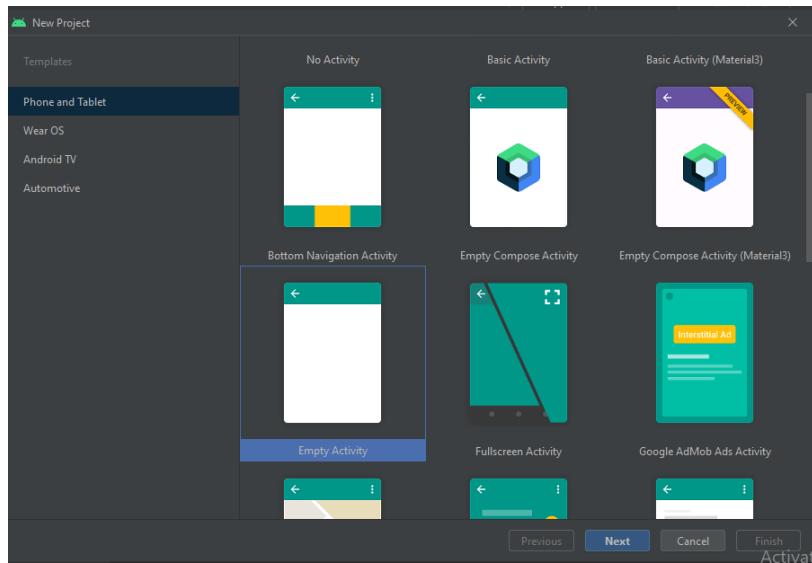
Buka Android Studio, kemudian Untuk memulai proyek baru pilihlah “**Create New Project**”.

Project Wizard

Setelah proses instalasi Android Studio, mari kita mulai membuat sebuah aplikasi *Hello World*, aplikasi pertama Anda.

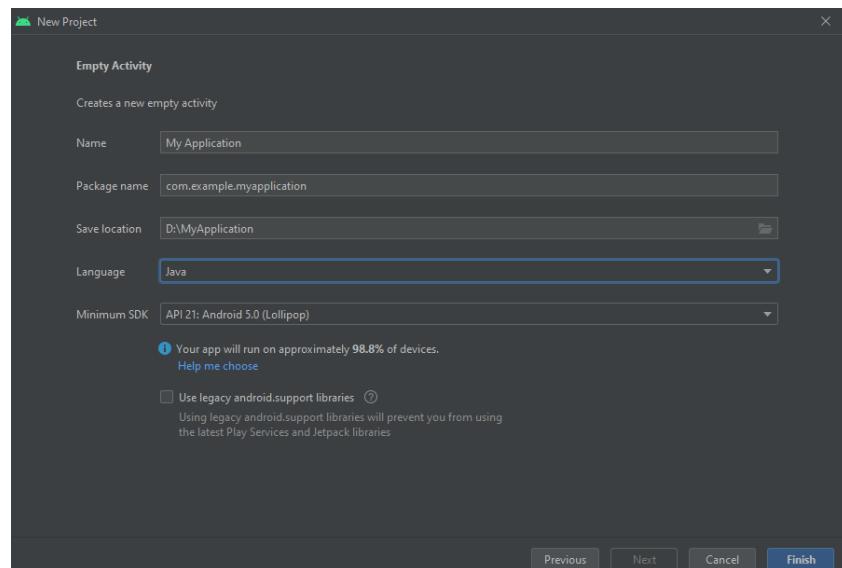
- i. Setelah melakukan **Create New Project**, Rekan diminta untuk melakukan konfigurasi dalam pembuatan proyek baru Anda. Dalam dialog ini Anda bisa memilih *template* dari Aplikasi yang akan Anda buat. Terdapat beberapa *template* yang bisa kita gunakan seperti Empty Activity, Login Activity, Navigation Drawer Activity dan lain-lain. Di dalam dialog ini kita bisa memilih target *devices* dari aplikasi yang akan kita buat.





Gambar 3 Tampilan untuk Membuat Projek Baru

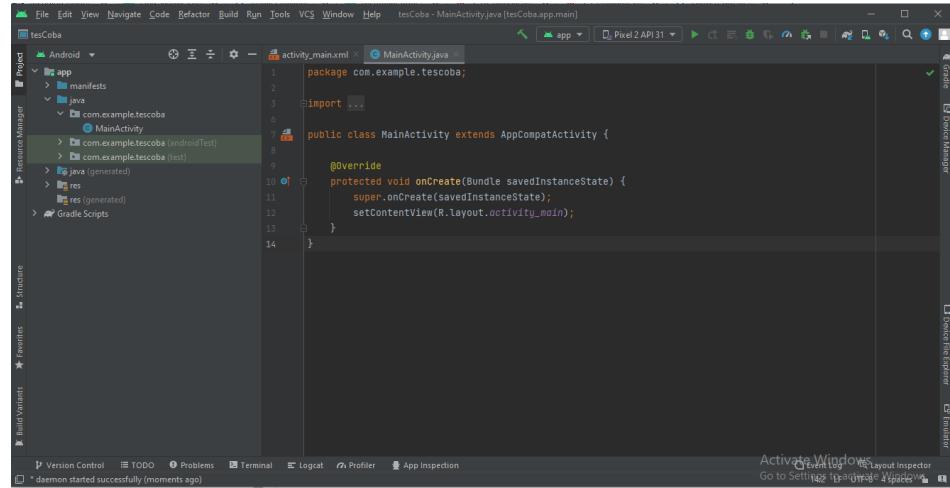
- ii. Dalam dialog ini Anda bisa memberi *nama dari aplikasi, lokasi proyek* dan *nama package*. Nama package akan digunakan dalam identifikasi unik dari aplikasi kita ketika sudah di-publish. Kita juga dapat mengganti dari direktori di mana proyek kita akan disimpan. Kita juga bisa mengganti nilai minimum SDK, yang berfungsi untuk membatasi penggunaan API pada sebuah aplikasi. Di dalam dialog ini juga Anda bisa memberi nama dari Activity yang pertama kali yang Anda buat. Selain itu Anda juga bisa mengganti bahasa *default* pada aplikasi tersebut menjadi bahasa Java atau Kotlin.



Gambar 4 Tampilan untuk Memberi Nama untuk Projek



membuat proyek pertama kali di Project Wizard, *done!*, hasilnya seperti gambar di bawah ini.



Gambar 5 Tampilan Project Wizard Pertama Kali di Android Studio

Menambahkan Code Sederhana pada Layout Activity

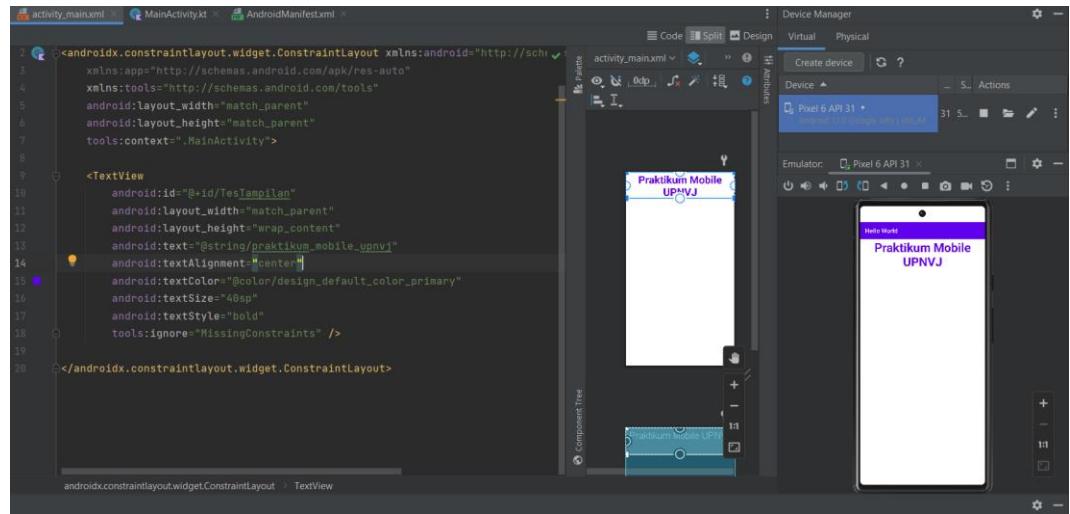
- 1) Silakan pilih berkas activity_main.xml pada *workspace* Anda (res/ layout/ activity_main.xml).

Ubah tulisan yang ada TextView seperti berikut:

A screenshot of the Android Studio code editor showing the XML code for 'activity_main.xml'. The code defines a ConstraintLayout with a single TextView inside. The TextView has an id of '@+id/TestTempilan', a font size of 40sp, and bold text style. It also has a center alignment and wrap content height.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8
9     <TextView
10         android:id="@+id/TestTempilan"
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:text="@string/praktikum_mobile_upnvi"
14         android:textAlignment="center"
15         android:textColor="@color/design_default_color_primary"
16         android:textSize="40sp"
17         android:textStyle="bold"
18         tools:ignore="MissingConstraints" />
19
20 </androidx.constraintlayout.widget.ConstraintLayout>
```





c) Jalankan Aplikasi di Emulator dan Smartphone

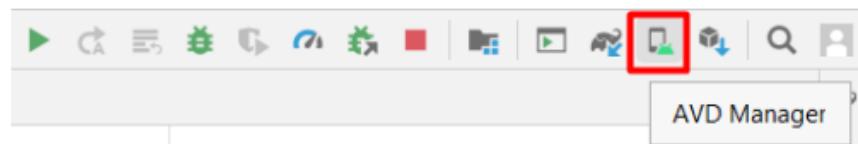
Untuk menjalankan atau running di android studio berbeda dengan di Idea yang lain sebab harus installasi emulator terlebih dahulu untuk bisa melihat aplikasi yang kita buat.

Installasi Emulator

Dalam menjalankan program di android studio, bisa menggunakan 2 cara, yaitu, Installasi emulator yang terdapat di Android Studio maupun menggunakan smartphone berbasis Android. Cara installasi bisa juga dilihat dari [video ini](#).

i. Instalasi emulator yang terdapat di Android studio

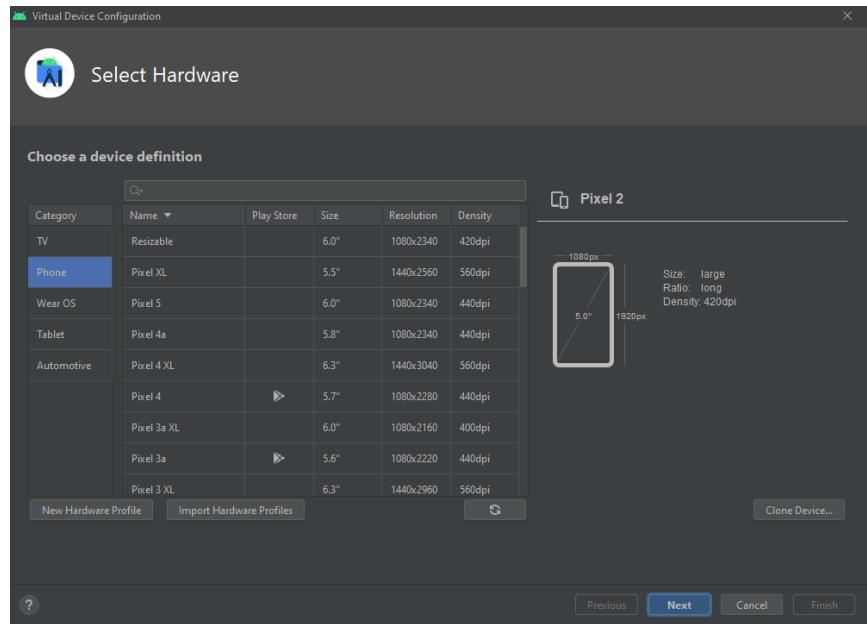
- Buka AVD Manager dengan klik icon berikut:



Gambar 6 Tampilan AVD

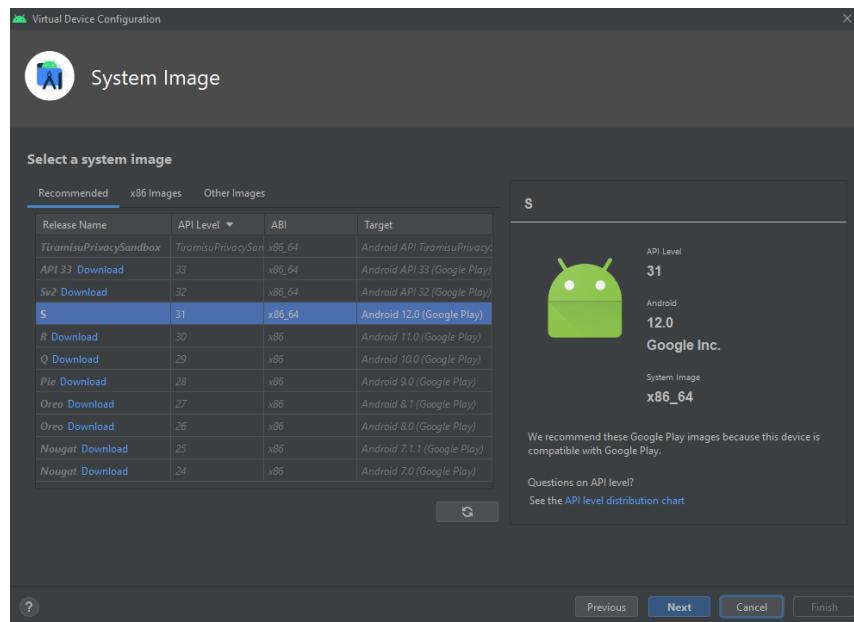
- Kemudian pilih **Virtual - Create Device**
- Akan muncul dialog dengan pilihan beberapa emulator yang bisa Anda pilih. Pilih salah satu dan klik **Next**.





Gambar 7 Tampilan Untuk Memilih Device Untuk Emulator

- Kemudian pilih satu system image, usahakan pilih versi Android yang terbaru dan klik Next



Gambar 8 Tampilan Untuk Memilih API Level

- Selanjutnya, maka akan download System Image tersebut. Jika sudah selesai, Anda dapat menekan tombol **Finish**.
- Pengaturan emulator sudah selesai dan bisa langsung dijalankan.



Konfigurasi Hardware Accelerated Execution Manager (HAXM)

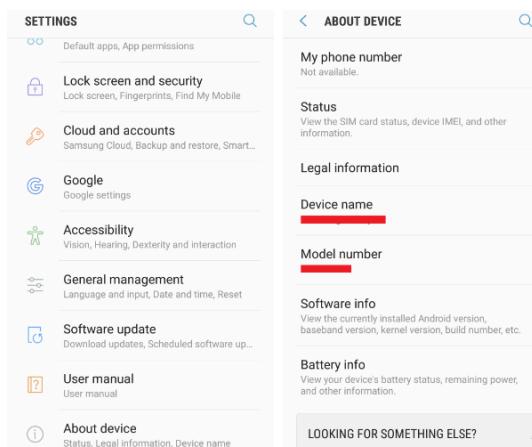
Dalam beberapa device yang tidak mendukung Emulator, Anda perlu menginstal HAXM. HAXM adalah hardware-assisted virtualization engine yang menggunakan teknologi VT dari Intel untuk mempercepat aplikasi Android yang diemulasi di mesin host. HAXM diperlukan untuk menjalankan emulator di Android Studio.

HAXM diperlukan jika sistem operasi yang Anda gunakan adalah Windows atau Mac. Untuk menginstalnya, ikuti petunjuk berikut ini:

1. Buka SDK Manager.
2. Pilih SDK Update Sites, kemudian hidupkan Intel HAXM.
3. Tekan OK.
4. Cari berkas installer-nya di directory folder sdk komputer Anda, ~sdk\extras\intel\Hardware_Accelerated_Execution_Manager\intelhaxm-android.exe.
5. Jalankan installer dan ikuti petunjuknya sampai selesai.

ii. Kemudian bisa juga dengan menggunakan smartphone berbasis Android

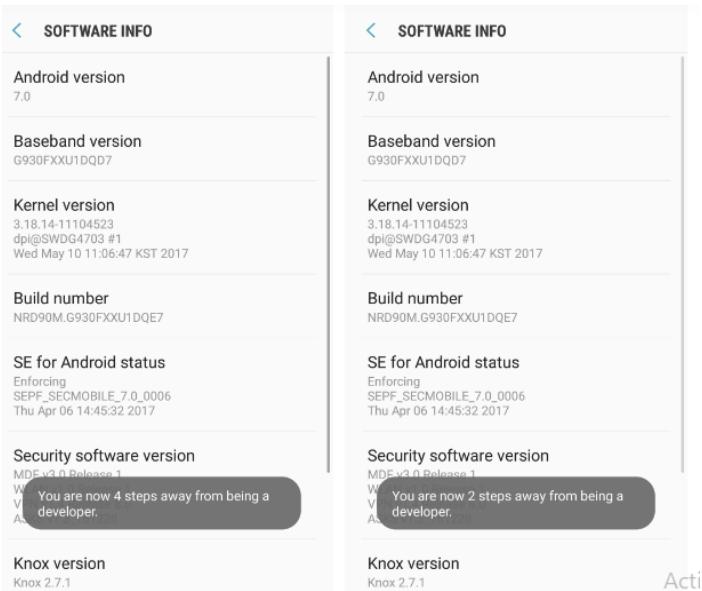
- a. Pastikan peranti yang akan dipakai sesuai dengan target SDK atau paling tidak mendukung versi SDK terendah yang digunakan aplikasi
- b. Buka *setting* dan masuk ke dalam menu **About** (karena letak about phone ini berbeda – beda tergantung jenis smartphone). Pada halaman menu ini, Anda perlu menemukan informasi tentang **Build number**.





Gambar 9 Cara setting Smartphone terhubung dengan Android Studio

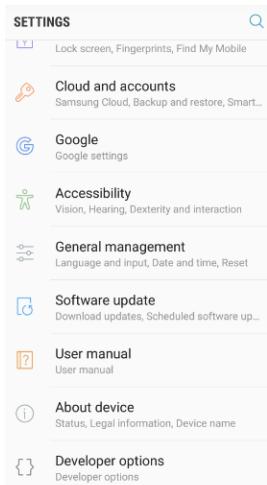
c. Kemudian tekan **Build number** sebanyak *7 kali*.





Gambar 10 Membuat Smartphone menjadi Developer

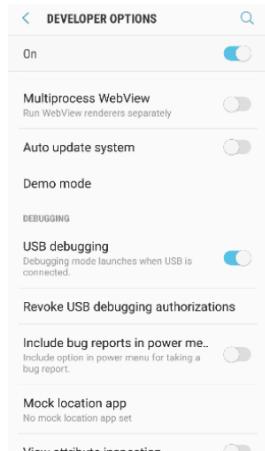
- d. Kembali ke **menu setting** di awal dan akan muncul menu baru di bawah **about** yaitu **Developer Options**.



Gambar 11 Tampilan di Smartphone setelah menjadi Developer

- e. Masuk ke dalam menu **Developer Options** dan pastikan opsi **USB Debugging Mode** sudah *on*.





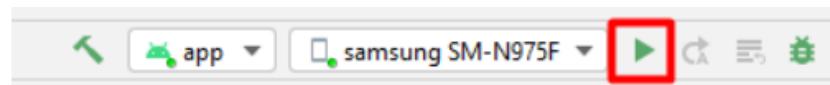
Gambar 12 Tampilan untuk USB Debugging Mode harus On

- f. Setelah menyelesaikan pengaturan pada peranti, peranti pun dapat dihubungkan dengan laptop atau komputer dengan menggunakan kabel data. Klik **Izinkan** dan centang untuk **selalu izinkan** apabila muncul pop up **Izinkan debugging USB**.



Gambar 13 tampilan Izinkan debugging USB

- g. Jika device sudah terdeteksi pada Android Studio, ia akan otomatis muncul pada daftar device dan Anda bisa menjalankan aplikasi dengan klik ikon segitiga hijau berikut:

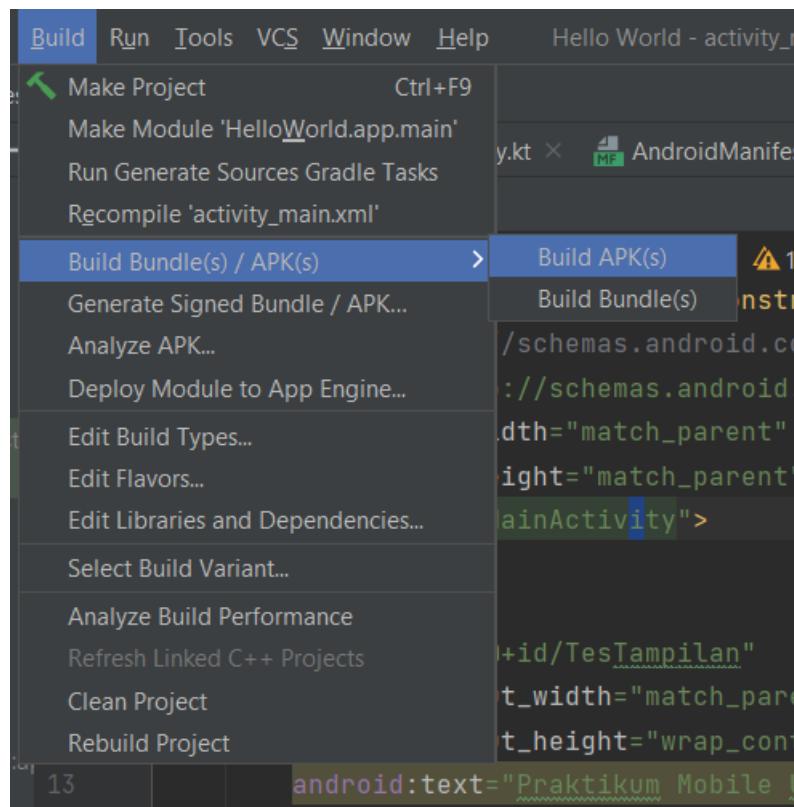


Gambar 14 Tanda untuk Menjalankan atau Running Aplikasi

d) Build APK

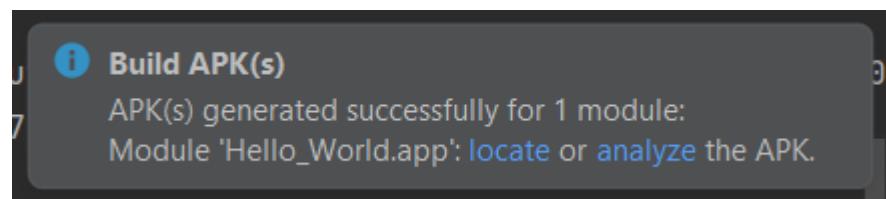
Sekarang klik menu Build → Build Bundles(s) / APK(s) → Build APK(s).





Gambar 15 Cara Build APK(s)

Ketika berhasil, Rekan dapat melihat notifikasi pada sudut kanan atas Android Studio:



Biasanya lokasi Build APK mengikuti struktur **project-name/module-name/outputs/apk/**. Jika proyek bernama Hello World, maka lokasinya adalah **HelloWorld/app/build/outputs/apk/apk-debug.apk**



5. Tugas Praktikum

- a. Buatlah Projek Android Pertama dengan ketentuan:

Nama Projek : Nama Masing-masing

Api : 21

Ketentuan yang dikumpulkan: output dalam bentuk emulator maupun smartphone (dalam bentuk video)

- b. Contoh Aplikasi Android (3) dengan tema yang berbeda dan masing-masing mahasiswa menjelaskan kelebihan & kekurangan (masing-masing 3) berbeda juga.
- c. Buatlah Kelompok dengan terdiri dari 3 mahasiswa untuk membuat Projek Akhir, yaitu membuat Aplikasi Mobile berbasis Android.



Praktikum 2

Merancang *Layout Application* dan Membuat *Splash Screen*

a) Sub Pokok Pembahasan

- a. Membuat Layout Application
- b. Membuat Splash Screen

b) Capaian Pembelajaran

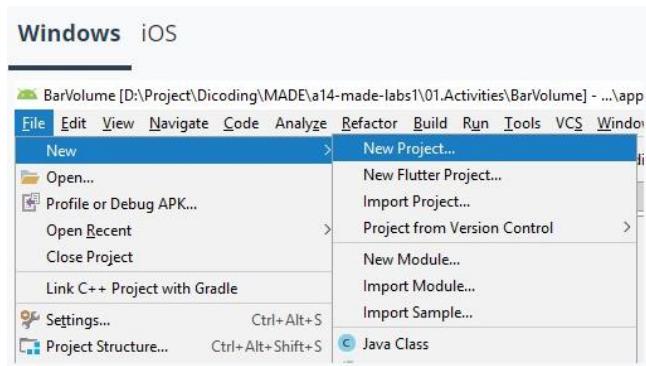
- a. Mahasiswa mampu menanamkan sikap jujur, bertanggung jawab, disiplin, saling menghargai dan berinovasi dengan tetap menjunjung tinggi nilai-nilai agama dan ketuhanan.
- b. Mahasiswa mampu menunjukkan sikap nasionalisme, mampu bekerja dalam tim sesuai kaidah keilmuan pada proyek pengembangan aplikasi mobile programming.
- c. Mahasiswa mampu berpartisipasi aktif, bertanggung jawab, dan memiliki motivasi mengembangkan diri dalam mendapatkan solusi permasalahan melalui aplikasi mobile programming
- d. Mahasiswa mampu menguasai, merancang dan mengembangkan bahasa pemrograman berbasis mobile, serta mampu menganalisis solusi dalam model bahasa pemrograman mobile.

c) Membuat Layout Application

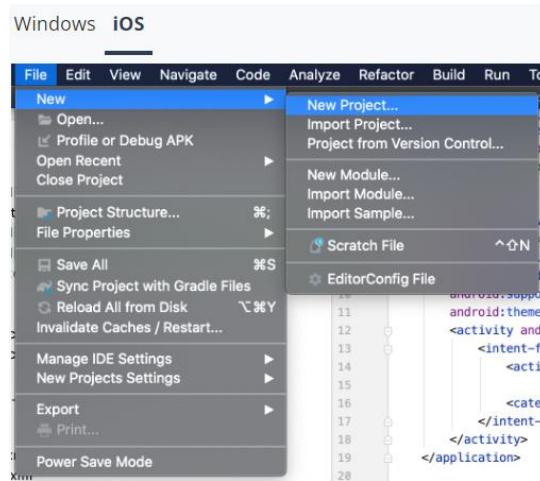
Layout yang dimaksud adalah mengatur tata letak suatu komponen pada pemrograman android, file yang diperlukan adalah sebuah file XML. File XML tersebut digunakan untuk mengatur antarmuka suatu aplikasi android. Untuk meletakkan komponen- komponen tersebut diperlukan sebuah cara agar komponen yang dibuat sesuai dengan kebutuhan user dan memudahkan user (user friendly). Cara yang digunakan untuk mengatur tata letak yaitu menggunakan layout. Nantinya File XML untuk membuat antarmuka tersebut diletakkan pada project_android -> res -> layout.

- a. Buat proyek baru dengan klik **File → New → New Project** pada Android Studio Anda atau Anda bisa memilih **Start a new Android Studio project** di bagian **dashboard**.



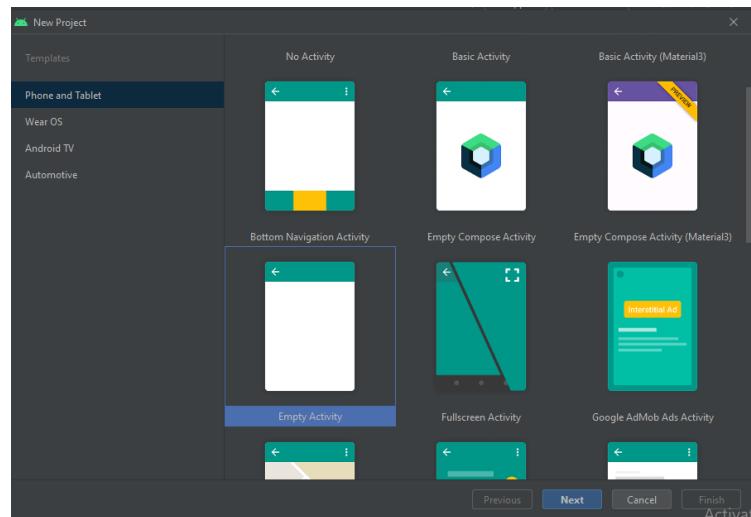


Gambar 16 Tampilan untuk Windows saat Membuat Project Baru



Gambar 17 Tampilan untuk Ios saat Membuat Project Baru

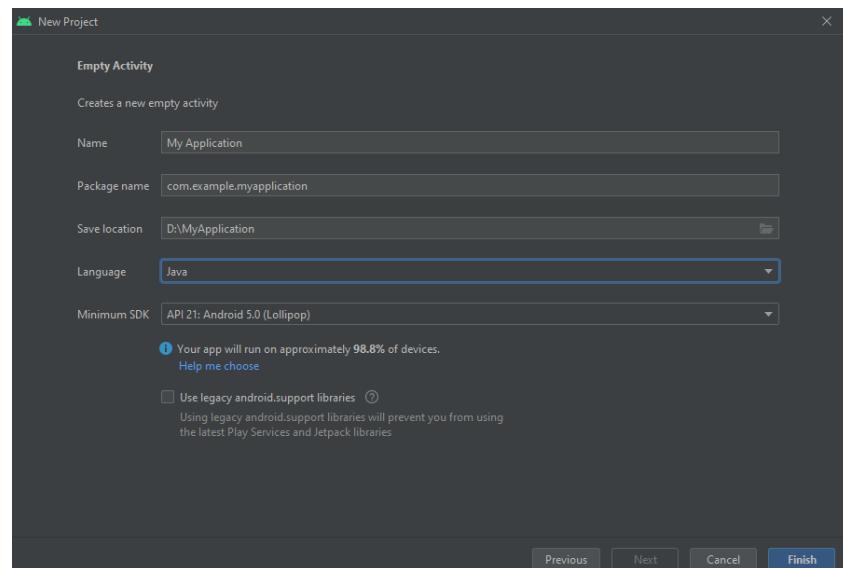
- b. Pada bagian ini kita akan memilih tipe activity awal dari *template* yang telah disediakan. Saat ini kita pilih tipe **Empty Activity** untuk belajar dari yang paling dasar, lalu klik **Next** untuk melanjutkan.



Gambar 18 Tampilan Template Untuk Memilih New Device

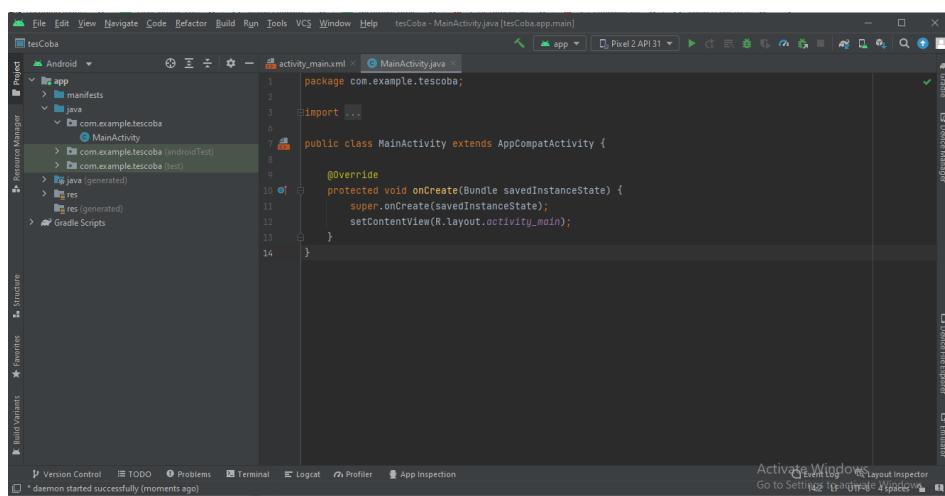


- c. Selanjutnya masukkan **nama aplikasi** dan nama *package* aplikasi Anda. Sebaiknya jangan sama dengan apa yang ada di contoh, karena ini berfungsi sebagai id dari aplikasi yang Anda buat. Kemudian Anda bisa menentukan lokasi proyek yang akan Anda buat. Setelah itu pilih tipe gawai/peranti (*device*) untuk aplikasi beserta target minimum SDK yang akan digunakan. Pilihan target Android SDK akan mempengaruhi banyaknya peranti yang dapat menggunakan aplikasi. Di sini kita memilih nilai minimum SDK kita pasang ke **Level 21 (Lollipop)**. Klik **Finish** untuk melanjutkan.



Gambar 19 Tampilan Android Studio dalam Memberikan Nama untuk Project

- d. Tampilan layar Anda akan seperti contoh di bawah ini



Gambar 20 Tampilan Layar Android Studio untuk Memulai Coding

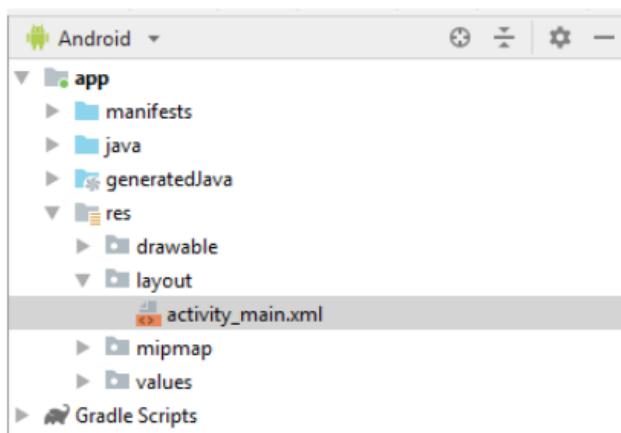


- e. Di sebelah kanan Anda adalah *workspace* di mana Activity Anda berada dan bernama MainActivity dengan layout-nya activity_main.xml. Di sebelah kiri Anda terdapat struktur proyek, di mana nanti kita akan banyak menambahkan berbagai komponen baru, *asset* dan *library*.

Layouting

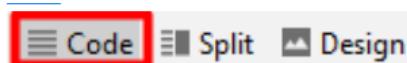
Menambahkan Code Sederhana pada Layout Activity

- 2) Silakan pilih berkas activity_main.xml pada *workspace* Anda (res/ layout/ activity_main.xml).



Gambar 21 Tampilan activity_main.xml di Android Studio

Maka akan ada tampilan seperti ini, kemudian pilih **tab Code** di sebelah pojok kanan atas.



Gambar 22 Tampilan untuk Coding di Layout .xml

Ubah layout dasar dari ConstraintLayout menjadi LinearLayout seperti berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/a
    xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MainActivity">

</LinearLayout>
```



```

1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    id=""
3.     xmlns:tools="http://schemas.android.com/tools"
4.     android:layout_width="match_parent"
5.     android:layout_height="match_parent"
6.     android:padding="16dp"
7.     android:orientation="vertical">
8.
9.     <TextView
10.         android:layout_width="match_parent"
11.         android:layout_height="wrap_content"
12.         android:text="Panjang" />
13.
14.     <EditText
15.         android:id="@+id/edt_length"
16.         android:layout_width="match_parent"
17.         android:layout_height="wrap_content"
18.         android:inputType="numberDecimal"
19.         android:lines="1" />
20.
21.     <TextView
22.         android:layout_width="match_parent"
23.         android:layout_height="wrap_content"
24.         android:text="Lebar" />
25.
26.     <EditText
27.         android:id="@+id/edt_width"
28.         android:layout_width="match_parent"
29.         android:layout_height="wrap_content"
30.         android:inputType="numberDecimal"
31.         android:lines="1" />
32.
33.     <TextView
34.         android:layout_width="match_parent"
35.         android:layout_height="wrap_content"
36.         android:text="Tinggi" />
37.
38.     <EditText
39.         android:id="@+id/edt_height"
40.         android:layout_width="match_parent"
41.         android:layout_height="wrap_content"
42.         android:inputType="numberDecimal"
43.         android:lines="1" />
44.
45.     <Button
46.         android:id="@+id/btn_calculate"
47.         android:layout_width="match_parent"
48.         android:layout_height="wrap_content"
49.         android:text="Hitung" />
50.

```



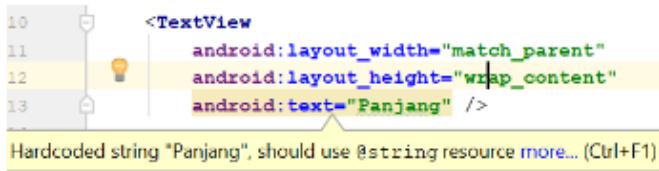
```

51.    <TextView
52.        android:id="@+id/tv_result"
53.        android:layout_width="match_parent"
54.        android:layout_height="wrap_content"
55.        android:gravity="center"
56.        android:text="Hasil"
57.        android:textSize="24sp"
58.        android:textStyle="bold" />
59.    </LinearLayout>

```

Perlu diperhatikan *root layout* (tag layout terluar) yang dipakai di sini adalah LinearLayout. Jika kita menggunakan Android Studio versi 3 ke atas, secara *default root* yang dipakai adalah ConstraintLayout. Agar sesuai dengan latihan ini, kita tinggal menggantinya menjadi LinearLayout.

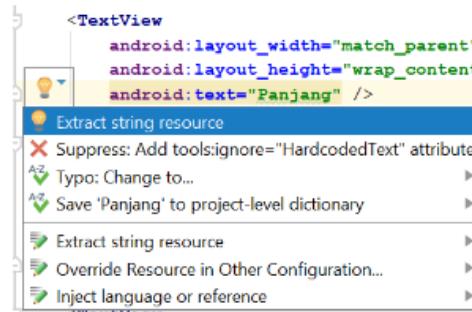
- 3) Perhatikan bahwa ada *warning* berwarna kuning pada atribut `android:text` di layout tersebut.



Gambar 23 Warning Kuning pada Atribut `android:txt`

Ini karena kita melakukan *hardcoding* (menuliskan teks secara langsung) pada nilai *string*-nya. Ini merupakan praktik yang kurang baik karena seharusnya kita menuliskan semua teks pada berkas `res/values/strings.xml` dan setelah itu baru memanggilnya.

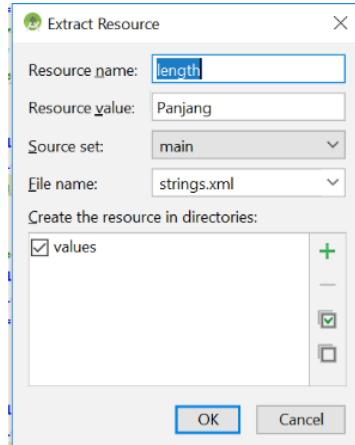
Okey, mari kita hilangkan *warning* tersebut dengan menekan **Alt+Enter** (option + return pada Mac) atau menekan lampu kuning yang muncul pada attribut `android:text`. Akan muncul dialog seperti ini, pilih **extract string resource**.



Gambar 24 Tampilan Muncul Dialog setelah klik Lampu Kuning

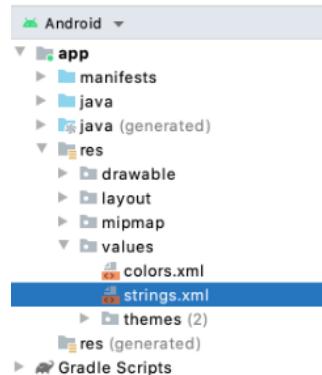


- 4) Kemudian akan muncul *dialog* seperti di bawah ini. Sesuaikan dengan nama yang dibuat.



Gambar 25 Tampilan Setelah Pilih **extract string resource**

- 5) Fungsi **extract string resource** akan secara otomatis menambahkan nilai dari **android:text** ke dalam berkas **res/values/strings.xml**.



Gambar 26 Tampilan Setelah Menambahkan strings.xml

Lakukan hal yang sama pada setiap text lainnya. Jika kita buka berkas **strings.xml** yang ada di folder **res/value**, maka isinya akan menjadi seperti ini:

```
1. <resources>
2.   <string name="app_name">BarVolume</string>
3.   <string name="width">Lebar</string>
4.   <string name="height">Tinggi</string>
5.   <string name="calculate">Hitung</string>
6.   <string name="result">Hasil</string>
7.   <string name="length">Panjang</string>
8. </resources>
```

- 6) Maka kode di dalam **activity_main.xml** akan menjadi seperti ini:



```

1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    id=""
3.     xmlns:tools="http://schemas.android.com/tools"
4.     android:layout_width="match_parent"
5.     android:layout_height="match_parent"
6.     android:padding="16dp"
7.     android:orientation="vertical">
8.
9.     <TextView
10.         android:layout_width="match_parent"
11.         android:layout_height="wrap_content"
12.         android:text="@string/length" />
13.
14.     <EditText
15.         android:id="@+id/edt_length"
16.         android:layout_width="match_parent"
17.         android:layout_height="wrap_content"
18.         android:inputType="numberDecimal"
19.         android:lines="1" />
20.
21.     <TextView
22.         android:layout_width="match_parent"
23.         android:layout_height="wrap_content"
24.         android:text="@string/width" />
25.
26.     <EditText
27.         android:id="@+id/edt_width"
28.         android:layout_width="match_parent"
29.         android:layout_height="wrap_content"
30.         android:inputType="numberDecimal"
31.         android:lines="1" />
32.
33.     <TextView
34.         android:layout_width="match_parent"
35.         android:layout_height="wrap_content"
36.         android:text="@string/height" />
37.
38.     <EditText
39.         android:id="@+id/edt_height"
40.         android:layout_width="match_parent"
41.         android:layout_height="wrap_content"
42.         android:inputType="numberDecimal"
43.         android:lines="1" />
44.
45.     <Button
46.         android:id="@+id/btn_calculate"
47.         android:layout_width="match_parent"
48.         android:layout_height="wrap_content"
49.         android:text="@string/calculate" />
50.

```

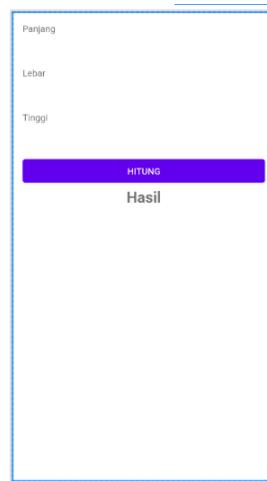


```

51.    <TextView
52.        android:id="@+id/tv_result"
53.        android:layout_width="match_parent"
54.        android:layout_height="wrap_content"
55.        android:gravity="center"
56.        android:text="@string/result"
57.        android:textSize="24sp"
58.        android:textStyle="bold" />
59.    </LinearLayout>

```

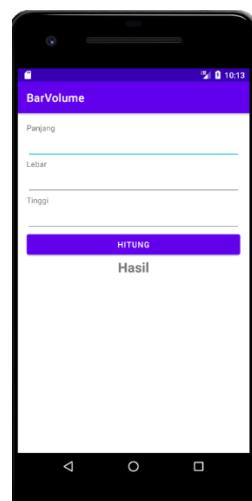
7) hasil layout sementara akan menjadi seperti ini:



Gambar 27 Tampilan Layout Sementara setelah di jalankan

Selain menggunakan *code* seperti di atas, Anda juga dapat membuat layout dengan menggunakan *design*. Untuk tutorialnya dapat Anda lihat di link video [berikut ini](#)

8) Kemudian jalankan aplikasi design tersebut, maka output akan terlihat di bawah ini:



Gambar 28 Tampilan Hasil dari Output



d) Membuat Splash Screen Android

Splashscreen adalah tampilan tambahan yang akan muncul saat pertama kali membuat suatu aplikasi

1) Buat project baru

Buka android studio kemudian klik create new project. Kemudian isi :

Application name : splasscreen

Phone and tablet : API 21 (disesuaikan)

Activity : empty activity

Activity name : mainactivity

2) Atur layout

Selanjutnya atur layout splash screennya pada **activity_main.xml**. Untuk logonya sendiri pada modul ini menggunakan logo dari aplikasi gojek. Bisa di download [disini](#). Jika sudah didownload masukan logo atau gambar yang ingin digunakan pada drawable dengan cara :

- klik kanan pada **drawable**
- copykan gambar ke folder **drawable**

Lalu ketikan kode-kode dibawah ini pada **activity_main.xml**.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    android:gravity="center"
    tools:context=".MainActivity">

    <ImageView
        android:layout_width="160dp"
        android:layout_height="60dp"
        android:id="@+id/logo" />
```



```
    android:src="@drawable/logogojek"  
    android:layout_centerVertical="true"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentStart="true" />  
  
  </RelativeLayout>
```

3) Atur style

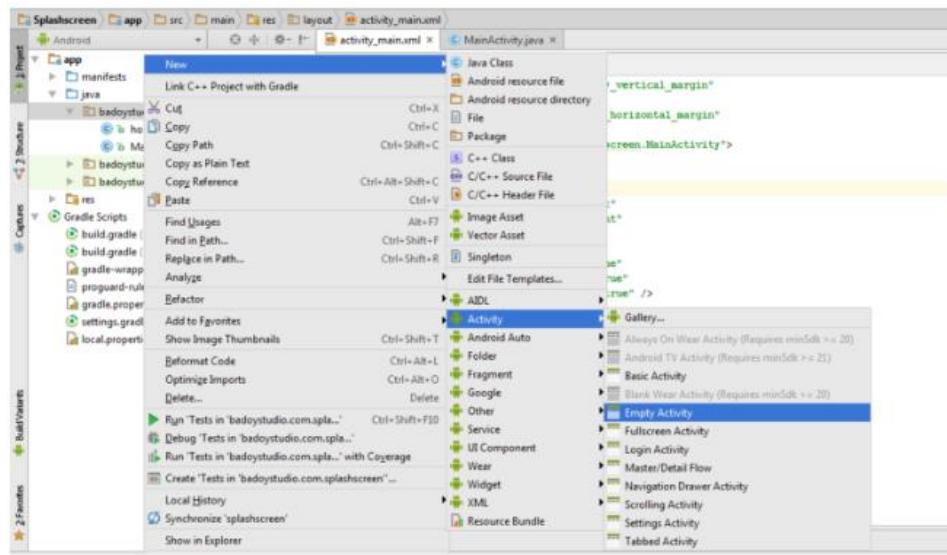
Pada **styles.xml** yang berada pada folder res kita akan menghilangkan app bar (action bar) dan mengganti warna primarynya dengan warna putih. Silahkan masukan kode-kode dibawah ini.

```
<resources>  
  
    <!-- Base application theme. -->  
    <style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">  
        <!-- Customize your theme here. -->  
        <item name="colorPrimary">@color/colorPrimary</item>  
        <item name="colorPrimaryDark">#FFFFFF</item>  
        <item name="colorAccent">@color/colorAccent</item>  
    </style>  
  
</resources>
```

4) Buat Activity_home

Splash screen yang dibuat pada tutorial ini hanya akan tampil selama 4 detik setelah itu akan langsung ditampilkan menu utama dari aplikasi android yang sudah diberi splash screen.





Gambar 29 Cara Membuat tambahan untuk Activity_home

Tambahkan kode-kode dibawah ini pada **activity_home.xml** :

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_home"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="30dp"

    tools:context=".home">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="halo ini adalah tutorial splash screen "
        android:textSize="30sp"/>

</RelativeLayout>
```

5) MainActivity.java

Buka **MainActivity.java** lalu tambahkan kode-kode java berikut.



```
package badoystudio.com.splashscreengojek;

import android.content.Intent;
import android.os.Handler;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    private int waktu_loading=4000;

    //4000=4 detik

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {

                //setelah loading maka akan langsung berpindah ke home activity
                Intent home=new Intent(MainActivity.this, home.class);
                startActivity(home);
                finish();
            }
        },waktu_loading);
    }
}
```

Perhatikan kode di atas **waktu_loading=4000** yaitu splash screen nantinya hanya akan muncul selama 4 detik saja kemudian langsung pindah ke activity yang lain.



6) **AndroidManifest.xml**

Terakhir tambahkan script dibawah ini pada **AndroidManifest.xml** yang ada pada manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="badoystudio.com.splashscreengojek">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".home"></activity>
    </application>

</manifest>
```

7) **Running Aplikasi**

Jalankan aplikasi tersebut sehingga nanti akan menampilkan seperti gambar di bawah ini





Gambar 30 Tampilan saat dijalankan hasil dari Splash Screen

e) Tugas Praktikum

- a. Buatlah layout aplikasi apa saja di Android Studio dengan ketentuan:
Nama Projek : (Masing-masing mahasiswa berbeda)
API : 21
File yang dikumpulkan ialah coding di screen shot dan output di videokan
- b. Membuat Splash Screen dengan logo setiap mahasiswa berbeda selama 10 detik, file yang dikumpulkan coding di screen shot dan output di videokan
- c. Progress project akhir dengan membuat abstrak dari projek tersebut dengan ketentuan (latar belakang dalam membuat aplikasi, kelebihan & kekurangan, tujuan, dan nama aplikasi tersebut)



Praktikum 3

View & Views Group dan Style & Theme

1. Sub Pokok Pembahasan

- a. View & Views Group
- b. Style & Theme

2. Capaian Pembelajaran

- a. Mahasiswa mampu menanamkan sikap jujur, bertanggung jawab, disiplin, saling menghargai dan berinovasi dengan tetap menjunjung tinggi nilai-nilai agama dan ketuhanan.
- b. Mahasiswa mampu menunjukkan sikap nasionalisme, mampu bekerja dalam tim sesuai kaidah keilmuan pada proyek pengembangan aplikasi mobile programming.
- c. Mahasiswa mampu berpartisipasi aktif, bertanggung jawab, dan memiliki motivasi mengembangkan diri dalam mendapatkan solusi permasalahan melalui aplikasi mobile programming
- d. Mahasiswa mampu menguasai, merancang dan mengembangkan bahasa pemrograman berbasis mobile, serta mampu menganalisis solusi dalam model bahasa pemrograman mobile.

3. View & Views Group

Tujuan

Pada latihan kali ini akan mengembangkan sebuah halaman detail yang di dalamnya terdiri dari beberapa komponen View dan ViewGroup layout.

Codelab Views and ViewGroup

- 1. Buat Project baru di Android Studio dengan kriteria sebagai berikut:

Nama Project	MyViewAndViews
Target & Minimum Target SDK	Phone and Tablet, Api level 21
Tipe Activity	Empty Activity
Activity Name	MainActivity
Use AndroidX artifacts	True
Language	Java / Kotlin



- 2) Lalu buka berkas build.gradle (Module: app) dan tambahkan satu baris ini di bagian **dependecies**:

implementation 'de.hdodenhof:circleimageview:3.1.0'

Sehingga berkas **build.gradle(Module: app)** sekarang menjadi seperti ini:

```
1. apply plugin: 'com.android.application'  
2.  
3. android {  
4.     ...  
5. }  
6.  
7. dependencies {  
8.     ...  
9.     implementation 'de.hdodenhof:circleimageview:3.1.0'}
```

Tekanlah **Sync Now** untuk men-download library dari Circle ImageView ke dalam proyek Android Studio.

- 3) Lalu buka berkas strings.xml di **res → values**. Sesuaikan isinya dengan seperti ini:

```
1) <resources>  
2) <string name="app_name">MyViewAndViews</string>  
3) <string name="content_text">Google officially  
announced its much-anticipated Pixel phones; the  
Pixel and Pixel XL, on October 4. We attended  
Google's London UK event, mirroring the main one  
taking place in San Francisco, US, where the firm  
unwrapped the new Android 7.1 Nougat devices which  
will apparently usurp Google's long-standing Nexus  
series.</string>  
4) <string name="content_specs_display">5.0 inches\n  
5) FHD AMOLED at 441ppi\n  
6) 2.5D Corning® Gorilla® Glass 4</string>  
7) <string name="content_specs_size">5.6 x 2.7 x 0.2 ~  
0.3 inches 143.8 x 69.5 x 7.3 ~ 8.5 mm</string>  
8) <string name="content_specs_battery">2,770 mAh  
battery\n  
9) Standby time (LTE): up to 19 days\n  
10) Talk time (3g/WCDMA): up to 26 hours\n  
11) Internet use time (Wi-Fi): up to 13 hours\n  
12) Internet use time (LTE): up to 13 hours\n  
13) Video playback: up to 13 hours\n  
14) Audio playback (via headset): up to 110 hours\n  
15) Fast charging: up to 7 hours of use from only  
15 minutes of charging</string>  
16) <string name="stock">Stock hanya 5  
buah</string>
```



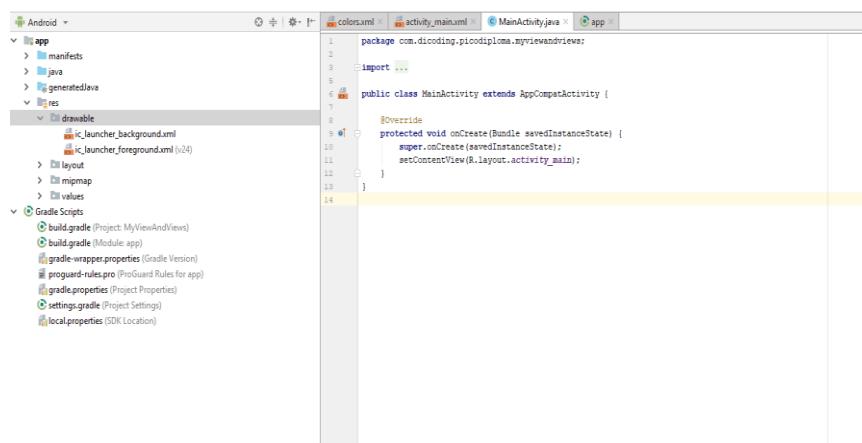
```

17)      <string
18)          name="specification">Spesifikasi</string>
19)      <string name="display">Display</string>
20)      <string name="size">Size</string>
21)      <string name="battery">Battery</string>
22)      <string name="seller">Dijual oleh</string>
23)      <string name="my_name">Narendra
24)          Wicaksono</string>
25)      <string name="buy">Beli</string>
26)      <string name="dummy_value">$735</string>
27)      <string name="dummy_photos">6 photos</string>

```

</resources>

- 4) Selanjutnya, unduh asetnya di [sini](#). Bila telah selesai, salin semua asset yang dibutuhkan ke dalam direktori **res → drawable**.



Gambar 31 Cara Memasukkan Gambar ke database Android Studio

- 5) Buka berkas **activity_main.xml** dan ubah layout utama menjadi seperti ini:

```

1)  <?xml version="1.0" encoding="utf-8"?>
2)  <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
3)      android:layout_width="match_parent"
4)      android:layout_height="match_parent">
5)  </ScrollView>

```

- 6) Kemudian tambahkan layout utama di dalam ScrollView.

```

1)  <?xml version="1.0" encoding="utf-8"?>
2)  <ScrollView
3)      xmlns:android="http://schemas.android.com/apk/res/android"
4)          android:layout_width="match_parent"
5)          android:layout_height="match_parent">

```



```

6)      <LinearLayout
7)          android:layout_width="match_parent"
8)          android:layout_height="wrap_content"
9)          android:orientation="vertical">
10)
11)      </LinearLayout>
</ScrollView>

```

- 7) Kemudian kita masukkan komponen di dalam LinearLayout yang sudah ditambahkan sebelumnya.

```

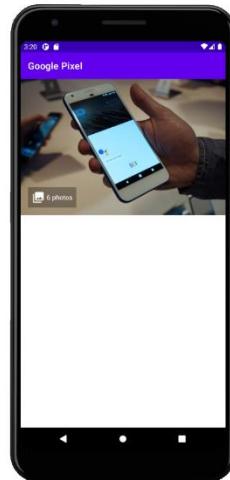
1) <?xml version="1.0" encoding="utf-8"?>
2) <ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
3)     android:layout_width="match_parent"
4)     android:layout_height="match_parent">
5)
6)     <LinearLayout
7)         android:layout_width="match_parent"
8)         android:layout_height="wrap_content"
9)         android:orientation="vertical">
10)
11)         <FrameLayout
12)             android:layout_width="match_parent"
13)             android:layout_height="wrap_content">
14)
15)             <ImageView
16)                 android:layout_width="match_parent"
17)                 android:layout_height="wrap_content"
18)                 android:adjustViewBounds="true"
19)                 android:scaleType="fitXY"
20)                 android:src="@drawable/pixel_google" />
21)
22)             <TextView
23)                 android:layout_width="wrap_content"
24)                 android:layout_height="wrap_content"
25)                 android:textAppearance="@style/TextAppearance.AppCompat.Small"
26)                     android:textColor="@android:color/white"
27)                     android:layout_gravity="bottom"
28)                     android:layout_marginStart="16dp"
29)                     android:layout_marginBottom="16dp"
30)                     android:background="#4D000000"
31)                     android:drawablePadding="4dp"
32)                     android:gravity="center_vertical"
33)                     android:padding="8dp"
34)                     android:text="@string/dummy_photos"
35)                     android:drawableStart="@drawable/ic_collections
    _white_18dp" />

```



```
36)      </FrameLayout>
37)      </LinearLayout>
        </ScrollView>
```

Untuk sementara tampilan akan seperti di bawah ini:



Gambar 32 Tampilan Sementara 1 Saat Dijalankan Aplikasi

- 8) Selanjutnya, tambahkan TextView di bawah FrameLayout, menjadi seperti ini:

```
1) <?xml version="1.0" encoding="utf-8"?>
2) <ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
3) android:layout_width="match_parent"
4) android:layout_height="match_parent">
5)
6) <LinearLayout
7) android:layout_width="match_parent"
8) android:layout_height="wrap_content"
9) android:orientation="vertical">
10)
11)     <FrameLayout ... >
12)
13)     <TextView
14)         android:layout_width="match_parent"
15)         android:layout_height="wrap_content"
16)         android:layout_marginLeft="16dp"
17)         android:layout_marginRight="16dp"
18)         android:layout_marginTop="16dp"
19)         android:layout_marginBottom="8dp"
20)         android:text="@string/dummy_value"
21)         android:textSize="32sp" />
22)
23)     <TextView
24)         android:layout_width="match_parent"
25)         android:layout_height="wrap_content"
```

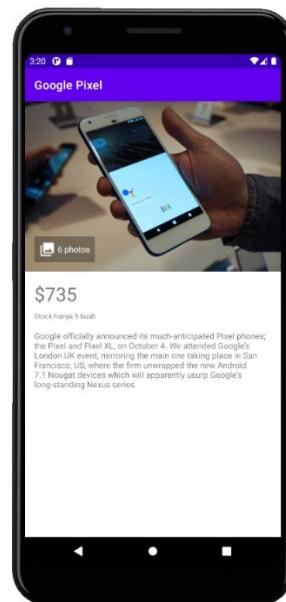


```

26)     android:textSize="12sp"
27)     android:layout_marginLeft="16dp"
28)     android:layout_marginRight="16dp"
29)     android:layout_marginBottom="16dp"
30)     android:text="@string/stock" />
31)
32) <TextView
33)     android:layout_width="match_parent"
34)     android:layout_height="wrap_content"
35)     android:layout_marginLeft="16dp"
36)     android:layout_marginRight="16dp"
37)     android:layout_marginBottom="16dp"
38)     android:lineSpacingMultiplier="1"
39)     android:text="@string/content_text" />
40)
41)   </LinearLayout>
</ScrollView>

```

Tampilan akan berubah seperti ini



Gambar 33 Tampilan Sementara 2 Saat Dijalankan Aplikasi

- 9) Setelah menambahkan beberapa TextView, selanjutnya akan ditambahkan TableLayout.

```

1) <?xml version="1.0" encoding="utf-8"?>
2) <ScrollView
  xmlns:android="http://schemas.android.com/apk/res/android"
3)   android:layout_width="match_parent"
4)   android:layout_height="match_parent">
5)
6)   <LinearLayout
7)     android:layout_width="match_parent"
8)     android:layout_height="wrap_content"

```



```

9) android:orientation="vertical">
10)
11)      <FrameLayout ...>
12)          <TextView ...>
13)              <TextView ...>
14)                  <TextView ...>
15)
16)                      <TextView
17)                          android:layout_width="match_parent"
18)                          android:layout_height="wrap_content"
19)                          android:textSize="12sp"
20)                          android:layout_marginLeft="16dp"
21)                          android:layout_marginRight="16dp"
22)                          android:layout_marginBottom="8dp"
23)                              android:text="@string/specification
" />
24)
25)      <TableLayout
26)          android:layout_width="match_parent"
27)          android:layout_height="wrap_content"
28)          android:layout_marginLeft="16dp"
29)          android:layout_marginRight="16dp"
30)          android:layout_marginBottom="16dp">
31)
32)          <TableRow
33)              android:layout_width="match_parent"
34)              android:layout_height="wrap_content"
35)              android:layout_marginBottom="8dp">
36)
37)              <TextView
38)                  android:layout_width="wrap_content"
39)                  android:layout_height="wrap_content"
40)                  android:textSize="14sp"
41)                  android:layout_marginEnd="16dp"
42)                  android:text="@string/display" />
43)
44)              <TextView
45)                  android:layout_width="match_parent"
46)                  android:layout_height="wrap_content"
47)                  android:textSize="14sp"
48)                  android:layout_weight="1"
49)                  android:text="@string/content_specs_display" />
50)          </TableRow>
51)
52)          <TableRow
53)              android:layout_width="match_parent"
54)              android:layout_height="wrap_content"
55)              android:layout_marginBottom="8dp">
56)
57)              <TextView
58)                  android:layout_width="wrap_content"
59)                  android:layout_height="wrap_content"

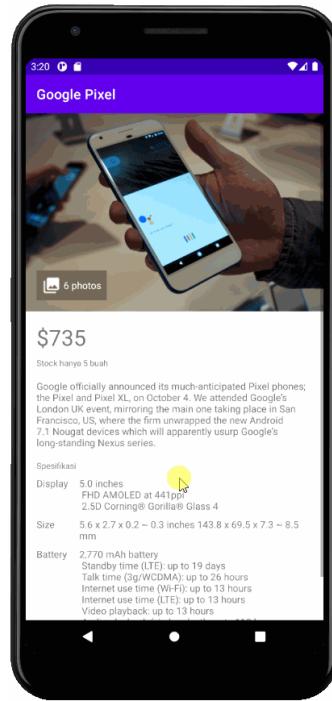
```



```
60)    android:layout_marginEnd="16dp"
61)    android:textSize="14sp"
62)    android:text="@string/size" />
63)
64)    <TextView
65)        android:layout_width="match_parent"
66)        android:layout_height="wrap_content"
67)        android:layout_weight="1"
68)        android:textSize="14sp"
69)        android:text="@string/content_specs_size" />
70)    </TableRow>
71)
72)    <TableRow
73)        android:layout_width="match_parent"
74)        android:layout_height="wrap_content"
75)        android:layout_marginBottom="8dp">
76)
77)        <TextView
78)            android:layout_width="wrap_content"
79)            android:layout_height="wrap_content"
80)            android:layout_marginEnd="16dp"
81)            android:textSize="14sp"
82)            android:text="@string/battery" />
83)
84)        <TextView
85)            android:layout_width="match_parent"
86)            android:layout_height="wrap_content"
87)            android:layout_weight="1"
88)            android:textSize="14sp"
89)            android:text="@string/content_specs_battery" />
90)        </TableRow>
91)    </TableLayout>
92)
93)    </LinearLayout>
</ScrollView>
```



Tampilan berubah kembali, jadi seperti ini



Gambar 34 Tampilan Sementara 3 Saat Dijalankan Aplikasi

10) Kemudian setelah menambahkan TableLayout, kita masukkan ViewGroup lagi.Tambahkan RelativeLayout dan Button di bawah TableLayout menjadi seperti ini.

```
1) <?xml version="1.0" encoding="utf-8"?>
2) <ScrollView
   xmlns:android="http://schemas.android.com/apk/res/android"
3)   xmlns:app="http://schemas.android.com/apk/res-auto"
4)   android:layout_width="match_parent"
5)   android:layout_height="match_parent">
6)
7)   <LinearLayout
8)     android:layout_width="match_parent"
9)     android:layout_height="wrap_content"
10)    android:orientation="vertical">
11)
12)      <FrameLayout ...>
13)      <TextView ...>
14)      <TextView ...>
15)          <TextView ...>
16)          <TableLayout...>
17)
18)      <RelativeLayout
19)        android:layout_width="match_parent"
20)        android:layout_height="wrap_content"
21)        android:layout_marginLeft="16dp"
```



```

22)     android:layout_marginRight="16dp"
23)     android:layout_marginBottom="16dp">
24)
25)         <de.hdodenhof.circleimageview.CircleImageView
26)             android:id="@+id/profile_image"
27)             android:layout_width="56dp"
28)             android:layout_height="56dp"
29)             android:layout_centerVertical="true"
30)             android:layout_marginEnd="16dp"
31)             android:src="@drawable/photo_2" />
32)
33)         <TextView
34)             android:layout_width="match_parent"
35)             android:layout_height="wrap_content"
36)             android:layout_centerVertical="true"
37)             android:layout_toEndOf="@+id/profile_image"
38)             android:text="@string/my_name" />
39)     </RelativeLayout>
40)
41)     <Button
42)         android:text="@string/buy"
43)         android:layout_width="match_parent"
44)         android:layout_height="wrap_content"
45)         android:layout_marginLeft="16dp"
46)         android:layout_marginRight="16dp"
47)         android:layout_marginBottom="16dp" />
48)     </LinearLayout>
    </ScrollView>

```

11) Maka hasil akhir kode pada activity_main.xml menjadi seperti ini:

```

1) <?xml version="1.0" encoding="utf-8"?>
2) <ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
3)   xmlns:app="http://schemas.android.com/apk/res-auto"
4)   android:layout_width="match_parent"
5)   android:layout_height="match_parent">
6)
7)   <LinearLayout
8)     android:layout_width="match_parent"
9)     android:layout_height="wrap_content"
10)      android:orientation="vertical">
11)
12)      <FrameLayout
13)          android:layout_width="match_parent"
14)          android:layout_height="wrap_content">
15)
16)          <ImageView
17)              android:layout_width="match_parent"
18)              android:layout_height="wrap_content"

```



```

19)        android:adjustViewBounds="true"
20)        android:scaleType="fitXY"
21)        android:src="@drawable/pixel_google" />
22)
23)        <TextView
24)            android:layout_width="wrap_content"
25)            android:layout_height="wrap_content"
26)            android:textAppearance="@style/TextAppearance.AppCompat.Small"
27)            android:textColor="@android:color/white"
28)            android:layout_gravity="bottom"
29)            android:layout_marginStart="16dp"
30)            android:layout_marginBottom="16dp"
31)            android:background="#4D000000"
32)            android:drawablePadding="4dp"
33)            android:gravity="center_vertical"
34)            android:padding="8dp"
35)            android:text="@string/dummy_photos"
36)            android:drawableStart="@drawable/ic_collections_white_18dp" />
37)        </FrameLayout>
38)
39)        <TextView
40)            android:layout_width="match_parent"
41)            android:layout_height="wrap_content"
42)            android:layout_marginLeft="16dp"
43)            android:layout_marginRight="16dp"
44)            android:layout_marginTop="16dp"
45)            android:layout_marginBottom="8dp"
46)            android:text="@string/dummy_value"
47)            android:textSize="32sp" />
48)
49)        <TextView
50)            android:layout_width="match_parent"
51)            android:layout_height="wrap_content"
52)            android:textSize="12sp"
53)            android:layout_marginLeft="16dp"
54)            android:layout_marginRight="16dp"
55)            android:layout_marginBottom="16dp"
56)            android:text="@string/stock" />
57)
58)        <TextView
59)            android:layout_width="match_parent"
60)            android:layout_height="wrap_content"
61)            android:layout_marginLeft="16dp"
62)            android:layout_marginRight="16dp"
63)            android:layout_marginBottom="16dp"
64)            android:lineSpacingMultiplier="1"
65)            android:text="@string/content_text" />
66)
67)        <TextView
68)            android:layout_width="match_parent"

```



```
69)     android:layout_height="wrap_content"
70)     android:textSize="12sp"
71)     android:layout_marginLeft="16dp"
72)     android:layout_marginRight="16dp"
73)     android:layout_marginBottom="8dp"
74)     android:text="@string/specification" />
75)
76)     <TableLayout
77)         android:layout_width="match_parent"
78)         android:layout_height="wrap_content"
79)         android:layout_marginLeft="16dp"
80)         android:layout_marginRight="16dp"
81)         android:layout_marginBottom="16dp">
82)
83)         <TableRow
84)             android:layout_width="match_parent"
85)             android:layout_height="wrap_content"
86)             android:layout_marginBottom="8dp">
87)
88)             <TextView
89)                 android:layout_width="wrap_content"
90)                 android:layout_height="wrap_content"
91)                 android:textSize="14sp"
92)                 android:layout_marginEnd="16dp"
93)                 android:text="@string/display" />
94)
95)             <TextView
96)                 android:layout_width="match_parent"
97)                 android:layout_height="wrap_content"
98)                 android:textSize="14sp"
99)                 android:layout_weight="1"
100)                android:text="@string/content_specs_display" />
101)            </TableRow>
102)
103)            <TableRow
104)                android:layout_width="match_parent"
105)                android:layout_height="wrap_content"
106)                android:layout_marginBottom="8dp">
107)
108)                <TextView
109)                    android:layout_width="wrap_content"
110)                    android:layout_height="wrap_content"
111)                    android:layout_marginEnd="16dp"
112)                    android:textSize="14sp"
113)                    android:text="@string/size" />
114)
115)                <TextView
116)                    android:layout_width="match_parent"
117)                    android:layout_height="wrap_content"
118)                    android:layout_weight="1"
119)                    android:textSize="14sp"
120)                    android:text="@string/content_specs_size" />
```



```

121)    </TableRow>
122)
123)    <TableRow
124)        android:layout_width="match_parent"
125)        android:layout_height="wrap_content"
126)        android:layout_marginBottom="8dp">
127)
128)        <TextView
129)            android:layout_width="wrap_content"
130)            android:layout_height="wrap_content"
131)            android:layout_marginEnd="16dp"
132)            android:textSize="14sp"
133)            android:text="@string/battery" />
134)
135)        <TextView
136)            android:layout_width="match_parent"
137)            android:layout_height="wrap_content"
138)            android:layout_weight="1"
139)            android:textSize="14sp"
140)            android:text="@string/content_specs_battery" />
141)    </TableRow>
142) </TableLayout>
143)
144) <TextView
145)     android:layout_width="match_parent"
146)     android:layout_height="wrap_content"
147)     android:textSize="12sp"
148)     android:layout_marginLeft="16dp"
149)     android:layout_marginRight="16dp"
150)     android:layout_marginBottom="8dp"
151)     android:text="@string/seller" />
152)
153) <RelativeLayout
154)     android:layout_width="match_parent"
155)     android:layout_height="wrap_content"
156)     android:layout_marginLeft="16dp"
157)     android:layout_marginRight="16dp"
158)     android:layout_marginBottom="16dp">
159)
160)    <de.hdodenhof.circleimageview.CircleImageView
161)        android:id="@+id/profile_image"
162)        android:layout_width="56dp"
163)        android:layout_height="56dp"
164)        android:layout_centerVertical="true"
165)        android:layout_marginEnd="16dp"
166)        android:src="@drawable/photo_2" />
167)
168)    <TextView
169)        android:layout_width="match_parent"
170)        android:layout_height="wrap_content"
171)        android:layout_centerVertical="true"
172)        android:layout_toEndOf="@+id/profile_image"

```



```

173)      android:text="@string/my_name" />
174)    </RelativeLayout>
175)
176)    <Button
177)      android:text="@string/buy"
178)      android:layout_width="match_parent"
179)      android:layout_height="wrap_content"
180)      android:layout_marginLeft="16dp"
181)      android:layout_marginRight="16dp"
182)      android:layout_marginBottom="16dp" />
183)    </LinearLayout>
      </ScrollView>

```

12) Terakhir, pada **MainActivity** tambahkan beberapa baris kode berikut:

Java:

```

1. if (getSupportActionBar() != null) {
2.     getSupportActionBar().setTitle("Google Pixel");
3. }

```

Kotlin:

```
supportActionBar?.title = "Google Pixel"
```

Sehingga kode yang ada di **MainActivity** menjadi seperti berikut:

Java:

```

1. public class MainActivity extends AppCompatActivity
{
2.
3.     @Override
4.     protected void onCreate(Bundle
5.         savedInstanceState) {
6.         super.onCreate(savedInstanceState);
7.         setContentView(R.layout.activity_main);
8.
9.         if (getSupportActionBar() != null) {
10.             getSupportActionBar().setTitle("Google
11.                 Pixel");
12.         }
13.     }
}

```

Kotlin:

```

1. class MainActivity : AppCompatActivity() {
2.
3.     override fun onCreate(savedInstanceState:
4.         Bundle?) {
5.         super.onCreate(savedInstanceState)
6.         setContentView(R.layout.activity_main)
}

```

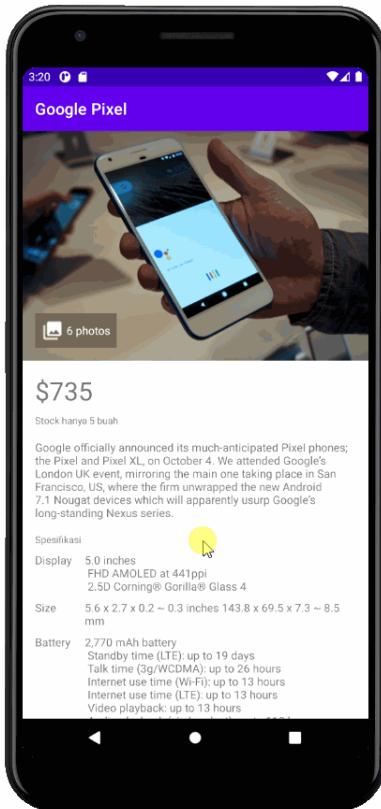


```

6.
7.         supportActionBar?.title = "Google Pixel"
8.     }
9. }

```

Maka, hasil output akan tampak seperti di bawah ini:



Gambar 35 Tampilan Akhir Saat Dijalankan Aplikasi

4. Style & Theme

Tujuan

Menerapkan style dan theme pada contoh aplikasi yang telah dibuat sebelumnya pada modul Views dan Viewgroup. Beberapa poin yang didapatkan pada materi ini adalah:

1. Bagaimana memanfaatkan style pada aplikasi?
2. Bagaimana memanfaatkan theme pada aplikasi?

Codelab Style dan Theme

- 1) Buka kembali proyek sebelumnya (**MyViewAndViews**)
- 2) Buka berkas **colors.xml** di **res → values → colors.xml**. Ubah setiap nilai yang ada saat ini dengan nilai-nilai berikut :



Android Studio 4.1 keatas

```
1. <resources>
2. <color name="gray">#607D8B</color>
3. <color name="gray_light">#B0BEC5</color>
4. <color name="gray_dark">#455A64</color>
5. <color name="orange">#FF5722</color>
6. <color name="orange_light">#FFAB91</color>
7. <color name="orange_dark">#E64A19</color>
8. <color name="black">#FF000000</color>
9. <color name="white">#FFFFFF</color>
10. <color name="colorSubtitle">#757575</color>
11. </resources>
```

Android Studio 4.0 kebawah

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <resources>
3.   <color name="colorPrimary">#607D8B</color>
4.   <color name="colorPrimaryDark">#455A64</color>
5.   <color name="colorAccent">#FF5722</color>
6.   <color name="colorSubtitle">#757575</color>
7. </resources>
```

- 3) Selanjutnya, buka berkas style/theme di res → values dan tambahkan beberapa *style* seperti di bawah ini:

Android Studio 4.1 keatas

Pada file themes.xml:

```
1. <resources
   xmlns:tools="http://schemas.android.com/tools">
2. <!-- Base application theme. -->
3. <style name="Theme.MyViewAndViews"
   parent="Theme.MaterialComponents.DayNight.DarkActionBar">
4. <!-- Primary brand color. -->
5. <item name="colorPrimary">@color/gray</item>
6. <item
   name="colorPrimaryVariant">@color/gray_dark</item>
7. <item name="colorOnPrimary">@color/white</item>
8. <!-- Secondary brand color. -->
9. <item
   name="colorSecondary">@color/orange_light</item>
10. <item
    name="colorSecondaryVariant">@color/orange_dark</item>
11. <item name="colorOnSecondary">@color/black</item>
12.
13. <!-- Status bar color. -->
14. <item name="android:statusBarColor"
   tools:targetApi="l">?attr/colorPrimaryVariant</item>
15. <!-- Customize your theme here. -->
16. </style>
```



```

17.
18. <style name="TextContent">
19.     <item
20.         name="android:layout_width">wrap_content</item>
21.     <item
22.         name="android:layout_height">wrap_content</item>
23.     <item
24.         name="android:layout_marginLeft">16dp</item>
25.     <item
26.         name="android:layout_marginRight">16dp</item>
27.     </style>
28.
29.     <style name="TextContent.Small">
30.         <item
31.             name="android:textAppearance">@style/TextAppearance.M
32.             aterialComponents.Subtitle1</item>
33.         </style>
34.     <style name="TextContent.Small.White">
35.         <item
36.             name="android:textColor">@android:color/white</item>
37.         </style>
38.     <style name="TextContent.Subtitle">
39.         <item
40.             name="android:textColor">@color/colorSubtitle</item>
41.         </style>
42.     <style name="TextSpec">
43.         <item
44.             name="android:layout_width">wrap_content</item>
45.         <item
46.             name="android:layout_height">wrap_content</item>
47.         <item
48.             name="android:layout_marginRight">16dp</item>
49.         <item
50.             name="android:textColor">@color/colorSubtitle</item>
51.         </style>
52.     <style name="TextSpec.Field">
53.         <item
54.             name="android:layout_width">wrap_content</item>
55.         <item
56.             name="android:layout_height">wrap_content</item>
57.         <item
58.             name="android:layout_weight">1</item>
59.         </style>

```



55.
56.</resources>
Android Studio 4.0 kebawah

Pada file styles.xml

```
1. <resources>
2.
3.     <!-- Base application theme. -->
4.     ...
5.     <style name="AppTheme"
6.         parent="Theme.AppCompat.Light.DarkActionBar">
7.         <!-- Customize your theme here. -->
8.         <item
9.             name="colorPrimary">@color/colorPrimary</item>
10.        <item
11.            name="colorPrimaryDark">@color/colorPrimaryDark</item>
12.        >
13.        <item
14.            name="colorAccent">@color/colorAccent</item>
15.        </style>
16.
17.        <style name="TextContent"
18.            parent="@style/TextAppearance.AppCompat">
19.            <item
20.                name="android:layout_width">wrap_content</item>
21.                <item
22.                    name="android:layout_height">wrap_content</item>
23.                    <item
24.                        name="android:layout_marginLeft">16dp</item>
25.                        <item
26.                            name="android:layout_marginRight">16dp</item>
27.                        </style>
28.
29.        <style name="TextContent.Small">
30.            <item
31.                name="android:textAppearance">@style/TextAppearance.AppCompat.Small</item>
32.            </style>
33.
34.        <style name="TextContent.Small.White">
35.            <item
36.                name="android:textColor">@android:color/white</item>
37.            </style>
38.
39.        <style name="TextContent.Subtitle">
40.            <item
41.                name="android:textColor">@color/colorSubtitle</item>
42.                    <item name="android:textSize">12sp</item>
43.            </style>
44.
45.        <style name="TextSpec">
```



```

33.      <item name="android:textSize">14sp</item>
34.    </style>
35.
36.    <style name="TextSpec.Field">
37.      <item
38.        name="android:layout_width">wrap_content</item>
39.      <item
40.        name="android:layout_height">wrap_content</item>
41.      <item
42.        name="android:layout_marginRight">16dp</item>
43.    <item
44.      name="android:textColor">@color/colorSubtitle</item>
45.    </style>
46.
47.    <style name="TextSpec.Value">
48.      <item
49.        name="android:layout_width">match_parent</item>
50.      <item
51.        name="android:layout_height">wrap_content</item>
52.      <item name="android:layout_weight">1</item>
53.    </style>
54.
55.  </resources>

```

- 4) Jika selesai, kita akan implementasikan beberapa style yang baru saja dibuat ke dalam berkas `activity_main.xml` dan kondisikan seperti ini:

```

1) <?xml version="1.0" encoding="utf-8"?>
2) <ScrollView
   xmlns:android="http://schemas.android.com/apk/res/android"
3)   android:id="@+id/activity_main"
4)   android:layout_width="match_parent"
5)   android:layout_height="match_parent"
6)   android:background="@android:color/white">
7)
8)   <LinearLayout
9)     android:layout_width="match_parent"
10)    android:layout_height="wrap_content"
11)    android:orientation="vertical">
12)
13)    <FrameLayout
14)      android:layout_width="match_parent"
15)      android:layout_height="wrap_content"
16)    <ImageView
17)      android:layout_width="match_par
18)      ent"

```



```

19)           android:layout_height="wrap_content"
20)           android:adjustViewBounds="true"
21)           android:scaleType="fitXY"
22)           android:src="@drawable/pixel_globe" />
23)
24)           <TextView
25)               style="@style/TextContent.SmallWhite"
26)               android:layout_gravity="bottom"
27)               android:layout_marginStart="16dp"
28)               android:layout_marginBottom="16dp"
29)               android:background="#4D000000"
30)               android:drawableStart="@drawable/ic_collections_white_18dp"
31)               android:drawablePadding="4dp"
32)               android:gravity="center_vertical"
33)               android:padding="8dp"
34)               android:text="@string/dummy_photos" />
35)           </FrameLayout>
36)
37)           <TextView
38)               style="@style/TextContent"
39)               android:layout_marginTop="16dp"
40)               android:layout_marginBottom="8dp"
41)               android:text="@string/dummy_value"
42)               android:textSize="32sp" />
43)
44)           <TextView
45)               style="@style/TextContent.Subtitle"
46)               android:layout_marginBottom="16dp"
47)               android:text="@string/stock" />
48)
49)           <TextView
50)               style="@style/TextContent"
51)               android:layout_marginBottom="16dp"
52)               android:lineSpacingMultiplier="1"
53)               android:text="@string/content_text" />
54)
55)           <TextView
56)               style="@style/TextContent.Subtitle"
57)               android:layout_marginBottom="8dp"
58)               android:text="@string/specification" />
59)
60)           <TableLayout
61)               android:layout_width="match_parent"
62)               android:layout_height="wrap_content"

```



```

63)             android:layout_marginLeft="16dp"
64)             android:layout_marginRight="16dp"
65)             android:layout_marginBottom="16dp">
66)
67)         <TableRow
68)             android:layout_width="match_par
ent"
69)             android:layout_height="wrap_con
tent"
70)             android:layout_marginBottom="8d
p">
71)
72)         <TextView
73)             style="@style/TextSpec.Field"
74)             android:text="@string/displ
ay" />
75)
76)         <TextView
77)             style="@style/TextSpec.Value"
78)             android:text="@string/conte
nt_specs_display" />
79)     </TableRow>
80)
81)     <TableRow
82)             android:layout_width="match_par
ent"
83)             android:layout_height="wrap_con
tent"
84)             android:layout_marginBottom="8d
p">
85)
86)         <TextView
87)             style="@style/TextSpec.Field"
88)             android:text="@string/size"
     />
89)
90)         <TextView
91)             style="@style/TextSpec.Value"
92)             android:text="@string/conte
nt_specs_size" />
93)     </TableRow>
94)
95)     <TableRow
96)             android:layout_width="match_par
ent"
97)             android:layout_height="wrap_con
tent"
98)             android:layout_marginBottom="8d
p">
99)
100)    <TextView
101)        style="@style/TextSpec.Field"

```



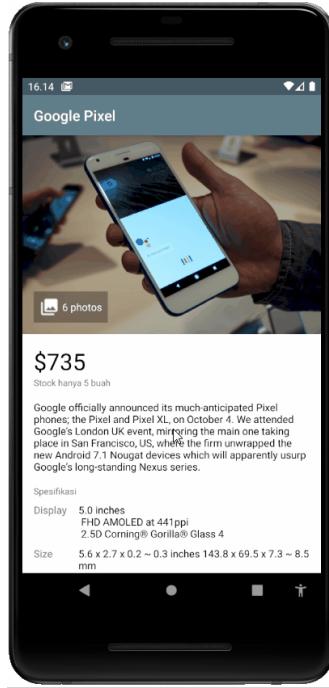
```

102)                               android:text="@string/batte
    ry" />
103)
104)             <TextView
105)                 style="@style/TextSpec.Value"
106)                 android:text="@string/conte
    nt_specs_battery" />
107)             </TableRow>
108)         </TableLayout>
109)
110)     <TextView
111)         style="@style/TextContent.Subtitle"
112)         android:layout_marginBottom="8dp"
113)         android:text="@string/seller" />
114)
115)     <RelativeLayout
116)         android:layout_width="match_parent"
117)         android:layout_height="wrap_content"
118)         android:layout_marginLeft="16dp"
119)         android:layout_marginRight="16dp"
120)         android:layout_marginBottom="16dp">
121)
122)         <de.hdodenhof.circleimageview.Circl
    eImageView
123)             android:id="@+id/profile_image"
124)             android:layout_width="56dp"
125)             android:layout_height="56dp"
126)             android:layout_centerVertical="
    true"
127)             android:layout_marginRight="16dp"
128)             android:src="@drawable/photo_2"
    />
129)
130)         <TextView
131)             style="@style/TextContent"
132)             android:layout_toEndOf="@+id/pr
    ofile_image"
133)             android:text="@string/my_name"
    />
134)         </RelativeLayout>
135)
136)     <Button
137)         android:layout_width="match_parent"
138)         android:layout_height="wrap_content"
139)         android:layout_marginLeft="16dp"
140)         android:layout_marginRight="16dp"
141)         android:layout_marginBottom="16dp"
142)         android:text="@string/buy" />
143)     </LinearLayout>
144) </ScrollView>

```



5) Jika selesai coba jalankan aplikasi Anda. Seharusnya hasilnya seperti ini



Gambar 36 Tampilan Akhir 1 Saat Dijalan Setelah Diberikan Style & Theme

Perhatikan bagaimana penulisan di setiap obyek **TextView** menjadi lebih sederhana:

```
1. <TextView  
2.     style="@style/TextContent.Subtitle"  
3.     android:layout_marginBottom="8dp"  
4.     android:text="@string/seller" />
```

6) Masih ada yang perlu kita perbaiki. Tombol masih berada dalam kondisi *default*. Yuk coba ubah agar lebih menyatu dengan tema aplikasi. Buka kembali themes.xml/styles.xml dan tambahkan kode berikut di atas tag </resources>

Android Studio 4.1 keatas

```
1. <style name="ButtonGeneral">  
2. <item  
    name="android:backgroundTint">@color/orange</item>  
3. <item name="android:layout_width">match_parent</item>  
4. <item name="android:layout_height">wrap_content</item>  
5. <item name="android:layout_marginRight">16dp</item>  
6. <item name="android:layout_marginLeft">16dp</item>  
7. <item name="android:layout_marginBottom">16dp</item>  
8. </style>
```



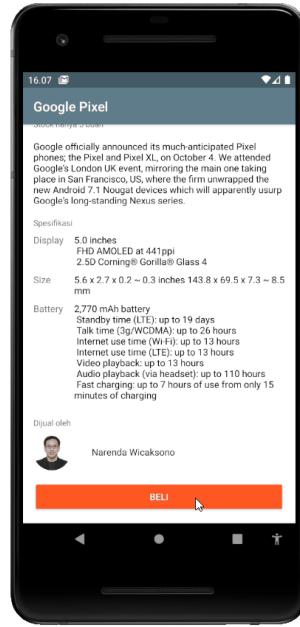
Android Studio 4.0 kebawah

```
1. <style name="ButtonGeneral"
       parent="@style/Widget.AppCompat.Button.Colored">
2.                                     <item
3.             name="android:layout_width">match_parent</item>
4.                                     <item
5.             name="android:layout_height">wrap_content</item>
6.             <item name="android:layout_marginRight">16dp</item>
7.             <item name="android:layout_marginLeft">16dp</item>
8.             <item
9.               name="android:layout_marginBottom">16dp</item>
10.      </style>
```

- 7) Pada `activity_main.xml` kondisikan bagian obyek Button menjadi seperti ini:

```
1) <Button
2)     android:text="@string/buy"
3)     style="@style/ButtonGeneral"/>
```

- 8) Jalankan aplikasi Anda. Seharusnya tampilannya akan menjadi seperti ini:



Gambar 37 Tampilan Akhir 2 Saat Dijalan Setelah Diberikan Style & Theme

Sekarang coba Anda klik dan tahan tombolnya. Bila perangkat Android Anda berada di bawah Lollipop (API level 21), seharusnya tombol



akan berwarna lebih gelap. Sedangkan jika di atas Lollipop, akan ada efek air (*ripple*) ketika tombol Anda tekan.

Sistem akan secara otomatis mengambil nilai `accentColor` sebagai nilai warna indikator aktif untuk `Button` dan nilai pada atribut `colorButtonNormal` sebagai nilai gelap ketika obyek `Button` ditekan.

5. Tugas Praktikum

- a. Buatlah View & View Group Profile kalian masing-masing dengan ketentuan:

Gambar bagian atas : Foto keluarga kalian

Gambar bagian bawah: Foto kalian

- b. Lanjutan dari file View & View Group, gunakan Style & Theme dari aplikasi tersebut sehingga akan tampak bagus dan tombol "button" bisa di klik dengan berwarna



Praktikum 4

Merancang Program Application

1. Sub Pokok Pembahasan

- a. Merancang Program Application

2. Capaian Pembelajaran

- a. Mahasiswa mampu menanamkan sikap jujur, bertanggung jawab, disiplin, saling menghargai dan berinovasi dengan tetap menjunjung tinggi nilai-nilai agama dan ketuhanan.
- b. Mahasiswa mampu menunjukkan sikap nasionalisme, mampu bekerja dalam tim sesuai kaidah keilmuan pada proyek pengembangan aplikasi mobile programming.
- c. Mahasiswa mampu berpartisipasi aktif, bertanggung jawab, dan memiliki motivasi mengembangkan diri dalam mendapatkan solusi permasalahan melalui aplikasi mobile programming
- d. Mahasiswa mampu menguasai, merancang dan mengembangkan bahasa pemrograman berbasis mobile, serta mampu menganalisis solusi dalam model bahasa pemrograman mobile.
- e. Mahasiswa mampu mengimplementasi pengetahuan dan teknologi menggunakan pendekatan *mobile programming*.

3. Merancang Program Application

Modul ini lanjutan dari membuat Layout Application sehingga bisa di buka kembali program tersebut untuk saat ini akan di buat codingan di Program tersebut

Codelab Kode Logika

- 1) Selanjutnya setelah selesai, lanjutkan dengan membuka berkas **MainActivity** dan lanjutkan ngoding baris-baris di bawah ini. Tambahkan beberapa variabel yang akan digunakan untuk menampung View.

Java

```
1. private EditText edtWidth;
2. private EditText edtHeight;
3. private EditText edtLength;
4. private Button btnCalculate;
5. private TextView tvResult;
```



Catatan:

Perhatikan bagaimana cara menuliskan kodenya. Biasakan menekan **Enter** ketika mengetik supaya komponen di-import secara otomatis, kecuali nama variabel yang Anda tentukan sendiri seperti **edtWidth** harus diketik satu per satu.

```
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Gambar 38 Cara Menginput **edtWidth** di Android Studio dengan Java

Kotlin

```
1. private lateinit var edtWidth: EditText
2. private lateinit var edtHeight: EditText
3. private lateinit var edtLength: EditText
4. private lateinit var btnCalculate: Button
5. private lateinit var tvResult: TextView
```

Catatan:

Perhatikan bagaimana cara menuliskan kodenya. Biasakan menekan **Enter** ketika mengetik supaya komponen di-*import* secara otomatis, kecuali nama variabel yang Anda tentukan sendiri seperti **edtWidth** harus diketik satu per satu.



```

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}

```

Gambar 39 Cara Menginput edtWidth di Android Studio dengan Kotlin

- 2) Kemudian inisiasi variabel yang telah kita buat dengan menambahkan kode berikut di dalam metode **onCreate**.

Java

```

1. @Override
2. protected void onCreate(Bundle savedInstanceState) {
3.     super.onCreate(savedInstanceState);
4.     setContentView(R.layout.activity_main);
5.
6.     edtWidth = findViewById(R.id.edt_width);
7.     edtHeight = findViewById(R.id.edt_height);
8.     edtLength = findViewById(R.id.edt_length);
9.     btnCalculate = findViewById(R.id.btn_calculate);
10.    tvResult = findViewById(R.id.tv_result);
11.
12.    btnCalculate.setOnClickListener(this);
13.
14. }

```

Kotlin

```

1. override fun onCreate(savedInstanceState: Bundle?) {
2.     super.onCreate(savedInstanceState)
3.     setContentView(R.layout.activity_main)
4.
5.     edtWidth = findViewById(R.id.edt_width)
6.     edtHeight = findViewById(R.id.edt_height)
7.     edtLength = findViewById(R.id.edt_length)
8.     btnCalculate = findViewById(R.id.btn_calculate)
9.     tvResult = findViewById(R.id.tv_result)
10.
11.    btnCalculate.setOnClickListener(this)
12. }

```



3) Akan muncul baris merah pada kata **this**. Hal ini karena ketika kita menggunakan **setOnClickListener(this)**, maka kita perlu menambahkan interface **View.OnClickListener** di kelas **MainActivity**. Silakan **klik di atas baris merah** tersebut, kemudian **tekan tombol Alt + Enter (option + return pada Mac)** atau **menekan lampu merah** yang muncul lalu pilih aksi berikut untuk implement interface.

Java



Gambar 40 Tampilan di MainActivity saat Terdapat setOnClickListener(this) dengan Java

Kotlin



Gambar 41 Tampilan di MainActivity saat Terdapat setOnClickListener(this) dengan Kotlin

Maka secara otomatis akan ada penambahan kode pada kelas **MainActivity** seperti berikut ini:

Java

```
1. public class MainActivity extends AppCompatActivity  
    implements View.OnClickListener {  
2.  
3.     ...  
4.  
5. }
```

Kotlin

```
1. class MainActivity : AppCompatActivity(),  
    View.OnClickListener {  
2.  
3.     ...  
4.  
5. }
```

4) Jika terdapat baris merah seperti ini:

class MainActivity

Gambar 42 Tampilan Ada Baris Merah di Codingan



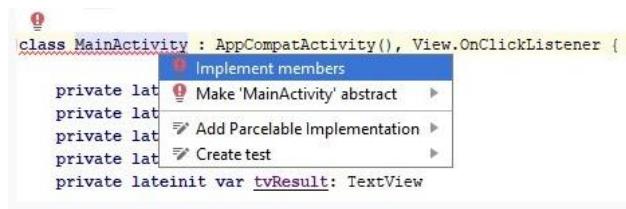
Jangan khawatir! Silakan klik di atas baris merah tersebut, kemudian tekan tombol Alt + Enter (option + return pada Mac) atau menekan lampu merah yang muncul lalu pilih **implement methods (Java)** atau **implement members (Kotlin)**.

Java



Gambar 43 Tampilan di MainActivity saat Terdapat AppCompatActivity dengan Java

Kotlin



Gambar 44 Tampilan di MainActivity saat Terdapat AppCompatActivity dengan Kotlin

Maka secara otomatis akan ada penambahan metode **onClick** di kelas **MainActivity**. Setelah itu, tambahkan kode berikut ke dalam metode **onClick**:

Java

```
1. @Override
2. protected void onCreate(Bundle savedInstanceState) {
3.     super.onCreate(savedInstanceState);
4.     setContentView(R.layout.activity_main);
5.
6.     ...
7.
8. }
9.
10. @Override
11. public void onClick(View v) {
12.     if (v.getId() == R.id.btn_calculate) {
13.         String inputLength =
14.             edtLength.getText().toString().trim();
15.         String inputWidth =
16.             edtWidth.getText().toString().trim();
17.         String inputHeight =
18.             edtHeight.getText().toString().trim();
19.     }
20. }
```



```

17.         double volume = Double.valueOf(inputLength) *
    Double.valueOf(inputWidth) *
    Double.valueOf(inputHeight);
18.         tvResult.setText(String.valueOf(volume));
19.     }
20. }
```

Kotlin

```

1. override fun onCreate(savedInstanceState: Bundle?) {
2.     super.onCreate(savedInstanceState)
3.     setContentView(R.layout.activity_main)
4.
5.     ...
6. }
7.
8. override fun onClick(v: View?) {
9.     if (v?.id == R.id.btn_calculate) {
10.         val inputLength =
    edtLength.text.toString().trim()
11.         val inputWidth =
    edtWidth.text.toString().trim()
12.         val inputHeight =
    edtHeight.text.toString().trim()
13.
14.         val volume = inputLength.toDouble() *
    inputWidth.toDouble() * inputHeight.toDouble()
15.         tvResult.text = volume.toString()
16.     }
17. }
```

5) Akhirnya kelas **MainActivity** akan memiliki kode seperti berikut ini:

Java

```

1. public class MainActivity extends AppCompatActivity
    implements View.OnClickListener {
2.     private EditText edtWidth, edtHeight, edtLength;
3.     private Button btnCalculate;
4.     private TextView tvResult;
5.
6.     @Override
7.     protected void onCreate(Bundle savedInstanceState) {
8.         super.onCreate(savedInstanceState);
9.         setContentView(R.layout.activity_main);
10.
11.         edtWidth = findViewById(R.id.edt_width);
12.         edtHeight = findViewById(R.id.edt_height);
13.         edtLength = findViewById(R.id.edt_length);
14.         btnCalculate =
    findViewById(R.id.btn_calculate);
15.         tvResult = findViewById(R.id.tv_result);
```



```

16.
17.        btnCalculate.setOnClickListener(this);
18.    }
19.
20.    @Override
21.    public void onClick(View v) {
22.        if (v.getId() == R.id.btn_calculate) {
23.            String inputLength =
24.                edtLength.getText().toString().trim();
25.            String inputWidth =
26.                edtWidth.getText().toString().trim();
27.            String inputHeight =
28.                edtHeight.getText().toString().trim();
29.            double volume = Double.valueOf(inputLength)
30.                * Double.valueOf(inputWidth)
31.                * Double.valueOf(inputHeight);
32.            tvResult.setText(String.valueOf(volume));
33.        }
34.    }

```

Kotlin

```

1. class MainActivity : AppCompatActivity(),
2.     View.OnClickListener {
3.     private lateinit var edtWidth: EditText
4.     private lateinit var edtHeight: EditText
5.     private lateinit var edtLength: EditText
6.     private lateinit var btnCalculate: Button
7.     private lateinit var tvResult: TextView
8.
9.     override fun onCreate(savedInstanceState: Bundle?) {
10.         super.onCreate(savedInstanceState)
11.         setContentView(R.layout.activity_main)
12.         edtWidth = findViewById(R.id.edt_width)
13.         edtHeight = findViewById(R.id.edt_height)
14.         edtLength = findViewById(R.id.edt_length)
15.         btnCalculate =
16.             findViewById(R.id.btn_calculate)
17.         tvResult = findViewById(R.id.tv_result)
18.         btnCalculate.setOnClickListener(this)
19.     }
20.
21.     override fun onClick(v: View?) {
22.         if (v?.id == R.id.btn_calculate) {
23.             val inputLength =
24.                 edtLength.text.toString().trim()
25.         }
26.     }

```



```

24.         val inputWidth =
    edtWidth.text.toString().trim()
25.         val inputHeight =
    edtHeight.text.toString().trim()
26.
27.         val volume = inputLength.toDouble() *
    inputWidth.toDouble() * inputHeight.toDouble()
28.             tvResult.text = volume.toString()
29.         }
30.     }
31. }
```

- 6) Jalankan aplikasi tersebut



Gambar 45 Tampilan Sementara Saat Aplikasi dijalankan

Silakan masukkan nilai panjang, lebar, dan tinggi kemudian tekan tombol **Hitung** dan hasilnya akan ditampilkan di objek textview **tvResult**. Namun masih ada sedikit masalah di sini, yaitu Anda tetap melakukan proses perhitungan walaupun salah satu nilainya kosong. Hal ini akan menyebabkan aplikasi *force close* karena perhitungan tidak dapat diproses. Maka untuk mengatasinya Anda akan menggunakan percabangan untuk mengecek apakah masing-masing EditText kosong atau tidak.

- 7) Silakan buka kembali kelas **MainActivity**. Tambahkan kode berikut ke dalam metode **onClick** sebelum melakukan perhitungan.



Java

```
1. @Override
2. public void onClick(View v) {
3.     if (v.getId() == R.id.btn_calculate) {
4.         String inputLength =
5.             edtLength.getText().toString().trim();
6.         String inputWidth =
7.             edtWidth.getText().toString().trim();
8.         String inputHeight =
9.             edtHeight.getText().toString().trim();
10.        boolean isEmptyFields = false;
11.
12.        if (TextUtils.isEmpty(inputLength)) {
13.            isEmptyFields = true;
14.            edtLength.setError("Field ini tidak boleh
15. kosong");
16.        }
17.
18.        if (TextUtils.isEmpty(inputWidth)) {
19.            isEmptyFields = true;
20.            edtWidth.setError("Field ini tidak boleh
21. kosong");
22.        }
23.
24.        if (TextUtils.isEmpty(inputHeight)) {
25.            isEmptyFields = true;
26.            edtHeight.setError("Field ini tidak boleh
27. kosong");
28.        }
29.    }
30. }
```

Kotlin

```
1. override fun onClick(v: View?) {
2.     if (v?.id == R.id.btn_calculate) {
3.         val inputLength =
4.             edtLength.text.toString().trim()
5.         val inputWidth =
6.             edtWidth.text.toString().trim()
7.         val inputHeight =
8.             edtHeight.text.toString().trim()
9.         var isEmptyFields = false
```



```

8.
9. if (inputLength.isEmpty()) {
10.    isEmptyFields = true
11.    edtLength.error = "Field ini tidak boleh kosong"
12. }
13. if (inputWidth.isEmpty()) {
14.    isEmptyFields = true
15.    edtWidth.error = "Field ini tidak boleh kosong"
16. }
17. if (inputHeight.isEmpty()) {
18.    isEmptyFields = true
19.    edtHeight.error = "Field ini tidak boleh kosong"
20. }
21.
22. if (!isEmptyFields) {
23.    val volume = inputLength.toDouble() *
    inputWidth.toDouble() * inputHeight.toDouble()
24.    tvResult.text = volume.toString()
25. }
26. }
27. }

```

8) Akhirnya kelas **MainActivity** akan memiliki kode seperti berikut ini:

Java

```

1. public class MainActivity extends AppCompatActivity
   implements View.OnClickListener {
2. private EditText edtWidth, edtHeight, edtLength;
3. private Button btnCalculate;
4. private TextView tvResult;
5.
6. @Override
7. protected void onCreate(Bundle savedInstanceState) {
8. super.onCreate(savedInstanceState);
9. setContentView(R.layout.activity_main);
10.
11. edtWidth = findViewById(R.id.edt_width);
12. edtHeight = findViewById(R.id.edt_height);
13. edtLength = findViewById(R.id.edt_length);
14. btnCalculate = findViewById(R.id.btn_calculate);
15. tvResult = findViewById(R.id.tv_result);
16.
17. btnCalculate.setOnClickListener(this);
18. }
19.
20. @Override
21. public void onClick(View v) {
22. if (v.getId() == R.id.btn_calculate) {
23. String inputLength =
    edtLength.getText().toString().trim();

```



```

24.     String inputWidth =
    edtWidth.getText().toString().trim();
25.     String inputHeight =
    edtHeight.getText().toString().trim();
26.
27.     boolean isEmptyFields = false;
28.
29.     if (TextUtils.isEmpty(inputLength)) {
30.         isEmptyFields = true;
31.         edtLength.setError("Field ini tidak boleh kosong");
32.     }
33.
34.     if (TextUtils.isEmpty(inputWidth)) {
35.         isEmptyFields = true;
36.         edtWidth.setError("Field ini tidak boleh kosong");
37.     }
38.
39.     if (TextUtils.isEmpty(inputHeight)) {
40.         isEmptyFields = true;
41.         edtHeight.setError("Field ini tidak boleh kosong");
42.     }
43.
44.     if (!isEmptyFields) {
45.         double volume = Double.valueOf(inputLength) *
    Double.valueOf(inputWidth) *
    Double.valueOf(inputHeight);
46.         tvResult.setText(String.valueOf(volume));
47.     }
48. }
49. }
50. }

```

Kotlin

```

1. class MainActivity : AppCompatActivity(),
    View.OnClickListener {
2.     private lateinit var edtWidth: EditText
3.     private lateinit var edtHeight: EditText
4.     private lateinit var edtLength: EditText
5.     private lateinit var btnCalculate: Button
6.     private lateinit var tvResult: TextView
7.
8.     override fun onCreate(savedInstanceState: Bundle?) {
9.         super.onCreate(savedInstanceState)
10.        setContentView(R.layout.activity_main)
11.
12.        edtWidth = findViewById(R.id.edt_width)
13.        edtHeight = findViewById(R.id.edt_height)
14.        edtLength = findViewById(R.id.edt_length)
15.        btnCalculate =
    findViewById(R.id.btn_calculate)

```



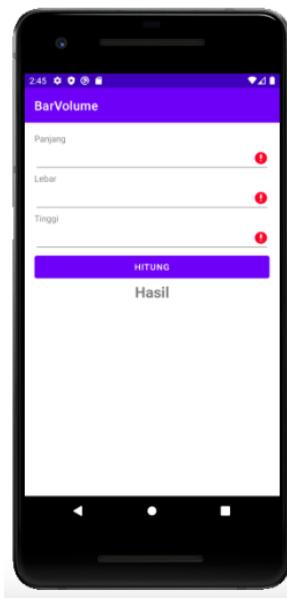
```

16.         tvResult = findViewById(R.id.tv_result)
17.
18.         btnCalculate.setOnClickListener(this)
19.     }
20.
21.     override fun onClick(v: View?) {
22.         if (v?.id == R.id.btn_calculate) {
23.             val inputLength =
24.                 edtLength.text.toString().trim()
25.             val inputWidth =
26.                 edtWidth.text.toString().trim()
27.             val inputHeight =
28.                 edtHeight.text.toString().trim()
29.             var isEmptyFields = false
30.
31.             if (inputLength.isEmpty()) {
32.                 isEmptyFields = true
33.                 edtLength.error = "Field ini tidak boleh kosong"
34.             }
35.             if (inputWidth.isEmpty()) {
36.                 isEmptyFields = true
37.                 edtWidth.error = "Field ini tidak boleh kosong"
38.             }
39.             if (inputHeight.isEmpty()) {
40.                 isEmptyFields = true
41.                 edtHeight.error = "Field ini tidak boleh kosong"
42.             }
43.             if (!isEmptyFields) {
44.                 val volume = inputLength.toDouble() *
45.                     inputWidth.toDouble() * inputHeight.toDouble()
46.                 tvResult.text = volume.toString()
47.             }
48.         }

```

- 9) Jalankan aplikasi tersebut dan cobalah langsung menekan tombol **HITUNG** tanpa mengisi EditText, maka aplikasi Anda tidak akan force close dan akan muncul peringatan bahwa "**Field ini tidak boleh kosong**".





Gambar 46 Tampilan Akhir Saat Aplikasi dijalankan

- 10) Apakah kita sudah selesai? Belum! Masih ada yang kurang. Ketika nilai volume sudah dihitung dan kemudian terjadi pergantian orientasi (portrait-landscape) pada peranti, maka hasil perhitungan tadi akan hilang.



Gambar 47 Tampilan Akhir Saat Aplikasi dijalankan dengan Pergantian Orientasi

Untuk mengatasinya, tambahkan metode `onSaveInstanceState()` pada `MainActivity` dan sesuaikan seperti berikut:

Java

```
1. public class MainActivity extends AppCompatActivity  
    implements View.OnClickListener{  
2.     private EditText edtWidth;  
3.     private EditText edtHeight;  
4.     private EditText edtLength;  
5.     private Button btnCalculate;  
6.     private TextView tvResult;  
7. }
```



```

8.     private static final String STATE_RESULT =
    "state_result";
9.
10.    @Override
11.    protected void onCreate(Bundle savedInstanceState) {
12.        ...
13.    }
14.
15.    @Override
16.    protected void onSaveInstanceState(Bundle outState) {
17.        super.onSaveInstanceState(outState);
18.        outState.putString(STATE_RESULT,
            tvResult.getText().toString());
19.    }
20.
21.    ...
22. }
```

Kotlin

```

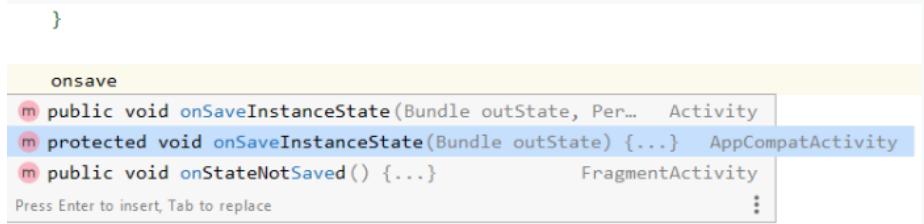
1. class MainActivity : AppCompatActivity(),
    View.OnClickListener {
2.
3.     private lateinit var edtWidth: EditText
4.     private lateinit var edtHeight: EditText
5.     private lateinit var edtLength: EditText
6.     private lateinit var btnCalculate: Button
7.     private lateinit var tvResult: TextView
8.
9.     companion object {
10.         private const val STATE_RESULT = "state_result"
11.     }
12.
13.     override fun onCreate(savedInstanceState: Bundle?) {
14.         ...
15.     }
16.
17.     override fun onSaveInstanceState(outState: Bundle) {
18.         super.onSaveInstanceState(outState)
19.         outState.putString(STATE_RESULT,
            tvResult.text.toString())
20.     }
21.
22.     ...
23. }
```

Catatan:

Karena `onSaveInstanceState` adalah class yang ada di superclass `AppCompatActivity`, Anda bisa membuat fungsi secara



otomatis dengan hanya mengetikkan huruf onSave... dan pilih yang hanya memiliki satu parameter.



Kemudian tambahkan juga beberapa baris berikut pada baris terakhir metode `onCreate`.

Java

```
1. @Override
2. protected void onCreate(Bundle savedInstanceState) {
3.     super.onCreate(savedInstanceState);
4.     setContentView(R.layout.activity_main);
5.
6.     ...
7.
8.     if (savedInstanceState != null) {
9.         String result =
10.            savedInstanceState.getString(STATE_RESULT);
11.         tvResult.setText(result);
12.     }
13. }
```

Kotlin

```
1. override fun onCreate(savedInstanceState: Bundle?) {
2.     super.onCreate(savedInstanceState)
3.     setContentView(R.layout.activity_main)
4.
5.     ...
6.
7.     if (savedInstanceState != null) {
8.         val result = savedInstanceState.getString(STATE_RESULT)
9.         tvResult.text = result
10.    }
11. }
```

- 11) Silakan jalankan kembali aplikasinya. Ulangi proses perhitungan seperti sebelumnya. Kemudian ganti orientasi peranti yang sebelumnya. Jika sudah benar maka hasil perhitungan tidak akan hilang.



4. Tugas Praktikum

Lanjutkan tugas Praktikum 2, yang sebelumnya hanya membuat layout saja, sekarang tambahkan program codingannya sehingga aplikasi sederhana tersebut bisa dijalankan



Praktikum 5

Intent

1. Sub Pokok Pembahasan

- a) Logika Dasar
- b) Eksplisit
- c) Implisit

2. Capaian Pembelajaran

- a. Mahasiswa mampu menanamkan sikap jujur, bertanggung jawab, disiplin, saling menghargai dan berinovasi dengan tetap menjunjung tinggi nilai-nilai agama dan ketuhanan.
- b. Mahasiswa mampu menunjukkan sikap nasionalisme, mampu bekerja dalam tim sesuai kaidah keilmuan pada projek pengembangan aplikasi mobile programming.
- c. Mahasiswa mampu berpartisipasi aktif, bertanggung jawab, dan memiliki motivasi mengembangkan diri dalam mendapatkan solusi permasalahan melalui aplikasi mobile programming
- d. Mahasiswa mampu menguasai, merancang dan mengembangkan bahasa pemrograman berbasis mobile, serta mampu menganalisis solusi dalam model bahasa pemrograman mobile.
- e. Mahasiswa mampu mengimplementasi pengetahuan dan teknologi menggunakan pendekatan *mobile programming*.

3. Intent

Tujuan

Codelab ini menitik beratkan pada implementasi Intent untuk melakukan perpindahan dari Activity ke Activity dengan atau tidak membawa data. Beberapa bagian dari code lab ini akan menjawab beberapa pertanyaan umum dalam pengembangan aplikasi Android sebagai berikut:

- 1) Bagaimana berpindah dari satu Activity ke Activity lain?
- 2) Bagaimana berpindah dari satu Activity ke Activity lain dengan membawa data?



3) Menjalankan komponen di aplikasi lain untuk keperluan membuka browser atau melakukan pemanggilan melalui aplikasi telepon bawaan?

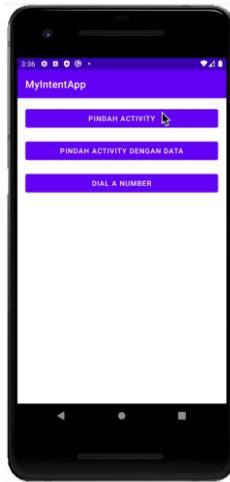
1) Logika Dasar

Berpindah dari satu Activity ke Activity lain dengan membawa data. Activity asal akan mengirimkan data melalui Intent dan Activity tujuan akan menerima data yang dikirimkan.

a) Buat Project baru di Android Studio dengan kriteria sebagai berikut:

- o Nama Project : **MyIntentApp**
- o Target & Minimum Target SDK : **Phone & Tablet; Api 21**
- o Tipe Activity : **Empty Activity**
- o Activity Name : **MainActivity**

b) Selanjutnya kita akan membangun **User Interface** menjadi seperti ini:



Gambar 48 Tampilan dari Pembuatan Intent

c) Kita akan memiliki **3 tombol** dengan fungsi yang berbeda-beda dan **1 buah TextView** untuk menampilkan data yang berasal dari Intent.

Kondisikan **activity_main.xml** menjadi seperti ini:

```
o  <?xml version="1.0" encoding="utf-8"?>
o  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
o    xmlns:tools="http://schemas.android.com/tools"
o    android:layout_width="match_parent"
o    android:layout_height="match_parent"
o    android:paddingBottom="@dimen/activity_vertical_margin"
o    android:paddingLeft="@dimen/activity_horizontal_margin"
o    android:paddingRight="@dimen/activity_horizontal_margin"
o    android:paddingTop="@dimen/activity_vertical_margin"
o    android:orientation="vertical"
```



```

○      tools:context="com.dicoding.associate.myintentapp.MainActivity">
○      <Button
○          android:id="@+id	btn_move_activity"
○          android:layout_width="match_parent"
○          android:layout_height="wrap_content"
○          android:text="Pindah Activity"
○          android:layout_marginBottom="@dimen/activity_vertical_margin"/>
○      </LinearLayout>

```

Jangan lupa untuk menambahkan file **dimens.xml** secara manual di dalam res values. Dan isikan file **dimens.xml** seperti berikut

```

○      <resources>
○          <!-- Default screen margins, per the Android Design guidelines. -->
○          <dimen name="activity_horizontal_margin">16dp</dimen>
○          <dimen name="activity_vertical_margin">16dp</dimen>
○      </resources>

```

- d) Untuk **MainActivity.java** tambahkan beberapa baris seperti ini.

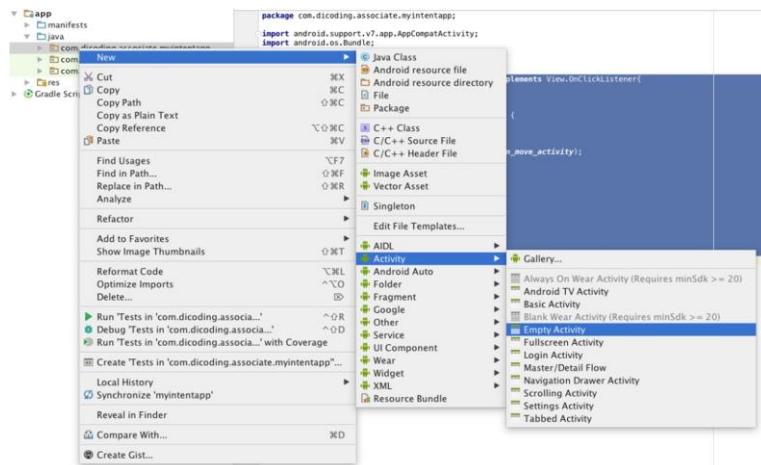
```

○      public class MainActivity extends AppCompatActivity implements View.OnClickListener{
○          private Button btnMoveActivity;
○          @Override
○          protected void onCreate(Bundle savedInstanceState) {
○              super.onCreate(savedInstanceState);
○              setContentView(R.layout.activity_main);
○
○              btnMoveActivity = (Button) findViewById(R.id.btn_move_activity);
○              btnMoveActivity.setOnClickListener(this);
○          }
○          @Override
○          public void onClick(View v) {
○              switch (v.getId()){
○                  case R.id.btn_move_activity:
○                      break;
○              }
○          }
○      }

```

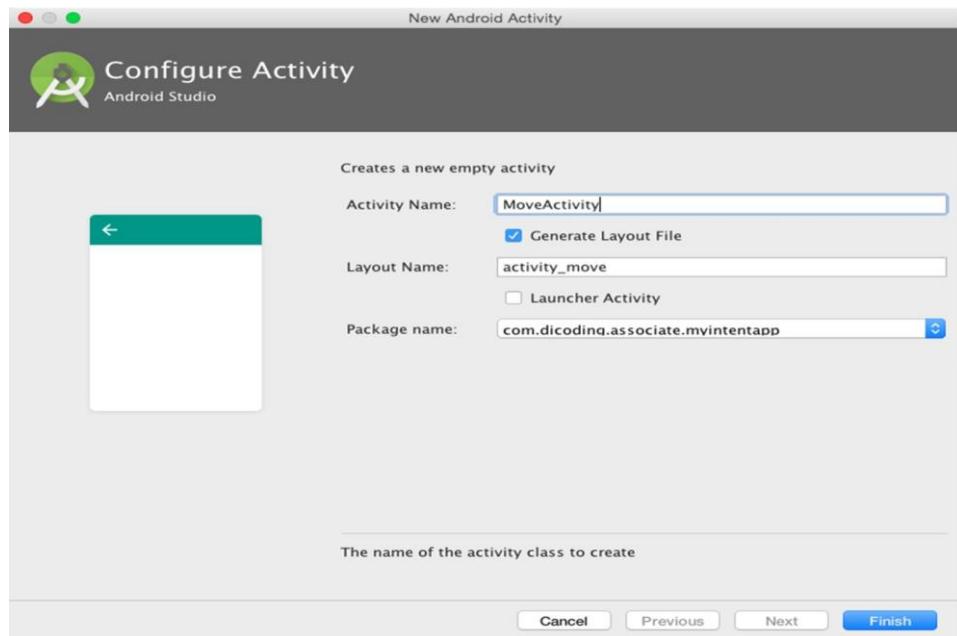
- e) Button **btnMoveActivity** akan memiliki fungsi untuk berpindah Activity ke Activity lain. Sekarang kita buat Activity baru dengan cara sebagai berikut: Klik kanan di package utama aplikasi **com.dicoding.associate.myintentapp** → **New** → **Activity** → **Empty Activity**





Gambar 49 Cara Membuat Package Utama di Android Studio

Lalu isikan **MoveActivity** pada dialog. Ketika sudah klik **finish**.



Gambar 50 Tampilan Untuk Pemberian Nama

- f) Untuk menandakan perpindahan Activity berhasil, silakan tambahkan satu buat **TextView** dan kondisikan **activity_move.xml** menjadi seperti berikut

```

○ <?xml version="1.0" encoding="utf-8"?>
○ <RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
○     xmlns:tools="http://schemas.android.com/tools"
○     android:layout_width="match_parent"
○     android:layout_height="match_parent"
○     android:paddingBottom="@dimen/activity_vertical_margin"

```



```

○ android:paddingLeft="@dimen/activity_horizontal_margin"
○ android:paddingRight="@dimen/activity_horizontal_margin"
○ android:paddingTop="@dimen/activity_vertical_margin"
○ tools:context="com.dicoding.associate.myintentapp.MoveActivity">
<TextView
○     android:layout_width="match_parent"
○     android:layout_height="wrap_content"
○     android:text="Ini MoveActivity"/>
○ </RelativeLayout>

```

- g) Setelah Activity tujuan sudah berhasil diciptakan, sekarang saatnya menambahkan sebuah Intent pada method **onClick()** di **MainActivity.java** menjadi sebagai berikut.

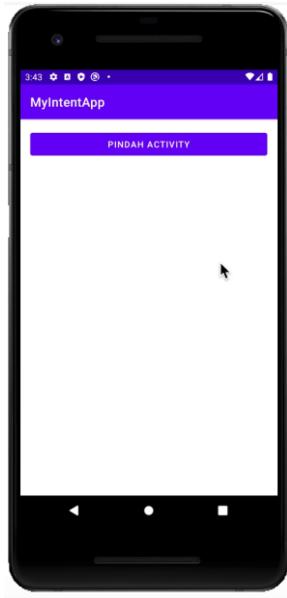
```

○ public class MainActivity extends AppCompatActivity implements View.OnClickListener{
○     private Button btnMoveActivity;
○     @Override
○     protected void onCreate(Bundle savedInstanceState) {
○         super.onCreate(savedInstanceState);
○         setContentView(R.layout.activity_main);
○         btnMoveActivity = (Button) findViewById(R.id.btn_move_activity);
○         btnMoveActivity.setOnClickListener(this);
○     }
○     @Override
○     public void onClick(View v) {
○         switch (v.getId()){
○             case R.id.btn_move_activity:
○                 Intent moveIntent = new Intent(MainActivity.this, MoveActivity.class);
○                 startActivity(moveIntent);
○                 break;
○         }
○     }
○ }

```

Selesai! Langkah pertama untuk memindahkan satu Activity ke Activity lain sudah selesai, sekarang silakan jalankan aplikasi anda dengan mengklik tombol pada menu bar. Seharusnya sekarang anda sudah bisa memindahkan Activity dengan mengklik tombol ‘**Pindah Activity**’.





Gambar 51 Tampilan saat dijalankan Tombol Pindah Activity

2) Eksplisit

Selanjutnya adalah kita akan membuat sebuah Intent yang didalamnya akan membawa data ke Activity tujuan

- a) Ayo buka **activity_main.xml** lagi dan tambahkan satu tombol lagi dibawah tombol sebelumnya sehingga kode **activity_main.xml** kita menjadi seperti ini.

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:tools="http://schemas.android.com/tools"
4.     android:layout_width="match_parent"
5.     android:layout_height="match_parent"
6.     android:paddingBottom="@dimen/activity_vertical_margin"
7.     android:paddingLeft="@dimen/activity_horizontal_margin"
8.     android:paddingRight="@dimen/activity_horizontal_margin"
9.     android:paddingTop="@dimen/activity_vertical_margin"
10.    android:orientation="vertical"
11.    tools:context="com.dicoding.associate.myintentapp.MainActivity">
12.    <Button
13.        android:id="@+id	btn_move_activity"
14.        android:layout_width="match_parent"
15.        android:layout_height="wrap_content"
16.        android:text="Pindah Activity"
17.        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
18.    <Button
19.        android:id="@+id	btn_move_activity_data"
20.        android:layout_width="match_parent"
21.        android:layout_height="wrap_content"
22.        android:text="Pindah Activity dengan Data"
23.        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
```



```
24.</LinearLayout>
```

Tambahkan kembali dimens.xml secara manual di dalam **res → values**.

Dan isikan filenya menjadi seperti berikut.

```
1.<resources>
2.    <!-- Default screen margins, per the Android Design guidelines. -->
3.    <dimen name="activity_horizontal_margin">16dp</dimen>
4.    <dimen name="activity_vertical_margin">16dp</dimen>
5.</resources>
```

- b) Setelah selesai dengan penambahan pada file **activity_main.xml**, maka dilanjutkan dengan menambahkan beberapa baris berikut di **MainActivity.java** (yang ditebalkan).

```
1. public class MainActivity extends AppCompatActivity implements View.OnClickListener{
2.     private Button btnMoveActivity;
3.     private Button btnMoveWithDataActivity;
4.     @Override
5.     protected void onCreate(Bundle savedInstanceState) {
6.         super.onCreate(savedInstanceState);
7.         setContentView(R.layout.activity_main);
8.         btnMoveActivity = (Button)findViewById(R.id.btn_move_activity);
9.         btnMoveActivity.setOnClickListener(this);
10.        btnMoveWithDataActivity =
11.            (Button)findViewById(R.id.btn_move_activity_data);
12.        btnMoveWithDataActivity.setOnClickListener(this);
13.    }
14.    @Override
15.    public void onClick(View v) {
16.        switch (v.getId()){
17.            case R.id.btn_move_activity:
18.                Intent moveIntent = new Intent(MainActivity.this,
19.                    MoveActivity.class);
20.                startActivity(moveIntent);
21.                break;
22.            case R.id.btn_move_activity_data:
23.                break;
24.        }
25.    }
26.}
```



- c) Selanjutnya, buat Activity baru lagi seperti cara sebelumnya dan namakan Activity-nya dengan **MoveWithDataActivity**. Lalu, pada **activity_move_with_data.xml** kita tambahkan sebuah TextView ber-ID untuk menampilkan data yang dikirimkan dari Activity asal.

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:tools="http://schemas.android.com/tools"
4.     android:layout_width="match_parent"
5.     android:layout_height="match_parent"
6.     android:paddingBottom="@dimen/activity_vertical_margin"
7.     android:paddingLeft="@dimen/activity_horizontal_margin"
8.     android:paddingRight="@dimen/activity_horizontal_margin"
9.     android:paddingTop="@dimen/activity_vertical_margin"
10.    tools:context="com.dicoding.associate.myintentapp.MoveWithDataActivity">
11.    <TextView
12.        android:id="@+id/tv_data_received"
13.        android:layout_width="match_parent"
14.        android:layout_height="wrap_content"
15.        android:text="Data Received"/>
16.</RelativeLayout>
```

- d) Selanjutnya pada **MoveWithDataActivity.java** kita kondisikan menjadi seperti berikut:

```

1. public class MoveWithDataActivity extends AppCompatActivity {
2.     public static String EXTRA_AGE = "extra_age";
3.     public static String EXTRA_NAME = "extra_name";
4.     private TextView tvDataReceived;
5.     @Override
6.     protected void onCreate(Bundle savedInstanceState) {
7.         super.onCreate(savedInstanceState);
8.         setContentView(R.layout.activity_move_with_data);
9.         tvDataReceived = (TextView) findViewById(R.id.tv_data_received);
10.        String name = getIntent().getStringExtra(EXTRA_NAME);
11.        int age = getIntent().getIntExtra(EXTRA_AGE, 0);
12.        String text = "Name : " + name + ", Your Age : " + age;
13.        tvDataReceived.setText(text);
14.    }
15.}
```

- e) Dan sekarang saatnya kita menambahkan obyek Intent pada **MainActivity.java** seperti baris yang ditebalkan.



```

1. public class MainActivity extends AppCompatActivity implements View.OnClickListener{
2.     private Button btnMoveActivity;
3.     private Button btnMoveWithDataActivity;
4.     @Override
5.     protected void onCreate(Bundle savedInstanceState) {
6.         super.onCreate(savedInstanceState);
7.         setContentView(R.layout.activity_main);
8.         btnMoveActivity = (Button) findViewById(R.id.btn_move_activity);
9.         btnMoveActivity.setOnClickListener(this);
10.        btnMoveWithDataActivity = (Button) findViewById(R.id.btn_move_activity_data);
11.        btnMoveWithDataActivity.setOnClickListener(this);
12.    }
13.    @Override
14.    public void onClick(View v) {
15.        switch (v.getId()){
16.            case R.id.btn_move_activity:
17.                Intent moveIntent = new Intent(MainActivity.this, MoveActivity.class);
18.                startActivity(moveIntent);
19.                break;
20.            case R.id.btn_move_activity_data:
21.                Intent moveWithDataIntent = new Intent(MainActivity.this,
22.                    MoveWithDataActivity.class);
23.                moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_NAME,
24.                    "DicodingAcademy Boy");
25.                moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_AGE,
26.                    5);
27.                startActivity(moveWithDataIntent);
28.                break;
29.        }
30.    }
31.}

```

- f) Sekarang, silakan jalankan aplikasi andalagi dan coba klik pada tombol ‘**Pindah Activity dengan Data**’ dan Selamat! Seharusnya sudah bisa memindahkan satu Activity ke Activity lain dengan membawa data.





Gambar 52 Tampilan Akhir saat Dijalankan Tombol Pindah Activity dengan Data

3) Implisit

Baik, anda sudah belajar bagaimana menggunakan Intent dengan tipe eksplisit, sekarang saatnya anda melanjutkan ke **Intent** dengan tipe **implicit**

- a) Buka kembali **activity_main.xml** tambahkan satu tombol lagi sebagai berikut :

```
1. <Button  
2.     android:id="@+id/btn_dial_number"  
3.     android:layout_width="match_parent"  
4.     android:layout_height="wrap_content"  
5.     android:text="Dial a Number"  
6.     android:layout_marginBottom="@dimen/activity_vertical_margin"/>
```

Sehingga file **activity_main.xml** kita sekarang menjadi:

```
1. <?xml version="1.0" encoding="utf-8"?>  
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
3.     xmlns:tools="http://schemas.android.com/tools"  
4.     android:layout_width="match_parent"  
5.     android:layout_height="match_parent"  
6.     android:paddingBottom="@dimen/activity_vertical_margin"  
7.     android:paddingLeft="@dimen/activity_horizontal_margin"  
8.     android:paddingRight="@dimen/activity_horizontal_margin"  
9.     android:paddingTop="@dimen/activity_vertical_margin"  
10.    android:orientation="vertical"  
11.    tools:context="com.dicoding.associate.myintentapp.MainActivity">
```



```

12.    <Button
13.        android:id="@+id	btn_move_activity"
14.        android:layout_width="match_parent"
15.        android:layout_height="wrap_content"
16.        android:text="Pindah Activity"
17.        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
18.    <Button
19.        android:id="@+id	btn_move_activity_data"
20.        android:layout_width="match_parent"
21.        android:layout_height="wrap_content"
22.        android:text="Pindah Activity dengan Data"
23.        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
24.    <Button
25.        android:id="@+id	btn_dial_number"
26.        android:layout_width="match_parent"
27.        android:layout_height="wrap_content"
28.        android:text="Dial a Number"
29.        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
30.</LinearLayout>

```

- b) Sekarang, buka kembali **MainActivity.java** dan lanjutkan dengan melakukan casting tombol Dial Number, set listener dan menambahkan action ketika tombol diklik.

```

1. public class MainActivity extends AppCompatActivity implements View.OnClickListener{
2.     private Button btnMoveActivity;
3.     private Button btnMoveWithDataActivity;
4.     private Button btnDialPhone;
5.     @Override
6.     protected void onCreate(Bundle savedInstanceState) {
7.         super.onCreate(savedInstanceState);
8.         setContentView(R.layout.activity_main);
9.         btnMoveActivity = (Button) findViewById(R.id.btn_move_activity);
10.        btnMoveActivity.setOnClickListener(this);
11.        btnMoveWithDataActivity =
12.            (Button) findViewById(R.id.btn_move_activity_data);
13.        btnMoveWithDataActivity.setOnClickListener(this);
14.        btnDialPhone = (Button) findViewById(R.id.btn_dial_number);
15.        btnDialPhone.setOnClickListener(this);
16.    }
17.    @Override
18.    public void onClick(View v) {
19.        switch (v.getId()){
20.            case R.id.btn_move_activity:
21.                Intent moveIntent = new Intent(MainActivity.this,
22.                    MoveActivity.class);
23.                startActivity(moveIntent);
24.                break;
25.            case R.id.btn_move_activity_data:
26.                Intent moveWithDataIntent = new
27.                Intent(MainActivity.this, MoveWithDataActivity.class);
28.                moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_

```



```

    TRA_NAME, "DicodingAcademy Boy");

26.                                moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA
TRA_AGE, 5);
27.                                startActivity(moveWithDataIntent);
28.                                break;
29.                case R.id.btn_dial_number:
30.                    String phoneNumber = "081210841382";
31.                    Intent dialPhoneIntent = new
Intent(Intent.ACTION_DIAL, Uri.parse("tel:" + phoneNumber));
32.                    startActivity(dialPhoneIntent);
33.                    break;
34.                }
35.            }
36.

```

Setelah selesai, silakan jalankan lagi aplikasi, sehingga sudah bisa mendial sebuah nomor telepon melalui aplikasi telepon bawaan dari Android/Device anda. Selamat!



Gambar 53 Tampilan Akhir saat Dijalankan Tombol Dial a Number

4. Tugas Praktikum

Buatlah program Intent untuk menghitung 3 luas bangun datar (masing-masing mahasiswa membuat berbeda bangun datar) dengan ketentuan, halaman awal adalah menampilkan button untuk menu bangun datar tersebut, kemudian jika memilih salah satu menu tersebut, maka akan tampil form untuk menghitung luas bangun datar tersebut!



Praktikum 6

Intent Lanjutan

1. Sub Pokok Pembahasan

- a) Intent Explicit dengan Parcelable
- b) Intent dengan ResultActivity

2. Capaian Pembelajaran

- a. Mahasiswa mampu menanamkan sikap jujur, bertanggung jawab, disiplin, saling menghargai dan berinovasi dengan tetap menjunjung tinggi nilai-nilai agama dan ketuhanan.
- b. Mahasiswa mampu menunjukkan sikap nasionalisme, mampu bekerja dalam tim sesuai kaidah keilmuan pada proyek pengembangan aplikasi mobile programming.
- c. Mahasiswa mampu berpartisipasi aktif, bertanggung jawab, dan memiliki motivasi mengembangkan diri dalam mendapatkan solusi permasalahan melalui aplikasi mobile programming
- d. Mahasiswa mampu menguasai, merancang dan mengembangkan bahasa pemrograman berbasis mobile, serta mampu menganalisis solusi dalam model bahasa pemrograman mobile.
- e. Mahasiswa mampu mengimplementasi pengetahuan dan teknologi menggunakan pendekatan *mobile programming*.

3. Intent Lanjutan

Tujuan

Codelab ini menitik beratkan pada implementasi Intent untuk melakukan perpindahan dari Activity ke Activity dengan atau tidak membawa data. Beberapa bagian dari code lab ini akan menjawab beberapa pertanyaan umum dalam pengembangan aplikasi Android sebagai berikut:

- 1) Bagaimana berpindah dari satu Activity ke Activity lain dengan membawa data? Obyek Model Plain Old Java Object (POJO)
- 2) Mengirimkan hasil nilai balik melalui Intent



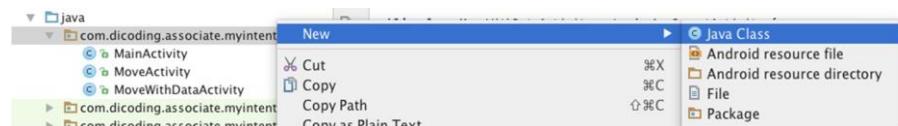
1) Intent Explicit dengan Parcelable

Bagi yang belum mengenal apa itu Plain Old Java Object (POJO), berikut penjelasan singkatnya. POJO adalah sebuah kelas Java biasa yang tidak bergantung dengan kelas lain. Umumnya kelas POJO ini disebut sebagai kelas Java yang memiliki properti/variabel dan metode *setter-getter*. Seperti dibawah ini.

```
1. public class MyBean {  
2.     private String someProperty;  
3.  
4.     public String getSomeProperty() {  
5.         return someProperty;  
6.     }  
7.  
8.  
9.     public void setSomeProperty(String someProperty) {  
10.        this.someProperty = someProperty;  
11.    }  
12.}
```

Lalu apa kegunaan obyek POJO? POJO akan membantu kita saat aplikasi semakin kompleks. Contohnya POJO bisa kita gunakan untuk melakukan koneksi ke server untuk *request API* atau akses ke *database* lokal dengan SQLite. Selain POJO, Android juga memiliki obyek parcelable POJO. Obyek ini akan lebih memudahkan kita untuk melakukan pengiriman data dari satu activity ke activity lainnya.

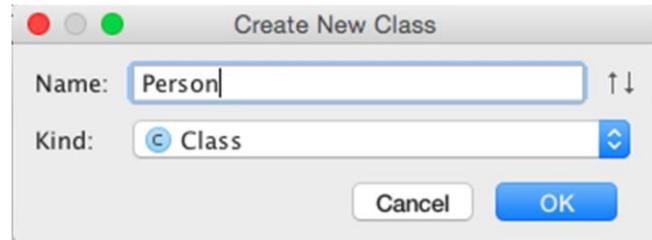
- a) Sebelum membuat obyek parcelable, kita buat kelas Java POJO terlebih dahulu. Caranya, klik kanan pada package utama proyek → **New** → **Java Class**



Gambar 54 Cara Membuat Kelas Java Di Intent Lanjutan



- b) Pada dialog, isikan nama kelas dengan ‘Person’. Lalu klik **OK**.



Gambar 55 Tampilan dalam Memberikan Nama Class

- c) Setelah selesai terbentuk kelas **Person**, kondisikan kelas tersebut menjadi seperti ini:

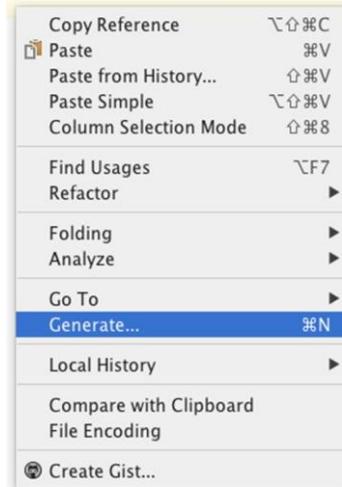
```
1. public class Person {  
2.     private String name;  
3.     private int age;  
4.     private String email;  
5.     private String city;  
6.     public String getName() {  
7.         return name;  
8.     }  
9.     public void setName(String name) {  
10.        this.name = name;  
11.    }  
12.    public int getAge() {  
13.        return age;  
14.    }  
15.    public void setAge(int age) {  
16.        this.age = age;  
17.    }  
18.    public String getEmail() {  
19.        return email;  
20.    }  
21.    public void setEmail(String email) {  
22.        this.email = email;  
23.    }  
24.    public String getCity() {  
25.        return city;  
26.    }  
27.    public void setCity(String city) {  
28.        this.city = city;  
29.    }  
30.}
```

Anda tidak perlu menulis satu per satu metode *setter-getter* nya, di Android Studio tersedia fasilitas *generator* untuk hal tersebut. Caranya,



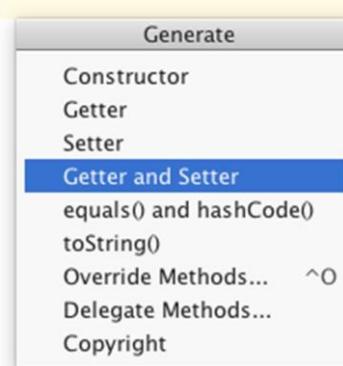
setelah Anda menentukan variabel apa saja yang akan digunakan lakukan, lakukan beberapa langkah sebagai berikut :

Klik kanan pada *workspace*, pilih **Generate**.



Gambar 56 Tampilan untuk Memilih Generate

d) Pilih **Getter and Setter**.



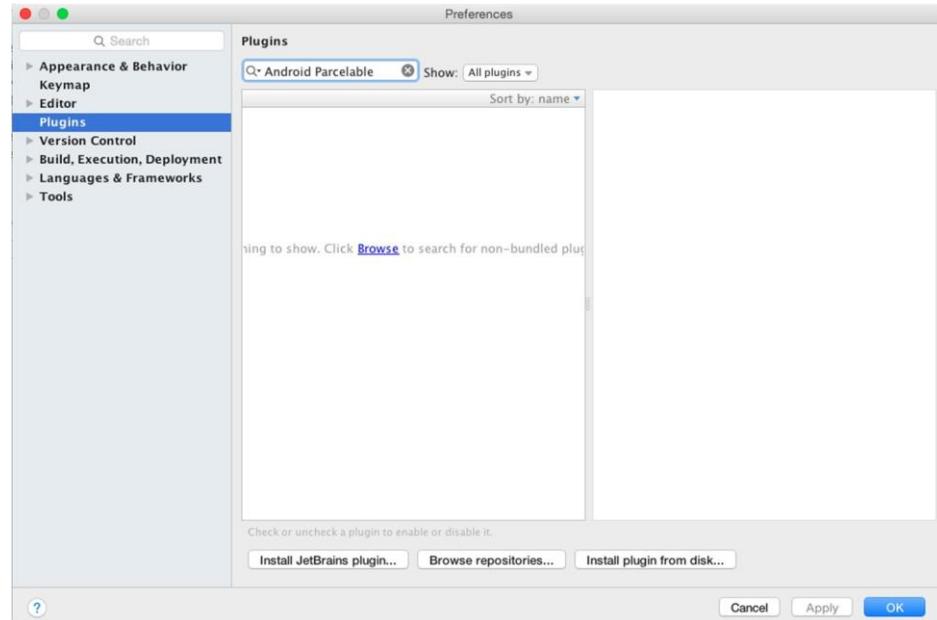
Gambar 57 Tampilan untuk Memilih Getter and Setter

e) Pilih semua variabel dengan cara **Ctrl+A** pada windows dan **Command+A** pada mac. Bila sudah, maka klik **OK**. Secara otomatis metode *setter* dan *getter* masing-masing variabel akan dibuatkan.

Selanjutnya kita akan menambahkan *plugin* Android Parcelable Generator pada Android Studio. *Plugin* ini berguna untuk mengubah kelas POJO menjadi obyek **Parcelable**. Langkah- langkahnya sebagai berikut :

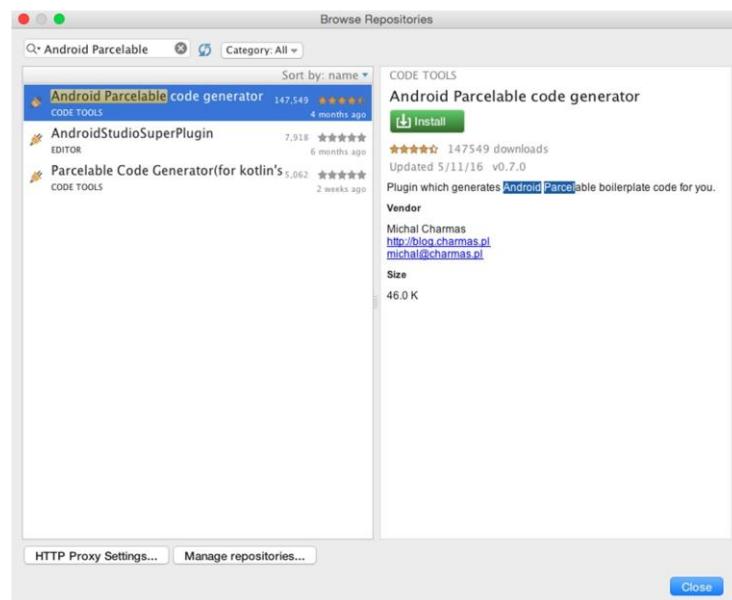


- a) Buka settings dengan cara **File** → **Settings** pada Windows dan **Android Studio** → **Preferences** pada Mac.
- b) Pilih menu **Plugin**. Dan inputkan "Android Parcelable" pada search box kemudian tekan **Enter**.



Gambar 58 Tampilan dalam Menu Plugin dan Android Parcelable

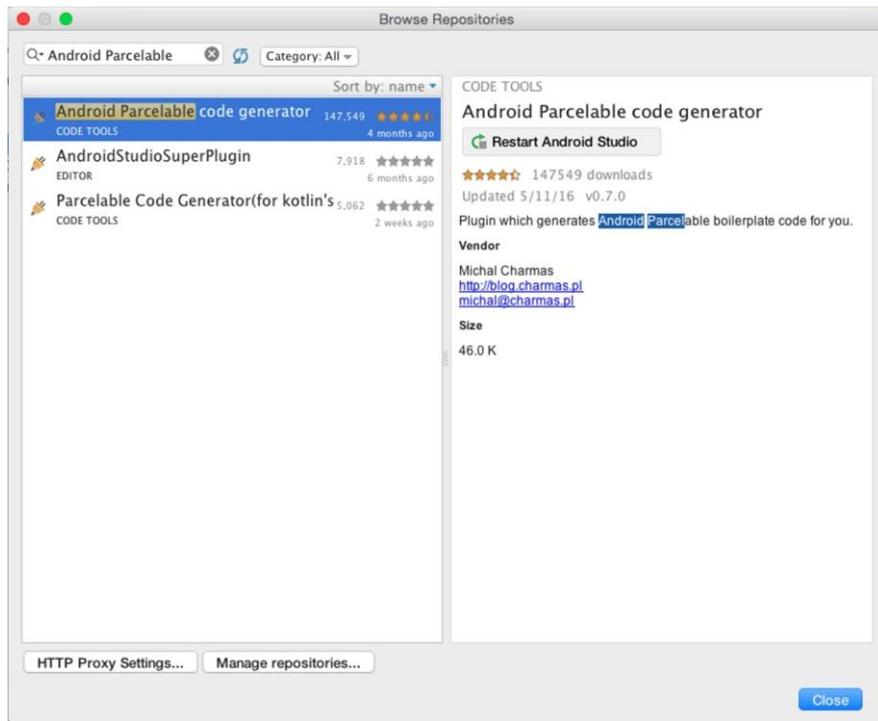
- c) Klik **browse**, dengan ini kita memerintahkan Android Studio untuk melakukan pencarian *plugin* di *server*. Setelah ketemu Android Parcelable code generator, klik **Install**.



Gambar 59 Tampilan Android Parcelable di Install



- d) Biarkan Android Studio melakukan proses instalasi *plugin*. Bila sukses, maka Anda akan diminta untuk melakukan *restart* Android Studio yang sedang berjalan.



Gambar 60 Tampilan saat Android Parcelable sudah Diinstall dan Restart Android Studio

Klik **Restart** → Klik **OK** dan jika ada pilihan untuk *restart* Android Studio maka pilih *restart*. Biarkan Android Studio untuk *restart* agar *plugin*-nya dapat kita gunakan setelahnya.

Catatan : Jika dialog search tidak berhasil mencari plugin, kosongkan kata pencarian dan masukkan kembali kata pencarian "Android Parcelable". Ini bisa terjadi karena ada *error* di dalam dialog Plugins.

- e) Selanjutnya, setelah plugin Parcelable generator terinstal saatnya kita implementasikan pada kelas **Person**. Dengan cara :

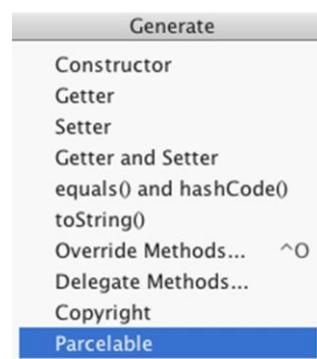


Klik kanan pada *workspace* (pastikan anda berada pada tab kelas Person) pilih menu **Generate**.



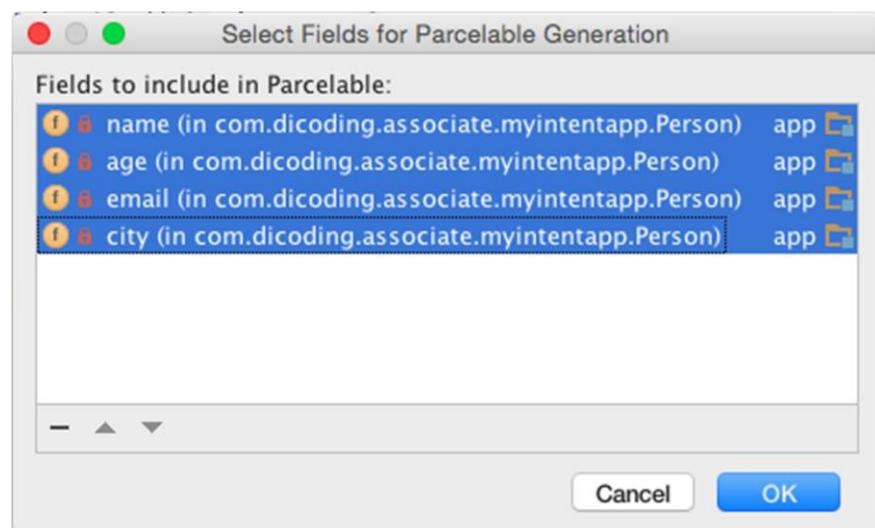
Gambar 61 Tampilan Untuk Memilih Generate

f) Pilih **Parcelable**



Gambar 62 Tampilan saat Memilih Parcelable

g) Pada dialog box yang tampil, pilih semua variable yang ada lalu klik **OK**



Gambar 63 Tampilan pada Dialog box



Sekarang Anda bisa menemukan beragam metode parcelable untuk kelas **Person**. Sekarang kelas **Person** sudah sah menjadi kelas parcelable dan dapat dikirimkan ke activity lain melalui intent.

- h) Sekarang waktunya kita implementasikan pada intent. Namun sebelumnya tambahkan satu tombol lagi pada **activity_main.xml** sebagai berikut:

```
1. <Button  
2.     android:id="@+id	btn_move_activity_object"  
3.     android:layout_width="match_parent"  
4.     android:layout_height="wrap_content"  
5.     android:text="Pindah Activity dengan Object"  
6.     android:layout_marginBottom="@dimen/activity_vertical_margin"/>
```

Sehingga **activity_main.xml** kita menjadi sebagai berikut :

```
1. <?xml version="1.0" encoding="utf-8"?>  
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
3.     xmlns:tools="http://schemas.android.com/tools"  
4.     android:layout_width="match_parent"  
5.     android:layout_height="match_parent"  
6.     android:paddingBottom="@dimen/activity_vertical_margin"  
7.     android:paddingLeft="@dimen/activity_horizontal_margin"  
8.     android:paddingRight="@dimen/activity_horizontal_margin"  
9.     android:paddingTop="@dimen/activity_vertical_margin"  
10.    android:orientation="vertical"  
11.    tools:context="com.dicoding.associate.myintentapp.MainActivity">  
12.        <Button  
13.            android:id="@+id	btn_move_activity"  
14.            android:layout_width="match_parent"  
15.            android:layout_height="wrap_content"  
16.            android:text="Pindah Activity"  
17.            android:layout_marginBottom="@dimen/activity_vertical_margin"/>  
18.        <Button  
19.            android:id="@+id	btn_move_activity_data"  
20.            android:layout_width="match_parent"  
21.            android:layout_height="wrap_content"  
22.            android:text="Pindah Activity dengan Data"  
23.            android:layout_marginBottom="@dimen/activity_vertical_margin"/>  
24.        <Button  
25.            android:id="@+id	btn_move_activity_object"  
26.            android:layout_width="match_parent"  
27.            android:layout_height="wrap_content"  
28.            android:text="Pindah Activity dengan Object"  
29.            android:layout_marginBottom="@dimen/activity_vertical_margin"/>  
30.</LinearLayout>
```



- i) Setelah selesai dengan layout xml, sekarang kita tambahkan (pada bagian yang ditebalkan) juga di kelas **MainActivity.java** sehingga menjadi sebagai berikut :

```

1. public class MainActivity extends AppCompatActivity implements View.OnClickListener{
2.     private Button btnMoveActivity;
3.     private Button btnMoveWithDataActivity;
4.     private Button btnMoveWithObject;
5.     @Override
6.     protected void onCreate(Bundle savedInstanceState) {
7.         super.onCreate(savedInstanceState);
8.         setContentView(R.layout.activity_main);
9.         btnMoveActivity = (Button) findViewById(R.id.btn_move_activity);
10.        btnMoveActivity.setOnClickListener(this);
11.        btnMoveWithDataActivity =
12.            (Button) findViewById(R.id.btn_move_activity_data);
13.            btnMoveWithDataActivity.setOnClickListener(this);
14.        btnMoveWithObject=
15.            (Button) findViewById(R.id.btn_move_activity_object);
16.            btnMoveWithObject.setOnClickListener(this);
17.    }
18.    @Override
19.    public void onClick(View v) {
20.        switch (v.getId()){
21.            case R.id.btn_move_activity:
22.                Intent moveIntent = new Intent(MainActivity.this,
23. MoveActivity.class);
24.                startActivity(moveIntent);
25.                break;
26.            case R.id.btn_move_activity_data:
27.                Intent moveWithDataIntent = new
28. Intent(MainActivity.this,MoveWithDataActivity.class);
29.                moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_NAME, "DicodingAcademy Boy");
30.                moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_AGE, 5);
31.                startActivity(moveWithDataIntent);
32.                break;
33.            case R.id.btn_move_activity_object:
34.                break;
35.        }

```



- j) Buat **Activity** baru dengan nama **MoveWithObjectActivity** lalu pada **activity_move_with_object.xml** tambahkan satu buah **TextView** untuk menampilkan data yang dikirimkan.

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:tools="http://schemas.android.com/tools"
4.     android:layout_width="match_parent"
5.     android:layout_height="match_parent"
6.     android:paddingBottom="@dimen/activity_vertical_margin"
7.     android:paddingLeft="@dimen/activity_horizontal_margin"
8.     android:paddingRight="@dimen/activity_horizontal_margin"
9.     android:paddingTop="@dimen/activity_vertical_margin"
10.    tools:context="com.dicoding.associate.myintentapp.MoveWithObjectActivity">
11.        <TextView
12.            android:id="@+id/tv_object_received"
13.            android:layout_width="match_parent"
14.            android:layout_height="wrap_content"
15.            android:text="Object Received"/>
16.</RelativeLayout>
```

- k) Sekarang, pada **MoveWithObjectActivity.java** tambahkan kodenya sehingga menjadi sebagai berikut:

```

1. public class MoveWithObjectActivity extends AppCompatActivity {
2.     public static String EXTRA_PERSON = "extra_person";
3.     private TextView tvObject;
4.     @Override
5.     protected void onCreate(Bundle savedInstanceState) {
6.         super.onCreate(savedInstanceState);
7.         setContentView(R.layout.activity_move_with_object);
8.         tvObject = (TextView) findViewById(R.id.tv_object_received);
9.         Person mPerson = getIntent().getParcelableExtra(EXTRA_PERSON);
10.        String text = "Name : "+mPerson.getName()+", Email : "+mPerson.get
11.          Email()+" , Age : "+mPerson.getAge()
12.          + ", Location : "+mPerson.getCity();
13.        tvObject.setText(text);
14.    }
```

- l) Setelah kita membuat activity tujuan dengan sempurna, sekarang kita hanya butuh menambahkan beberapa baris pada bagian **switch... case** sebagai berikut:



```

1. Person mPerson = new Person();
2. mPerson.setName("DicodingAcademy");
3. mPerson.setAge(5);
4. mPerson.setEmail("academy@dicoding.com");
5. mPerson.setCity("Bandung");
6. Intent moveWithObjectIntent = new Intent(MainActivity.this, MoveWithObjectActivity.class);
7. moveWithObjectIntent.putExtra(MoveWithObjectActivity.EXTRA_PERSON, mPerson);
8. startActivity(moveWithObjectIntent);

```

m) Sehingga kode **MainActivity.java** kita menjadi sebagai berikut :

```

1. public class MainActivity extends AppCompatActivity implements View.OnClickListener{
2.     private Button btnMoveActivity;
3.     private Button btnMoveWithDataActivity;
4.     private Button btnMoveWithObject;
5.     @Override
6.     protected void onCreate(Bundle savedInstanceState) {
7.         super.onCreate(savedInstanceState);
8.         setContentView(R.layout.activity_main);
9.         btnMoveActivity = (Button)findViewById(R.id.btn_move_activity);
10.        btnMoveActivity.setOnClickListener(this);
11.        btnMoveWithDataActivity =
12.            (Button)findViewById(R.id.btn_move_activity_data);
13.        btnMoveWithDataActivity.setOnClickListener(this);
14.        btnMoveWithObject =
15.            (Button)findViewById(R.id.btn_move_activity_object);
16.        btnMoveWithObject.setOnClickListener(this);
17.    }
18.    @Override
19.    public void onClick(View v) {
20.        switch (v.getId()){
21.            case R.id.btn_move_activity:
22.                Intent moveIntent = new Intent(MainActivity.this,
23.                    MoveActivity.class);
24.                startActivity(moveIntent);
25.                break;
26.            case R.id.btn_move_activity_data:
27.                Intent moveWithDataIntent = new
28.                    Intent(MainActivity.this, MoveWithDataActivity.class);
29.                    moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_NAME, "DicodingAcademy Boy");
30.                    moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_AGE, 5);
31.                    startActivity(moveWithDataIntent);
32.                    break;
33.            case R.id.btn_move_activity_object:
34.        }
35.    }
36. }

```



```

30.         Person mPerson = new Person();
31.         mPerson.setName("DicodingAcademy");
32.         mPerson.setAge(5);
33.         mPerson.setEmail("academy@dicoding.com");
34.         mPerson.setCity("Bandung");
35.         Intent moveWithObjectIntent = new
Intent(MainActivity.this, MoveWithObjectActivity.class);
36.         moveWithObjectIntent.putExtra(MoveWithObjectActivi
ty.EXTRA_PERSON, mPerson);
37.         startActivity(moveWithObjectIntent);
38.         break;
39.     }
40. }
41.}

```

Setelah semua komponen selesai dibuat, saatnya menjalankan kembali aplikasinya. Cobalah menakan tombol '**Pindah Activity dengan Object**'. Sekarang Anda sudah berhasil memindahkan satu obyek secara utuh, Selamat!

2) Intent dengan ResultActivity

Pada materi ini, kita akan membedah hubungan activity dan intent dalam menerima nilai balik. Kadang kala, ketika kita menjalankan sebuah activity dari activity lain, kita mengharapkan ada nilai hasil balik dari activity yang dijalankan ketika ia ditutup.

Contohnya kita memiliki activity A yang menjalankan activity B untuk melakukan sebuah proses. Lalu nilai hasil dari proses tersebut dikirimkan kembali ke activity A sebelum activity B ditutup dengan pemanggilan metode `finish()`. Itulah yang dinamakan sebuah activity menerima nilai hasil balik dari activity yang dia jalankan. Untuk memperjelas, ayo kita lanjutkan lagi proses ngodingnya.

- Sekarang untuk menerapkan konsep diatas, buat sebuah activity baru dengan nama `MoveForResultActivity`. Setelah itu, pada `activity_move_for_result.xml` silakan kondisikan layout kita dengan **1**



buah TextView, 1 buah Radio Group, 3 buah Radio Button dan 2 buah Button sebagai berikut :

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:tools="http://schemas.android.com/tools"
4.     android:layout_width="match_parent"
5.     android:layout_height="match_parent"
6.     android:paddingBottom="@dimen/activity_vertical_margin"
7.     android:paddingLeft="@dimen/activity_horizontal_margin"
8.     android:paddingRight="@dimen/activity_horizontal_margin"
9.     android:paddingTop="@dimen/activity_vertical_margin"
10.    android:orientation="vertical"
11.    tools:context="com.dicoding.associate.myintentapp.MoveForResultActivi
ty">
12.        <TextView
13.            android:layout_width="match_parent"
14.            android:layout_height="wrap_content"
15.            android:text="Pilih angka yang kamu suka"
16.            android:layout_marginBottom="@dimen/activity_vertical_margin"/>
17.        <RadioGroup
18.            android:id="@+id/rg_number"
19.            android:layout_width="match_parent"
20.            android:layout_height="wrap_content"
21.            android:orientation="vertical">
22.            <RadioButton
23.                android:id="@+id/rb_50"
24.                android:layout_width="match_parent"
25.                android:layout_height="wrap_content"
26.                android:text="50"
27.                android:layout_marginBottom="@dimen/activity_vertical_margin"/>
28.            <RadioButton
29.                android:id="@+id/rb_100"
30.                android:layout_width="match_parent"
31.                android:layout_height="wrap_content"
32.                android:text="100"
33.                android:layout_marginBottom="@dimen/activity_vertical_margin"/>
34.            <RadioButton
35.                android:id="@+id/rb_150"
36.                android:layout_width="match_parent"
37.                android:layout_height="wrap_content"
38.                android:text="150"
39.                android:layout_marginBottom="@dimen/activity_vertical_margin"/>
40.            <RadioButton
41.                android:id="@+id/rb_200"
42.                android:layout_width="match_parent"
43.                android:layout_height="wrap_content"
```



```

44.         android:text="200"
45.         android:layout_marginBottom="@dimen/activity_vertical_margin"/>
46.     
```

47. <Button
48. android:id="@+id btn_choose"
49. android:layout_width="match_parent"
50. android:layout_height="wrap_content"
51. android:text="Pilih"/>

```

52.</LinearLayout>

```

- b) Setelah selesai, lanjutkan ke berkas **MoveForResultActivity.java** dengan menuliskan kode sebagai berikut :

```

1. public class MoveForResultActivity extends AppCompatActivity
2.     implements View.OnClickListener{
3.     private Button btnChoose;
4.     private RadioGroup rgNumber;
5.     public static String EXTRA_SELECTED_VALUE = "extra_selected_value";
6.     public static int RESULT_CODE = 110;
7.     @Override
8.     protected void onCreate(Bundle savedInstanceState) {
9.         super.onCreate(savedInstanceState);
10.        setContentView(R.layout.activity_move_for_result);
11.        btnChoose = (Button) findViewById(R.id.btn_choose);
12.        btnChoose.setOnClickListener(this);
13.        rgNumber = (RadioGroup) findViewById(R.id.rg_number);
14.    }
15.    @Override
16.    public void onClick(View v) {
17.        if (v.getId() == R.id.btn_choose){
18.            if (rgNumber.getCheckedRadioButtonId() != 0){
19.                int value = 0;
20.                switch(rgNumber.getCheckedRadioButtonId()){
21.                    case R.id.rb_50:
22.                        value = 50;
23.                        break;
24.                    case R.id.rb_100:
25.                        value = 100;
26.                        break;
27.                    case R.id.rb_150:
28.                        value = 150;
29.                        break;
30.                    case R.id.rb_200:
31.                        value = 200;
32.                        break;
33.                }
34.            }
35.        }
36.    }
37. }

```



```

34.         Intent resultIntent = new Intent();
35.         resultIntent.putExtra(EXTRA_SELECTED_VALUE,value);
36.         setResult(RESULT_CODE,resultIntent);
37.         finish();
38.     }
39. }
40. }
41.

```

- c) Pada `activity_main.xml` tambahkan lagi satu **Button** dan satu **TextView** yang akan kita gunakan untuk menjalankan **MoveForResultActivity** sebagai berikut:

```

1. <Button
2.     android:id="@+id/btn_move_for_result"
3.     android:layout_width="match_parent"
4.     android:layout_height="wrap_content"
5.     android:text="Pindah Activity untuk Result"
6.     android:layout_marginBottom="@dimen/activity_vertical_margin"/>
7. <TextView
8.     android:id="@+id/tv_result"
9.     android:layout_width="match_parent"
10.    android:layout_height="wrap_content"
11.    android:textSize="24sp"
12.    android:gravity="center"
13.    android:text="Hasil"/>

```

Sehingga layout `activity_main.xml` yang kita punya menjadi seperti ini:

```

1.<?xml version="1.0" encoding="utf-8"?>
2.<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.    xmlns:tools="http://schemas.android.com/tools"
4.    android:layout_width="match_parent"
5.    android:layout_height="match_parent"
6.    android:paddingBottom="@dimen/activity_vertical_margin"
7.    android:paddingLeft="@dimen/activity_horizontal_margin"
8.    android:paddingRight="@dimen/activity_horizontal_margin"
9.    android:paddingTop="@dimen/activity_vertical_margin"
10.   android:orientation="vertical"
11.   tools:context="com.dicoding.associate.myintentapp.MainActivity">
12.   <Button
13.     android:id="@+id/btn_move_activity"
14.     android:layout_width="match_parent"
15.     android:layout_height="wrap_content"
16.     android:text="Pindah Activity"
17.     android:layout_marginBottom="@dimen/activity_vertical_margin"/>
18.   <Button
19.     android:id="@+id/btn_move_activity_data"
20.     android:layout_width="match_parent"
21.     android:layout_height="wrap_content"

```



```

22.    android:text="Pindah Activity dengan Data"
23.    android:layout_marginBottom="@dimen/activity_vertical_margin"/>
24.    <Button
25.        android:id="@+id/btn_move_activity_object"
26.        android:layout_width="match_parent"
27.        android:layout_height="wrap_content"
28.        android:text="Pindah Activity dengan Object"
29.        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
30.    <Button
31.        android:id="@+id/btn_dial_number"
32.        android:layout_width="match_parent"
33.        android:layout_height="wrap_content"
34.        android:text="Dial a Number"
35.        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
36.    <Button
37.        android:id="@+id/btn_move_for_result"
38.        android:layout_width="match_parent"
39.        android:layout_height="wrap_content"
40.        android:text="Pindah Activity untuk Result"
41.        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
42.    <TextView
43.        android:id="@+id/tv_result"
44.        android:layout_width="match_parent"
45.        android:layout_height="wrap_content"
46.        android:textSize="24sp"
47.        android:gravity="center"
48.        android:text="Hasil"/>
49.</LinearLayout>

```

- d) Setelah selesai, kita lanjut garap berkas **MainActivity.java**. Buka berkasnya dan sesuaikan kodenya sebagai berikut:

```

1. package com.dicoding.associate.myintentapp;
2. import android.content.Intent;
3. import android.net.Uri;
4. import android.support.v7.app.AppCompatActivity;
5. import android.os.Bundle;
6. import android.view.View;
7. import android.widget.Button;
8. import android.widget.TextView;
9. public class MainActivity extends AppCompatActivity implements View.OnClickListener{
10.     private Button btnMoveActivity;
11.     private Button btnMoveWithDataActivity;
12.     private Button btnMoveWithObject;
13.     private Button btnDialPhone;

```



```

14.     private Button btnMoveForResult;
15.     private TextView tvResult;
16.     private int REQUEST_CODE = 100;
17.     @Override
18.     protected void onCreate(Bundle savedInstanceState) {
19.         super.onCreate(savedInstanceState);
20.         setContentView(R.layout.activity_main);
21.         btnMoveActivity = (Button) findViewById(R.id.btn_move_activity);
22.         btnMoveActivity.setOnClickListener(this);
23.         btnMoveWithDataActivity = (Button) findViewById(R.id.btn_move_activity_data);
24.         btnMoveWithDataActivity.setOnClickListener(this);
25.         btnMoveWithObject = (Button) findViewById(R.id.btn_move_activity_object);
26.         btnMoveWithObject.setOnClickListener(this);
27.         btnDialPhone = (Button) findViewById(R.id.btn_dial_number);
28.         btnDialPhone.setOnClickListener(this);
29.         btnMoveForResult =
30.             (Button) findViewById(R.id.btn_move_for_result);
31.         btnMoveForResult.setOnClickListener(this);
32.     }
33.     @Override
34.     public void onClick(View v) {
35.         switch (v.getId()) {
36.             case R.id.btn_move_activity:
37.                 Intent moveIntent = new Intent(MainActivity.this, MoveActivity.class);
38.                 startActivity(moveIntent);
39.                 break;
40.             case R.id.btn_move_activity_data:
41.                 Intent moveWithDataIntent = new Intent(MainActivity.this,
42.                     MoveWithDataActivity.class);
43.                 moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_NAME,
44.                     "DicodingAcademy Boy");

```



```

        moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_AGE, 5);
44.        startActivity(moveWithDataIntent);
45.        break;
46.        case R.id.btn_move_activity_object:
47.            Person mPerson = new Person();
48.            mPerson.setName("DicodingAcademy");
49.            mPerson.setAge(5);
50.            mPerson.setEmail("academy@dicoding.com");
51.            mPerson.setCity("Bandung");
52.            Intent moveWithObjectIntent = new Intent(MainActivity.this, MoveWithObjectActivity.class);
53.            moveWithObjectIntent.putExtra(MoveWithObjectActivity.EXTRA_PERSON, mPerson);
54.            startActivity(moveWithObjectIntent);
55.            break;
56.            case R.id.btn_dial_number:
57.                String phoneNumber = "081210841382";
58.                Intent dialPhoneIntent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:"+phoneNumber));
59.                startActivity(dialPhoneIntent);
60.                break;
61.                case R.id.btn_move_for_result:
62.                    Intent moveForResultIntent = new Intent(MainActivity.this, MoveForResultActivity.class);
63.                    startActivityForResult(moveForResultIntent, REQUEST_CODE);
64.                    break;
65.                }
66.            }
67.            @Override
68.            protected void onActivityResult(int requestCode, int resultCode, Intent data) {
69.                super.onActivityResult(requestCode, resultCode, data);
70.                if (requestCode == REQUEST_CODE){
71.                    if (resultCode == MoveForResultActivity.RESULT_CODE){
72.                        int selectedValue = data.getIntExtra(MoveForResultActivit

```



```

y.EXTRA_SELECTED_VALUE, 0);
73.          tvResult.setText("Hasil : "+selectedValue);
74.      }
75.  }
76. }
77.}

```

- e) Setelah selesai, pastikan tidak ada bagian kode yang tertinggal. Jalankan kembali aplikasinya, dan coba Anda pilih tombol ‘Pindah Activity untuk Result’. Pilih angka yang Anda suka dan lihat hasilnya. Di **MainActivity.java** sudah ada angka yang tadi Anda pilih di obyek textview **tvResult**.



Gambar 64 Tampilan Akhir saat Dijalankan Aplikasi pada Intent Lanjutan

4. Tugas Praktikum

Buatlah aktiviti Intent, dengan ketentuan sebagai berikut:

Halaman Pertama : Nama Lengkap Kalian

Halaman Kedua : Nama, Nim, Alamat, Hobi, Cita-cita



File yang dikumpul, semua codingan baik .xml maupun .java di screen shot dan di pdf kan kemudian output di Videokan (.mp4) (misal: setelah di running klik bagian "nama lengkap kalian" lalu pindah ke halaman kedua berisi Nama, Nim, Alamat, Hobi, Cita-cita kemudian hasil pdf dan video tsb di upload di leads dengan format nama_nim)



Praktikum 7

Fragment dan RecyclerView Mode List

1. Sub Pokok Pembahasan

- a. Fragment
- b. RecyclerView Mode List

2. Capaian Pembelajaran

- a. Mahasiswa mampu menanamkan sikap jujur, bertanggung jawab, disiplin, saling menghargai dan berinovasi dengan tetap menjunjung tinggi nilai-nilai agama dan ketuhanan.
- b. Mahasiswa mampu menunjukkan sikap nasionalisme, mampu bekerja dalam tim sesuai kaidah keilmuan pada proyek pengembangan aplikasi mobile programming.
- c. Mahasiswa mampu berpartisipasi aktif, bertanggung jawab, dan memiliki motivasi mengembangkan diri dalam mendapatkan solusi permasalahan melalui aplikasi mobile programming
- d. Mahasiswa mampu menguasai, merancang dan mengembangkan bahasa pemrograman berbasis mobile, serta mampu menganalisis solusi dalam model bahasa pemrograman mobile.
- e. Mahasiswa mampu mengimplementasi pengetahuan dan teknologi menggunakan pendekatan *mobile programming*.

3. Fragment

Bagi kalian yang baru mengenal fragment, simpelnya Fragment itu bagian dari Activity yang memiliki siklus hidupnya sendiri, meskipun begitu Fragment berjalan di atas suatu Activity. Jika Activity-nya di destroy maka fragment juga ikut menghilang. Nah mungkin itu sekilas penjelasan tentang Fragment, lebih jelasnya langsung kita coba membuat Fragment sederhana.

Praktikum ini memiliki 4 Fragment dalam sebuah activity. Disini kita memiliki komponen atau view yang sama pada setiap layout fragment, sehingga kita tidak perlu membuat layout untuk masing-masing fragment, cukup satu saja.

- a) Tentunya mari memulai dengan membuat sebuah project baru, dalam contoh ini kita menggunakan tema **NoActionBar**. Pada **activity_main.xml**,



akan terdapat sebuah tombol yang berupa ImageView yang nantinya akan menampilkan **PopUpMenu** untuk navigasi antar fragment.

activity_main.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <LinearLayout
        android:id="@+id/layoutHeader"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/teal_700"
        android:gravity="center_vertical"
        android:orientation="horizontal"
        android:padding="10dp">

        <TextView
            android:id="@+id/textTitle"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_marginEnd="10dp"
            android:layout_weight="1"
            android:fontFamily="serif"
            android:text="Android Fragments"
            android:textColor="@color/white"
            android:textSize="25sp" />

        <ImageView
            android:id="@+id/imageMenu"
            android:layout_width="30dp"
            android:layout_height="30dp"
            android:background="?selectableItemBackground"
```



```

        android:contentDescription="@string/app_name"
        android:src="@drawable/ic_menu"
        app:tint="@color/white" />

    </LinearLayout>

    <FrameLayout
        android:id="@+id/layoutFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>

```

Pada layout diatas, kita memiliki sebuah **Framelayout** dengan id nya *layoutFragment*. Framelayout tersebut nantinya yang akan digantikan oleh fragment. Saat kita bernavigasi ke fragment 1, maka layout fragment 1 akan menggantikan **Framelayout** tersebut, begitupun untuk ketiga fragment lainnya.

- b) Baik, sekarang kita persiapkan sebuah layout untuk ke-empat fragment tersebut. Disini kita buat dengan nama **fragment_layout.xml**, yang mana berisi 3 buah komponen view : **TextView**, **ImageView**, dan **Button**. Seperti yang sudah dikatakan diatas, karena kita memiliki view yang sama pada semua fragment sehingga kita tidak perlu membuat layout baru untuk setiap fragment, cukup satu untuk semua.

Namun seandainya kita akan memiliki view yang berbeda atau view dengan jumlah yang berbeda pada setiap fragment, tentunya kita perlu membuat layout nya berbeda-beda juga, atau bisa saja dengan layout yang sama namun menambahkan view secara terprogram melalui file java class dari fragment. Tentunya akan lebih sedikit merepotkan daripada membuat layout berbeda.

fragment_layout.xml :

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"

```



```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="20dp">>

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"
        android:fontFamily="serif"
        android:gravity="center"
        android:text="Contoh Fragment"
        android:textColor="@color/white"
        android:textSize="25sp" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_marginBottom="20dp"
        android:contentDescription="@string/app_name" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="sans-serif-black"
        android:text="Button Click"
        android:textColor="@color/black"
        android:textSize="20sp" />

</LinearLayout>
```



Selanjutnya kita membuat 4 file java class yang memperluas (extends) kelas Fragment. Disini kita buat dengan nama: **Fragment1**, **Fragment2**, **Fragment3**, dan **Fragment4**. Pada sebuah activity kita menggunakan argumen *setContentView* untuk menautkan konteks dengan layout, namun pada fragment kita menggunakan object *View* dengan *LayoutInflater* untuk menautkan ke layout, sama seperti kelas adapter ataupun kelas lainnya jika ingin menginflate layout.

c) **Fragment1.java :**

```
package com.gwnbs.(NamaKalian);

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class Fragment1 extends Fragment {

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
        View view =
inflater.inflate(R.layout.fragment_layout, container, false);

        TextView textView =
view.findViewById(R.id.textView);
        Button button = view.findViewById(R.id.button);
        ImageView imageView =
view.findViewById(R.id.imageView);

        imageView.setImageResource(R.drawable.ic_1);
        textView.setText("Ini adalah fragment 1");
        button.setText("Click 1");

        button.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        Toast.makeText(requireActivity(), "Sekarang
Anda Berada di Halaman Fragment 1",
Toast.LENGTH_SHORT).show();
    }
});
```



```

    } );

    return view;
}
}

```

d) **Fragment2.java :**

```

package com.gwnbs.(NamaKalian);

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class Fragment2 extends Fragment {

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater
    inflater, @Nullable ViewGroup container,
                           @Nullable Bundle
    savedInstanceState) {
        View view =
        inflater.inflate(R.layout.fragment_layout, container,
        false);

        TextView textView =
        view.findViewById(R.id.textView);
        Button button = view.findViewById(R.id.button);
        ImageView imageView =
        view.findViewById(R.id.imageView);

        imageView.setImageResource(R.drawable.ic_2);
        textView.setText("Ini adalah fragment 2");
        button.setText("Click 2");

        button.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View view) {
                Toast.makeText(requireActivity(), " Sekarang
Anda Berada di Halaman Fragment 2",
Toast.LENGTH_SHORT).show();
            }
        });
    }

    return view;
}
}

```



e) **Fragment3.java :**

```
package com.gwnbs.(NamaKalian);

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class Fragment3 extends Fragment {

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,
                            @Nullable Bundle savedInstanceState) {
        View view =
inflater.inflate(R.layout.fragment_layout, container,
false);

        TextView textView =
view.findViewById(R.id.textView);
        Button button = view.findViewById(R.id.button);
        ImageView imageView =
view.findViewById(R.id.imageView);

        imageView.setImageResource(R.drawable.ic_3);
        textView.setText("Ini adalah fragment 3");
        button.setText("Click 3");

        button.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        Toast.makeText(requireActivity(), " Sekarang
Anda Berada di Halaman Fragment 3",
Toast.LENGTH_SHORT).show();
    }
});

        return view;
    }
}
```

f) **Fragment4.java :**

```
package com.gwnbs.(NamaKalian);

import android.os.Bundle;
```



```

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class Fragment4 extends Fragment {

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
        View view =
        inflater.inflate(R.layout.fragment_layout, container, false);

        TextView textView =
        view.findViewById(R.id.textView);
        Button button = view.findViewById(R.id.button);
        ImageView imageView =
        view.findViewById(R.id.imageView);

        imageView.setImageResource(R.drawable.ic_4);
        textView.setText("Ini adalah fragment 4");
        button.setText("Click 4");

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(requireActivity(), " Sekarang Anda Berada di Halaman Fragment 4", Toast.LENGTH_SHORT).show();
            }
        });

        return view;
    }
}

```

- g) Selanjutnya kita membuat file resource **menu**, pada contoh ini yaitu **menu_fragment.xml**, untuk di inflate nantinya menggunakan **PopUpMenu**, yang berisi 4 item untuk bernavigasi antar fragment. Caranya klik kanan pada folder **res > New > Android Resource File**, kemudian pada resource type pilih menu.

menu_fragment.xml :



```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/menuFragment1"
        android:title="Fragment 1"/>

    <item
        android:id="@+id/menuFragment2"
        android:title="Fragment 2"/>

    <item
        android:id="@+id/menuFragment3"
        android:title="Fragment 3"/>

    <item
        android:id="@+id/menuFragment4"
        android:title="Fragment 4"/>

</menu>

```

- h) Terakhir di **MainActivity.java**, pertama-tama kita buat sebuah metode private untuk berganti dari satu fragment ke fragment lainnya. Di dalam metode ini kita berikan 2 argumen yaitu Fragment dan String. Saat berganti fragment, disini kita juga merubah nilai string dari **TextView** pada actionBar, sehingga kita menggunakan argumen String supaya mudah. Kelas yang diperlukan untuk dapat berganti antar fragment yaitu *FragmentManager* dan *FragmentTransaction*.
- Kemudian didalam onCreate kita menentukan metode klik listener untuk tombol yang ada di actionBar, yang mana didalam metode klik tersebut kita memanggil **PopUpMenu**, lalu menentukan kondisi berdasarkan *id* dari item PopUpMenu tersebut. Di setiap kondisi kita memanggil metode private yang sudah dibuat sebelumnya dengan **Fragment** (Fragment1, Fragment2, Fragment3, dan Fragment4) dan nilai String yang berbeda.

MainActivity.java :

```

package com.gwnbs.(NamaKalian);

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

import android.os.Bundle;

```



```

import android.view.MenuItem;
import android.view.View;
import android.widget.PopupMenu;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    private TextView textTitle;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        textTitle = findViewById(R.id.textTitle);

        findViewById(R.id.imageMenu).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                PopupMenu popupMenu = new
PopupMenu(MainActivity.this, view);
                popupMenu.inflate(R.menu.menu_fragment);
                popupMenu.setOnMenuItemClickListener(new
PopupMenu.OnMenuItemClickListener() {
                    @Override
                    public boolean onMenuItemClick(MenuItem
menuItem) {
                        int id = menuItem.getItemId();
                        if (id == R.id.menuFragment1) {
                            replaceFragment(new Fragment1(),
"Fragment 1");
                        } else if (id == R.id.menuFragment2) {
                            replaceFragment(new Fragment2(),
"Fragment 2");
                        } else if (id == R.id.menuFragment3) {
                            replaceFragment(new Fragment3(),
"Fragment 3");
                        } else {
                            replaceFragment(new Fragment4(),
"Fragment 4");
                        }
                        return true;
                    }
                });
                popupMenu.show();
            }
        });
    }

    private void replaceFragment(Fragment fragment, String
title) {
        FragmentManager manager =
getSupportFragmentManager();
        FragmentTransaction transaction =
manager.beginTransaction();
        transaction.replace(R.id.layoutFragment, fragment);
    }
}

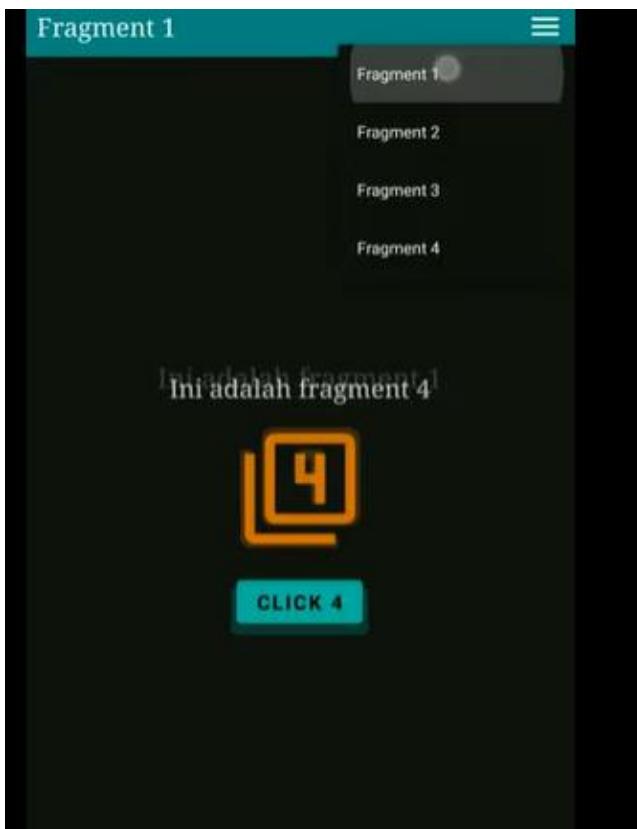
```



```
transaction.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_OPEN);
        transaction.commit();

        textTitle.setText(title);
    }
}
```

- i) Kemudian jalankan aplikasi tersebut, akan terlihat seperti di bawah ini



Gambar 65 Tampilan Akhir saat Aplikasi dijalankan Pada Fragment

4. Recycler View Mode List

Pada dasarnya interaksi umum antara pengguna dengan aplikasi dalam menampilkan data dengan jumlah yang banyak adalah dengan menggunakan list yang bisa di-scroll ke atas dan ke bawah. Nah, untuk membuat list tersebut, Anda dapat menggunakan komponen RecyclerView.

RecyclerView adalah sebuah komponen tampilan (widget) yang lebih canggih ketimbang pendahulunya, yakni **ListView**. Gambar di bawah menunjukkan perbedaan ListView dan RecyclerView, jika Anda menggunakan ListView

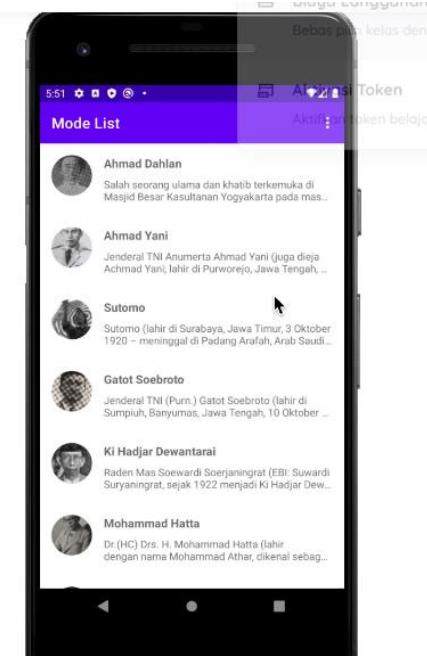


objek dari tiap item akan dibuat semuanya dari awal sampai akhir. Sedangkan jika Anda menggunakan RecyclerView, objek yang dibuat hanya sebatas ukuran layar dan beberapa di atas dan di bawahnya saja. Selanjutnya ia menggunakan kembali item yang sudah tidak terlihat.

Video Perbedaan antara ListView dengan RecyclerView bisa dilihat dari [link ini](#)

Pada codelab kali ini akan mempelajari cara menampilkan data Pahlawan ke dalam sebuah RecyclerView. Beberapa poin yang akan diulas dalam materi ini adalah :

1. Bagaimana menggunakan Recyclerview?
2. Recyclerview seperti *list*?



Gambar 66 Tampilan Akhir yang Nanti Dibuat

Codelab RecyclerView Mode List

- a) Buat Project baru di Android Studio dengan kriteria sebagai berikut:



Nama Project	MyRecyclerView
Target & Minimum Target SDK	Phone and Tablet, Api level 21
Tipe Activity	Empty Activity
Activity Name	MainActivity
Use AndroidX artifacts	True
Language	Java / Kotlin

- b) Setelah terbentuk, tambahkan beberapa dependensi yang akan kita gunakan pada berkas `build.gradle` (module: `app`) di bagian *dependencies* seperti berikut:

```

1. implementation 'androidx.recyclerview:recyclerview:1.1.0'
2. implementation 'androidx.cardview:cardview:1.0.0'
3. implementation 'de.hdodenhof:circleimageview:3.1.0'
4. implementation 'com.github.bumptech.glide:glide:4.11.0'

```

Sehingga secara keseluruhan berkas `build.gradle`(module: `app`) Anda akan seperti ini:

```

1. apply plugin: 'com.android.application'
2.
3. android {
4.     compileSdkVersion 30
5.     defaultConfig {
6.         applicationId "com.dicoding.myrecyclerview"
7.         minSdkVersion 19
8.         targetSdkVersion 30
9.         versionCode 1
10.        versionName "1.0"
11.        testInstrumentationRunner
12.            "androidx.test.runner.AndroidJUnitRunner"
13.        }
14.        buildTypes {
15.            release {
16.                minifyEnabled false
17.                proguardFiles getDefaultProguardFile('proguard-
18.                    android-optimize.txt'), 'proguard-rules.pro'
19.            }
20.
21.            dependencies {
22.                implementation 'androidx.appcompat:appcompat:1.2.0'

```

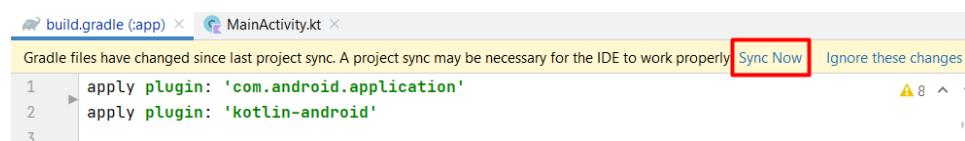


```

23.     implementation
        'com.google.android.material:material:1.2.1'
24.     implementation
        'androidx.constraintlayout:constraintlayout:2.0.4'
25.     implementation
        'androidx.recyclerview:recyclerview:1.1.0'
26.     implementation 'androidx.cardview:cardview:1.0.0'
27.     implementation 'de.hdodenhof:circleimageview:3.1.0'
28.     implementation
        'com.github.bumptech.glide:glide:4.11.0'
29.     androidTestImplementation
        'androidx.test:runner:1.3.0'
30.     androidTestImplementation
        'androidx.test.espresso:espresso-core:3.3.0'
31. }

```

Setelah itu klik **Sync Now** untuk mengunduh *library* tersebut:



Gambar 67 Tampilan untuk Klik Sync Now

Jika menggunakan Kotlin, maka ditambahkan juga 'kotlin-android' di `build.gradle(module: app)`

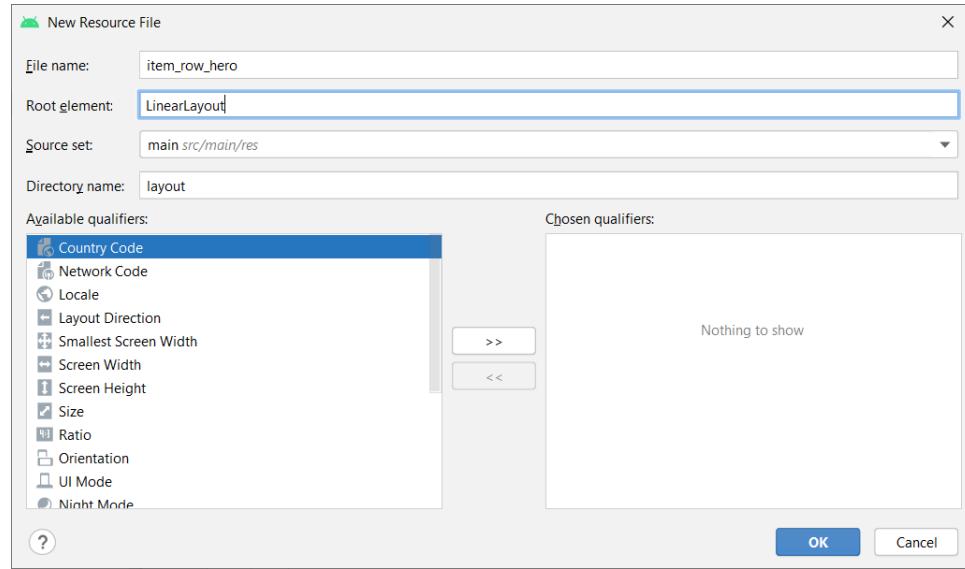
- c) Selanjutnya pada `activity_main.xml` lengkapi kodenya menjadi seperti berikut:

```

1.  <?xml version="1.0" encoding="utf-8"?>
2.  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.      xmlns:tools="http://schemas.android.com/tools"
4.      android:id="@+id/activity_main"
5.      android:layout_width="match_parent"
6.      android:layout_height="match_parent">
7.
8.      <androidx.recyclerview.widget.RecyclerView
9.          android:id="@+id/rv_heroes"
10.         android:layout_width="match_parent"
11.         android:layout_height="match_parent"
12.         tools:listitem="@layout/item_row_hero" />
13. 
```



- d) Saatnya kita membuat sebuah item tampilan dalam bentuk berkas layout xml yang akan ditampilkan di RecyclerView. Karena data pertama kali akan ditampilkan dalam bentuk list, maka kita buat layout dengan cara klik kanan pada direktori **layout** → **new** → **layout resource file** dan kemudian beri nama **item_row_hero**. Jangan lupa untuk mengubah **RootElement** menjadi **LinearLayout**.



Gambar 68 Tampilan untuk Membuat Layout Resource File

- e) Setelah terbentuk lengkapi tampilan tersebut menjadi seperti ini:

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:tools="http://schemas.android.com/tools"
4.     android:layout_width="match_parent"
5.     android:layout_height="wrap_content"
6.     android:orientation="horizontal"
7.     android:padding="16dp">
8.
9.     <de.hdodenhof.circleimageview.CircleImageView
10.        android:id="@+id/img_item_photo"
11.        android:layout_width="55dp"
12.        android:layout_height="55dp"
13.        android:layout_marginEnd="16dp"
14.        tools:src="@tools:sample/avatars" />
15.

```



```

16.      <LinearLayout
17.          android:layout_width="match_parent"
18.          android:layout_height="wrap_content"
19.          android:orientation="vertical">
20.
21.          <TextView
22.              android:id="@+id/tv_item_name"
23.              android:layout_width="match_parent"
24.              android:layout_height="wrap_content"
25.              android:layout_marginBottom="8dp"
26.              android:textSize="16sp"
27.              android:textStyle="bold"
28.              tools:text="@string/heroes_name" />
29.
30.          <TextView
31.              android:id="@+id/tv_item_detail"
32.              android:layout_width="match_parent"
33.              android:layout_height="wrap_content"
34.              android:ellipsize="end"
35.              android:maxLines="2"
36.              tools:text="@string/detail" />
37.      </LinearLayout>
38.  </LinearLayout>

```

tools:text bisa kita gunakan untuk placeholder di dalam editor layout. Atribut ini tidak akan terbawa saat run-time dan hanya akan tampak di dalam preview editor saja.

Catatan:

Pastikan untuk ukuran **android:layout_height** menggunakan **wrap_content** semua, supaya ketika dijalankan tidak ada layout kosong / space lebar karena menggunakan **match_parent**, hal ini karena 1 item nya dihitung 1 layar.



Akan ada yang eror pada bagian @string/heroes_name dan @string/detail. Seperti pada modul sebelumnya, Anda perlu menambahkan data String pada bagian **res → values → strings.xml**.

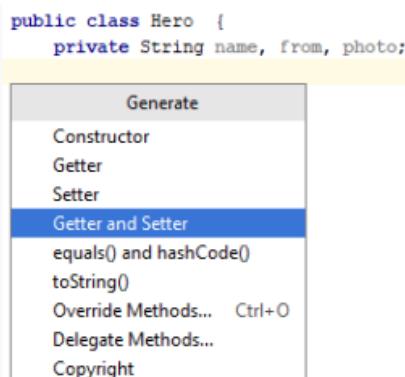
```
1. <resources>
2.   <string name="app_name">My Application</string>
3.   <string name="heroes_name">Nama Pahlawan</string>
4.   <string name="detail">Detail</string>
5. </resources>
```

- f) Kemudian buat sebuah kelas model data bernama Hero dengan cara **klik kanan pada nama package → new → Java/Kotlin class**.

Java

```
1. public class Hero {
2.   private String name;
3.   private String detail;
4.   private int photo;
5. }
```

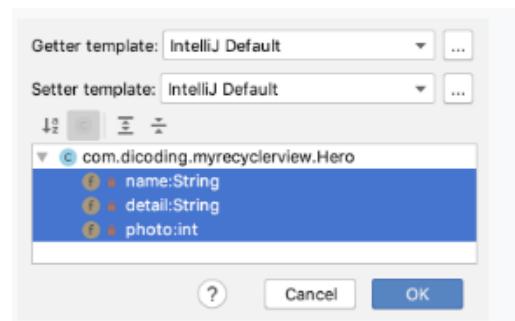
Setelah itu, kita akan buat setter getter, seperti sebelumnya. Bisa kita lakukan dengan **Alt+Insert** atau **Klik Kanan Generate→ Getter and Setter**.



Gambar 69 Tampilan Untuk Memilih Getter and Setter di Java



Kemudian pilih semua variabel yang sudah dibuat.



Gambar 70 Tampilan setelah Klik Getter and Setter

Maka kode **Hero** akan menjadi seperti ini:

```
1. public class Hero {  
2.     private String name;  
3.     private String detail;  
4.     private int photo;  
5.  
6.     public String getName() {  
7.         return name;  
8.     }  
9.  
10.    public void setName(String name) {  
11.        this.name = name;  
12.    }  
13.  
14.    public String getDetail() {  
15.        return detail;
```



```

16. }
17.
18. public void setDetail(String detail) {
19.     this.detail = detail;
20. }
21.
22. public int getPhoto() {
23.     return photo;
24. }
25.
26. public void setPhoto(int photo) {
27.     this.photo = photo;
28. }
29. }
```

Kotlin:

```

1. data class Hero(
2.     var name: String = "",
3.     var detail: String = "",
4.     var photo: Int = 0
5. )
```

Catatan:

Untuk bahasa Kotlin, sudah bisa memanfaatkan constructor, getter, dan setter secara otomatis dari sebuah data classes. Jadi tidak perlu menambahkan kode apapun lagi.

- g) Karena di latihan ini akan menampilkan foto pahlawan, maka perlu mengunduh berkas fotonya [di sini](#). Setelah selesai *extract* berkas tersebut dan *copy paste* semua berkasnya ke folder ***drawable***.

Catatan:

Ingat bahwa nama file yang ada di dalam resource drawable harus menggunakan huruf kecil dan underscore saja.

- h) Setelah selesai kita akan membuat beberapa kelas terkait koleksi data yang ingin ditampilkan.

Java

Buat class baru dengan nama **HeroesData** dengan cara yang **klik kanan pada nama package → new → Java class**.



Setelah itu kita bisa menambahkan kode menjadi seperti berikut:

```
1. public class HeroesData {  
2.     private static String[] heroNames = {  
3.         "Ahmad Dahlan",  
4.         "Ahmad Yani",  
5.         "Sutomo",  
6.         "Gatot Soebroto",  
7.         "Ki Hadjar Dewantara",  
8.         "Mohammad Hatta",  
9.         "Soedirman",  
10.        "Soekarno",  
11.        "Soepomo",  
12.        "Tan Malaka"  
13.    };  
14.  
15.    private static String[] heroDetails = {  
16.        "Salah seorang ulama dan khatib terkemuka di Masjid  
            Besar Kasultanan Yogyakarta pada masa itu, dan ibu dari  
            K.H. Ahmad Dahlan adalah puteri dari H. Ibrahim yang  
            juga menjabat penghulu Kesultanan Ngayogyakarta  
            Hadiningrat pada masa itu. KH. Ahmad Dahlan telah  
            mempelopori kebangkitan ummat Islam untuk menyadari  
            nasibnya sebagai bangsa terjajah yang masih harus  
            belajar dan berbuat.",  
17.        "Jenderal TNI Anumerta Ahmad Yani (juga dieja Achmad  
            Yani; lahir di Purworejo, Jawa Tengah, 19 Juni 1922 –  
            meninggal di Lubang Buaya, Jakarta, 1 Oktober 1965 pada  
            umur 43 tahun) adalah komandan Tentara Nasional  
            Indonesia Angkatan Darat, dan dibunuh oleh anggota  
            Gerakan 30 September saat mencoba untuk menculik dia  
            dari rumahnya.",  
18.        "Sutomo (lahir di Surabaya, Jawa Timur, 3 Oktober  
            1920 – meninggal di Padang Arafah, Arab Saudi, 7  
            Oktober 1981 pada umur 61 tahun) lebih dikenal dengan  
            sapaan akrab oleh rakyat sebagai Bung Tomo, adalah  
            pahlawan yang terkenal karena peranannya dalam  
            membangkitkan semangat rakyat untuk melawan kembalinya  
            penjajah Belanda melalui tentara NICA, yang berakhir  
            dengan pertempuran 10 November 1945 yang hingga kini  
            diperingati sebagai Hari Pahlawan.",  
19.        "Jenderal TNI (Purn.) Gatot Soebroto (lahir di  
            Sumpiuh, Banyumas, Jawa Tengah, 10 Oktober 1907 –  
            meninggal di Jakarta, 11 Juni 1962 pada umur 54 tahun)  
            adalah tokoh perjuangan militer Indonesia dalam merebut  
            kemerdekaan dan juga pahlawan nasional Indonesia. Ia  
            dimakamkan di Ungaran, kabupaten Semarang.",  
20.        "Raden Mas Soewardi Soerjaningrat (EBI: Suwardi  
            Suryaningrat, sejak 1922 menjadi Ki Hadjar Dewantara,  
            EBI: Ki Hajjar Dewantara, beberapa menuliskan bunyi  
            bahasa Jawanya dengan Ki Hajjar Dewantoro; lahir di  
            Pakualaman, 2 Mei 1889 – meninggal di Yogyakarta, 26
```



April 1959 pada umur 69 tahun; selanjutnya disingkat sebagai \"Soewardi\" atau \"KHD\") adalah aktivis pergerakan kemerdekaan Indonesia, kolumnis, politisi, dan pelopor pendidikan bagi kaum pribumi Indonesia dari zaman penjajahan Belanda. Ia adalah pendiri Perguruan Taman Siswa, suatu lembaga pendidikan yang memberikan kesempatan bagi para pribumi untuk bisa memperoleh hak pendidikan seperti halnya para priyayi maupun orang-orang Belanda.",

21. "Dr.(HC) Drs. H. Mohammad Hatta (lahir dengan nama Mohammad Athar, dikenal sebagai Bung Hatta; lahir di Fort de Kock (sekarang Bukittinggi, Sumatera Barat), Hindia Belanda, 12 Agustus 1902 – meninggal di Jakarta, 14 Maret 1980 pada umur 77 tahun) adalah tokoh pejuang, negarawan, ekonom, dan juga Wakil Presiden Indonesia yang pertama. Ia bersama Soekarno memainkan peranan penting untuk memerdekakan bangsa Indonesia dari penjajahan Belanda sekaligus memproklamirkannya pada 17 Agustus 1945. Ia juga pernah menjabat sebagai Perdana Menteri dalam Kabinet Hatta I, Hatta II, dan RIS. Ia mundur dari jabatan wakil presiden pada tahun 1956, karena berselisih dengan Presiden Soekarno. Hatta juga dikenal sebagai Bapak Koperasi Indonesia.",
22. "Jenderal Besar Raden Soedirman (EYD: Sudirman; lahir 24 Januari 1916 – meninggal 29 Januari 1950 pada umur 34 tahun) adalah seorang perwira tinggi Indonesia pada masa Revolusi Nasional Indonesia. Menjadi panglima besar Tentara Nasional Indonesia pertama, ia secara luas terus dihormati di Indonesia. Terlahir dari pasangan rakyat biasa di Purbalingga, Hindia Belanda, Soedirman diadopsi oleh pamannya yang seorang priyayi. Setelah keluarganya pindah ke Cilacap pada tahun 1916, Soedirman tumbuh menjadi seorang siswa rajin; ia sangat aktif dalam kegiatan ekstrakurikuler, termasuk mengikuti program kepanduan yang dijalankan oleh organisasi Islam Muhammadiyah. Saat di sekolah menengah, Soedirman mulai menunjukkan kemampuannya dalam memimpin dan berorganisasi, dan dihormati oleh masyarakat karena ketaatannya pada Islam. Setelah berhenti kuliah keguruan, pada 1936 ia mulai bekerja sebagai seorang guru, dan kemudian menjadi kepala sekolah, di sekolah dasar Muhammadiyah; ia juga aktif dalam kegiatan Muhammadiyah lainnya dan menjadi pemimpin Kelompok Pemuda Muhammadiyah pada tahun 1937. Setelah Jepang menduduki Hindia Belanda pada 1942, Soedirman tetap mengajar. Pada tahun 1944, ia bergabung dengan tentara Pembela Tanah Air (PETA) yang disponsori Jepang, menjabat sebagai komandan batalion di Banyumas. Selama menjabat, Soedirman bersama rekannya sesama prajurit melakukan pemberontakan, namun kemudian diasangkan ke Bogor.",



```

23. "Dr.(H.C.) Ir. H. Soekarno (ER, EYD: Sukarno, nama
    lahir: Koesno Sosrodihardjo) (lahir di Surabaya, Jawa
    Timur, 6 Juni 1901 – meninggal di Jakarta, 21 Juni 1970
    pada umur 69 tahun) adalah Presiden pertama Republik
    Indonesia yang menjabat pada periode 1945–1967. Ia
    memainkan peranan penting dalam memerdekakan bangsa
    Indonesia dari penjajahan Belanda. Ia adalah
    Proklamator Kemerdekaan Indonesia (bersama dengan
    Mohammad Hatta) yang terjadi pada tanggal 17 Agustus
    1945. Soekarno adalah yang pertama kali mencetuskan
    konsep mengenai Pancasila sebagai dasar negara
    Indonesia dan ia sendiri yang menamainya.",

24. "Prof. Mr. Dr. Soepomo (Ejaan Soewandi: Supomo; lahir
    di Sukoharjo, Jawa Tengah, 22 Januari 1903 – meninggal
    di Jakarta, 12 September 1958 pada umur 55 tahun)
    adalah seorang pahlawan nasional Indonesia. Soepomo
    dikenal sebagai arsitek Undang-undang Dasar 1945,
    bersama dengan Muhammad Yamin dan Soekarno.",

25. "Tan Malaka atau Ibrahim gelar Datuk Sultan Malaka
    (lahir di Nagari Pandam Gadang, Suliki, Lima Puluh
    Kota, Sumatera Barat, 2 Juni 1897 – meninggal di Desa
    Selopanggung, Kediri, Jawa Timur, 21 Februari 1949 pada
    umur 51 tahun) adalah seorang pembela kemerdekaan
    Indonesia, tokoh Partai Komunis Indonesia, juga pendiri
    Partai Murba, dan merupakan salah satu Pahlawan
    Nasional Indonesia."
26.    };
27.
28.    private static int[] heroesImages = {
29.        R.drawable.ahmad_dahlan,
30.        R.drawable.ahmad_yani,
31.        R.drawable.bung_tomo,
32.        R.drawable.gatot_subroto,
33.        R.drawable.ki_hadjar_dewantara,
34.        R.drawable.mohammad_hatta,
35.        R.drawable.sudirman,
36.        R.drawable.sukarno,
37.        R.drawable.supomo,
38.        R.drawable.tan_malaka
39.    };
40.
41.    static ArrayList<Hero> getListData() {
42.        ArrayList<Hero> list = new ArrayList<>();
43.        for (int position = 0; position <
    heroNames.length; position++) {
44.            Hero hero = new Hero();
45.            hero.setName(heroNames[position]);
46.            hero.setDetail(heroDetails[position]);
47.            hero.setPhoto(heroesImages[position]);
48.            list.add(hero);
49.        }
50.        return list;

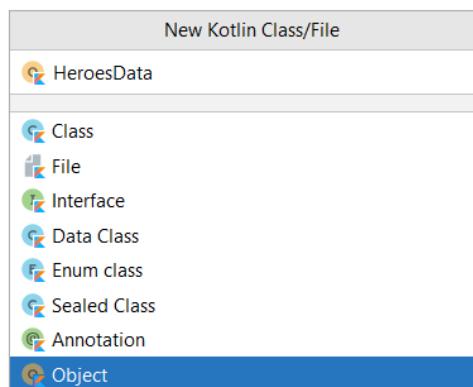
```



```
51.     }
}
```

Kotlin

Buat object baru dengan nama HeroesData dengan cara yang **klik kanan pada nama package → new → Kotlin Class/File.**



Gambar 71 Tampilan untuk Membuat Object di Kotlin

Setelah itu kita bisa menambahkan kode menjadi seperti berikut:

```
1. object HeroesData {
2.     private val heroNames = arrayOf("Ahmad Dahlan",
3.         "Ahmad Yani",
4.         "Sutomo",
5.         "Gatot Soebroto",
6.         "Ki Hadjar Dewantara",
7.         "Mohammad Hatta",
8.         "Soedirman",
9.         "Soekarno",
10.        "Soepomo",
11.        "Tan Malaka")
12.
13.     private val heroDetails = arrayOf("Salah seorang
        ulama dan khatib terkemuka di Masjid Besar Kasultanan
        Yogyakarta pada masa itu, dan ibu dari K.H. Ahmad Dahlan
        adalah puteri dari H. Ibrahim yang juga menjabat
        penghulu Kesultanan Ngayogyakarta Hadiningrat pada masa
        itu. KH. Ahmad Dahlan telah mempelopori kebangkitan
        ummat Islam untuk menyadari nasibnya sebagai bangsa
        terjajah yang masih harus belajar dan berbuat.",
14.        "Jenderal TNI Anumerta Ahmad Yani (juga dieja Achmad
        Yani; lahir di Purworejo, Jawa Tengah, 19 Juni 1922 –
        meninggal di Lubang Buaya, Jakarta, 1 Oktober 1965 pada
        umur 43 tahun) adalah komandan Tentara Nasional
        Indonesia Angkatan Darat, dan dibunuh oleh anggota
        Gerakan 30 September saat mencoba untuk menculik dia
        dari rumahnya.",
15.        "Sutomo (lahir di Surabaya, Jawa Timur, 3 Oktober
        1920 – meninggal di Padang Arafah, Arab Saudi, 7 Oktober
```



1981 pada umur 61 tahun) lebih dikenal dengan sapaan akrab oleh rakyat sebagai Bung Tomo, adalah pahlawan yang terkenal karena peranannya dalam membangkitkan semangat rakyat untuk melawan kembalinya penjajah Belanda melalui tentara NICA, yang berakhir dengan pertempuran 10 November 1945 yang hingga kini diperingati sebagai Hari Pahlawan.",

16. "Jenderal TNI (Purn.) Gatot Soebroto (lahir di Sumpiuh, Banyumas, Jawa Tengah, 10 Oktober 1907 – meninggal di Jakarta, 11 Juni 1962 pada umur 54 tahun) adalah tokoh perjuangan militer Indonesia dalam merebut kemerdekaan dan juga pahlawan nasional Indonesia. Ia dimakamkan di Ungaran, kabupaten Semarang.",
17. "Raden Mas Soewardi Soerjaningrat (EBI: Suwardi Suryaningrat, sejak 1922 menjadi Ki Hadjar Dewantara, EBI: Ki Hajar Dewantara, beberapa menuliskan bunyi bahasa Jawanya dengan Ki Hajar Dewantoro; lahir di Pakualaman, 2 Mei 1889 – meninggal di Yogyakarta, 26 April 1959 pada umur 69 tahun; selanjutnya disingkat sebagai \"Soewardi\" atau \"KHD\") adalah aktivis pergerakan kemerdekaan Indonesia, kolumnis, politisi, dan pelopor pendidikan bagi kaum pribumi Indonesia dari zaman penjajahan Belanda. Ia adalah pendiri Perguruan Taman Siswa, suatu lembaga pendidikan yang memberikan kesempatan bagi para pribumi untuk bisa memperoleh hak pendidikan seperti halnya para priyayi maupun orang-orang Belanda.",
18. "Dr.(HC) Drs. H. Mohammad Hatta (lahir dengan nama Mohammad Athar, dikenal sebagai Bung Hatta; lahir di Fort de Kock (sekarang Bukittinggi, Sumatera Barat), Hindia Belanda, 12 Agustus 1902 – meninggal di Jakarta, 14 Maret 1980 pada umur 77 tahun) adalah tokoh pejuang, negarawan, ekonom, dan juga Wakil Presiden Indonesia yang pertama. Ia bersama Soekarno memainkan peranan penting untuk memerdekakan bangsa Indonesia dari penjajahan Belanda sekaligus memproklamirkannya pada 17 Agustus 1945. Ia juga pernah menjabat sebagai Perdana Menteri dalam Kabinet Hatta I, Hatta II, dan RIS. Ia mundur dari jabatan wakil presiden pada tahun 1956, karena berselisih dengan Presiden Soekarno. Hatta juga dikenal sebagai Bapak Koperasi Indonesia.",
19. "Jenderal Besar Raden Soedirman (EYD: Sudirman; lahir 24 Januari 1916 – meninggal 29 Januari 1950 pada umur 34 tahun) adalah seorang perwira tinggi Indonesia pada masa Revolusi Nasional Indonesia. Menjadi panglima besar Tentara Nasional Indonesia pertama, ia secara luas terus dihormati di Indonesia. Terlahir dari pasangan rakyat biasa di Purbalingga, Hindia Belanda, Soedirman diadopsi oleh pamannya yang seorang priyayi. Setelah keluarganya pindah ke Cilacap pada tahun 1916, Soedirman tumbuh menjadi seorang siswa rajin; ia sangat aktif dalam kegiatan ekstrakurikuler, termasuk mengikuti program



kepanduan yang dijalankan oleh organisasi Islam Muhammadiyah. Saat di sekolah menengah, Soedirman mulai menunjukkan kemampuannya dalam memimpin dan berorganisasi, dan dihormati oleh masyarakat karena ketaatannya pada Islam. Setelah berhenti kuliah keguruan, pada 1936 ia mulai bekerja sebagai seorang guru, dan kemudian menjadi kepala sekolah, di sekolah dasar Muhammadiyah; ia juga aktif dalam kegiatan Muhammadiyah lainnya dan menjadi pemimpin Kelompok Pemuda Muhammadiyah pada tahun 1937. Setelah Jepang menduduki Hindia Belanda pada 1942, Soedirman tetap mengajar. Pada tahun 1944, ia bergabung dengan tentara Pembela Tanah Air (PETA) yang disponsori Jepang, menjabat sebagai komandan batalion di Banyumas. Selama menjabat, Soedirman bersama rekannya sesama prajurit melakukan pemberontakan, namun kemudian diasingkan ke Bogor.",

20. "Dr.(H.C.) Ir. H. Soekarno (ER, EYD: Sukarno, nama lahir: Koesno Sosrodihardjo) (lahir di Surabaya, Jawa Timur, 6 Juni 1901 - meninggal di Jakarta, 21 Juni 1970 pada umur 69 tahun) adalah Presiden pertama Republik Indonesia yang menjabat pada periode 1945-1967. Ia memainkan peranan penting dalam memerdekakan bangsa Indonesia dari penjajahan Belanda. Ia adalah Proklamator Kemerdekaan Indonesia (bersama dengan Mohammad Hatta) yang terjadi pada tanggal 17 Agustus 1945. Soekarno adalah yang pertama kali mencetuskan konsep mengenai Pancasila sebagai dasar negara Indonesia dan ia sendiri yang menamainya.",
21. "Prof. Mr. Dr. Soepomo (Ejaan Soewandi: Supomo; lahir di Sukoharjo, Jawa Tengah, 22 Januari 1903 - meninggal di Jakarta, 12 September 1958 pada umur 55 tahun) adalah seorang pahlawan nasional Indonesia. Soepomo dikenal sebagai arsitek Undang-undang Dasar 1945, bersama dengan Muhammad Yamin dan Soekarno.",
22. "Tan Malaka atau Ibrahim gelar Datuk Sutan Malaka (lahir di Nagari Pandam Gadang, Suliki, Lima Puluh Kota, Sumatera Barat, 2 Juni 1897 - meninggal di Desa Selopanggung, Kediri, Jawa Timur, 21 Februari 1949 pada umur 51 tahun) adalah seorang pembela kemerdekaan Indonesia, tokoh Partai Komunis Indonesia, juga pendiri Partai Murba, dan merupakan salah satu Pahlawan Nasional Indonesia.")
- 23.
24.

```
private val heroesImages =  
intArrayOf(R.drawable.ahmad_dahlan,
```
25.

```
R.drawable.ahmad_yani,
```
26.

```
R.drawable.bung_tomo,
```
27.

```
R.drawable.gatot_subroto,
```
28.

```
R.drawable.ki_hadjar_dewantara,
```
29.

```
R.drawable.mohammad_hatta,
```
30.

```
R.drawable.sudirman,
```



```

31.     R.drawable.sukarno,
32.     R.drawable.supomo,
33.     R.drawable.tan_malaka)
34.
35.     val listData: ArrayList<Hero>
36.     get() {
37.         val list = arrayListOf<Hero>()
38.         for (position in heroNames.indices) {
39.             val hero = Hero()
40.             hero.name = heroNames[position]
41.             hero.detail = heroDetails[position]
42.             hero.photo = heroesImages[position]
43.             list.add(hero)
44.         }
45.     return list
46. }
47. }
```

- i) Sekarang akan dibuat sebuah adapter yang akan memformat bagaimana tiap elemen dari koleksi data ditampilkan. Buat kelas adapter secara manual dengan klik kanan pada **package utama** → **new** → **Kotlin/Java Class** dan beri nama **ListHeroAdapter**.

Setelah kelas adapter berhasil dibuat, lengkapi kodennya menjadi seperti berikut:

Java

```

1. public class ListHeroAdapter extends
   RecyclerView.Adapter<ListHeroAdapter.ListViewHolder> {
2.
3. }
```

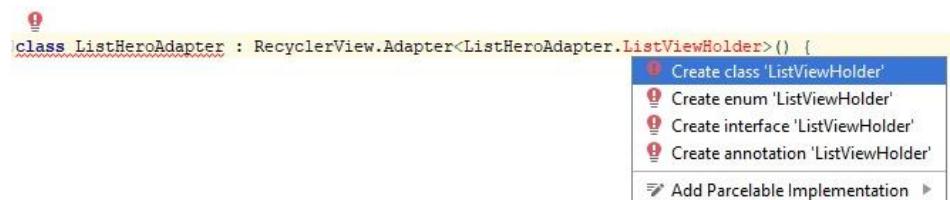
Kotlin

```

1. class ListHeroAdapter
   : RecyclerView.Adapter<ListHeroAdapter.ListViewHolder>{
2.
3. }
```

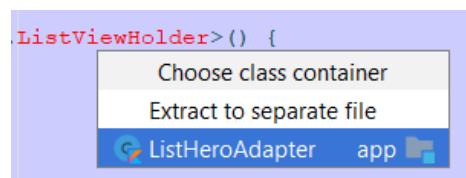
Maka akan ada garis merah dan **ListViewHolder** juga akan merah. Kita *resolve* satu-persatu. Pertama adalah tekan **alt+enter/klik tombol merah** pada **ListViewHolder** dan pilih **Create class 'ListViewHolder'**:





Gambar 72 Tampilan di ViewHolder saat ada Garis Merah

Kemudian pilih **ListHeroAdapter** untuk membuat kelas tersebut di dalam ListHeroAdapter.



Gambar 73 Tampilan di ListHeroAdapter

Maka kode saat ini menjadi seperti ini:

Java

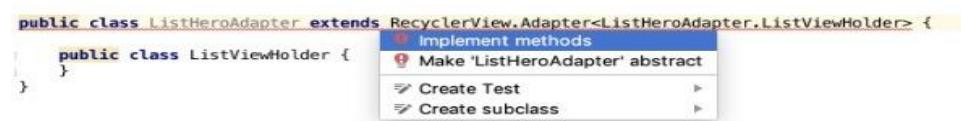
```
1. public class ListHeroAdapter extends
   RecyclerView.Adapter<ListHeroAdapter.ListViewHolder> {
2.     public class ListViewHolder {
3.     }
4. }
```

Kotlin

```
1. class ListHeroAdapter :
   RecyclerView.Adapter<ListHeroAdapter.ListViewHolder>{
2.     inner class ListViewHolder()
3. }
```

Masih ada garis merah, arahkan kurSOR ke teks class **ListHeroAdapter** kita lakukan lagi alt+enter / klik tombol merah dan pilih **implement members (Kotlin)** atau **implement methods(Java)**.

Java



Gambar 74 Tampilan di Adapter saat ada Garis Merah dengan Java

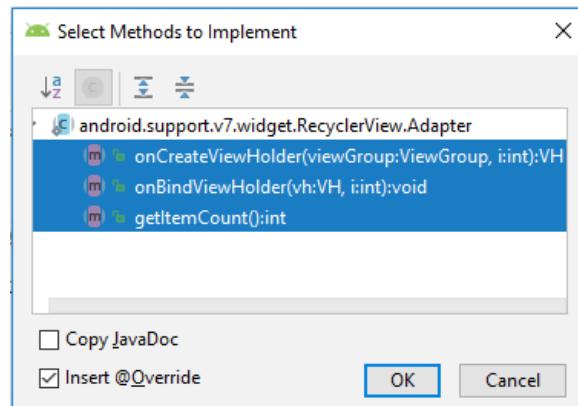


Kotlin



Gambar 75 Tampilan di Adapter saat ada Garis Merah dengan Kotlin

Kemudian pilih ketiga metode dari `RecyclerView.Adapter`.



Gambar 76 Tampilan dari RecyclerViewAdapter

Maka kode saat ini menjadi seperti ini:

Java

```
1. public class ListHeroAdapter extends
   RecyclerView.Adapter<ListHeroAdapter.ListViewHolder> {
2.
3.     @NonNull
4.     @Override
5.     public ListViewHolder onCreateViewHolder(@NonNull
   ViewGroup parent, int viewType) {
6.         return null;
7.     }
8.
9.     @Override
10.    public void onBindViewHolder(@NonNull ListViewHolder
   listViewHolder, int position) {
11.
12.    }
13.
14.    @Override
15.    public int getItemCount() {
16.        return 0;
17.    }
18.
19.    public class ListViewHolder {
```



```
20.        }
}
```

Kotlin

```
1. class ListHeroAdapter :
    RecyclerView.Adapter<ListHeroAdapter.ListViewHolder>() {
2.
3.     override fun onCreateViewHolder(parent: ViewGroup,
    viewType: Int): ListViewHolder {
4.         TODO("not implemented") //To change body of
    created functions use File | Settings | File Templates.
5.     }
6.
7.     override fun onBindViewHolder(holder:
    ListViewHolder, position: Int) {
8.         TODO("not implemented") //To change body of
    created functions use File | Settings | File Templates.
9.     }
10.
11.    override fun getItemCount(): Int {
12.        TODO("not implemented") //To change body of
    created functions use File | Settings | File Templates.
13.    }
14.
15.    inner class ListViewHolder(
16.    }
```

Masih ada garis merah, hal ini karena **ListViewHolder** seharusnya inherit ke RecyclerView. ViewHolder, maka ubah kodennya menjadi seperti ini :

Java

```
1. public class ListHeroAdapter extends
    RecyclerView.Adapter<ListHeroAdapter.ListViewHolder> {
2.     ...
3.
4.     public class ListViewHolder extends
    RecyclerView.ViewHolder {
5.     }
6. }
```

Kotlin

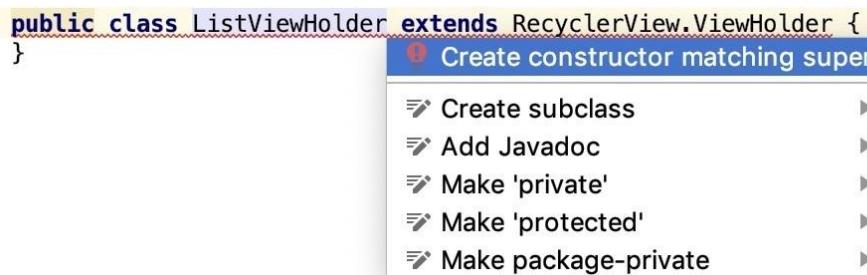
```
1. class ListHeroAdapter :
    RecyclerView.Adapter<ListHeroAdapter.ListViewHolder>() {
2.     ...
3.
4.     inner class ListViewHolder() :
    RecyclerView.ViewHolder {
```



```
5.      }
6. }
```

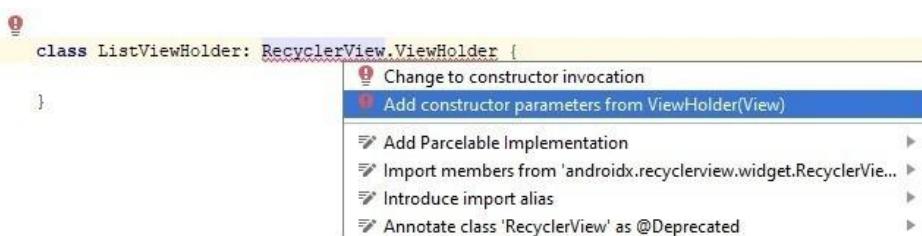
- j) Masih ada garis merah kita lakukan lagi **alt+enter** / **klik tombol merah** dan pilih **Create constructor matching super(Java)** atau **Add constructor parameter from ViewHolder(View)**.

Java



Gambar 77 Tampilan ViewHolder saat ada Garis merah di Java

Kotlin



Gambar 78 Tampilan ViewHolder saat ada Garis merah di Kotlin

- k) Maka kode saat ini menjadi seperti ini:

Java

```
1. public class ListHeroAdapter extends
   RecyclerView.Adapter<ListHeroAdapter.ListViewHolder> {
2.     ...
3.
4.     class ListViewHolder extends
   RecyclerView.ViewHolder {
5.         public ListViewHolder(@NonNull View itemView) {
6.             super(itemView);
7.         }
8.     }
9. }
```

Kotlin

```
1. class ListHeroAdapter :
   RecyclerView.Adapter<ListHeroAdapter.ListViewHolder>()
```



```

2. ...
3.
4.     inner class ListViewHolder(itemView: View) :
    RecyclerView.ViewHolder(itemView) {
5.
6. }
7. }
```

Kemudian isi ListViewHolder dengan kode berikut untuk inisialisasi view yang ada di dalam layout item:

Java

```

1. public class ListHeroAdapter extends
    RecyclerView.Adapter<ListHeroAdapter.ListViewHolder> {
2. ...
3.
4.     class ListViewHolder extends RecyclerView.ViewHolder
    {
5.         ImageView imgPhoto;
6.         TextView tvName, tvDetail;
7.
8.         ListViewHolder(View itemView) {
9.             super(itemView);
10.            imgPhoto =
11.                itemView.findViewById(R.id.img_item_photo);
12.            tvName =
13.                itemView.findViewById(R.id.tv_item_name);
14.            tvDetail =
15.                itemView.findViewById(R.id.tv_item_detail);
16.        }
17.    }
18. }
```

Kotlin

```

1. class ListHeroAdapter :
    RecyclerView.Adapter<ListHeroAdapter.ListViewHolder>() {
2.
3. ...
4.     inner class ListViewHolder(itemView: View) :
    RecyclerView.ViewHolder(itemView) {
5.         var tvName: TextView =
    itemView.findViewById(R.id.tv_item_name)
6.         var tvDetail: TextView =
    itemView.findViewById(R.id.tv_item_detail)
7.         var imgPhoto: ImageView =
    itemView.findViewById(R.id.img_item_photo)
8.     }
```



```
9. }
```

Kemudian buatlah dan *constructor* untuk *list* seperti ini:

Java

```
1. public class ListHeroAdapter extends
   RecyclerView.Adapter<ListHeroAdapter.ListViewHolder> {
2.     private ArrayList<Hero> listHero;
3.
4.     public ListHeroAdapter(ArrayList<Hero> list) {
5.         this.listHero = list;
6.     }
7.
8.     ...
9. }
```

Kotlin

```
1. class ListHeroAdapter(private val listHero:
   ArrayList<Hero>) :
   RecyclerView.Adapter<ListHeroAdapter.ListViewHolder>() {
2. ...
3. }
```

Setelah itu kita bisa melengkapi kode-kode hasil dari turunan `RecyclerView.Adapter`. Maka kode lengkap dari kelas `ListHeroAdapter` sebagai berikut:

Java

```
1. public class ListHeroAdapter extends
   RecyclerView.Adapter<ListHeroAdapter.ListViewHolder> {
2.     private ArrayList<Hero> listHero;
3.
4.     public ListHeroAdapter(ArrayList<Hero> list) {
5.         this.listHero = list;
6.     }
7.
8.
9.     @NonNull
10.    @Override
11.    public ListViewHolder onCreateViewHolder(@NonNull
   ViewGroup parent, int viewType) {
12.        View view =
   LayoutInflater.from(parent.getContext()).inflate(R.layo
   ut.item_row_hero, parent, false);
13.        return new ListViewHolder(view);
14.    }
15.
16.    @Override
```



```

17.     public void onBindViewHolder(@NonNull final
18.         ViewHolder holder, int position) {
19.             Hero hero = listHero.get(position);
20.             Glide.with(holder.itemView.getContext())
21.                 .load(hero.getPhoto())
22.                 .apply(new
23.                     RequestOptions().override(55, 55))
24.                         .into(holder.imgPhoto);
25.             holder.tvName.setText(hero.getName());
26.             holder.tvDetail.setText(hero.getDetail());
27.         }
28.     }
29.     @Override
30.     public int getItemCount() {
31.         return listHero.size();
32.     }
33.     class ViewHolder extends
34.         RecyclerView.ViewHolder {
35.         ImageView imgPhoto;
36.         TextView tvName, tvDetail;
37.         ViewHolder(View itemView) {
38.             super(itemView);
39.             imgPhoto =
40.                 itemView.findViewById(R.id.img_item_photo);
41.             tvName =
42.                 itemView.findViewById(R.id.tv_item_name);
43.             tvDetail =
44.                 itemView.findViewById(R.id.tv_item_detail);
45.         }
46.     }
47. }

```

Kotlin

```

1. class ListHeroAdapter(val listHero: ArrayList<Hero>) :
2.     RecyclerView.Adapter<ListHeroAdapter.ListViewHolder>() {
3.
4.     override fun onCreateViewHolder(parent: ViewGroup,
5.         viewType: Int): ListViewHolder {
6.         val view: View =
7.             LayoutInflater.from(parent.context).inflate(R.layout.it
8.             em_row_hero, parent, false)
9.         return ListViewHolder(view)
10.
11.    }
12.
13.    override fun onBindViewHolder(holder: ListViewHolder,
14.        position: Int) {
15.        val hero = listHero[position]
16.
17.        holder.imgPhoto.setImageResource(hero.getPhoto())
18.        holder.tvName.text = hero.getName()
19.        holder.tvDetail.text = hero.getDetail()
20.    }
21.
22.    companion object {
23.        const val ITEM_TYPE_IMAGE = 0
24.        const val ITEM_TYPE_DETAIL = 1
25.    }
26.
27.    class ListViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
28.        val imgPhoto: ImageView = itemView.findViewById(R.id.img_item_photo)
29.        val tvName: TextView = itemView.findViewById(R.id.tv_item_name)
30.        val tvDetail: TextView = itemView.findViewById(R.id.tv_item_detail)
31.    }
32.
33. }

```



```

11.         Glide.with(holder.itemView.context)
12.             .load(hero.photo)
13.             .apply(RequestOptions().override(55,
      55))
14.             .into(holder.imgPhoto)
15.
16.         holder.tvName.text = hero.name
17.         holder.tvDetail.text = hero.detail
18.     }
19.
20.     override fun getItemCount(): Int {
21.         return listHero.size
22.     }
23.
24.     inner class ListViewHolder(itemView: View) :
25.         RecyclerView.ViewHolder(itemView) {
26.         var tvName: TextView =
27.             itemView.findViewById(R.id.tv_item_name)
28.         var tvDetail: TextView =
29.             itemView.findViewById(R.id.tv_item_detail)
30.         var imgPhoto: ImageView =
31.             itemView.findViewById(R.id.img_item_photo)
32.     }
33. }
```

Catatan:

Variabel parent didapat dari parameter yang ada pada fungsi onCreateViewHolder. Apabila hasil generate pada Android Studio Anda berbeda (misal ViewGroup, bukan parent). Silakan sesuaikan saja dengan parameternya atau nama variabelnya.

- 1) Kemudian kita modifikasi berkas kelas **MainActivity** kita inisiasikan **RecyclerView** yang sudah dibuat di **activity_main**.



Java

```
1. public class MainActivity extends AppCompatActivity {  
2.     private RecyclerView rvHeroes;  
3.  
4.     @Override  
5.     protected void onCreate(Bundle savedInstanceState) {  
6.         super.onCreate(savedInstanceState);  
7.         setContentView(R.layout.activity_main);  
8.  
9.         rvHeroes = findViewById(R.id.rv_heroes);  
10.        rvHeroes.setHasFixedSize(true);  
11.    }  
12. }
```

Kotlin

```
1. class MainActivity : AppCompatActivity() {  
2.     private lateinit var rvHeroes: RecyclerView  
3.  
4.     override fun onCreate(savedInstanceState: Bundle?) {  
5.         super.onCreate(savedInstanceState)  
6.         setContentView(R.layout.activity_main)  
7.  
8.         rvHeroes = findViewById(R.id.rv_heroes)  
9.         rvHeroes.setHasFixedSize(true)  
10.    }  
11. }
```

m) Selanjutnya setelah diinisiasi, kita akan panggil data yang sudah kita buat di kelas **HeroesData**.



Java

```
1.  public class MainActivity extends AppCompatActivity {  
2.      private RecyclerView rvHeroes;  
3.      private ArrayList<Hero> list = new ArrayList<>();  
4.  
5.      @Override  
6.      protected void onCreate(Bundle savedInstanceState) {  
7.          super.onCreate(savedInstanceState);  
8.          setContentView(R.layout.activity_main);  
9.  
10.         rvHeroes = findViewById(R.id.rv_heroes);  
11.         rvHeroes.setHasFixedSize(true);  
12.  
13.         list.addAll(HeroesData.getListData());  
14.         showRecyclerList();  
15.     }  
  
16.  
17.     private void showRecyclerList(){  
18.         rvHeroes.setLayoutManager(new LinearLayoutManager(this));  
19.         ListHeroAdapter listHeroAdapter = new ListHeroAdapter(list);  
20.         rvHeroes.setAdapter(listHeroAdapter);  
21.     }  
22. }
```

Acti

Kotlin

```
1.  class MainActivity : AppCompatActivity() {  
2.      private lateinit var rvHeroes: RecyclerView  
3.      private var list: ArrayList<Hero> = arrayListOf()  
4.  
5.      override fun onCreate(savedInstanceState: Bundle?) {  
6.          super.onCreate(savedInstanceState)  
7.          setContentView(R.layout.activity_main)  
8.  
9.          rvHeroes = findViewById(R.id.rv_heroes)  
10.         rvHeroes.setHasFixedSize(true)  
11.  
12.         list.addAll(HeroesData.listData)  
13.         showRecyclerList()  
14.     }  
15.
```

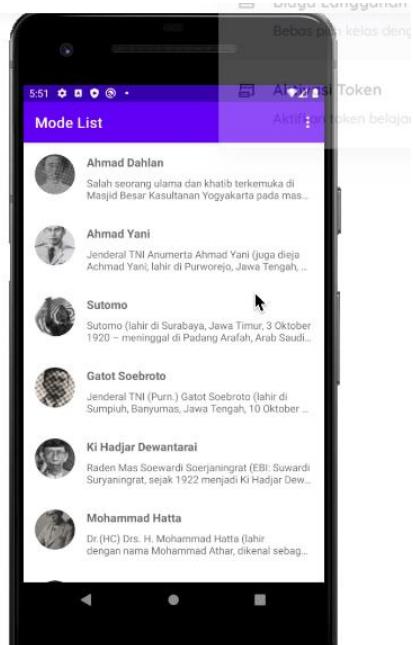


```

16.     private fun showRecyclerList() {
17.         rvHeroes.layoutManager = LinearLayoutManager(this)
18.         val listHeroAdapter = ListHeroAdapter(list)
19.         rvHeroes.adapter = listHeroAdapter
20.     }
21. }

```

n) Menjalankan Aplikasi, jika sesuai akan tampak seperti di bawah ini

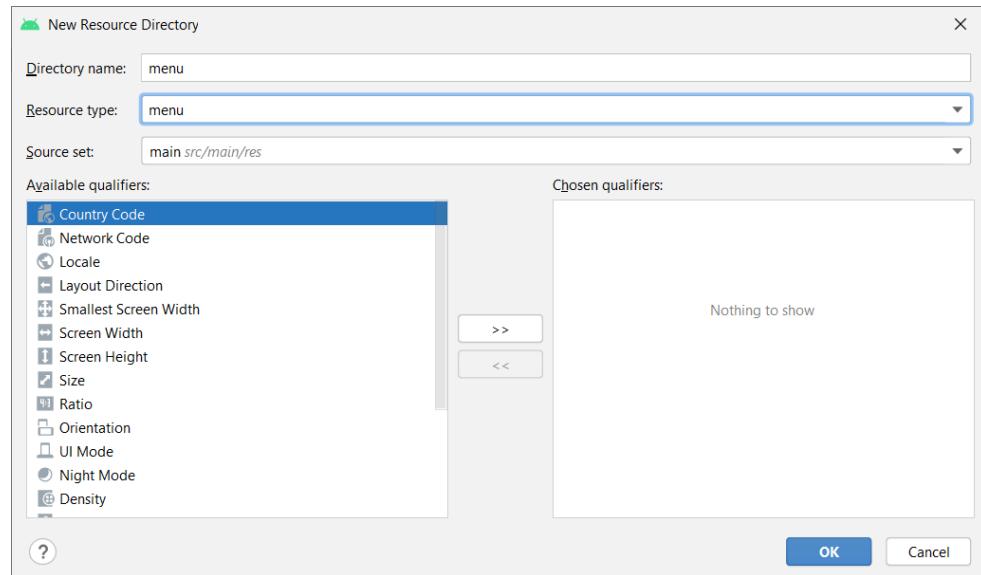


Gambar 79 Tampilan Sementara saat Aplikasi dijalankan mada RecyclerView Mode List

Sekarang mari kita buat sebuah **RecyclerView** dengan memanfaatkan fasilitas menu.

o) Langkah pertama adalah dengan membuat *resource directory* terkait. *Resource directory* menu secara bawaan tidak disediakan. Kita harus membuatnya terlebih dahulu. Klik kanan pada direktori **res** → **new** → **Android Resource Directory**. Setelah muncul dialog box seperti di bawah ini, isikan menu pada *field directory name*. Klik **OK** untuk menyelesaiakannya.

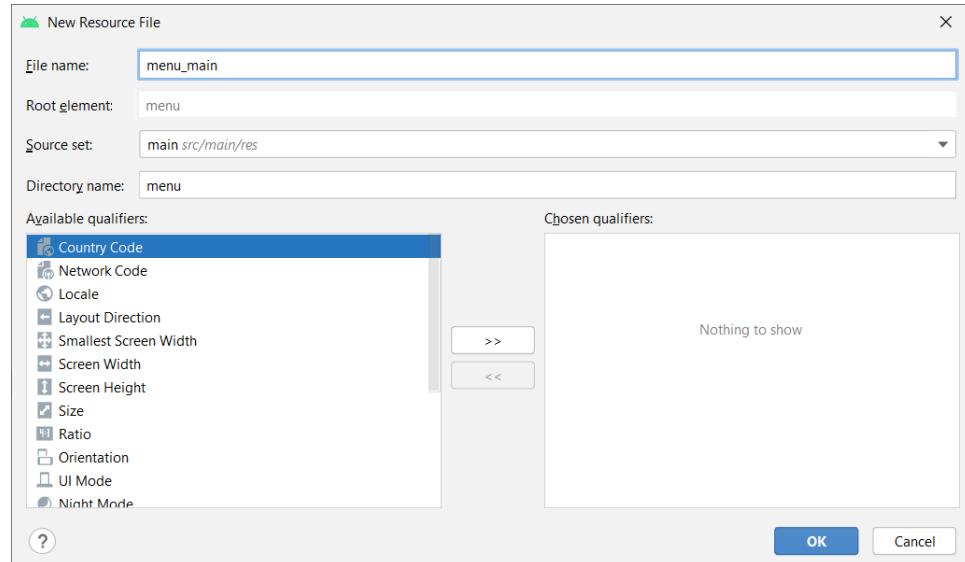




Gambar 80 Tampilan untuk Membuat Menu

Sebuah **RecyclerView** untuk menampilkan data Pahlawan Republik Indonesia sudah tercipta.

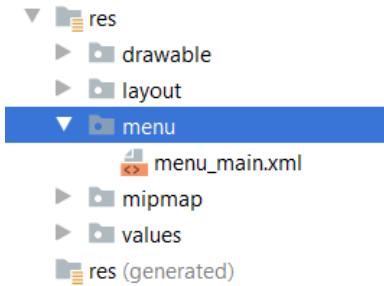
- p) Kemudian, klik kanan pada direktori tersebut → new → **Menu Resource File**. Beri nama **menu_main** pada *field file name*.



Gambar 81 Tampilan untuk Membuat Menu Resource File



Maka hasilnya akan terlihat seperti ini:

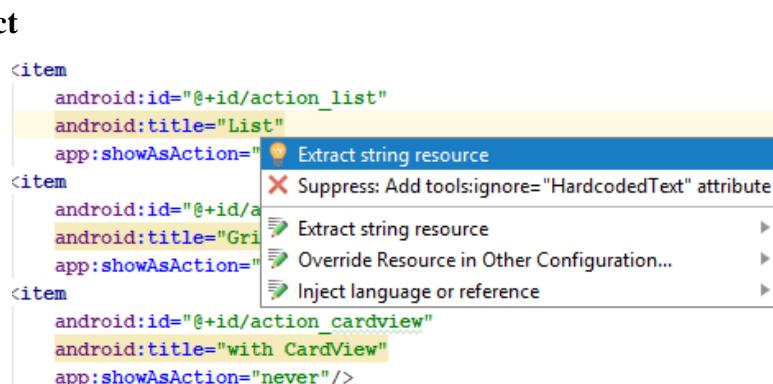


Gambar 82 Tampilan setelah Menu Dibuat

Setelah itu, lengkapi kodennya menjadi seperti berikut:

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <menu xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:app="http://schemas.android.com/apk/res-auto">
4.     <item
5.         android:id="@+id/action_list"
6.         android:title="List"
7.         app:showAsAction="never"/>
8.     <item
9.         android:id="@+id/action_grid"
10.        android:title="Grid"
11.        app:showAsAction="never"/>
12.    <item
13.        android:id="@+id/action_cardview"
14.        android:title="with CardView"
15.        app:showAsAction="never"/>
```

Jangan lupa untuk memindahkan text ke **strings.xml**. Caranya **alt+enter** → **Extract**



Gambar 83 Tampilan saat Mengextract strings.xml



Maka `menu_main` menjadi seperti berikut:

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <menu xmlns:android="http://schemas.android.com/apk/res/android"
3.      xmlns:app="http://schemas.android.com/apk/res-auto">
4.     <item
5.         android:id="@+id/action_list"
6.         android:title="@string/list"
7.         app:showAsAction="never" />
8.     <item
9.         android:id="@+id/action_grid"
10.        android:title="@string/grid"
11.        app:showAsAction="never" />
12.    <item
13.        android:id="@+id/action_cardview"
14.        android:title="@string/with_cardview"
15.        app:showAsAction="never" />
16. </menu>
```

Dan pada bagian `res → values → strings.xml`, tambahkan juga string berikut:

```
1. <resources>
2.   <string name="app_name">My Application</string>
3.   <string name="heroes_name">Nama Pahlawan</string>
4.   <string name="detail">Detail</string>
5.   <string name="list">List</string>
6.   <string name="grid">Grid</string>
7.   <string name="with_cardview">with CardView</string>
8.   <string name="favorite">Favorite</string>
9.   <string name="share">Share</string>
10.  </resources>
```

- q) Setelah selesai, saatnya pasang menu tersebut di `MainActivity` dengan menambahkan metode berikut:



Java

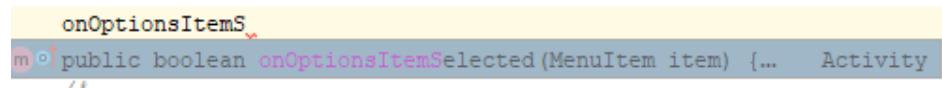
```
1. public class MainActivity extends AppCompatActivity {  
2.     ...  
3.  
4.     @Override  
5.     public boolean onCreateOptionsMenu(Menu menu) {  
6.         getMenuInflater().inflate(R.menu.menu_main, menu);  
7.         return super.onCreateOptionsMenu(menu);  
8.     }  
9.  
10.    @Override  
11.    public boolean onOptionsItemSelected(MenuItem item) {  
12.        setMode(item.getItemId());  
13.        return super.onOptionsItemSelected(item);  
14.    }  
15.  
16.    public void setMode(int selectedMode) {  
17.        switch (selectedMode) {  
18.            case R.id.action_list:  
19.                break;  
20.  
21.            case R.id.action_grid:  
22.                break;  
23.  
24.            case R.id.action_cardview:  
25.                break;  
26.        }  
27.    }  
28. }
```



Kotlin

```
1. class MainActivity : AppCompatActivity() {  
2.     ...  
3.  
4.     override fun onCreateOptionsMenu(menu: Menu): Boolean {  
5.         menuInflater.inflate(R.menu.menu_main, menu)  
6.         return super.onCreateOptionsMenu(menu)  
7.     }  
8.  
9.     override fun onOptionsItemSelected(item: MenuItem): Boolean {  
10.        setMode(item.itemId)  
11.        return super.onOptionsItemSelected(item)  
12.    }  
13.  
14.    private fun setMode(selectedMode: Int) {  
15.        when (selectedMode) {  
16.            R.id.action_list -> {  
17.                ...  
18.  
19.                R.id.action_grid -> {  
20.                    ...  
21.  
22.                    R.id.action_cardview -> {  
23.                        ...  
24.                    }  
25.                }  
26.            }  
27.        }  
28.    }  
29.  
30.}
```

Untuk metode di atas, Anda bisa menggunakan **Ctrl (tahan) + Spasi** untuk menampilkan *code assistant* pada Android Studio.



A screenshot of the Android Studio code editor. The cursor is at the start of the word 'onOptionsItemSelected' in the line 'm o t public boolean onOptionsItemSelected(MenuItem item) { ... Activity'. A yellow tooltip-like box is displayed above the cursor, containing the text 'onOptionsItemSelected' with a small downward arrow indicating it's a suggestion or part of a larger list.

Gambar 84 Tampilan untuk menggunakan **Ctrl (tahan) + Spasi** saat menampilkan code assistant



Kode di kelas **MainActivity** akan berubah menjadi seperti ini:

Java

```
1. public class MainActivity extends AppCompatActivity {  
2.     private RecyclerView rvHeroes;  
3.     private ArrayList<Hero> list = new ArrayList<>();  
4.  
5.     @Override  
6.     protected void onCreate(Bundle savedInstanceState) {  
7.         super.onCreate(savedInstanceState);  
8.         setContentView(R.layout.activity_main);  
9.  
10.        rvHeroes = findViewById(R.id.rv_heroes);  
11.        rvHeroes.setHasFixedSize(true);  
12.  
13.        list.addAll(HeroesData.getListData());  
14.        showRecyclerList();  
15.    }  
16.  
17.    private void showRecyclerList(){  
18.        rvHeroes.setLayoutManager(new LinearLayoutManager(this));  
19.        ListHeroAdapter listHeroAdapter = new ListHeroAdapter(list);  
20.        rvHeroes.setAdapter(listHeroAdapter);  
21.    }  
22.  
23.    @Override  
24.    public boolean onCreateOptionsMenu(Menu menu) {  
25.        getMenuInflater().inflate(R.menu.menu_main, menu);  
26.        return super.onCreateOptionsMenu(menu);  
27.    }  
28.
```



```
29.     @Override
30.     public boolean onOptionsItemSelected(MenuItem item) {
31.         setMode(item.getItemId());
32.         return super.onOptionsItemSelected(item);
33.     }
34.
35.     public void setMode(int selectedMode) {
36.         switch (selectedMode) {
37.             case R.id.action_list:
38.                 break;
39.
40.             case R.id.action_grid:
41.                 break;
42.
43.             case R.id.action_cardview:
44.                 break;
45.         }
46.     }
47. }
```



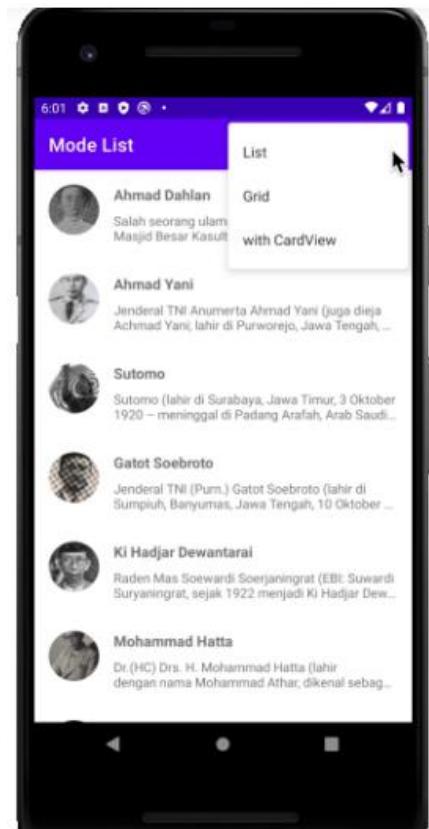
Kotlin

```
1. class MainActivity : AppCompatActivity() {  
2.     private lateinit var rvHeroes: RecyclerView  
3.     private var list: ArrayList<Hero> = arrayListOf()  
4.  
5.     override fun onCreate(savedInstanceState: Bundle?) {  
6.         super.onCreate(savedInstanceState)  
7.         setContentView(R.layout.activity_main)  
8.  
9.         rvHeroes = findViewById(R.id.rv_heroes)  
10.        rvHeroes.setHasFixedSize(true)  
11.  
12.        list.addAll(HeroesData.listData)  
13.        showRecyclerList()  
14.    }  
15.  
16.    private fun showRecyclerList() {  
17.        rvHeroes.layoutManager = LinearLayoutManager(this)  
18.        val listHeroAdapter = ListHeroAdapter(list)  
19.        rvHeroes.adapter = listHeroAdapter  
20.    }  
21.  
22.    override fun onCreateOptionsMenu(menu: Menu): Boolean {  
23.        menuInflater.inflate(R.menu.menu_main, menu)  
24.        return super.onCreateOptionsMenu(menu)  
25.    }  
26.  
27.    override fun onOptionsItemSelected(item: MenuItem): Boolean {  
28.        setMode(item.itemId)  
29.        return super.onOptionsItemSelected(item)  
30.    }
```



```
31.  
32.    private fun setMode(selectedMode: Int) {  
33.        when (selectedMode) {  
34.            R.id.action_list -> {  
35.                }  
36.  
37.            R.id.action_grid -> {  
38.                }  
39.  
40.            R.id.action_cardview -> {  
41.                }  
42.            }  
43.        }  
44.    }
```

- r) Running Aplikasinya sehingga output akan seperti gambar di bawah



Gambar 85 Tampilan Akhir Saat RecyclerView ditambahkan Menu





5. Tugas Praktikum

- a. Buatlah Perpindahan Fragment dengan menu, dengan ketentuan:

Package : Nama Kalian

Fragment 1 : Nama

Fragment 2 : Nim

Fragment 3 : Kelas

Fragment 4 : Prodi

- b. Buatlah RecylerView di Android Studio dengan Ketentuan:

Nama Project : Nama Kalian

Foto : kalian dan Anggota Kelompok Projek Akhir
(Foto Pertama di list atas Nama Kalian)

Detail : Ceritakan Masing-masing Profile Anggota
Kelompok Kalian (Seperti, lahir dimana,
anak keberapa, tanggal lahir, alamat,
hobby, cita-cita)

file yang dikumpulkan, Semua Codingan (di Screen Shot) output
(Videokan)



DAFTAR REFERENSI

1. Dicoding, “Belajar Membuat Aplikasi Android Untuk Pemula”, <https://www.dicoding.com/academies/51> , diakses pada tanggal 3 November 2022.
2. Dicoding, “Belajar Fundamental Aplikasi Android”, <https://www.dicoding.com/academies/14> , diakses pada tanggal 3 November 2022.
3. Wirawan, Rio, dkk. ”Mobile Programming”, 2021. Universitas Pembangunan Nasional “Veteran” Jakarta.

