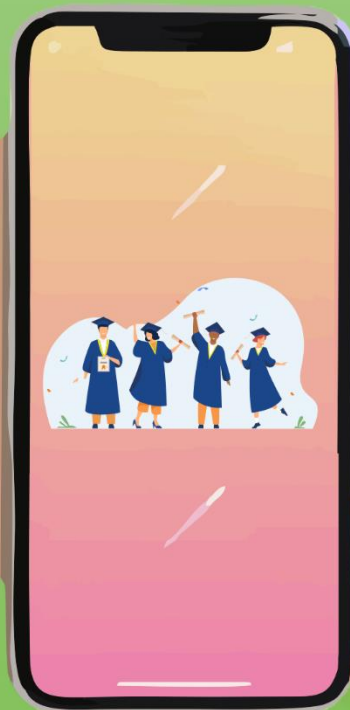




# MODUL MOBILE PROGRAMMING I



**Android  
Studio**

**PROGRAM STUDI REKAYASA PERANGKAT LUNAK (S1)  
FAKULTAS TEKNIK DAN INFORMATIKA  
UNIVERSITAS BINA SARANA INFORMATIKA**

Disusun Oleh :  
Unit Pengembangan Akademik

## **KATA PENGANTAR**

Puji syukur penulis panjatkan kehadirat Allah SWT, yang telah memberikan rahmat dan hidayahnya sehingga modul Mobile Programming I ini dapat terselesaikan dengan baik. Selanjutnya modul ini disusun untuk memberikan gambaran bagi mahasiswa yang mempelajari Mobile Programming I. Dengan menggunakan metode “Belajar Cepat Dan Menyenangkan” karena modul ini disertai contoh kasus, sehingga lebih memudahkan dalam memahami pembuatan aplikasi Android.

Akhir kata penulis berharap semoga modul Mobile Programming ini dapat dipergunakan sebaik-baiknya dan dapat dijadikan referensi untuk mahasiswa umum yang ingin mempelajari pembuatan aplikasi *mobile*. Tak lupa penulis mengucapkan banyak terima kasih kepada semua pihak yang telah membantu dengan tenaga dan pikirannya, terima kasih juga kepada rekan-rekan instruktur, dosen dan semuanya yang tidak bisa disebutkan satu persatu, yang selalu mendukung penulis sehingga modul ini dapat selesai sesuai yang kita inginkan.

Penulis menyadari masih banyak kekurangan dalam penyusunan modul ini. Untuk itu saran dan kritik yang membangun sangat penulis harapkan guna perbaikan dan pengembangan modul ini ke depan.

**Jakarta, Maret 2022**

**Unit Pengembangan Akademik  
Program Studi Rekayasa Perangkat Lunak**

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>ii</b>
<b>DAFTAR ISI.....</b>	<b>iii</b>
<b>BAB 1 PENGENALAN ANROID .....</b>	<b>1</b>
A. Mengetahui Android .....	1
B. Logo Android.....	2
<b>BAB 2 MENGENAL ANDROID STUDIO .....</b>	<b>4</b>
A. Pengantar Android Studio.....	4
B. Menginstall Android Studio.....	4
C. Memulai Proyek Android Studio.....	6
D. Antarmuka Android Studio.....	9
E. Shortcut di Anroid Studio .....	11
<b>BAB 3 KONSEP RUNNING APLICATION .....</b>	<b>13</b>
A. Aspek Pada Emulator.....	13
B. Menjalankan Pada Emulator .....	14
C. Menjalankan Pada Device.....	16
<b>BAB 4 KOMPONEN DASAR ANDROID – ACTIVITY .....</b>	<b>19</b>
A. Pengenalan Activity .....	19
B. Latihan Activity .....	23
<b>BAB 5 KOMPONEN DASAR ANDROID – INTENT .....</b>	<b>32</b>
A. Pengenalan Intent.....	32
B. Latihan Intent .....	32
<b>BAB 6 KOMPONEN DASAR ANDROID – SERVICE.....</b>	<b>64</b>
A. Pengenalan Service .....	64
B. Latihan Service .....	66
C. Penggunaan Service Di Alarm Manager.....	72
<b>BAB 7 CONTOH APLIKASI BERBASIS ANDROID .....</b>	<b>102</b>
A. Aplikasi Penghitung Resistor CMD.....	102
B. Aplikasi Pengenalan Nama Binatang.....	104
C. Aplingasi Menghitung Waktu Belajar .....	105
<b>DAFTAR PUSTAKA.....</b>	<b>106</b>

## BAB 1 PENGENALAN ANDROID

### A. Mengetahui Android

Android merupakan sebuah sistem operasi dan platform pemrograman yang dikembangkan oleh perusahaan Google yang diperuntukan untuk ponsel pintar dan perangkat seluler lain. Android dapat berfungsi dalam berbagai macam perangkat dengan kit development sebagai penulisan kode dan perakitan modul untuk membangun aplikasi bagi pengguna Android.

Android bermula dibawah naungan Android, Inc sebuah perusahaan yang didirikan oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White yang berlokasi di Palo Alto, California pada tahun 2003. Android pada awalnya bertujuan untuk pengembangan sistem operasi kamera digital. Pada 17 Agustus 2005 Google mengakuisisi Android, Inc. Disamping itu ada beberapa hal yang membuat Android sebagai sistem operasi yang memang layak digunakan oleh pengguna atau dikembangkan para developer. Diantaranya:

1. Sistem Operasi Smartphone Terpopuler

Sejak dirilis tahun 2008, Google telah mengeluarkan beberapa versi, dengan "Oreo" sebagai versi yang terbaru. Pada tahun 2013, Android menjadi *operation system* (OS) terlaris pada *tablet* dan *smartphone*. Tercatat pada tahun 2016, store Android memiliki lebih dari 2.8 juta aplikasi. Android menarik bagi perusahaan teknologi yang membutuhkan barang siap jadi, biaya rendah dan kustomisasi OS untuk perangkat teknologi tinggi mereka. Hal ini menjadi daya tarik bagi banyak perusahaan, sehingga mereka memilih Android.

2. *Open Source*

*Source code* dari Android bersifat *open source*. Ini adalah hal menarik bagi komunitas *developer*, karena lisensi *open source* sangat mendukung untuk mengembangkan produknya dengan aman.

3. *Mempunyai Store tersendiri*

Aplikasi Android bisa di distribusikan menggunakan *web*, *copy APK*, dan *store*. *Store* Android, yaitu Google Play, merupakan cara termudah bagi para *developer* untuk mendistribusikan aplikasinya ke pasar yang memiliki miliaran pengguna. Google play merupakan *store* resmi Android yang dikelola oleh Google. Pengguna bisa mencari dan mengunduh aplikasi yang dikembangkan dengan menggunakan Android Software Development Kit. Tidak hanya aplikasi yang ditawarkan di Google Play. Ada beragam konten lainnya yang dapat dinikmati pengguna, misalnya media digital, musik, buku, majalah, film dan program televisi.

4. *Mempunyai Development Kit untuk Developer*

Android Software Development Kit (SDK) merupakan *kit* yang bisa digunakan oleh para *developer* untuk mengembangkan aplikasi berbasis Android. Di dalamnya, terdapat

beberapa *tools* seperti *debugger*, *software libraries*, *emulator*, dokumentasi, *sample code* dan tutorial.

##### 5. Bahasa Pemograman yang mendukung

Bahasa pemrograman yang sering digunakan untuk mengembangkan aplikasi Android adalah Java. Namun ada beberapa bahasa lainnya yang dapat digunakan, seperti C++, dan Go dan Kotlin.

##### 6. Perkembangan Android

Saat ini versi yang sudah ada yaitu:

Code name	Version number	Linux Kernel version	Initasi release date	API level
(No codename)	1.0	?	September 23, 2008	1
Petit Four	1.1	2.6	February 9, 2009	2
Cupcake	1.5	2.6.27	April 27,2009	3
Donut	1.6	2.6.29	September 15, 2009	4
Eclair	2.0 -2.1	2.6.29	Oktober 26,2009	5-7
Froyo	2.2-2.2.3	2.6.32	May 20, 2010	8
Gingerbead	2.3 – 2.3.7	2.6.35	December 6, 2010	9-10
Honeycomb	3.0 – 3.2.6	2.6.36	February 22, 2011	11-13
Ice Cream Sandwich	4.0 – 4.0.4	3.0.1	October 10.2011	14 – 15
Jelly Bean	4.1 -4.3.1	3.0.31 to 3.4.39	July 9, 2012	16-18
Kitkat	4.4 – 4.4.4	3.10	October 31, 2013	19-20
Lollipop	5.0 - 5.1.1	3.16	November 12, 2014	21 – 22
Marshmallow	6.0 – 6.0.1	3.1.8	October 5, 2015	23
Nougat	7.0 – 7.1.2	4.4	Agust 22, 2016	24-25
Oreo	8.0 – 8.1	4.10	Agust 21, 2017	26 – 27
Pie	9.0	4.4.107, 4.9.84, and 4.14.42	Agust 6, 2018	28
Android 10	10	4.9.191 4.14.142 4.19.71	September 3, 2019	29

## B. Logo Android

Logo yang sekarang akrab untuk OS Android, yang terlihat seperti kombinasi robot dan bug hijau, diciptakan oleh Irina Blok saat dipekerjakan oleh Google. Dalam obrolan dengan The New York Times pada tahun 2013, Blok mengatakan bahwa satu-satunya petunjuk yang diberikan kepada tim desainnya oleh Google adalah membuat logo terlihat seperti robot. Dia mengklaim

bahwa desain akhir diilhami sebagian dengan melihat logo toilet yang dikenal seperti “Pria” dan “Wanita”.



Satu hal yang coba dilakukan Blok dan Google adalah menamai robot Android itu sendiri sebagai proyek open source. Hampir setiap perusahaan besar lainnya akan melindungi logo atau maskot semacam itu agar tidak didesain ulang dan digunakan oleh orang lain. Namun, robot Android kini telah dimodifikasi dan digunakan oleh banyak orang, semuanya karena Google mengizinkan perubahan tersebut di bawah Creative Commons 3.0 Attribution License.

## **BAB 2**

### **MENGENAL ANDROID STUDIO**

#### **A. Pengantar Android Studio**

Android Studio merupakan Integrated Development Environment (IDE) untuk pengembangan aplikasi Android yang diperkenalkan google pada acara I/O 2013. Android Studio berbasiskan JetBrains IntelliJ IDEA, dan dikhususkan untuk mengembangkan aplikasi berplatform Android. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas saat membuat aplikasi Android, Diantaranya:

1. Sistem versi berbasis Gradle yang fleksibel
2. Emulator yang cepat dan kaya fitur
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
4. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
5. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
6. Alat pengujian dan kerangka kerja yang ekstensif
7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
8. Dukungan Bahasa C++ dan NDK
9. Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine

Sebelum munculnya Android Studio, para programmer Android sebelumnya mengenal dan menggunakan Eclipse. Eclipse adalah IDE pemrograman Android sebelum munculnya Android Studio tahun 2013. Saat ini Google telah berpaling dari Eclipse dan menjadikan Android Studio sebagai IDE resminya. yaitu pada tanggal 16 Mei 2013, Google menghentikan support Android Development Tools (ADT) ke Eclipse, ADT resmi hanya didapatkan di IDE Android Studio.

#### **B. Menginstall Android Studio**

Untuk melakukan instalasi Android Studio maka pertama sekali harus mengetahui bahasa pemrograman yang digunakan. Salah satu bahasa yang bisa digunakan untuk mengembangkan Android adalah Java. Bahasa lain yang bisa digunakan seperti C/C++, Go, Kotlin. Untuk membuat program dengan Bahasa pemrograman java maka perlu instalasi JDK pertama kali. JDK adalah singkatan dari Java SE Development Kit. JDK merupakan perangkat lunak yang digunakan untuk melakukan kompilasi dari kode - kode Java yang dibuat oleh pengembang aplikasi lalu menerjemahkannya ke dalam bytecode untuk dijalankan oleh JRE. JRE sendiri adalah singkatan dari Java Runtime Environment. JRE merupakan perangkat lunak yang digunakan untuk menjalankan program/ aplikasi yang berbasis Java. Apabila kode - kode Java yang dibuat oleh pengembang telah selesai dikompilasi oleh JDK, maka tugas JRE lah yang

menjalankan aplikasi berbasis Java tersebut. Untuk menjalankan aplikasi Java, versi JRE harus sama atau lebih tinggi dari versi JDK yang terinstal di komputer pengembang.

<http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>

berikut halaman tautan setelah diklik:

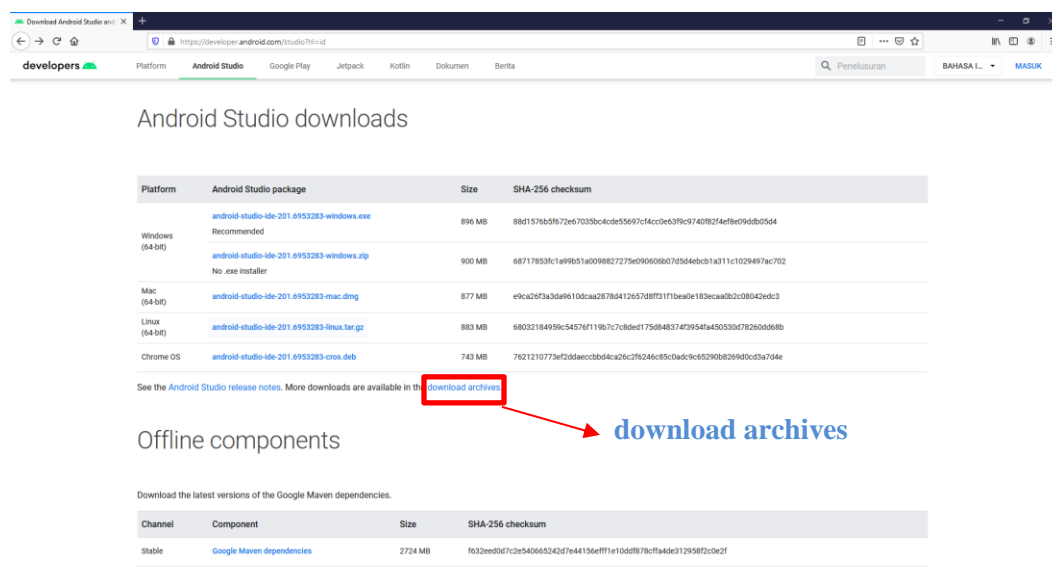
Langkah Instalasi JDK:

1. Klik Java SE Download pada tautan yang sudah diklik  
Maka muncul pilihan Java SE development Kit yang akan digunakan.
2. Pilih sesuai Sistem Operasi yang digunakan, download dan lakukan instalasi dan ikuti petunjuk sampai selesai dan dapat digunakan.
3. Setelah selesai instalasi JDK, Selanjutnya melakukan instalasi IDE Android Studio.

Adroid Studio dapat diunduh pada tautan berikut:

<https://developer.android.com/studio/archive.html>

pada tautan tersebut tersedia versi yang terbaru dan versi yang sebelumnya. Namun sebaiknya mendownload versi yang stabil. Pada Buku ini menggunakan versi 4.0.1



**Gambar. 1 Download Android Versi 4.0.1**



## Arsip download Android Studio

Halaman ini menyediakan arsip semua rilis Android Studio.

Namun, sebaiknya download [versi stabil terbaru](#) atau [versi pratinjau terbaru](#).

Sebelum mendownload, Anda harus menyetujui ketentuan dan persyaratan berikut.



**Gambar. 2 Persetujuan Download Arsip**

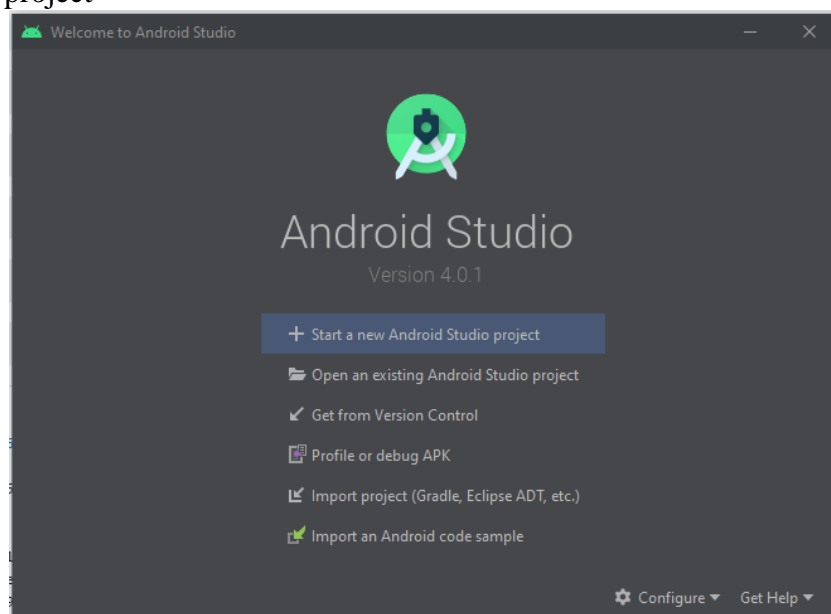
4. Setelah mendownload lakukan instalasi sampai selesai. Pada saat instalasi membutuhkan koneksi internet untuk proses install.

## C. Memulai Proyek Android Studio

Setelah berhasil memasang Android Studio IDE, klik dua kali ikon aplikasi Android Studio untuk memulainya. Pilih Start a new Android Studio project di jendela Welcome seperti gambar berikut:

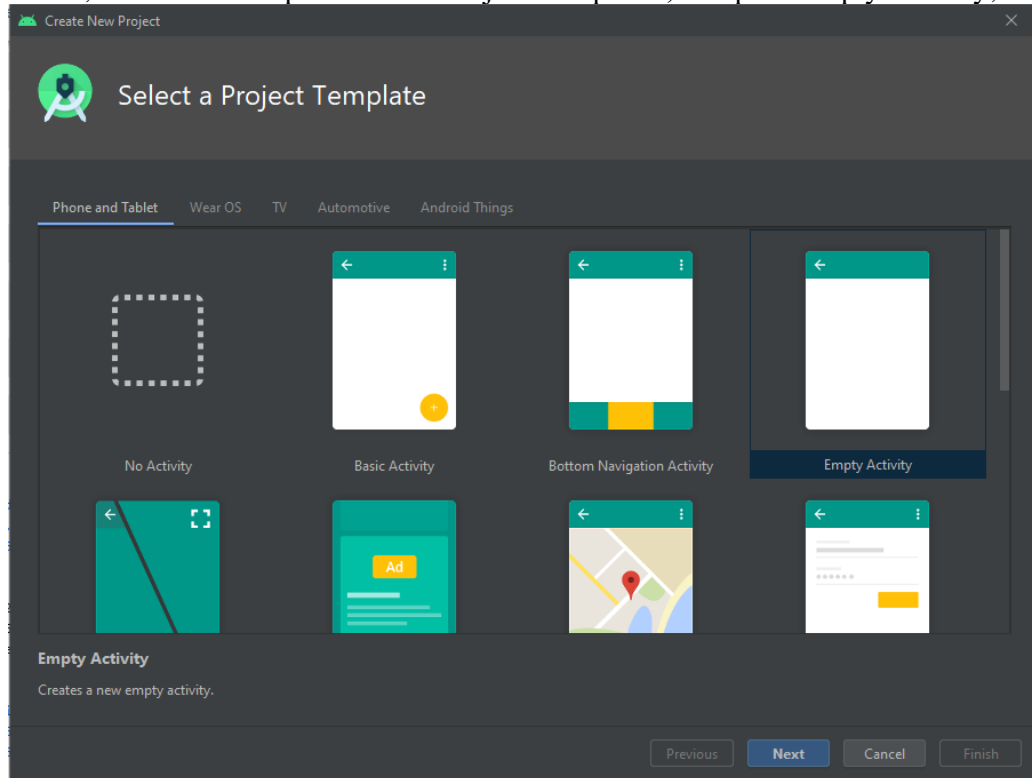
Studio Langkah-langkah membuat project awal di Android Studio:

1. Pada halaman welcome, pilih Start a new android studio project untuk memulai membuat project



Dalam tampilan ini berikan nama aplikasi di application name company domain. Company domain akan digunakan sebagai alat identifikasi ketika aplikasi akan dipublikasikan nantinya. Lokasi project silahkan sesuaikan di mana proyek akan disimpan.

2. Klik next, maka akan tampil “Select a Project Template”, lalu pilih Empty Activity,



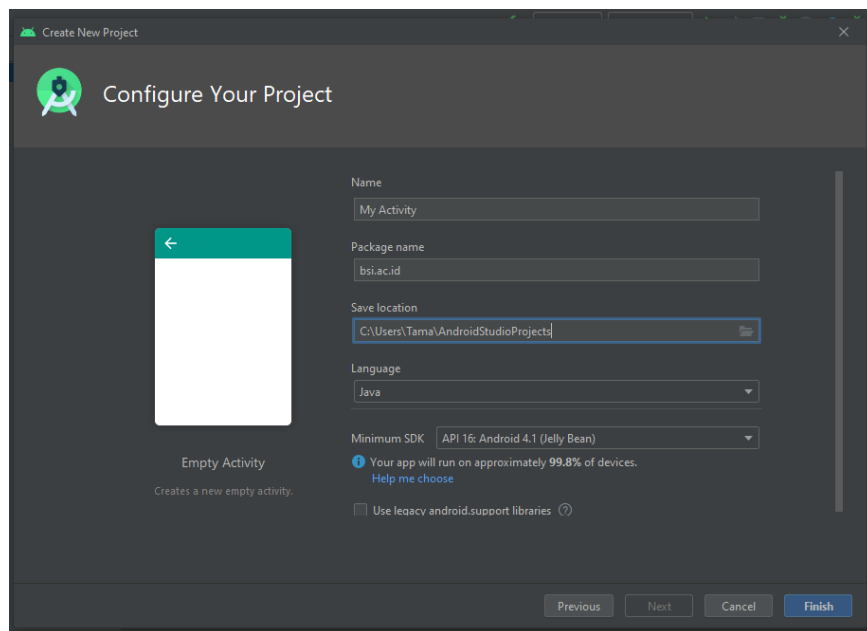
Berikut Penjelasan tipe activity awal dari template yang telah disediakan secara mendasar, diantaranya:

a	Add No Activity	Tidak ada activity yang ditambahkan
b	Basic Activity	dengan template komponen material design seperti FloatingActionButton
c	Empty Activity	Activity dalam bentuk yang paling dasar
d	Fullscreen Activity	Activity fullscreen tanpa status bar
e	Google AdMob Ads Activity	Activity dengan konfigurasi default iklan Admob
f	Google Maps Activity	Activity dengan menyediakan konfigurasi dasar Google Maps
g	Login Activity	Activity untuk halaman login
h	Master / Detail Flow	Activity yang diperuntukan untuk alur aplikasi
i	Navigation Drawer Activity	Activity dengan tampilan side bar menu
j	Scrolling Activity	Activity dengan kemampuan scroll konten didalamnya

		secara vertical
k	Settings Activity	
l	Tabbed Activity	Activity yang diperuntukan untuk konfigurasi

Pada tampilan ini memilih peranti target dari aplikasi yang akan dibuat. Dalam target devices juga bisa menentukan nilai minimum SDK yang akan didukung oleh aplikasi. Alasan memilih minimum JDK adalah untuk memastikan aplikasi yang dibuat kompatibel dengan perangkat android yang aktif di google play store saat ini. Setiap versi Android yang berurutan menyediakan kompatibilitas untuk aplikasi yang dibangun menggunakan API dari versi sebelumnya, sehingga aplikasi yang dibuat akan selalu kompatibel dengan versi Android mendatang.

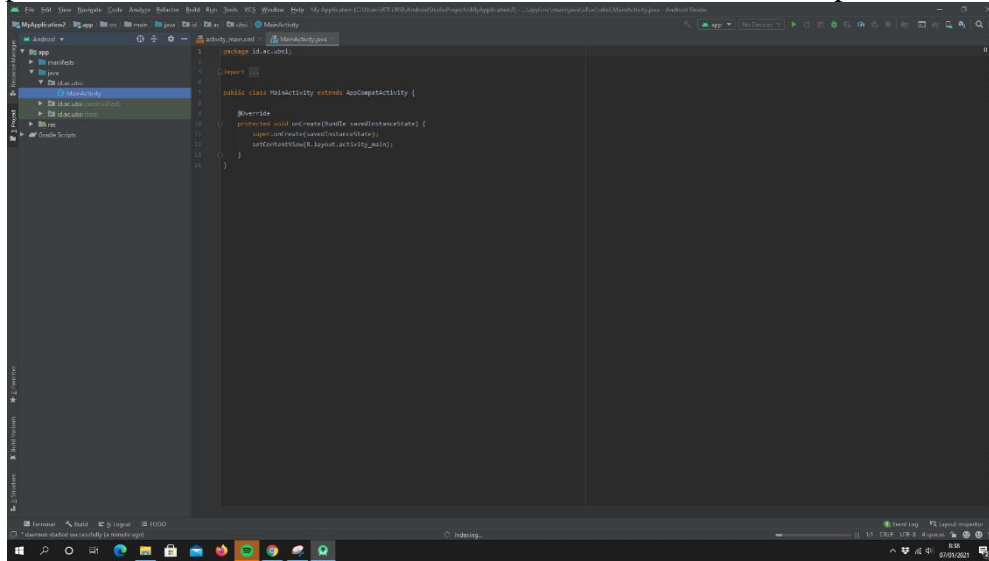
3. Klik next maka muncul jendela Configure Your Project. Sesuaikan Nama sampai dengan Minimum SDK yang diinginkan.



4. Klik Finish untuk memulai Menggunakan IDE Android Studio

## D. Antarmuka Android Studio

Tampilan penuh IDE Android Studio berbasis IntelliJ IDEA ketika dibuka pertama kali.



### 1. Jendela Tools

Merupakan Tools yang sering digunakan dalam development untuk memungkinkan dalam melakukan berbagai jenis tindakan seperti copy/paste, build, menjalankan aplikasi, hingga menjalankan emulator.



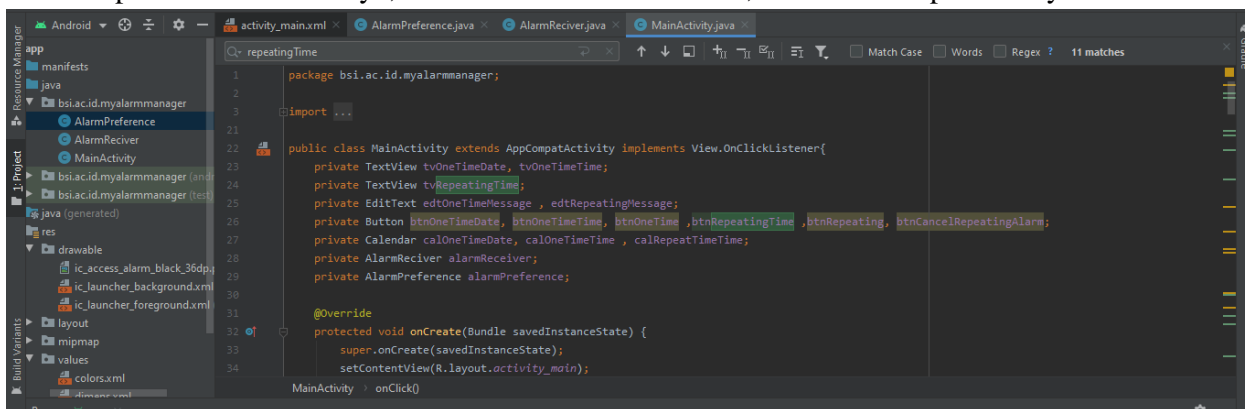
### 2. Jendela Navigasi

Jendela navigasi ini memberikan tampilan struktur yang terlihat lebih ringkas dalam jendela Project. Tampilan ini menunjukkan posisi proyek yang sedang aktif.

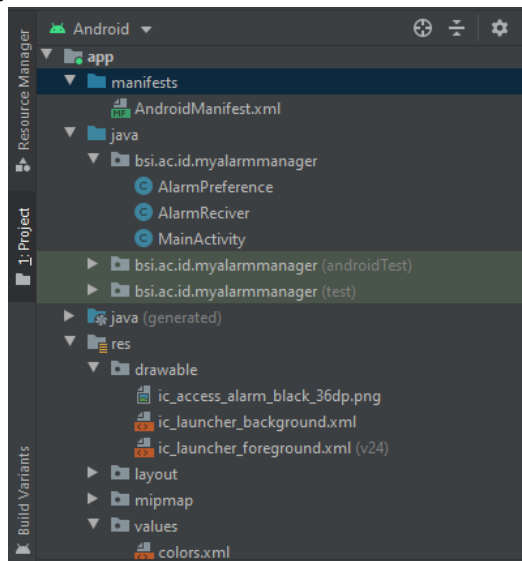


### 3. Jendela Editor

Merupakan tempat untuk membuat dan memodifikasi kode. Bergantung pada jenis file saat ini, editor dapat berubah. Misalnya, ketika melihat file tata letak, editor menampilkan Layout Editor.



#### 4. Tampilan Struktur Project



##### a. Manifests

Manifests digunakan untuk menetapkan informasi tentang aplikasi bagi lingkungan waktu proses Android. Manifest merupakan salah satu berkas yang harus ada di dalam proyek android . Manifest bertugas untuk memberikan beragam informasi penting kepada sistem Android. Sistem perlu mengetahui apa yang akan digunakan oleh aplikasi sebelum dirunning.

##### b. Java

Folder ini menyertakan aktivitas, pengujian, dan komponen lain di kode sumber Java. Setiap aktivitas, layanan, dan komponen lain didefinisikan sebagai kelas Java. Nama aktivitas pertama (layar) yang pengguna lihat, yang juga melakukan inisialisasi sumber daya seluruh aplikasi, biasanya dinamakan MainActivity secara otomatis. Java Berisi berkas source code yang ditulis dalam bahasa Java, termasuk didalamnya juga kode Unit Test dan androidTest.

##### c. Res

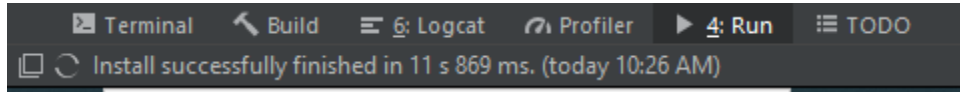
Folder res ini berisikan sumber daya, seperti layout XML, string UI, dan gambar. Aktivitas biasanya dikaitkan dengan file sumber daya XML yang menetapkan layout tampilannya. File ini biasanya diberi nama setelah aktivitas atau fungsinya. Mengatur resource di dalamnya, yang mana bukan berupa kode, melainkan layout aplikasi, sumber gambar, ikon, hingga style. Di dalam folder res ini juga terdapat sejumlah folder yang sudah diatur dan dikategorikan sesuai kebutuhan.

##### d. Gradle

Secara *default* Gradle merupakan build tools yang digunakan oleh Android Studio. Fungsinya adalah untuk membantu mengkompilasi dan menjalankan source code aplikasi yang sedang dikembangkan berdasarkan konfigurasi di Gradle. Gradle itu sendiri juga mendukung manajemen proyek dalam hal penambahan library di luar framework Android

## 5. Jendela Status

Jendela status berada diposisi bawah. Jendela status menampilkan status proyek dan IDE itu sendiri, serta setiap peringatan atau pesan yang terjadi pada saat proyek sedang dibuat atau sedang berjalan.



## E. Shortcut di Android Studio

IDE Android Studio menyediakan fasilitas yang powerful di bawah IntelliJ IDEA ini. Banyak tools yg dimiliki Android yang membantu para programmer saat mengembangkan Aplikasi yang dibuat. Berikut penjelasan shortcut yang ada di Android Studio:

No	Fungsi	Tombol	Keterangan
1	Pencarian	Ctrl+F	Berfungsi untuk <i>Find</i> , pencarian teks dalam salah satu berkas
		Shift+Shift	Berfungsi untuk <i>Search Everywhere</i> , atau dapat dikatakan pencarian semua jenis berkas yang masih dalam 1 proyek
		Ctrl+Shift+F	Berfungsi untuk <i>Find in path</i> , pencarian teks di seluruh berkas proyek
		Ctrl+R	Berfungsi untuk <i>Replace</i> , mengganti teks di dalam berkas
2	Navigasi	Ctrl+N	Berfungsi untuk <i>Find Class</i> , navigasi ke kelas tertentu
		Ctrl+Shift+N	Berfungsi untuk <i>Find file</i> , navigasi ke berkas
		Alt+	Berfungsi untuk Lompat ke method sebelumnya
		Ctrl+B	Berfungsi untuk <i>Go to declaration</i> , lompat ke deklarasi yang dipilih
		Alt+↓	Berfungsi untuk Lompat ke method sesudahnya
		Ctrl+G	Berfungsi untuk <i>Go to line</i> , lompat ke baris tertentu
		Ctrl+E	Berfungsi untuk Membuka berkas teranyar ( <i>recent file</i> ).
		Alt + F7 / Ctrl + F7	Berfungsi untuk Melihat penggunaan variabel/objek yang dipilih di seluruh berkas proyek.
		Ctrl+Shift+B	Berfungsi untuk Mencari tahu implementasi dari variabel/objek yang dipilih.
		Ctrl+Left Mouse (or) Ctrl+Alt+F7	Berfungsi untuk Melihat penggunaan pada variabel/objek yang diklik
3	Redaksi	Ctrl+D	Berfungsi untuk Menggandakan bagian yang dipilih
		Ctrl+Q	Berfungsi untuk Melihat dokumentasi dengan tampilan minimal.

		Ctrl+P	Berfungsi untuk Melihat isi dari parameter, penting ketika melihat method dari Android atau library lain
		Ctrl	Berfungsi untuk <i>Basic code completion</i> , menampilkan saran untuk melengkapi kode Anda
		Ctrl+Shift+Space	Berfungsi untuk <i>Smart code completion</i> , menampilkan saran kode untuk melengkapi kode Anda dengan lebih pintar (menampilkan apa yang benar-benar terkait dengan kode Anda)
		Alt+Insert	Berfungsi untuk <i>Generate code</i> , menghasilkan ( <i>generate</i> ) kode. Perintah ini sangat memudahkan ketika membuat constructor dan setter/getter
		Ctrl+Alt+L	Berfungsi untuk Memformat ulang kode, merapikan kode
4	Run	Ctrl + F9	Berfungsi untuk Menjalankan Project
		Ctrl + Shift + F9	Berfungsi untuk Melakukan kompilasi pada berkas, package atau modul
		Shift + F10	Berfungsi untuk <i>Running</i>
		Shift + F9	Berfungsi untuk <i>Melakukan Debug</i>
5	Basic Code Completion	Ctrl + Space	Berfungsi untuk Pemanggilan Code Completion standar untuk membantu melengkapi kode yang akan dibuat
6	Statement Completion	Ctrl + Shift + Enter	Berfungsi untuk Perintah ini sangat membantu karena menyelesaikan kode tanpa harus mengetik lengkap dan tanpa tanda kurung, kurung siku, kurung kurawal, dan banyak macam pemformatan lainnya.

## **BAB 3**

### **KONSEP RUNNING APP**

#### **A. Aspek Penting di Emulator**

Menjalankan App di Emulator harus memperhatikan beberapa aspek:

##### **1. Aspek Virtualization**

Virtualization merupakan sebuah teknik untuk menyembunyikan karakteristik fisik dari sumber daya komputer dari bagaimana cara system lain, aplikasi atau pengguna berinteraksi dengan sumber daya tersebut. Hal ini termasuk membuat sebuah sumber daya tunggal (seperti server, sebuah sistem operasi, sebuah aplikasi, atau peralatan penyimpanan terlihat berfungsi sebagai beberapa sumber daya logikal. Pada saat ingin menjalankan emulator di dalam Android Studio, pertama sekali harus memastikan aspek virtualization sesuai dimana sistem yang digunakan harus memenuhi persyaratannya, yakni ketentuan prosesor dan sistem operasi dari perangkat laptop/ PC yang digunakan dalam pembuatan project.

##### **Persyaratan Processor:**

- a. Untuk Prosesor Intel pastikan sudah mendukung Intel VT-x, Intel EM64T (Intel 64), dan Execute Disable (XD) Bit functionality.
- b. Untuk Prosesor AMD pastikan sudah support dengan AMD Virtualization (AMDV) dan Supplemental Streaming SIMD Extensions 3 (SSSE3).

##### **Persyaratan Sistem Operasi:**

- a. Untuk prosesor Intel dapat menjalankan Android Studio di sistem operasi Windows, Linux, maupun Mac
  - b. Untuk prosesor AMD hanya bisa menjalankan android Studio di sistem operasi Linux.
2. Hardware Accelerated Execution Manager (HAXM)

Dalam pengembangan aplikasi mobile dalam Android, Android Studio dan Android SDK menyediakan tool yang disebut AVD (Android Virtual Device) sehingga nantinya bisa menggunakan emulator Android untuk melakukan pengujian terhadap aplikasi mobile yang kita kembangkan. Supaya emulator Android bisa berjalan dengan cepat dan lancar di PC atau laptop yang digunakan, maka Intel menyediakan Hardware Accelerated Execution Manager (Intel HAXM) yang mampu mempercepat kinerja dari emulator Android termasuk akselerasi 3D melalui OpenGL. Untuk menggunakan Intel HAXM, maka ada beberapa syarat yang harus terpenuhi di PC atau laptop yang digunakan, diantaranya:



- a. Prosesor Intel dengan dukungan 64-bit (EM64T), virtualisasi (VT-x), dan Execute Disable (XD). Kadang-kadang fitur VT-x dan Execute Disable perlu diaktifkan dahulu melalui konfigurasi BIOS di motherboard.
- b. RAM 8GB sangat disarankan. Bila RAM di PC hanya 4GB, maka sebaiknya menggunakan HP Android aktual dan bukan emulator. Intel HAXM sendiri akan membutuhkan RAM sebesar 2GB untuk mendukung emulator Android.

HAXM diperlukan apabila sistem operasi yang digunakan adalah Windows atau Mac. Untuk menginstalnya, ikuti petunjuk berikut ini.

- a. Buka SDK Manager.
- b. Pilih SDK Update Sites, kemudian hidupkan Intel HAXM.
- c. Tekan OK.
- d. Cari berkas installer-nya di directory folder sdk computeranda, ~sdk\extras\intel\Hardware\_Accelerated\_Execution\_Manager\intelhaxmandroid.exe
- e. Jalankan installer dan ikuti petunjuknya sampai selesai.

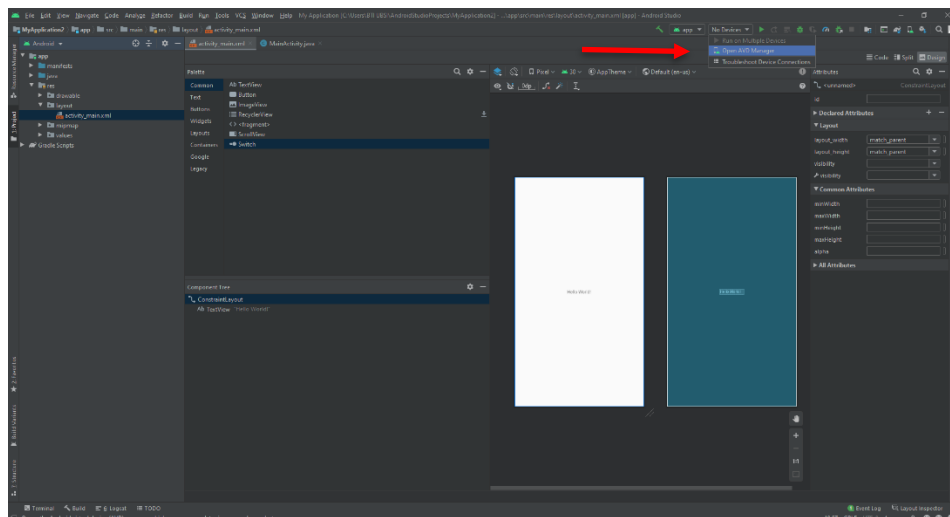
Untuk pengguna linux, maka untuk menggantikan HAXM lakukan instalasi Kernel-based Virtual Machine (KVM) Sistem operasi Linux dapat support accelerated virtual machine dengan menggunakan KVM. Untuk instal KVM, lakukan dengan menggunakan perintah berikut ini.

**\$ sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils ia32-libs-multiarch**

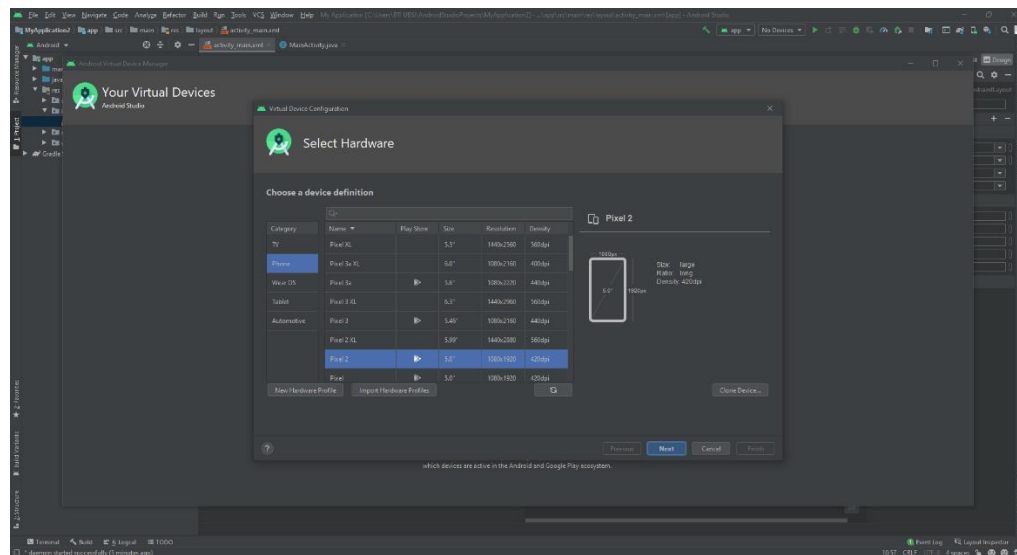
## B. Menjalankan Pada Emulator

Langkah-Langkah menggunakan Emulator:

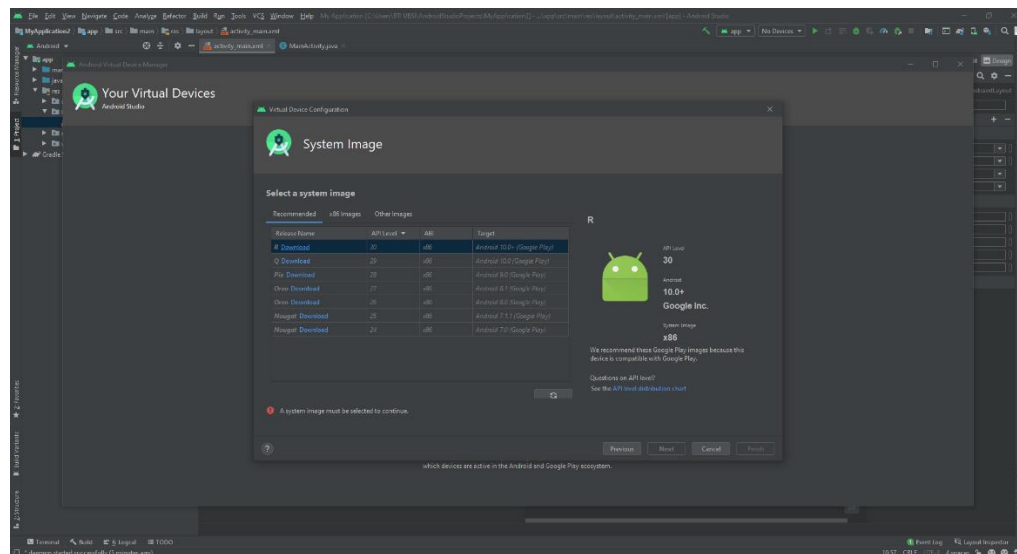
1. Pilih Open AVD Manager



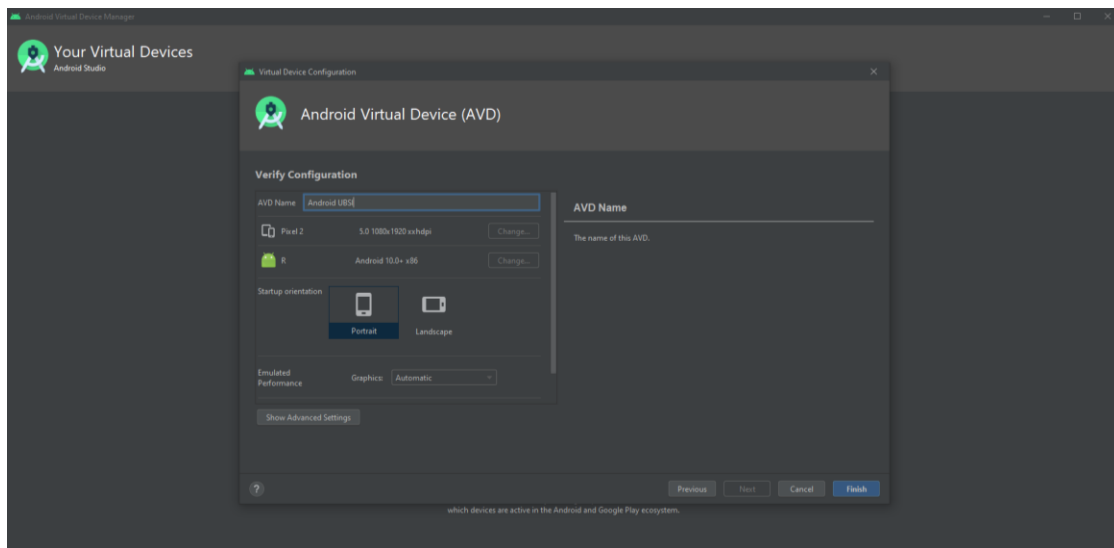
2. Pilih tipe perangkat yang akan digunakan sebagai emulator



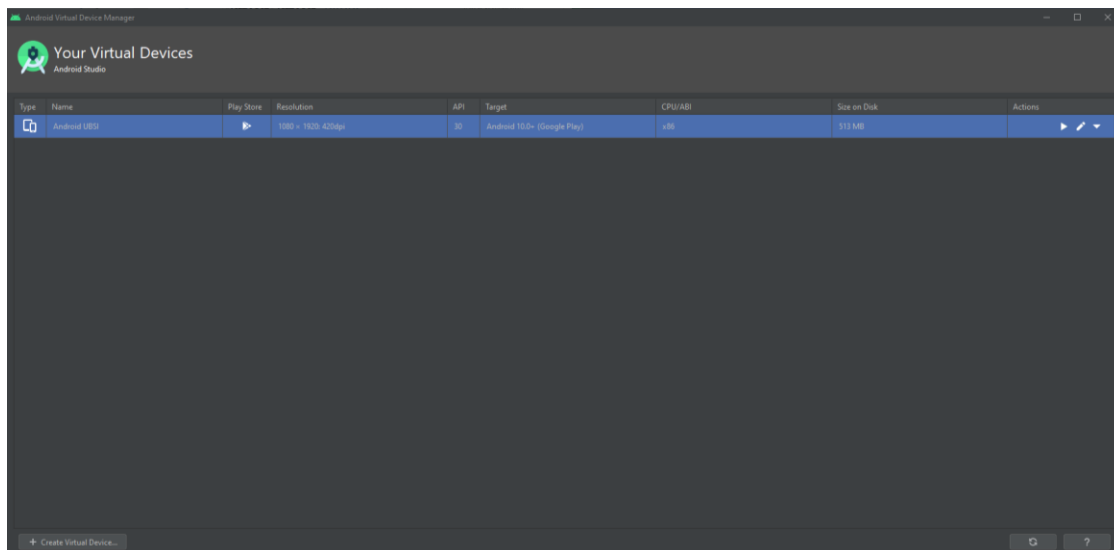
3. Pilih versi android yang akan diinstall pada perangkat emulator



4. Lalu klik Next, Setelah itu untuk AVD Name sesuaikan dengan keinginan (Contoh : Android UBSI)



5. Jika sudah berhasil maka akan muncul list emulator yang sudah dibuat



### C. Menjalankan Pada Device

Selain menggunakan emulator, menggunakan peranti smartphone dengan system operasi Android. Running dengan menggunakan peranti smartphone memiliki beberapa kelebihan jika dibandingkan dengan emulator diantaranya:

- a. Lebih cepat dan lebih mudah;
- b. Fitur seperti geo-location, push notif bisa digunakan;
- c. Bisa mengetahui daya serap baterai terhadap aplikasi;

- d. Aplikasi dapat berjalan dengan wajar dan bisa dianalisa ketika sudah sampai tangan pengguna.

Dengan menggunakan peranti smartphone asli, kita dapat memastikan bahwa aplikasi kita berjalan dengan wajar ketika sudah sampai di tangan pengguna. Kendala dari pendekatan ini adalah beragamnya model peranti yang ada di pasaran. Namun, pembahasan mengenai hal tersebut berada diluar cakupan kelas ini.

Mari ikuti langkah-langkah untuk menjalankan proses *run* atau *debugging*. Tampilan dari langkah berikut bisa dipastikan akan berbeda dengan peranti yang Anda pakai. Akan tetapi secara garis besar langkahnya akan sama.

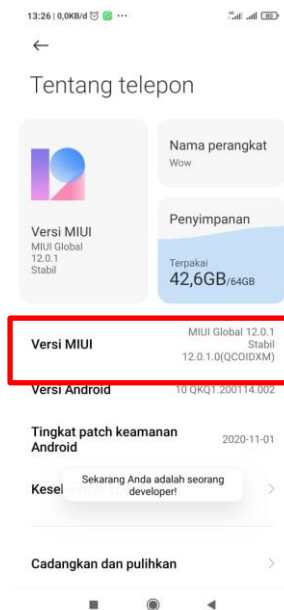
1. Pastikan peranti yang akan dipakai sesuai dengan target SDK atau paling tidak mendukung versi SDK terendah yang digunakan aplikasi.
2. Buka *setting* dan masuk ke dalam menu About. Pada halaman menu ini, Anda perlu menemukan informasi tentang Build number.
3. Kemudian tekan Build number sebanyak 7 kali
4. Kembali ke menu setting di awal dan akan muncul menu baru di bawah about yaitu Developer Options
5. Masuk ke menu Developer Options dan pastikan opsi USB Debugging Mode sudah aktif
6. Setelah selesai maka peranti dapat dihubungkan dengan Android Studio

**Catatan :** Beberapa vendor smartphone memiliki sistem operasi yang unik. Tampilan setting dan letak opsi bisa jadi tidak sama dengan gambar di atas.

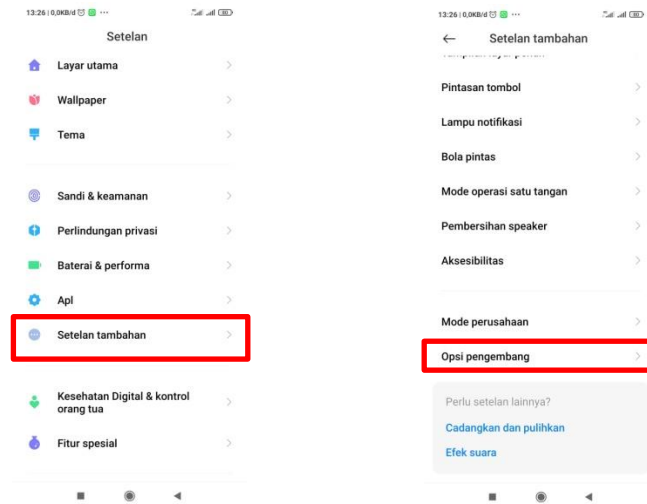
Beberapa vendor juga mengharuskan Anda untuk mengunduh *driver* khusus sebelum bisa menghubungkannya ke Android Studio. Kami sarankan untuk mengunjungi website / membaca petunjuk yang sesuai dengan vendor dari peranti yang digunakan.

### Contoh Pada Xiaomi Redmi Note 8

1. Pilih Pengaturan lalu klik Tentang telepon



2. Ketuk sebanyak Tujuh kali (7x) pada versi MIUI
3. Setelah mengaktifkan mode pengembang pilih pengaturan tambahan lalu pilih opsi pengembang



4. Pada opsi pengembang hidubkan USB Debuging dan Install via USB



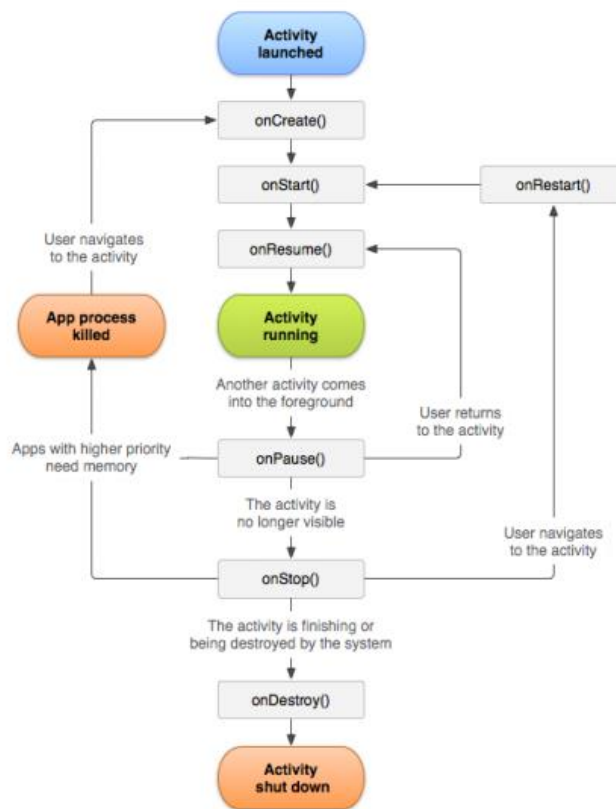
## BAB 4

### KOMPONEN DASAR ANDROID-ACTIVITY

#### A. Pengenalan Activity

Activity merupakan satu komponen yang berhubungan dengan pengguna. Activity menangani *window* (tampilan) mana yang akan di tampilkan ke dalam *interface* (antarmuka). Activity merupakan komponen penting dari aplikasi Android, dan cara aktivitas diluncurkan dan disatukan adalah bagian mendasar dari model aplikasi platform. Tidak seperti paradigma pemrograman di mana aplikasi yang diluncurkan dengan metode `main()`, sistem Android memulai kode dalam contoh Activity dengan menerapkan metode callback tertentu yang sesuai dengan tahap-tahap tertentu pada siklus hidup. Activity merupakan salah satu komponen penting Android yang memiliki daur hidup (*life cycle*) dalam sebuah *stack* pada *virtual sandbox* yang disiapkan oleh *Dalvik Virtual Machine* (DVM) atau *Android Runtime* (ART) yang bersifat *last in first out*. Pada implementasinya, activity selalu memiliki satu layout *user interface* dalam bentuk berkas xml. Suatu aplikasi Android bisa memiliki lebih dari satu activity dan harus terdaftar di berkas **AndroidManifest.xml** sebagai sub aplikasi.

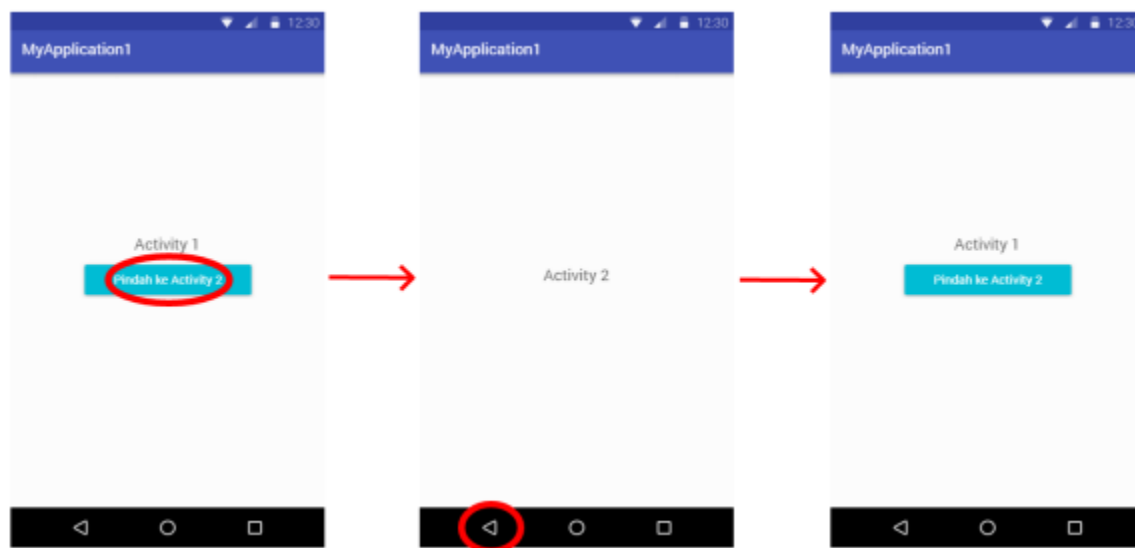
#### Activity Lifecycle



Developer yang baik harus mengetahui secara detail tentang *life cycle* sebuah activity. Terutama untuk melakukan aksi yang tepat, saat terjadi perubahan *state* activity. *Callback*

*methods* yang ada dapat digunakan untuk melakukan beragam proses terkait state dari activity. Misalnya melakukan semua inisialisasi komponen di **onCreate()**, melakukan *disconnect* terhadap koneksi ke *server* pada **onStop()** atau **onDestroy()** dan lain sebagainya. Pemahaman yang baik tentang daur hidup activity akan membuat implementasi rancangan aplikasi Anda menjadi lebih baik. Hal ini juga akan meminimalisir terjadinya *error/bug/force close* yang tidak diinginkan.

## Last In, First Out (LIFO)



Gambar 1	Gambar 2	Gambar 3
Aktif : Activity 1  <b>onCreate() → onStart() → onResume →</b>	Aktif : Activity 2  Stack append: Activity 2 [ <b>onResume()</b> ]	Aktif : Activity 1  <b>onStop() → onRestart() → onStart () → onResume()</b>
Aksi: Klik Button1 (Pindah)	Aksi : Klik Hardware Back Button	Aktif : Activity 1
Stack append: Activity 1 [ <b>onStop()</b> ]	Activity 2 [ finish() ]  Stack pop: Activity 2 [ <b>onDestroy()</b> ]	

### 1. Gambar 1

Jika Anda memiliki sebuah aplikasi yang terdiri dari 2 activity, maka activity pertama akan dijalankan setelah pengguna meluncurkan aplikasi melalui ikon aplikasi di layar

device. Activity yang ada saat ini berada pada posisi activity *running* setelah melalui beberapa *state* onCreate (*created*) → onStart (*started*) → onResume (*resumed*) dan masuk ke dalam sebuah stack activity.

Bila pada activity pertama Anda menekan sebuah tombol untuk menjalankan activity kedua, maka posisi *state* dari activity pertama berada pada posisi *stop*. Saat itu, *callback* onStop() pada activity pertama akan dipanggil.

Ini terjadi karena activity pertama sudah tidak berada pada layar *foreground* / tidak lagi ditampilkan. Semua informasi terakhir pada activity pertama akan disimpan secara otomatis.

Sementara itu, **activity** kedua masuk ke dalam *stack* dan menjadi activity terakhir yang masuk.

## 2. Gambar 2

Activity kedua sudah muncul di layar sekarang. Ketika Anda menekan tombol *back* pada *physical button* menu utama atau menjalankan metode finish(), maka activity kedua Anda akan dikeluarkan dari *stack*.

Pada kondisi di atas, *state* activity kedua akan berada pada *destroy*. Oleh karenanya, metode onDestroy() akan dipanggil.

Kejadian keluar dan masuk *stack* pada proses di atas menandakan sebuah model *Last In, First Out*. Activity kedua menjadi yang terakhir masuk *stack* (*Last In*) dan yang paling pertama keluar dari *stack* (*First Out*).

## 3. Gambar 3

Activity Pertama akan dimunculkan kembali di layar setelah melalui beberapa *state* dengan rangkaian *callback method* yang terpanggil, onStop → onRestart → onStart → onResume.

## Saving Activity State

Ketika sebuah activity mengalami *pause* kemudian *resume*, maka *state* dari sebuah activity tersebut dapat terjaga. Sebabnya, obyek activity masih tersimpan di *memory* sehingga dapat dikembalikan *state*-nya.

Dengan menjaga *state* dari activity, maka ketika activity tersebut ditampilkan, kondisinya akan tetap sama dengan kondisi sebelumnya.

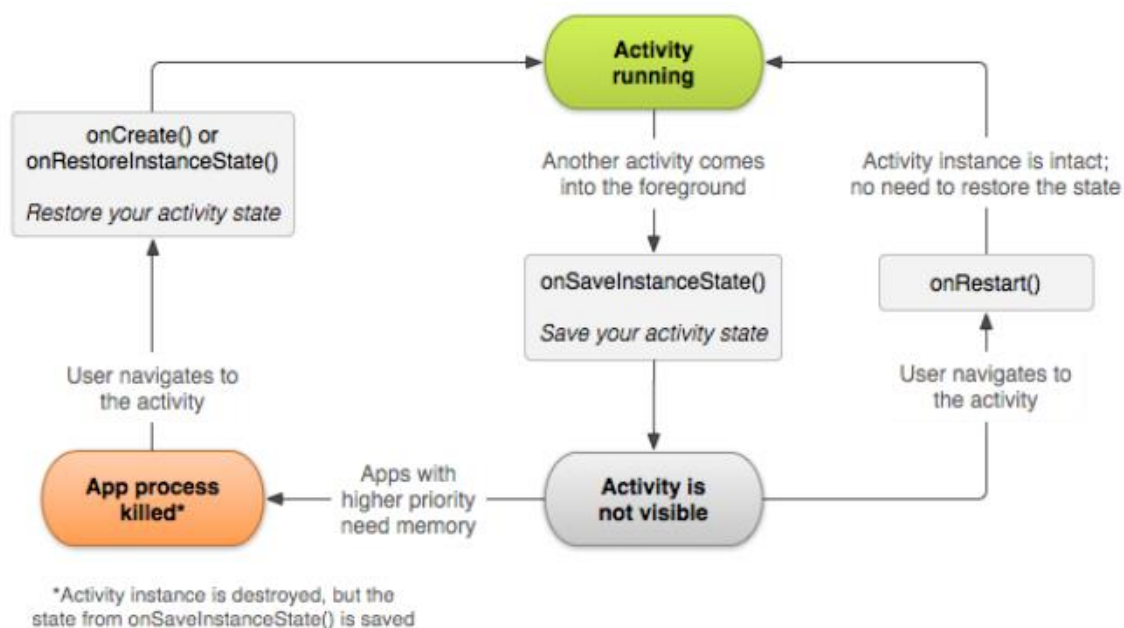
Akan tetapi ketika sistem menghancurkan activity untuk keperluan memori misalnya karena memori habis, maka obyek activity dihancurkan. Alhasil, ketika activity ingin ditampilkan kembali diperlukan proses *re-create* activity yang dihancurkan tadi.



Kejadian di atas adalah hal yang lumrah terjadi. Oleh karena itu, perubahan yang terjadi pada activity perlu disimpan terlebih dahulu sebelum ia dihancurkan. Disinilah metode `onSaveInstanceState()` digunakan.

Dalam `onSaveInstanceState` terdapat bundle yang dapat digunakan untuk menyimpan informasi. Informasi dapat disimpan dengan memanfaatkan fungsi seperti `putString()` dan `putInt()`.

Ketika activity di-*restart*, bundle akan diberikan kepada metode `onCreate` dan `onRestoreInstanceState`. Bundle tersebut akan dimanfaatkan untuk mengembalikan kembali perubahan yang telah terjadi sebelumnya.



Proses penghancuran activity dapat juga terjadi ketika terdapat perubahan konfigurasi seperti perubahan orientasi layar (*portrait-landscape*), *keyboard availability*, dan perubahan bahasa. Penghancuran ini akan menjalankan *callback method* `onDestroy` dan kemudian menjalankan `onCreate`. Penghancuran ini dimaksudkan agar activity dapat menyesuaikan diri dengan konfigurasi baru yang muncul pada kejadian-kejadian sebelumnya.

Hal yang perlu diingat ketika menggunakan `onSaveInstanceState` adalah untuk tidak menyimpan data yang besar pada bundle. Contohnya, hindari penyimpanan data *bitmap* pada bundle. Bila data pada bundle berukuran besar, proses serialisasi dan deserialisasi akan memakan banyak memori.

## B. Latihan Sederhana penggunaan Activity

### Tujuan

Penerapan komponen activity pada aplikasi pertama yang akan bangun. Harapannya aktifitas ini dapat memberi gambaran yang jelas tentang cara kerja activity.

### Logika Dasar

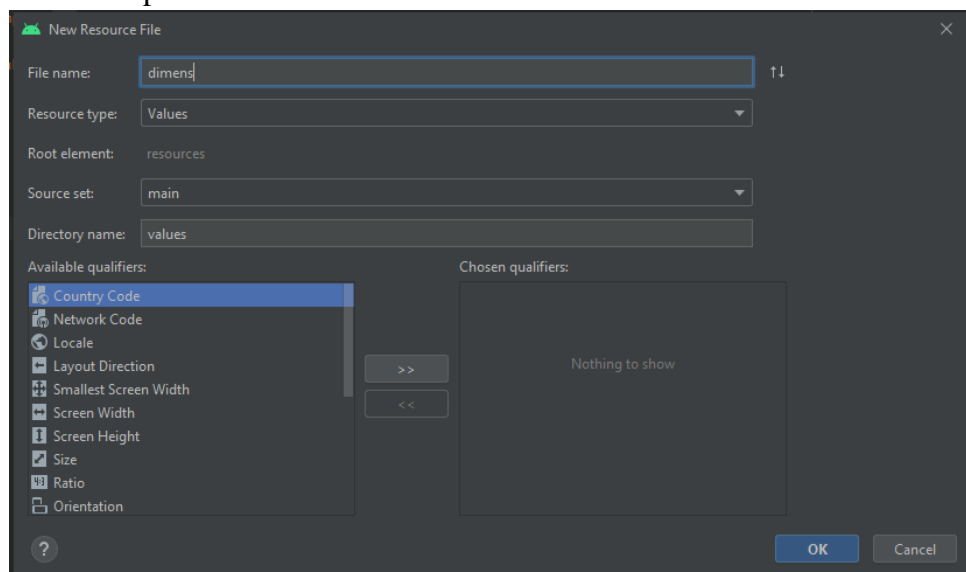
Melakukan input ke dalam obyek `TextBox` → melakukan validasi input → melakukan perhitungan volume balok ketika tombol hitung diklik maka tampil hasil perhitungan.

Berikut flow umumnya.

1. **Kode Layout** untuk user interface aplikasi
2. **Kode Activity** untuk menambahkan logika aplikasi

### Langkah Pengerjaan :

1. Buat Project baru di Android Studio dengan kriteria sebagai berikut:
  - Nama Project : LatihanActivity
  - Target & Minimum Target SDK : Phone and Tablet, API level 16
  - Tipe Activity : Empty Activity
  - Activity Name : MainActivity
2. Buatlah file `dimens.xml` di dalam `res/values/dimens` dengan cara klik kanan pada folder `res` lalu pilih `New` dan pilih `Android Resource Files`



3. Jika file `dimens.xml` sudah dibuat, lalu tambahkan kode berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
</resources>
```

### Menambahkan Code Sederhana Pada Layout Activity

4. Pada tab file `activity_main.xml` pada *workspace* Anda (`res/Layout/activity_main.xml`), dan tambahkan baris-baris berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:orientation="vertical"
    tools:context="bsi.ac.id.MainActivity">

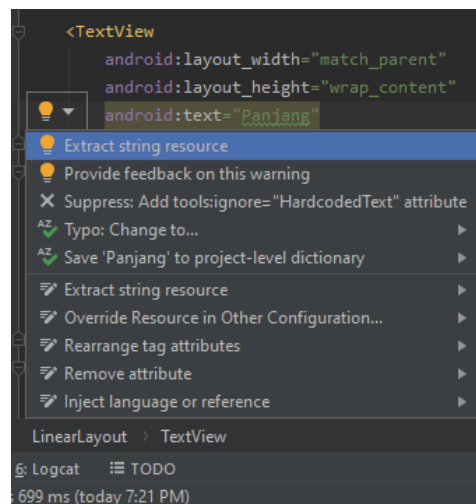
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/panjang"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
    <EditText
        android:id="@+id/edit_lenght"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:lines="1"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/lebar"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
    <EditText
        android:id="@+id/edit_width"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:lines="1"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/tinggi"
```

```

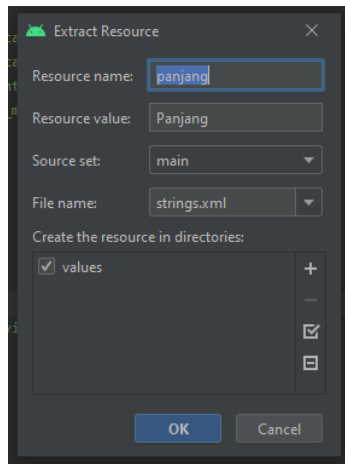
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
<EditText
    android:id="@+id/edit_height"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal"
    android:lines="1"
    android:layout_marginBottom="@dimen/activity_vertical_margin"/>
<Button
    android:id="@+id/btn_calculate"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/hitung"
    android:layout_marginBottom="@dimen/activity_vertical_margin"/>
<TextView
    android:id="@+id/tv_result"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/hasil"
    android:gravity="center"
    android:textSize="24sp"
    android:textStyle="bold"
    android:layout_marginBottom="@dimen/activity_vertical_margin"/>
</LinearLayout>

```

5. Akan muncul *warning* pada atribut `android:text` pada layout tersebut. Ini karena kita melakukan *hardcoding* pada nilai *string*-nya. Mari kita hilangkan *code warning* tersebut dengan menekan **Alt+Enter** pada attribut `android:text`. Akan muncul dialog seperti ini, pilihlah *extract string resource*.



6. Kemudian akan muncul dialog seperti dibawah, sesuaikan dengan nama yang ada



7. Fungsi *extract string resource* akan secara otomatis menambahkan nilai dari android:text ke dalam berkas res/values/strings.xml. Lakukan hal yang sama pada view lainnya hingga tidak ada warning lagi. Jika kita buka file strings.xml maka isinya menjadi :

```
<resources>
  <string name="app_name">My Activity</string>
  <string name="panjang">Panjang</string>
  <string name="lebar">Lebar</string>
  <string name="tinggi">Tinggi</string>
  <string name="hitung">Hitung</string>
  <string name="hasil">Hasil</string>
</resources>
```

## Menambahkan Kode Logika Sederhana pada MainActivity

1. Selanjutnya setelah selesai. Lanjutkan dengan membuka berkas MainActivity.java dan lanjutkan coding dibawah ini.

```
package bsi.ac.id;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import org.w3c.dom.Text;

public class MainActivity extends AppCompatActivity implements View.OnClickListener
{
    private EditText edtWidth, edtHeight, edtLenght;
```

```

private Button btnCalculate;
private TextView tvResult;
private static final String STATE_HASIL = "state_hasil";

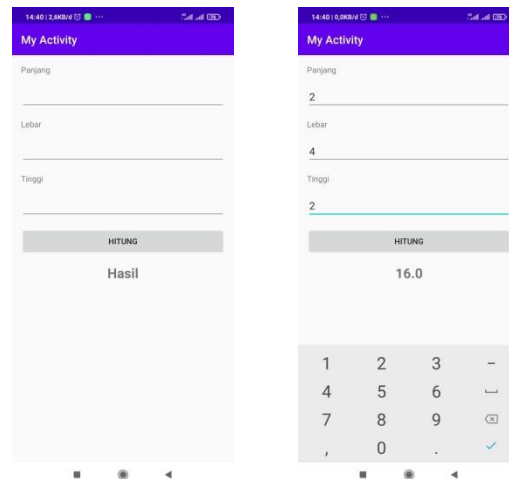
@Override
protected void onSaveInstanceState(Bundle outState) {
    outState.putString(STATE_HASIL, tvResult.getText().toString());
    super.onSaveInstanceState(outState);
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    edtWidth=(EditText)findViewById(R.id.edit_width);
    edtHeight=(EditText)findViewById(R.id.edit_height);
    edtLenght=(EditText)findViewById(R.id.edit_lenght);
    btnCalculate=(Button)findViewById(R.id.btn_calculate);
    tvResult=(TextView)findViewById(R.id.tv_result);
    btnCalculate.setOnClickListener(this);
    if (savedInstanceState !=null) {
        String hasil = savedInstanceState.getString(STATE_HASIL);
        tvResult.setText(hasil);
    }
}

@Override
public void onClick(View v) {
    if(v.getId()==R.id.btn_calculate){
        String lenght = edtLenght.getText().toString().trim();
        String width = edtWidth.getText().toString().trim();
        String height = edtHeight.getText().toString().trim();
        boolean isEmptyFields = false;
        if (TextUtils.isEmpty(lenght)){
            isEmptyFields=true;
            edtWidth.setError("Field ini tidak boleh kosong");
        }
        if (TextUtils.isEmpty(width)){
            isEmptyFields=true;
            edtWidth.setError("Field ini tidak boleh kosong");
        }
        if (TextUtils.isEmpty(height)){
            isEmptyFields = true;
            edtHeight.setError("Field ini tidak boleh kosong");
        }
        if(!isEmptyFields){
            double l = Double.parseDouble(lenght);
            double w = Double.parseDouble(width);
            double h = Double.parseDouble(height);
            double volume = l*w*h;
            tvResult.setText(String.valueOf(volume));
        }
    }
}
}

```

## Hasil



## Pembahasan :

### Pembahasan tentang layout xml

```
<?xml version="1.0" encoding="utf-8"?>
```

Baris ini mengidentifikasi bahwa berkas ini berformat xml.

```
xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:tools="http://schemas.android.com/tools"
```

Kode tersebut sebagai tanda namespace yang digunakan dalam keseluruhan berkas xml ini.

Dalam aplikasi ini digunakan beberapa komponen user interface yang disebut view, diantaranya:

- TextView : Komponen view untuk menampilkan teks ke layar
- EditText : Komponen view untuk memberikan input teks
- Button : Komponen View untuk melakukan sebuah aksi klik
- LinearLayout : Komponen view bertipe viewgroup yang menjadi *parent* dari semua sub komponen view (sub view) di dalamnya. Komponen ini bersifat sebagai kontainer untuk komponen lain dengan orientasi secara vertikal atau horizontal.

Cara membaca :

```
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="@string/panjang"  
    android:layout_marginBottom="@dimen/activity_vertical_margin"/>
```

Komponen diatas adalah sebuah TextView. Perhatikan gambar di bawah ini, warna ungu menandakan *namespace* yang digunakan; warna biru adalah atribut dari komponen dan warna hijau adalah nilai dari atribut. Penjelasannya seperti dibawah ini:

**<TextView**  
Nama Komponen View

**android:layout\_width="match\_parent"**  
Namespace      Attribute      Value

**match\_parent:** Ini berarti bahwa ukuran dimensi sebuah View disesuaikan dengan ukuran layar secara horizontal jika pada `layout_width` dan vertikal jika pada `layout_height`.

**wrap\_content :** Ini berarti bahwa ukuran dimensi sebuah View disesuaikan dengan ukuran konten didalamnya baik secara horizontal pada `layout_width` dan vertikal jika pada `layout_height`.

**@dimen/activity\_vertical\_margin:** Value `activity_vertical_margin` berasal dari berkas **dimens.xml** yang bisa Anda lihat dengan cara menekan dan tahan tombol **Ctrl** (atau command) + arahkan kursor keatasnya dan kemudian klik sekali. Penggunaan *centralize resource value* akan memudahkan Anda sewaktu mengembangkan aplikasi Android. Cara tersebut digunakan agar Anda tidak menulis nilai yang sama berulang-ulang.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
</resources>
```

**Apa itu '@+id/' ?**

Salah satu contoh penerapan penggunaan **@+id/** sebagai berikut:

```
<Button
    android:id="@+id/btn_calculate"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/hitung"
    android:layout_marginBottom="@dimen/activity_vertical_margin"/>
```

Penjelasannya adalah :

```
android:id="@+id/btn_calculate"
```

Jika kita memberikan id pada sebuah view maka kita telah memberikan *identifier* untuk view tersebut. Pemberian id ini dimaksudkan agar kita bisa melakukan manipulasi/pengendalian pada *level logic* di komponen seperti activity atau fragment.



Id di atas akan diciptakan di berkas R.java dan disimpan dalam bentuk *hexa* bertipe data integer **public static final int btn\_calculate=0x7f0b0057.**

## Pembahasan tentang Logika Kode

### 1. Public class MainActivity extends AppCompatActivity

Menandakan bahwa kelas Java di atas merupakan sebuah activity karena *inherit* ke *superclass* bernama AppCompatActivity.

### 2. Implementasi View.OnClickListener

Ini adalah *listener* yang kita implementasikan untuk memantau kejadian klik pada komponen tombol (button).

### 3. private EditText edtWidth, edtHeigh, EdtLength;

### 4. private Button btnCalculate;

### 5. private Text tvResult;

Kode di atas mendeklarasikan semua komponen view yang akan dimanipulasi. Kita deklarasikan secara global agar bisa dikenal di keseluruhan bagian kelas.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    edtWidth=(EditText)findViewById(R.id.edit_width);
    edtHeight=(EditText)findViewById(R.id.edit_height);
    edtLenght=(EditText)findViewById(R.id.edit_lenght);
    btnCalculate=(Button)findViewById(R.id.btn_calculate);
    tvResult=(TextView)findViewById(R.id.tv_result);
    btnCalculate.setOnClickListener(this);
    if (savedInstanceState !=null) {
        String hasil = savedInstanceState.getString(STATE_HASIL);
        tvResult.setText(hasil);
    }
}
```

Metode onCreate() merupakan metode utama pada activity. Di sinilah kita dapat mengatur layout xml. Semua proses inisialisasi komponen yang digunakan akan dijalankan disini.. Pada metode ini kita casting semua komponen view yang kita telah deklarasikan sebelumnya, agar dapat kita manipulasi.

```
setContentView(R.layout.activity_main);
```

Maksud baris diatas adalah kelas MainActivity akan menampilkan tampilan yang berasal dari layout activity\_main.xml.

```
edtWidth=(EditText)findViewById(R.id.edit_width);
```

Maksud dari baris diatas adalah obyek editText edtWidth disesuaikan (cast) dengan komponen editText ber-ID edt\_width di layout xml melalui metode findViewById().

```
btnCalculate=(Button)findViewById(R.id.btn_calculate);
```

Kita memasang *event click listener* untuk obyek btnCalculate sehingga sebuah aksi dapat dijalankan ketika obyek tersebut diklik. Keyword this merujuk pada obyek Activity saat ini yang telah mengimplementasikan listener OnClickListener sebelumnya.

## Pembahasan saveInstanceState

```
@Override
protected void onSaveInstanceState(Bundle outState) {
    outState.putString(STATE_HASIL, tvResult.getText().toString());
    super.onSaveInstanceState(outState);
}
```

Perhatikan metode onSaveInstanceState. Di dalam metode tersebut, hasil perhitungan yang ditampilkan pada tvResult dimasukkan pada bundle kemudian disimpan isinya. Setelah onSaveInstanceState berhasil dijalankan, maka activity akan melakukan proses onDestroy dan menjalankan kembali onCreate secara otomatis.

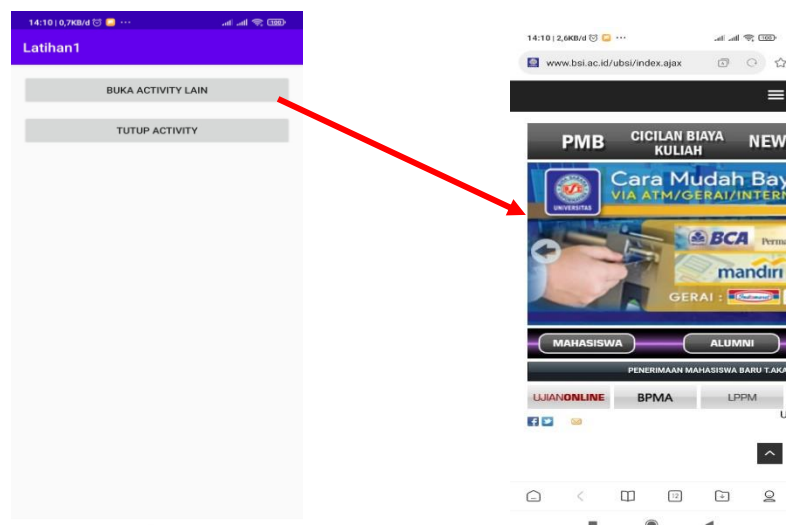
```
if (savedInstanceState !=null) {
    String hasil = savedInstanceState.getString(STATE_HASIL);
    tvResult.setText(hasil);
}
```

Pada onCreate inilah kita menggunakan nilai bundle yang telah kita simpan sebelumnya pada onSaveInstanceState. Nilai tersebut kita isikan kembali pada tvResult.

## Tugas 1

Buatlah aplikasi android yang menggambarkan proses activity dengan ketentuan sebagai berikut:

1. Buatlah layout yang terdapat 2 tombol yang terdiri dari :
  - a. Buka Activity Lain
  - b. Tutup Activity
2. Jika klik pada Buka Activity Lain = Maka akan membuka halaman [www.bsi.ac.id](http://www.bsi.ac.id)
3. Jika klik pada Tutup Activity = Maka aplikasi akan keluar



## **BAB 5**

### **INTENT**

#### **A. Pengenalan Intent**

Intent adalah mekanisme untuk melakukan sebuah *action* dan komunikasi antar komponen aplikasi misal activity, services, dan broadcast receiver. Ada tiga penggunaan umum intent dalam aplikasi Android yaitu:

- Memindahkan satu activity ke activity lain dengan atau tidak membawa data.
- Menjalankan background service, misalnya melakukan sinkronisasi ke server dan menjalankan proses berulang (*periodic/scheduler task*).
- Mengirimkan obyek broadcast ke aplikasi yang membutuhkan. Misal, ketika aplikasi membutuhkan proses menjalankan sebuah background service setiap kali aplikasi selesai melakukan *booting*. Aplikasi harus bisa menerima obyek *broadcast* yang dikirimkan oleh sistem Android untuk *event booting* tersebut.

Intent memiliki dua bentuk yaitu:

##### **1. Explicit Intent**

Adalah tipe Intent yang digunakan untuk menjalankan komponen dari dalam sebuah aplikasi. Explicit intent bekerja dengan menggunakan nama kelas yang dituju misal : `ac.id.nusamandiri.activity.DetailActivity`. Umumnya intent ini digunakan untuk mengaktifkan komponen pada satu aplikasi.

##### **2. Implicit Intent.**

Adalah tipe intent yang tidak memerlukan detail nama kelas yang ingin diaktifkan. Model ini memungkinkan komponen dari aplikasi lain bisa merespon *request* intent yang dijalankan.

Penggunaan tipe intent ini umumnya diperuntukan untuk menjalankan fitur/fungsi dari komponen aplikasi lain. Contohnya ketika kita membutuhkan fitur untuk mengambil foto. Daripada membuat sendiri fungsi kamera, lebih baik kita menyerahkan proses tersebut pada aplikasi kamera bawaan dari peranti atau aplikasi kamera lain yang telah terinstal sebelumnya di peranti.

Hal yang sama misalnya ketika kita membutuhkan fungsi berbagi konten. Kita bisa memanfaatkan intent untuk menampilkan aplikasi mana saja yang bisa menangani fitur tersebut.

Implementasi implicit intent ini akan sangat memudahkan bagi pengembang agar tetap fokus pada proses bisnis inti dari aplikasi yang dikembangkan.

#### **B. Latihan Intent**

Latihan ini menitik beratkan pada implementasi intent untuk melakukan perpindahan dari activity ke activity lain, dengan atau tidak membawa data. Beberapa bagian dari latihan ini akan menjawab beberapa pertanyaan umum dalam pengembangan aplikasi Android sebagai berikut:

- Bagaimana berpindah dari satu activity ke activity lain?

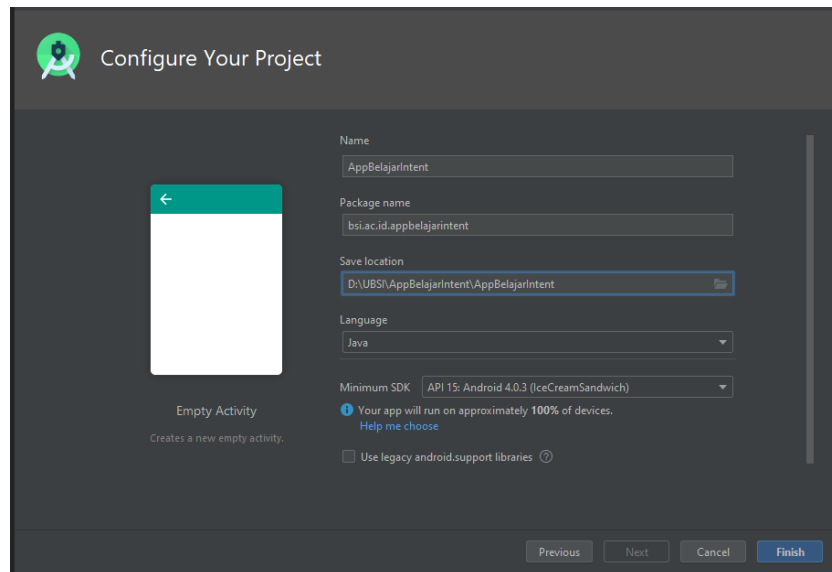
- Bagaimana berpindah dari satu activity ke activity lain dengan membawa data?
  - Single value dari suatu variabel.
  - Obyek model Plain Old Java Object (POJO).
- Menjalankan komponen di aplikasi lain untuk keperluan membuka *browser* atau melakukan pemanggilan melalui aplikasi telepon bawaan?
- Mengirimkan hasil nilai balik melalui Intent

## Logika Dasar

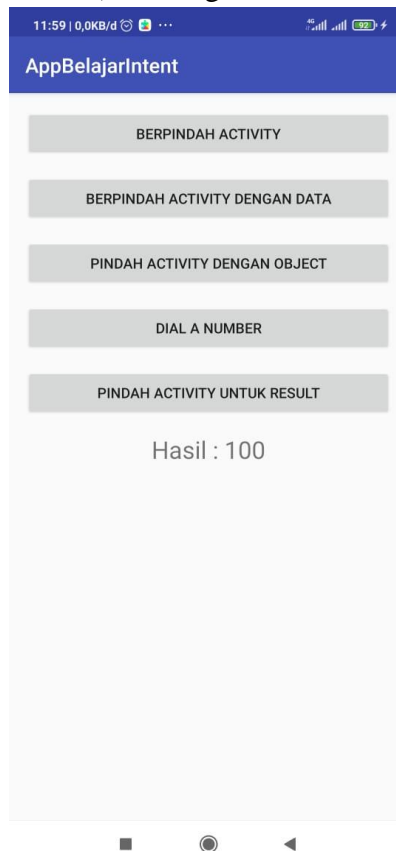
Berpindah dari satu Activity ke Activity lain dengan membawa data Activity asal akan mengirimkan data melalui Intent dan Activity tujuan akan menerima data yang dikirimkan.

## Latihan Intent Sederhana

1. Buat Project baru di Android Studio dengan kriteria sebagai berikut:
  - Nama Project : AppBelajarIntent
  - Target & Minimum Target SDK : Phone and Tablet, API level 16
  - Tipe Activity : Empty Activity
  - Activity Name : MainActivity



2. Selanjutnya buat antarmuka (interface) sesuai gambar berikut:



3. Pada tampilan tersebut memiliki 5 tombol dengan masing-masing fungsi yang berbeda dan 1 buah TextView untuk menampilkan data yang berasal dari Intent. Baik, kita akan mulai selangkah demi selangkah dimulai dari tombol yang paling atas. Sehingga `activity.main.xml` menjadi seperti ini:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context="bsi.ac.id.appbelajarintent.MainActivity">

    <Button
        android:id="@+id/btn_move_activity"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="BERPINDAH ACTIVITY"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>

</LinearLayout>
```

4. Buat file **dimens.xml** dalam res/value dan tuliskan kode dibawah ini

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <!-- Default screen margins, per the Android Design guidelines -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
</resources>
```

5. Untuk MainActivity.java tambahkan beberapa baris berikut

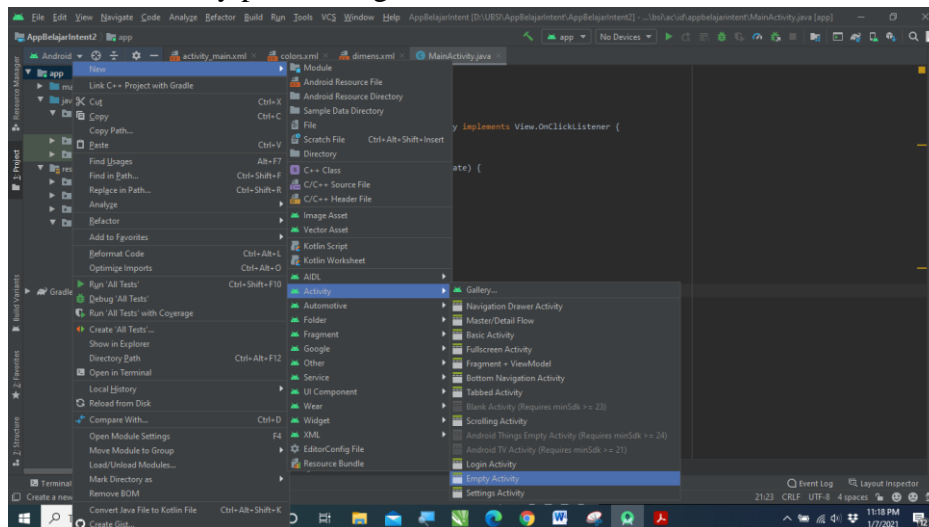
```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

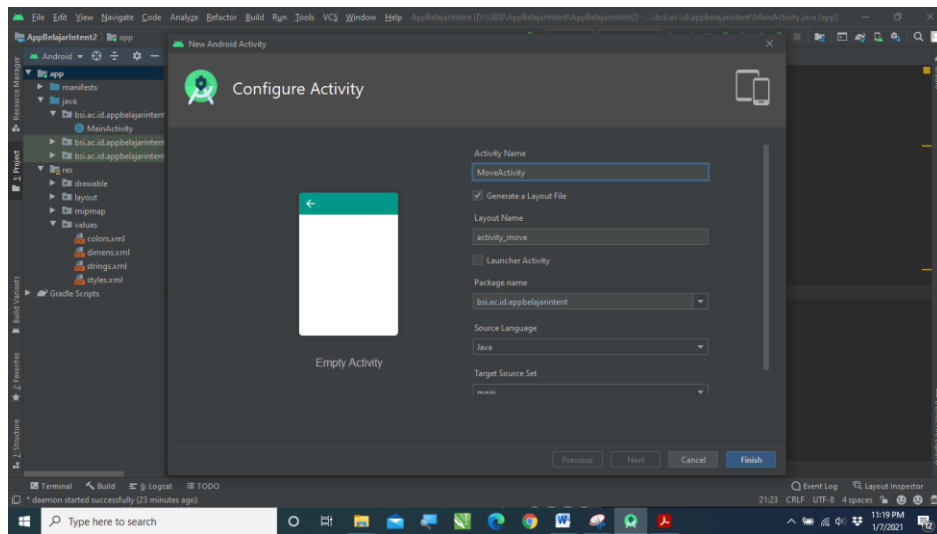
public class MainActivity extends AppCompatActivity implements
View.OnClickListener {
    private Button btnMoveActivity;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnMoveActivity.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {

    }
}
```

6. *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 package name → New → Activity → Empty Activity. Lalu isikan MoveActivity pada dialog. Ketika sudah klik Finish.





7. Untuk menandakan bahwa perpindahan activity berhasil, silakan tambahkan satu **TextView** dan kondisikan **activity\_move.xml** menjadi seperti berikut.

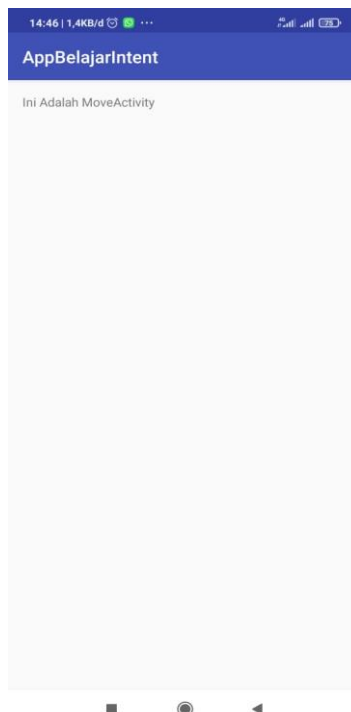
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="bsi.ac.id.appbelajarintent.MoveActivity">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Ini Adalah MoveActivity"/>
</RelativeLayout>
```

8. Setelah activity tujuan sudah berhasil diciptakan, sekarang saatnya menambahkan suatu 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 ‘Berpindah Activity’.

Hasil:





## Bedah Kode

Kita telah belajar bagaimana membuat suatu activity baru. Di materi sebelumnya, syarat suatu activity haruslah terdaftar pada berkas AndroidManifest.xml. Karena menggunakan Android Studio, proses pendaftaran tersebut dilakukan secara otomatis.

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

    <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=".MoveActivity"></activity>
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Perhatikan bahwa MoveActivity sudah teregistrasi di AndroidManifest.xml dan sekarang sudah aman jika kita melakukan perpindahan activity dari MainActivity ke MoveActivity. Umumnya, jika kita lupa meregistrasikan Activity baru ke dalam berkas AndroidManifest.xml akan terjadi *error* seperti ini "android.content.ActivityNotFoundException: Unable to find explicit activity class".

Berikutnya, apakah Anda memahami maksud dari baris ini?

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

Intent-filter merupakan mekanisme untuk menentukan bagaimana suatu activity dijalankan oleh *Android Runtime* (ART) atau *Dalvik Virtual Machine* (DVM).

```
<action android:name="android.intent.action.MAIN" />
```

Baris di atas bermakna bahwa MainActivity memiliki intent action utama dari keseluruhan activity yang ada di suatu Aplikasi.

```
<category android:name="android.intent.category.LAUNCHER" />
```

Kode di atas menandakan bahwa MainActivity akan dikategorikan sebagai *activity launcher*. Activity tersebut akan diluncurkan pertama kali ketika pengguna menekan ikon aplikasi pada peranti Android.

Selanjutnya.

```
Intent moveIntent = new Intent(MainActivity.this, MoveActivity.class);
```

Kita membuat suatu obyek Intent dengan cara seperti di atas dengan memberikan kelas Activity asal (MainActivity.this) dan kelas Activity tujuan (MoveActivity.class) pada konstruktor kelas Intent.

**Intent(MainActivity.this, MoveActivity.class);**

Kelas Asal

Kelas Tujuan

Kelas asal selalu menggunakan .this yang menandakan obyek kelas saat ini. Kelas tujuan selalu menggunakan .class.

startActivity(moveIntent) : metode ini akan menjalankan activity baru tanpa membawa data. Obyek intent yang diinputkan adalah obyek moveIntent yang akan menjalankan **MoveActivity**.

Pada modul ini kita telah berhasil memindahkan satu Activity ke Activity lain dengan tidak membawa data. Pada bagian selanjutnya kita akan membuat suatu Intent yang di dalamnya akan membawa data ke Activity tujuan.

### LATIHAN (Intent Explicit)

Selanjutnya kita akan membuat sebuah intent yang didalamnya akan membawa data ke activity tujuan. Anda Siap?

1. Buka **activity\_main.xml**, kemudian tambahkan satu tombol lagi dibawah tombol sebelumnya, sehingga kode activity\_main.xml kita menjadi seperti ini.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context="bsi.ac.id.appbelajarintent.MainActivity">

    <Button
        android:id="@+id/btn_move_activity"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="BERPINDAH ACTIVITY"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>

    <Button
        android:id="@+id/btn_move_activity_data"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="BERPINDAH ACTIVITY DENGAN DATA"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>

</LinearLayout>
```

2. Setelah selesai dengan penambahan pada berkas `activity_main.xml`, maka lanjutkan dengan menambahkan beberapa baris berikut di `MainActivity.java`.

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
    private Button btnMoveActivity;  
    private Button btnMoveWithDataActivity;  
    @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);  
        btnMoveWithDataActivity=(Button)findViewById(R.id.btn_move_activity_data);  
        btnMoveWithDataActivity.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;  
            case R.id.btn_move_activity_data:  
                break;  
        }  
    }  
}
```

3. Selanjutnya, buat Activity baru lagi seperti cara sebelumnya dan beri nama `MoveWithDataActivity`. Lalu, pada layout `activity_move_with_data.xml` kita tambahkan sebuah `TextView` ber-ID untuk menampilkan data yang dikirimkan dari Activity asal.

```
<?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="bsi.ac.id.appbelajarintent.MoveWithDataActivity">  
    <TextView  
        android:id="@+id/tv_data_received"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:text="Data recived"/>  
</RelativeLayout>
```

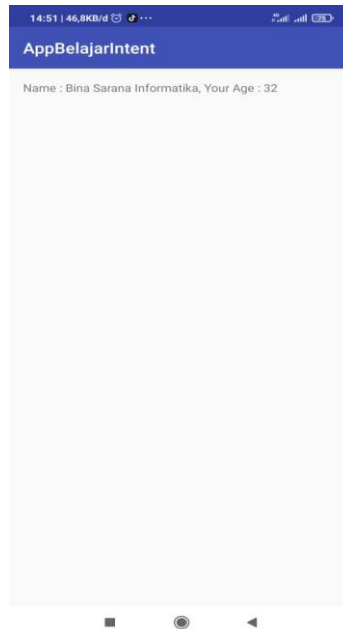
4. Selanjutnya pada MoveWithDataActivity.java kita kondisikan menjadi seperti berikut:

```
public class MoveWithDataActivity extends AppCompatActivity {
    public static String EXTRA_AGE = "extra_age";
    public static String EXTRA_NAME = "extra_name";
    private TextView tvDataReceived;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_move_with_data);
        tvDataReceived=(TextView)findViewById(R.id.tv_data_received);
        String name = getIntent().getStringExtra(EXTRA_NAME);
        int age = getIntent().getIntExtra(EXTRA_AGE, 0);
        String text = "Name : "+name+", Your Age : "+age;
        tvDataReceived.setText(text);
    }
}
```

5. Dan sekarang saatnya kita menambahkan obyek Intent pada MainActivity.java

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    private Button btnMoveActivity;
    private Button btnMoveWithDataActivity;
    @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);
        btnMoveWithDataActivity=(Button)findViewById(R.id.btn_move_activity_data);
        btnMoveWithDataActivity.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;
            case R.id.btn_move_activity_data:
                Intent moveWithDataIntent = new
Intent(MainActivity.this,MoveWithDataActivity.class);
                moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_NAME, "Bina
Sarana Informatika");
                moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_AGE,32);
                startActivity(moveWithDataIntent);
                break;
        }
    }
}
```

6. Silakan jalankan aplikasi Anda, kemudian coba klik pada tombol 'Pindah Activity dengan Data'. Seharusnya Anda sudah bisa memindahkan satu activity ke activity lain dengan membawa data.



## Bedah Kode

Pada bagian sebelumnya Anda sudah mempelajari bagaimana memindahkan satu activity ke activityactivity lain dengan membawa data. Dan itu sangat penting karena ketika kita mengembangkan suatu aplikasi Android yang kompleks. Akan ada banyak sekali activity yang terlibat. Lancarnya perpindahan activity dengan data menjadi hal yang penting untuk memberikan pengalaman yang baik kepada pengguna.

```
Intent moveWithDataIntent = new Intent(MainActivity.this, MoveWithDataActivity.class);
moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_NAME, "Bina Sarana Informatika");
moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_AGE, 32);
startActivity(moveWithDataIntent);
```

Perbedaan mendasar antara memindahkan activity dengan membawa data dan tidak adalah dengan menempatkan data yang akan dikirim ke activity tujuan pada baris ini.

```
moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_NAME, "Bina Sarana Informatika");
moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_AGE, 32);
```

Kita memanfaatkan metode `putExtra()` untuk mengirimkan data bersamaan dengan obyek `Intent`. Sedangkan metode `putExtra()` itu sendiri merupakan metode yang menampung pasangan *key-value* dan memiliki beberapa pilihan tipe *input* seperti berikut :

Hampir semua tipe data untuk *input value* didukung oleh metode `putExtra()`.

```
moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_NAME, "Bina Sarana Informatika");
```

Name : `MoveWithDataActivity.EXTRA_NAME` dimana `EXTRA_NAME` adalah variabel *static* bertipe data string dan bernilai “extra\_name” pada `MoveWithDataActivity.java`. Penentuan nilai untuk *key parameter* untuk intent adalah bebas, di sini kami merekomendasikan format terbaik yang biasa diimplementasikan.

Value : Bina Sarana Informatika.

Di sini kita akan mengirimkan data bertipe string ke `MoveWithDataActivity`. Di dalam `MoveWithdataActivity` kita akan mengambil nilai data berdasarkan *key* yang dikirimkan dengan menggunakan metode `getIntent().getStringExtra(key)`. Implementasinya sebagai berikut:

```
String name = getIntent().getStringExtra(EXTRA_NAME);
```

Catatan : *Key* yang dikirimkan melalui `putExtra()` harus sama dengan *key* sewaktu mengambil nilai dari data yang dikirimkan melalui `getStringExtra()`.

Dalam konteks di atas, *key* yang digunakan untuk mengirim dan mengambil nilai data adalah sama, yaitu `EXTRA_NAME` (yang bernilai “extra\_name”). Nilai dari data yang dikirimkan melalui intent disimpan ke dalam variabel `name` bertipe data string.

Begitu juga dengan,

```
int age = getIntent().getIntExtra(EXTRA_AGE, 0);
```

Nilai dari variabel `age` yang bertipe data integer berasal dari `getIntent().getIntExtra(Key, nilai default)`. *Key* yang digunakan untuk mengirimkan dan mengambil data adalah `EXTRA_AGE` (yang bernilai “extra\_age”).

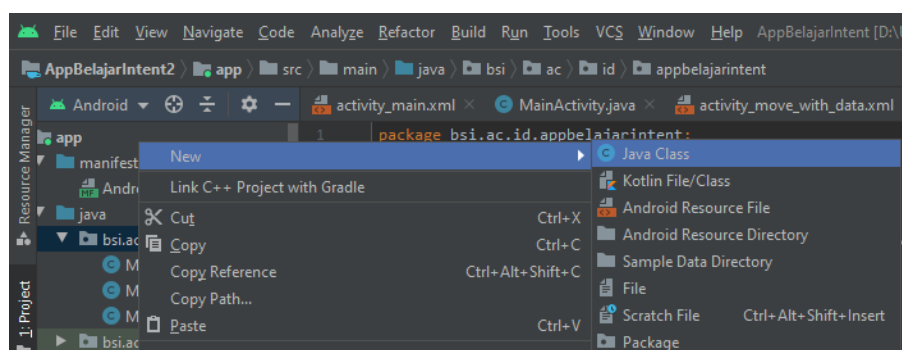
Selanjutnya, kita akan membuat sebuah intent yang di dalamnya akan membawa data dengan Plain Old Java Object (POJO) ke activity tujuan.

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*.

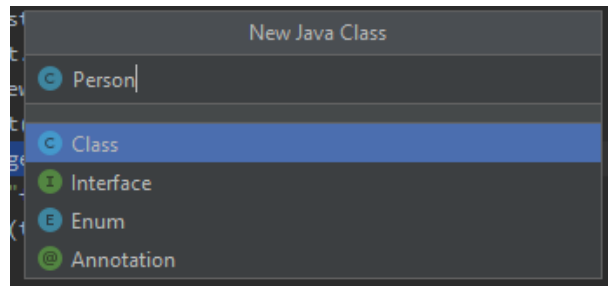
Lalu apa kegunaan obyek POJO? POJO akan membantu kita saat aplikasi semakin kompleks. Contohnya POJO bisa kita gunakan untuk melakukan koneksi ke server untuk *request* API atau akses ke *database* lokal dengan SQLite.

Selain POJO, Android juga memiliki obyek *parcelable* POJO. Obyek ini akan lebih memudahkan kita untuk melakukan pengiriman data dari satu activity ke activity lainnya.

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



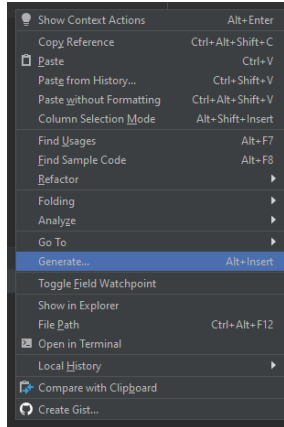
2. Pada dialog, isikan nama kelas dengan 'Person'. Lalu tekan Enter



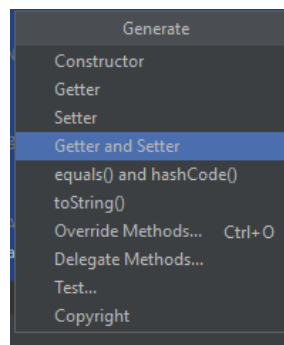
3. Setelah selesai terbentuk kelas Person, kondisikan kelas tersebut menjadi seperti ini.

```
public class Person {  
    private String name;  
    private int age;  
    private String email;  
    private String city;  
    public String getName() {  
        return name;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
    public int getAge() {  
        return age;  
    }  
    public void setAge(int age) {  
        this.age = age;  
    }  
    public String getEmail() {  
        return email;  
    }  
    public void setEmail(String email) {  
        this.email = email;  
    }  
    public String getCity() {  
        return city;  
    }  
    public void setCity(String city) {  
        this.city = city;  
    }  
}
```

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.



4. Pilih Getter and Setter.

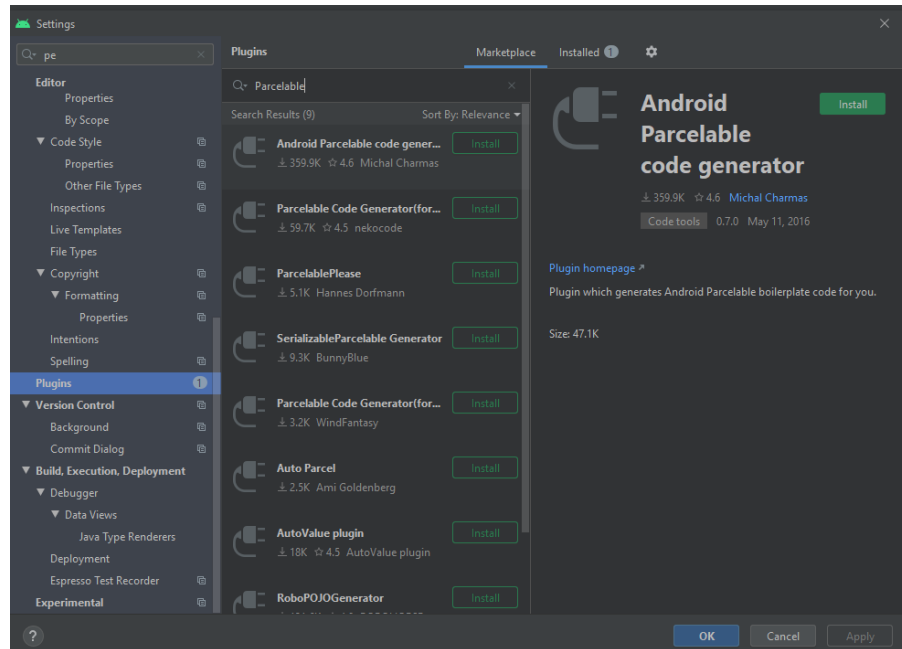


5. Pilih semua variabel dengan cara Ctrl+A pada windows dan Command+A pada mac. Bila sudah, maka klik OK. Secara otomatis metode *setter* dan *getter* masing-masing variabel akan dibuatkan.

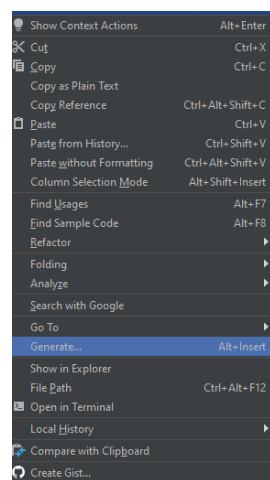
Selanjutnya kita akan menambahkan *plugin* Android Parcelable Generator pada Android Studio. *Plugin* ini berguna untuk mengubah kelas POJO menjadi obyek Parcelable. Langkah-langkahnya sebagai berikut :

1. Buka settings dengan cara File → Settings - Plugin pada Windows. Sedangkan pada Mac OS Pilih menu Plugin. Dan inputkan "Android Parcelable" pada search box kemudian tekan Enter.
2. Klik browse, dengan ini kita memerintahkan Android Studio untuk melakukan pencarian *plugin* di *server*. Setelah ketemu Android Parcelable code generator, klik Install.

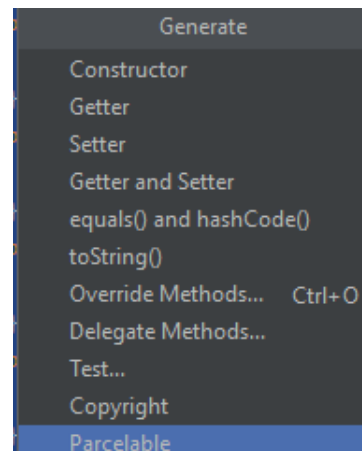




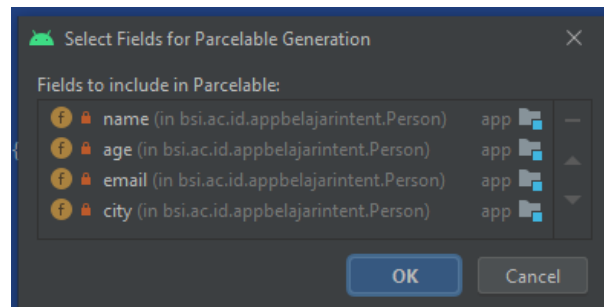
3. Biarkan Android Studio melakukan proses instalasi *plugin*. Bila sukses, maka Anda akan diminta untuk melakukan *restart* Android Studio yang sedang berjalan.
4. Klik OK dan jika ada pilihan untuk *restart* Android Studio maka pilih *restart*. Biarkan Android Studio untuk *restart* agar *plugin*-nya 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.
5. Selanjutnya, setelah plugin Parcelable generator terinstal saatnya kita implementasikan pada kelas Person.:



>>>>>>



6. Pada dialog box yang tampil, pilih semua variable yang ada lalu klik OK. 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.



7. Sekarang waktunya kita implementasikan pada intent. Namun sebelumnya tambahkan satu tombol lagi pada activity\_main.xml sebagai berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context="bsi.ac.id.appbelajarintent.MainActivity">

    <Button
        android:id="@+id/btn_move_activity"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="BERPINDAH ACTIVITY"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>

    <Button
        android:id="@+id/btn_move_activity_data"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="BERPINDAH ACTIVITY DENGAN DATA"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>

    <Button
        android:id="@+id/btn_move_activity_object"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Pindah Activity Dengan Object"
        android:layout_marginBottom="@dimen/activity_vertical_margin"
    />

</LinearLayout>
```

8. Setelah selesai dengan `activity_main.xml`, sekarang kita tambahkan juga di kelas `MainActivity.java` sehingga menjadi sebagai berikut :

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    private Button btnMoveActivity;
    private Button btnMoveWithDataActivity;
    private Button btnMoveWithObject;
    @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);
        btnMoveWithDataActivity=(Button)findViewById(R.id.btn_move_activity_data);
        btnMoveWithDataActivity.setOnClickListener(this);
        btnMoveWithObject=(Button)findViewById(R.id.btn_move_activity_object);
        btnMoveWithObject.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;
            case R.id.btn_move_activity_data:
                Intent moveWithDataIntent = new
Intent(MainActivity.this,MoveWithDataActivity.class);
                moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_NAME, "Bina
Sarana Informatika");
                moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_AGE,32);
                startActivity(moveWithDataIntent);
                break;
            case R.id.btn_move_activity_object:
                break;
        }
    }
}
```

9. Buat Activity baru dengan nama MoveWithObjectActivity lalu pada activity\_move\_with\_object.xml tambahkan satu buah TextView untuk menampilkan data yang dikirimkan.

```
<?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="bsi.ac.id.appbelajarintent.MoveWithObjectActivity">
    <TextView
        android:id="@+id/tv_object_received"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Object Recived"/>
</RelativeLayout>
```

10. Sekarang, pada MoveWithObjectActivity.java tambahkan kodenya sehingga menjadi sebagai

```
public class MoveWithObjectActivity extends AppCompatActivity {
    public static String EXTRA_PERSON = "extra_person";
    private TextView tvObject;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_move_with_object);
        tvObject = (TextView)findViewById(R.id.tv_object_received);
        Person mPerson = getIntent().getParcelableExtra(EXTRA_PERSON);
        String text = "Name : "+mPerson.getName()+", Email : "+mPerson.getEmail()+",
Age : "+mPerson.getAge()+ " , Location : "+mPerson.getCity();
        tvObject.setText(text);
    }
}
```

11. Setelah kita membuat activity tujuan dengan sempurna, sekarang kita hanya butuh menambahkan beberapa baris pada bagian switch... case sehingga kode MainActivity.java kita menjadi sebagai berikut :

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    private Button btnMoveActivity;
    private Button btnMoveWithDataActivity;
    private Button btnMoveWithObject;
    @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);
        btnMoveWithDataActivity=(Button)findViewById(R.id.btn_move_activity_data);
        btnMoveWithDataActivity.setOnClickListener(this);
        btnMoveWithObject=(Button)findViewById(R.id.btn_move_activity_object);
    }
}
```

```

        btnMoveWithObject.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;
            case R.id.btn_move_activity_data:
                Intent moveWithDataIntent = new
Intent(MainActivity.this,MoveWithDataActivity.class);
                moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_NAME, "Bina
Sarana Informatika");
                moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_AGE,32);
                startActivity(moveWithDataIntent);
                break;
            case R.id.btn_move_activity_object:
                Person mPerson = new Person();
                mPerson.setName("Bina Sarana Informatika");
                mPerson.setAge(32);
                mPerson.setEmail("ubsi@bsi.ac.id");
                mPerson.setCity("DKI Jakarta");
                Intent moveWithObjectIntent = new Intent(MainActivity.this,
MoveWithObjectActivity.class);
                moveWithObjectIntent.putExtra(MoveWithObjectActivity.EXTRA_PERSON,
mPerson);
                startActivity(moveWithObjectIntent);
                break;
        }
    }
}

```

## Latihan : Intent Implicit

Anda sudah belajar bagaimana menggunakan intent dengan tipe eksplisit. Sekarang saatnya Anda melanjutkan ke intent dengan tipe implicit.

1. Buka kembali activity\_main.xml tambahkan satu tombol lagi sebagai berikut :

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context="bsi.ac.id.appbelajarintent.MainActivity">

```

```

<Button
    android:id="@+id/btn_move_activity"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="BERPINDAH ACTIVITY"
    android:layout_marginBottom="@dimen/activity_vertical_margin"/>
<Button
    android:id="@+id/btn_move_activity_data"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="BERPINDAH ACTIVITY DENGAN DATA"
    android:layout_marginBottom="@dimen/activity_vertical_margin"/>
<Button
    android:id="@+id/btn_move_activity_object"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Pindah Activity Dengan Object"
    android:layout_marginBottom="@dimen/activity_vertical_margin"/>
<Button
    android:id="@+id/btn_dial_number"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Dial a Number"
    android:layout_marginBottom="@dimen/activity_vertical_margin"/>
</LinearLayout>

```

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

```

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    private Button btnMoveActivity;
    private Button btnMoveWithDataActivity;
    private Button btnMoveWithObject;
    private Button btnDialPhone;
    @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);
        btnMoveWithDataActivity=(Button)findViewById(R.id.btn_move_activity_data);
        btnMoveWithDataActivity.setOnClickListener(this);
        btnMoveWithObject=(Button)findViewById(R.id.btn_move_activity_object);
        btnMoveWithObject.setOnClickListener(this);
        btnDialPhone=(Button)findViewById(R.id.btn_dial_number);
        btnDialPhone.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;
            case R.id.btn_move_activity_data:

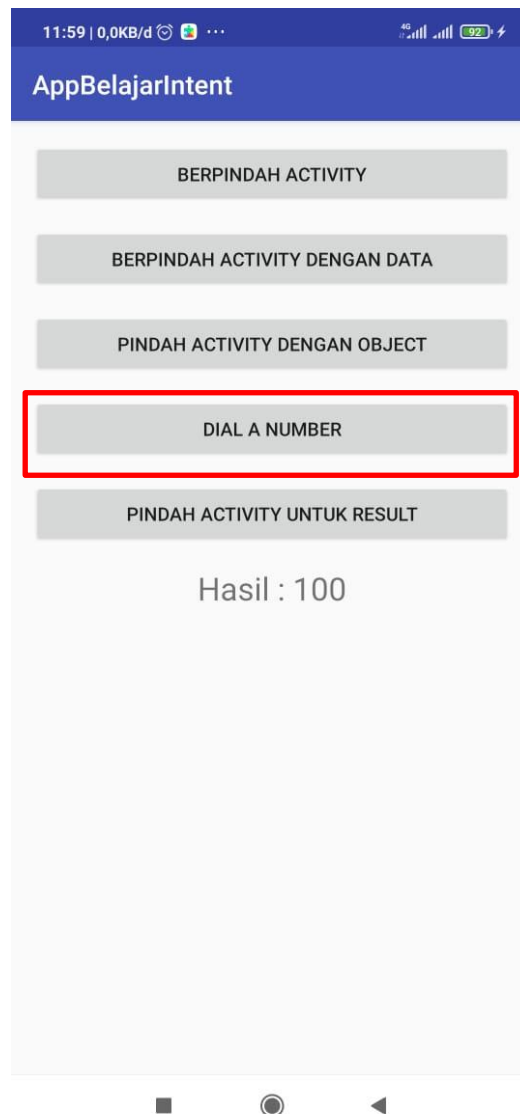
```

```

        Intent moveWithDataIntent = new
Intent(MainActivity.this, MoveWithDataActivity.class);
        moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_NAME, "Bina
Sarana Informatika");
        moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_AGE, 32);
        startActivity(moveWithDataIntent);
        break;
    case R.id.btn_move_activity_object:
        Person mPerson = new Person();
        mPerson.setName("Bina Sarana Informatika");
        mPerson.setAge(32);
        mPerson.setEmail("ubsj@bsi.ac.id");
        mPerson.setCity("DKI Jakarta");
        Intent moveWithObjectIntent = new Intent(MainActivity.this,
MoveWithObjectActivity.class);
        moveWithObjectIntent.putExtra(MoveWithObjectActivity.EXTRA_PERSON,
mPerson);
        startActivity(moveWithObjectIntent);
        break;
    case R.id.btn_dial_number:
        String phoneNumber = "08123456789";
        Intent dialPhoneIntent = new Intent(Intent.ACTION_DIAL,
Uri.parse("tel:"+phoneNumber));
        startActivity(dialPhoneIntent);
        break;
    }
}
}

```

Setelah selesai, silakan jalankan lagi aplikasi. dan aplikasi yang digunakan bisa *men-dial* sebuah nomor telepon melalui aplikasi bawaan dari peranti yang dibuat.



### Bedah Kode

```
String phoneNumber = "08123456789";
Intent dialPhoneIntent = new Intent(Intent.ACTION_DIAL,
Uri.parse("tel:"+phoneNumber));
startActivity(dialPhoneIntent);
```

Baru saja kita mengimplementasikan penggunaan intent secara implicit untuk melakukan proses *dial* sebuah nomor telepon. Pada bagian `new Intent(Intent.ACTION_DIAL, Uri.parse("tel:"+phoneNumber));` kita menggunakan inputan `new Intent(ACTION, Uri);` pada konstruktor sewaktu menciptakan obyek Intent dimana :

Action : `Intent.ACTION_DIAL`  
Uri : `Uri.parse("tel:"+phoneNumber)`

Variabel `ACTION_DIAL` menentukan intent filter dari aplikasi-aplikasi yang bisa menangani *action* tersebut. Di sini aplikasi yang memiliki kemampuan untuk komunikasi akan muncul pada opsi pilihan, sebagaimana ditampilkan ke pengguna.



Selain **ACTION\_DIAL**, di Android sudah tersedia berbagai action yang tinggal didefinisikan sewaktu menciptakan obyek **Intent** untuk mengakomodir berbagai tujuan.

Silakan cek link berikut untuk detailnya : [Intent](#)

Buat yang belum tau apa itu Uri, berikut penjelasan singkatnya :

Uri adalah sebuah untaian karakter yang digunakan untuk mengidentifikasi nama, sumber, atau layanan di internet sesuai dengan RFC 2396. Pada **Uri.parse("tel:"+phoneNumber)** kita melakukan parsing Uri dari bentuk teks string menjadi sebuah obyek uri dengan menggunakan metode **static parse(String)**. Secara struktur, dibagi menjadi :

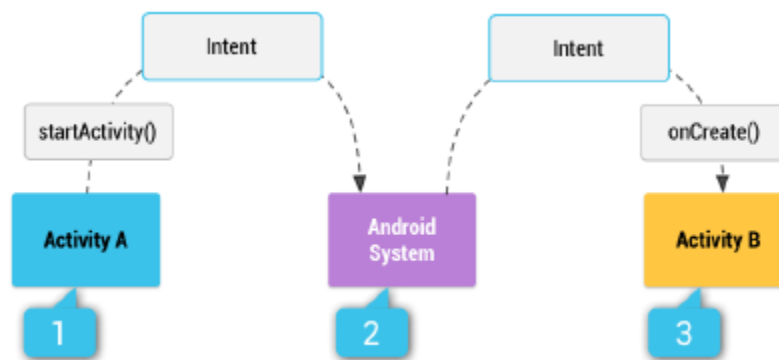
## Uri.parse("tel:"+phoneNumber)

Schema

teks

Dimana “tel” adalah sebuah skema yang disepakati untuk sumber daya telepon dan **phoneNumber** adalah variabel string yang bernilai “08123456789”. Skema lain dari Uri seperti “geo” untuk peta, “http” untuk browser sisanya bisa dilihat di halaman ini : [Common Intents](#)

Pada prosesnya, pemanggilan intent secara implicit akan berjalan sesuai dengan diagram dibawah ini.



1. Aplikasi kita menjalankan intent untuk **ACTION\_DIAL** melalui **startActivity()**.
2. Sistem Android akan melakukan seleksi terhadap semua aplikasi yang memiliki kemampuan untuk menangani action tersebut. Sistem Android akan menentukan aplikasi mana saja yang bisa memproses action berdasarkan intent filter yang telah ditentukan di berkas **AndroidManifest.xml** pada masing-masing aplikasi. Sistem Android akan menampilkan opsi aplikasi-aplikasi mana saja yang bisa menangani action tersebut ke pengguna.

Pengguna memilih salah satu opsi aplikasi dan kemudian sistem Android akan me-routing ke activity pada aplikasi yang dipilih yang memiliki intent-filter untuk aksi **ACTION\_DIAL**.

3. Aplikasi yang dipilih pun muncul di layar dengan nomor telepon yang sudah diatur.

**Anda sudah mempelajari 3 buah intent secara eksplisit dan 1 buah intent secara implicit. Selamat!**

Selanjutnya, kita akan membuat sebuah Intent yang didalamnya akan membawa data dengan **ResultActivity**.

Pada modul 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 dijalankan.

1. 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 :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context="bsi.ac.id.appbelajarintent.MoveForResultActivity">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Pilih Angka Yang Disuka"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
    <RadioGroup
        android:id="@+id/rg_number"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <RadioButton
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="50"
            android:layout_marginBottom="@dimen/activity_vertical_margin"/>
        <RadioButton
            android:id="@+id/rb_100"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="100"
            android:layout_marginBottom="@dimen/activity_vertical_margin"/>
        <RadioButton
            android:id="@+id/rb_150"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="150"
            android:layout_marginBottom="@dimen/activity_vertical_margin"/>
        <RadioButton
            android:id="@+id/rb_200"
```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="200"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
    </RadioGroup>
    <Button
        android:id="@+id/btn_choose"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Pilih"/>
</LinearLayout>

```

2. Setelah selesai, lanjutkan ke berkas **MoveForResultActivity.java** dengan menuliskan kode sebagai berikut

```

public class MoveForResultActivity extends AppCompatActivity implements
View.OnClickListener {
    private Button btnChoose;
    private RadioGroup rgNumber;
    public static String EXTRA_SELECTED_VALUE = "extra_selected_value";
    public static int RESULT_CODE = 110;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_move_for_result);
        btnChoose = (Button)findViewById(R.id.btn_choose);
        btnChoose.setOnClickListener(this);
        rgNumber = (RadioGroup)findViewById(R.id.rg_number);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_choose){
            if (rgNumber.getCheckedRadioButtonId() != 0){
                int value = 0;
                switch (rgNumber.getCheckedRadioButtonId()){
                    case R.id.rb_50:
                        value = 50;
                        break;
                    case R.id.rb_100:
                        value =100;
                        break;
                    case R.id.rb_150:
                        value=150;
                        break;
                    case R.id.rb_200:
                        value=200;
                        break;
                }
                Intent resultIntent = new Intent();
                resultIntent.putExtra(EXTRA_SELECTED_VALUE, value);
                setResult(RESULT_CODE, resultIntent);
                finish();
            }
        }
    }
}

```

```

    }
}

```

3. Pada **activity\_main.xml** tambahkan lagi satu **Button** dan satu **TextView** yang akan kita gunakan untuk menjalankan **MoveForResultActivity** sehingga layout activity\_main.xml yang kita punya menjadi seperti ini :

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context="bsi.ac.id.appbelajarintent.MainActivity">

    <Button
        android:id="@+id/btn_move_activity"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="BERPINDAH ACTIVITY"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>

    <Button
        android:id="@+id/btn_move_activity_data"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="BERPINDAH ACTIVITY DENGAN DATA"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>

    <Button
        android:id="@+id/btn_move_activity_object"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Pindah Activity Dengan Object"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>

    <Button
        android:id="@+id/btn_dial_number"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Dial a Number"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>

```

```

<Button
    android:id="@+id/btn_move_for_result"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Pindah Activity untuk Result"
    android:layout_marginBottom="@dimen/activity_vertical_margin"/>
<TextView
    android:id="@+id/tv_result"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="24sp"
    android:gravity="center"
    android:text="Hasil"/>
</LinearLayout>

```

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

```

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    private Button btnMoveActivity;
    private Button btnMoveWithDataActivity;
    private Button btnMoveWithObject;
    private Button btnDialPhone;
    private Button btnMoveForResult;
    private TextView tvResult;
    private int REQUEST_CODE = 100;

    @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);
        btnMoveWithDataActivity=(Button)findViewById(R.id.btn_move_activity_data);
        btnMoveWithDataActivity.setOnClickListener(this);
        btnMoveWithObject=(Button)findViewById(R.id.btn_move_activity_object);
        btnMoveWithObject.setOnClickListener(this);
        btnDialPhone=(Button)findViewById(R.id.btn_dial_number);
        btnDialPhone.setOnClickListener(this);
        btnMoveForResult = (Button)findViewById(R.id.btn_move_for_result);
        btnMoveForResult.setOnClickListener(this);
        tvResult = (TextView)findViewById(R.id.tv_result);
    }

    @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;
            case R.id.btn_move_activity_data:
                Intent moveWithDataIntent = new
Intent(MainActivity.this,MoveWithDataActivity.class);
                moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_NAME, "Bina

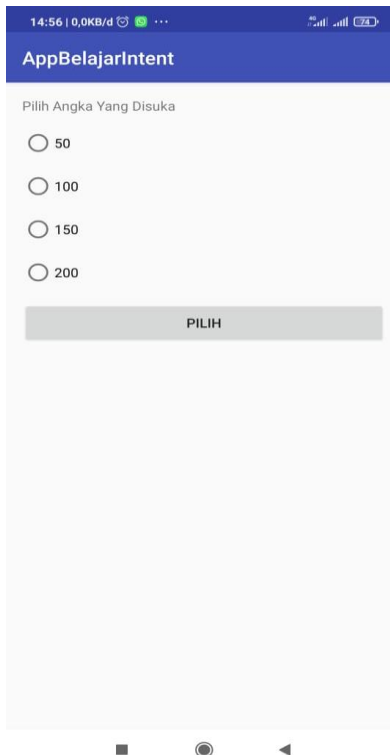
```

```

Sarana Informatika");
        moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_AGE, 32);
        startActivity(moveWithDataIntent);
        break;
    case R.id.btn_move_activity_object:
        Person mPerson = new Person();
        mPerson.setName("Bina Sarana Informatika");
        mPerson.setAge(32);
        mPerson.setEmail("ubsi@bsi.ac.id");
        mPerson.setCity("DKI Jakarta");
        Intent moveWithObjectIntent = new Intent(MainActivity.this,
MoveWithObjectActivity.class);
        moveWithObjectIntent.putExtra(MoveWithObjectActivity.EXTRA_PERSON,
mPerson);
        startActivity(moveWithObjectIntent);
        break;
    case R.id.btn_dial_number:
        String phoneNumber = "08123456789";
        Intent dialPhoneIntent = new Intent(Intent.ACTION_DIAL,
Uri.parse("tel:"+phoneNumber));
        startActivity(dialPhoneIntent);
        break;
    case R.id.btn_move_for_result:
        Intent moveForResultIntent = new Intent(MainActivity.this,
MoveForResultActivity.class);
        startActivityForResult(moveForResultIntent, REQUEST_CODE);
        break;
    }
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_CODE){
        if (resultCode == MoveForResultActivity.RESULT_CODE){
            int selectedValue =
data.getIntExtra(MoveForResultActivity.EXTRA_SELECTED_VALUE, 0);
            tvResult.setText("Hasil : "+selectedValue);
        }
    }
}
}
}

```

5. 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 textiew **tvResult**.



## Bedah Kode

Perbedaan mendasar antara perpindahan activity untuk menghasilkan nilai balik dengan tidak adalah pada metode untuk menjalankan obyek intent-nya. Sebelumnya kita menggunakan `startActivity(Intent)` untuk berpindah activity. Kali ini kita menggunakan `startActivityForResult(Intent, RequestCode)`.

```
Intent moveForResultIntent = new Intent(MainActivity.this,
MoveForResultActivity.class);
startActivityForResult(moveForResultIntent, REQUEST_CODE);
```

Pada baris diatas, kita akan menjalankan sebuah activity melalui intent untuk nilai balik ke activity yang menjalankan di mana nilai REQUEST\_CODE adalah 110. Penentuan nilai 110 itu dibebaskan dan kalau bisa disesuaikan dengan kebutuhan pengembangan aplikasi.

Kemudian, pada MoveForResultActivity kita memilih satu angka yang kita suka, Sebagai contoh angka 150. Kemudian tekanlah tombol 'Pilih'. Maka baris kode dibawah ini akan dijalankan.

```
@Override
public void onClick(View v) {
    if (v.getId() == R.id.btn_choose){
        if (rgNumber.getCheckedRadioButtonId() != 0){
            int value = 0;
            switch (rgNumber.getCheckedRadioButtonId()){
                case R.id.rb_50:
                    value = 50;
                    break;
                case R.id.rb_100:
                    value =100;
                    break;
                case R.id.rb_150:
                    value=150;
```

```

        break;
    case R.id.rb_200:
        value=200;
        break;
    }
    Intent resultIntent = new Intent();
    resultIntent.putExtra(EXTRA_SELECTED_VALUE, value);
    setResult(RESULT_CODE, resultIntent);
    finish();
}
}
}
}
}

```

Di mana akan divalidasi terlebih dahulu apakah ada nilai dari obyek **RadioButton** yang dipilih. Bila ada, maka proses selanjutnya menentukan obyek **RadioButton** mana yang diklik berdasarkan nilai dari **rgNumber.getCheckedRadioButtonId()**.

Mengapa kita tidak memeriksa langsung ke obyek **RadioButton**? Karena kita menggunakan **RadioGroup** sebagai *parent* pada obyek-obyek **RadioButton**. Secara otomatis kita bisa mendapatkan mana obyek **RadioButton** yang dipilih dengan **rgNumber.getCheckedRadioButtonId()**.

```

<RadioGroup
    android:id="@+id/rg_number"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <RadioButton
        android:id="@+id/rb_50"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="50"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
    <RadioButton
        android:id="@+id/rb_100"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="100"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
    <RadioButton
        android:id="@+id/rb_150"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="150"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
    <RadioButton
        android:id="@+id/rb_200"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="200"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
</RadioGroup>

```

Selanjutnya, ketika sudah didapatkan nilainya, maka baris ini akan dieksekusi:



```
Intent resultIntent = new Intent();
resultIntent.putExtra(EXTRA_SELECTED_VALUE, value);
setResult(RESULT_CODE, resultIntent);
finish();
```

Kita membuat sebuah intent tanpa ada inputan apapun di konstruktornya. Kemudian kita meletakkan variabel `value` kedalam metode `putExtra(Key, Value)` dengan `EXTRA_SELECTED_VALUE` bertipekan `static string` dan bernilai `"extra_selected_value"`. Kemudian kita jadikan obyek `resultIntent` yang telah dibuat sebelumnya menjadi parameter dari `setResult(RESULT_CODE, Intent)`.

Setelah itu, kita panggil method `finish()` untuk menutup `MoveForResultActivity`.

Ketika `MoveForResultActivity` telah tertutup sempurna, maka metode `onActivityResult()` pada `MainActivity` akan dijalankan.

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_CODE){
        if (resultCode == MoveForResultActivity.RESULT_CODE){
            int selectedValue =
data.getIntExtra(MoveForResultActivity.EXTRA_SELECTED_VALUE, 0);
            tvResult.setText("Hasil : "+selectedValue);
        }
    }
}
```

Di sinilah `MainActivity` akan merespon terhadap nilai balik yang dikirimkan oleh `MoveForResultActivity`. Pada baris 4 di atas, dilakukan perbandingan apakah `requestCode` sama dengan yang dikirimkan oleh `MainActivity`.

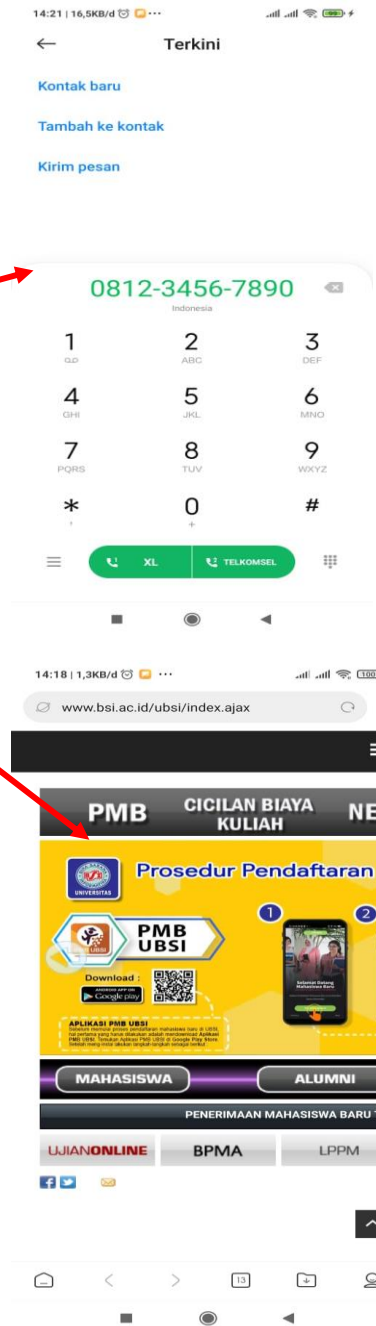
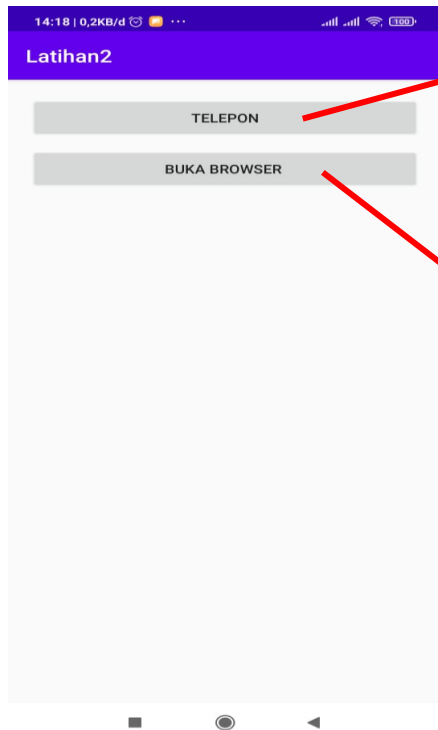
Kemudian pada baris 5, diperiksa apakah nilai `resultCode` sama dengan `resultCode` yang dikirim oleh `MoveForResultActivity`. Bila iya, maka data `RadioButton` yang dipilih akan ditampilkan di textview `tvResult`.

Pada realita sehari-hari, konsep yang barusan kita pelajari ini akan bersinggungan dengan aplikasi *native* lainnya. Misalnya, jika aplikasi kita membutuhkan gambar yang diambil dari kamera atau gallery photo. Tentu kita mengharapkan nilai balik berupa alamat foto yang dapat diterima oleh activity yang menjalankan. Begitu juga jika kita membutuhkan data kontak yang berasal dari aplikasi phonebook bawaan peranti Android.

## Tugas 2

Buatlah aplikasi android yang menggambarkan proses Intent dengan ketentuan sebagai berikut:

1. Buatlah layout yang terdapat 2 tombol yang terdiri dari :
  - c. Telepon
  - d. Buka Browser
2. Jika klik pada Telepon = Maka akan membuka halaman dial pada smartphone
3. Jika klik pada Buka Browser = Maka akan berpindah ke halaman [www.bsi.ac.id](http://www.bsi.ac.id)



## BAB VII SERVICE

### A. Pengenalan Service

Sebelumnya telah belajar mengenai activity dan implementasinya. Activity adalah komponen yang memberikan pengalaman kepada pengguna secara langsung. Pengguna dapat melihat dan berinteraksi di atasnya.

Service berada pada sisi yang lain, komponen ini tidak memiliki antarmuka dan bahkan pengguna tidak akan tahu bagaimana dia bekerja. Pengalaman yang diberikan oleh service hanya berupa proses yang tidak terlihat. Ia digunakan untuk menjalankan beragam macam proses yang memakan waktu lama.

Walaupun berjalan secara *background*, pada dasarnya service dan komponen Android lainnya berjalan pada satu proses dan *thread* yang sama yaitu *main thread* atau *ui thread*. Bekerja di *background* bukan berarti ia bekerja secara *asynchronous*. Service tetap membutuhkan *thread* terpisah jika kita ingin melakukan proses yang membutuhkan komputasi intensif atau yang memakan waktu.

Contoh pemanfaatan service antara lain:

- Aplikasi sosial media atau aplikasi yang memiliki kemampuan untuk menerima *push notification*. Aplikasi semacam ini pasti memiliki sebuah service yang berjalan dalam posisi *stand by* untuk selalu menerima pesan yang masuk.
- Aplikasi *chat* juga membutuhkan service untuk melakukan pengiriman dan menerima pesan yang dikirimkan oleh pengguna.
- Aplikasi pemutar musik juga melakukan hal yang sama. Untuk memberikan pengalaman yang lebih baik kepada pengguna, aplikasi pemutar musik biasanya meletakkan proses *streaming* atau memainkan musik di komponen service dengan tetap mempertahankan integrasi dengan komponen lain, misalnya notifikasi.

Secara umum, terdapat dua bentuk dari service :

#### 1. Started

Service berjenis ini adalah tipe yang dijalankan oleh komponen lain, misal activity. Sekali dijalankan, service ini akan berjalan selama belum dimatikan atau proses yang dijalankan selesai. Service akan tetap berjalan walaupun komponen yang lain dimatikan oleh sistem Android. Umumnya penggunaan service ini adalah untuk melakukan proses yang tidak memberikan nilai balik ke komponen yang memanggilnya. Contohnya adalah, mengunduh atau mengunggah berkas.

#### 2. Bound

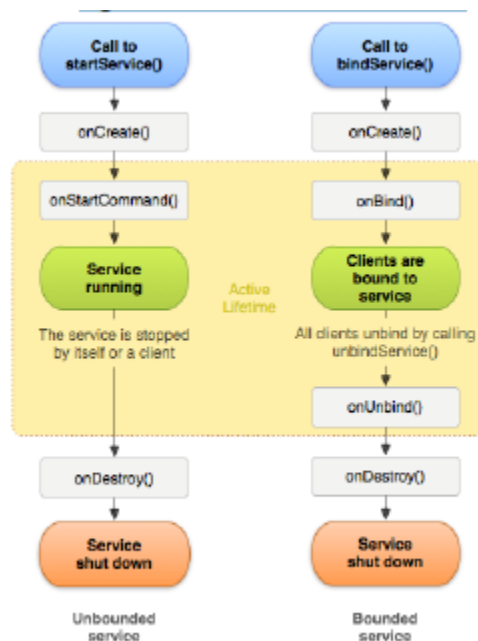
Service jenis ini merupakan tipe service yang dijalankan oleh komponen lain, namun saling mengikat. Hubungan yang terjadi antar kedua komponen tersebut seperti *client-server*. Bisa saling menerima hasil dan menerima *request* yang ada. Pada service ini dimungkinkan terjadi proses IPC (*Interprocess Communication*). Service ini akan tetap berjalan di background selama masih ada komponen lain yang mengikatnya. Jika tidak, maka Service akan dimatikan oleh sistem. Aplikasi pemutar musik merupakan salah satu jenis aplikasi yang mengimplementasikan service jenis ini.

Pada bagian ini kita akan sepenuhnya membahas service berjenis started. Tipe service tersebut akan dibagi menjadi dua bagian dalam implementasinya:

- Kelas service yang inherit langsung kepada kelas Service  
Ketika sebuah kelas java *inherit* ke service ingin menjalankan proses yang memakan waktu lama, maka kelas tersebut diharuskan membuat thread terpisah agar tidak memblok ui thread yang ada. Service ini akan selalu hidup di *background* selama tidak ada komponen yang memanggil `stopService()` atau dimatikan oleh sistem.
- Kelas service yang inherit ke `IntentService`  
Ini adalah kelas yang sangat memudahkan hidup kita. Dia bersifat *fire and forget*, ketika ia telah menyelesaikan tugasnya, ia akan mematikan dirinya.

Poin-poin penting lain tentang service diantaranya:

- Setiap kelas Java dinyatakan sebuah service bila kelas tersebut inherit/extends ke kelas Service atau `IntentService`.
- Service memiliki *life cycle*-nya sendiri dan bergantung pada tipe service apa yang digunakan, started atau bound service.
- Untuk menjalankan service dari komponen lain seperti activity, cukup menggunakan `startService(Intent)`. Sebaliknya untuk mematikan/stop service terdapat dua cara. Pertama `stopService(Intent)` dijalankan dari komponen yang memanggil dan `stopSelf()` dari kelas Service itu sendiri.



## B. Latihan Service

Anda sudah paham service secara garis besar berikut pemanfaatannya. Sekarang saatnya kita menerapkannya.

1. Baik, buat proyek baru dengan nama MyService. Pilih Empty Activity dengan pilihan *default* pada *set up* proyek. Setelah proyek tercipta, Tambahkan kembali *dimens.xml* secara manual di dalam *res* → *values*. Isi berkasnya sehingga menjadi seperti berikut.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
</resources>
```

2. Lengkapi *activity\_main.xml* dengan contoh seperti ini:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context="bsi.ac.id.myservice.MainActivity">
    <Button
        android:id="@+id/btn_start_service"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Start Service"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
    <Button
        android:id="@+id/btn_start_intent_service"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Start Intent Service"/>
</LinearLayout>
```

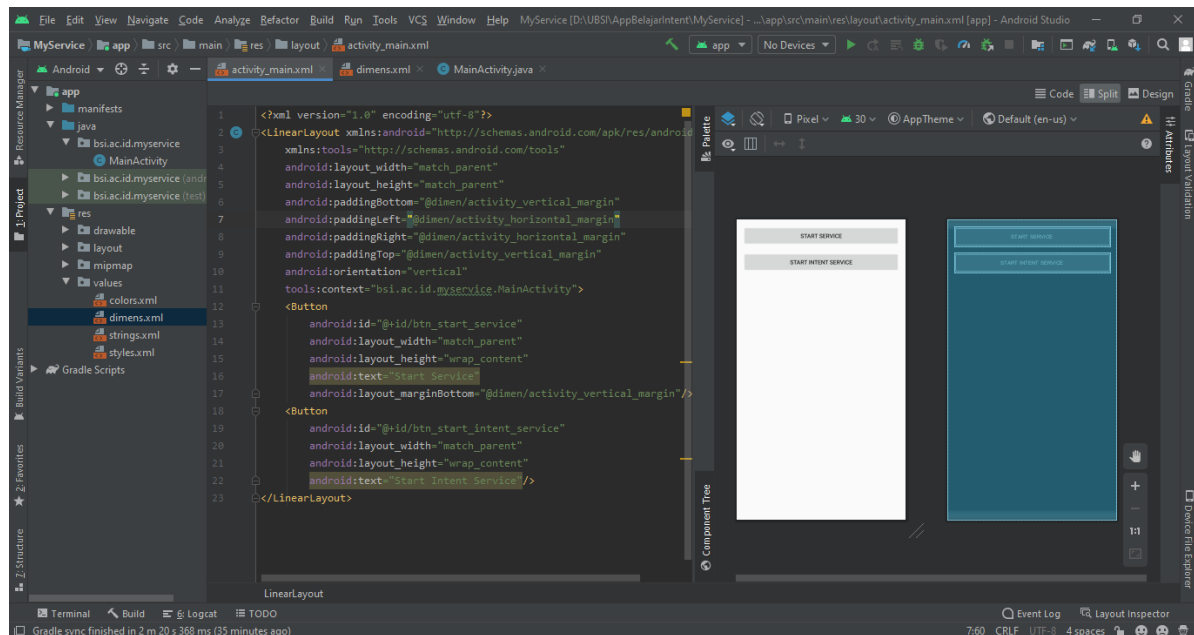
3. Pada MainActivity.java silakan lengkapi kode-nya menjadi sebagai berikut:

```
public class MainActivity extends AppCompatActivity implements
View.OnClickListener {
    private Button btnStartService;
    private Button btnStartIntentService;

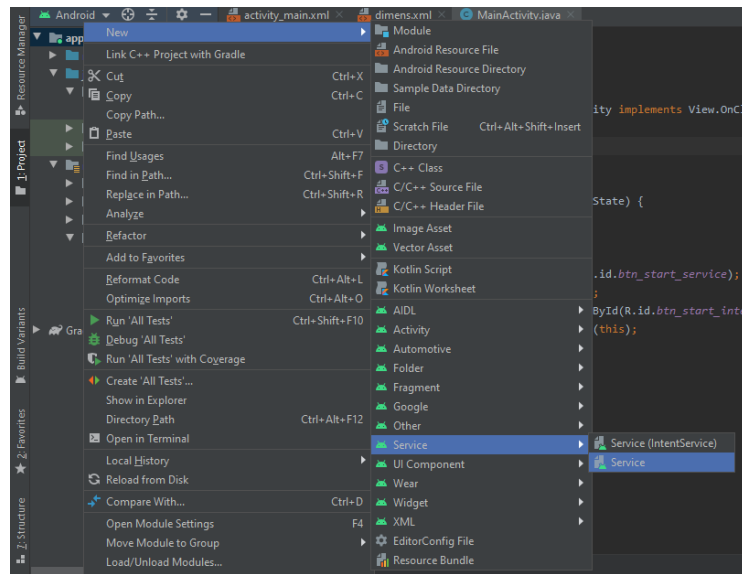
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnStartService = (Button)findViewById(R.id.btn_start_service);
        btnStartService.setOnClickListener(this);
        btnStartIntentService =
        (Button)findViewById(R.id.btn_start_intent_service);
        btnStartIntentService.setOnClickListener(this);
    }
    @Override
    public void onClick(View v) {
        switch (v.getId()){
            case R.id.btn_start_service:
                break;
            case R.id.btn_start_intent_service:
                break;
        }
    }
}
```

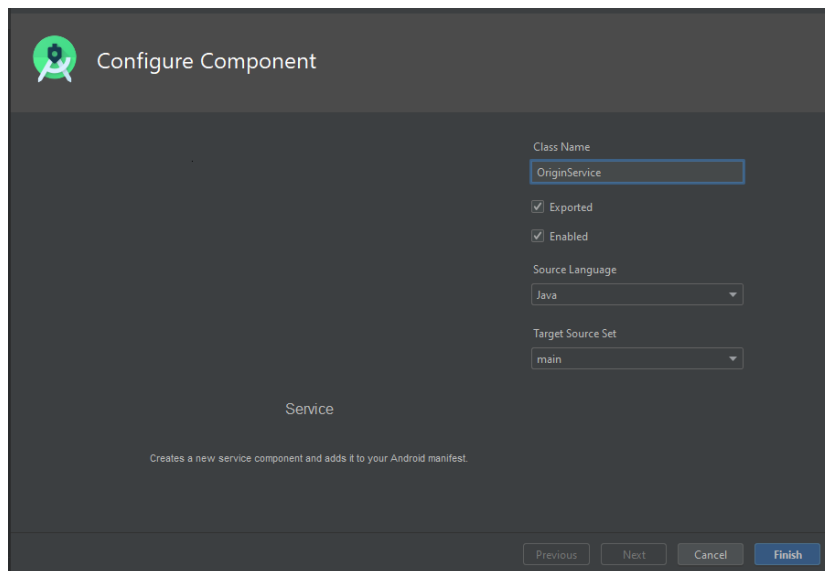
4. Tampilan yang seharusnya ada pada MainActivity, adalah seperti ini.



5. Lanjut, buat kelas service bernama OriginService dengan cara klik kanan pada package project → New → Service → Service. OriginService akan *inherit* (*extends*) kepada kelas Service.



6. Selanjutnya pada dialog yang tampil, isikan nama kelas service yang diinginkan. Di sini kita menamainya sebagai **OriginService** dan biarkan *exported* dan *enabled* tercentang. Klik Finish untuk menyelesaikan proses.



7. Selanjutnya, buka berkas AndroidManifest.xml pada package manifest dan perhatikan isi berkas tersebut. Service yang baru saja kita buat sudah ada didalam tag <application>

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="bsi.ac.id.myservice">

    <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">
        <service
            android:name=".OriginService"
            android:enabled="true"
            android:exported="true"></service>

        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

8. Berkas **AndroidManifest** sudah dibuat secara otomatis. Dengan demikian kita sudah bisa menjalankan kelas service tersebut. Namun, sebelum menjalankan aplikasi, lengkapi kode pada **OriginService** menjadi seperti berikut :

```
public class OriginService extends Service {
    public static final String ORIGIN_SERVICE = "OriginService";
    public OriginService() {
    }

    @Override
    public IBinder onBind(Intent intent) {
        // TODO: Return the communication channel to the service.
        throw new UnsupportedOperationException("Not yet implemented");
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        Log.d(ORIGIN_SERVICE, "OriginService dijalankan");
        return START_STICKY;
    }
}
```



9. Selanjutnya pada **MainActivity.java** di metode **onClick()** pada case **R.id.btn\_start\_service** tambahkan baris berikut, Sehingga kode pada metode **onClick()** menjadi seperti ini :

```
public class MainActivity extends AppCompatActivity implements
View.OnClickListener {
    private Button btnStartService;
    private Button btnStartIntentService;

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

        btnStartService = (Button)findViewById(R.id.btn_start_service);
        btnStartService.setOnClickListener(this);
        btnStartIntentService =
(Button)findViewById(R.id.btn_start_intent_service);
        btnStartIntentService.setOnClickListener(this);
    }
    @Override
    public void onClick(View v) {
        switch (v.getId()){
            case R.id.btn_start_service:
                Intent mStartServiceIntent = new Intent(MainActivity.this,
OriginService.class);
                startService(mStartServiceIntent);
                break;
            case R.id.btn_start_intent_service:
                break;
        }
    }
}
```

10. Baik, sekarang kita akan menambahkan sebuah inner class **AsyncTask**. Ia seakan-akan menjalankan sebuah proses secara *asynchronous* dan mematikan/menghentikan dirinya sendiri dengan memanggil metode **stopSelf()**. Lengkapi kodenya menjadi sebagai berikut pada file **OriginService.java**:

```
public class OriginService extends Service {
    public static final String ORIGIN_SERVICE = "OriginService";
    public OriginService() {
    }

    @Override
    public IBinder onBind(Intent intent) {
        // TODO: Return the communication channel to the service.
        throw new UnsupportedOperationException("Not yet implemented");
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        Log.d(ORIGIN_SERVICE, "OriginService dijalankan");
        ProcessAsync mProcessAsync = new ProcessAsync();
        mProcessAsync.execute();
        return START_STICKY;
    }

    private class ProcessAsync extends AsyncTask<Void, Void, Void> {
        @Override
        protected Void doInBackground(Void... params) {
            try {
                Thread.sleep(3000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            return null;
        }

        @Override
        protected void onPostExecute(Void aVoid) {
            super.onPostExecute(aVoid);
            Log.d(ORIGIN_SERVICE, "StopService");
            stopSelf();
        }
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
        Log.d(ORIGIN_SERVICE, "onDestroy()");
    }
}
```

11. Jalankan aplikasinya. Klik tombol ‘start service’ dan perhatikan *log*-nya. Service dijalankan secara *asynchronous* dan mematikan dirinya sendiri setelah proses selesai.

### C. Bedah Kode

Kita baru saja berkenalan dengan service melalui kelas **OriginService**. Kita *inherit* kelas dasar dari service seperti yang dijelaskan sebelumnya. Service jenis ini berada pada *thread* yang sama yaitu *ui thread*.

Service di atas dijalankan dengan cara berikut ini :

```
Intent mStartServiceIntent = new Intent(MainActivity.this, OriginService.class);
startService(mStartServiceIntent);
```

Ingat! Bukan `startActivity()` seperti pada contoh sebelumnya. Namun `startService()` karena kita menginginkan sebuah service yang berjalan. Setelah dijalankan, metode `onStartCommand()` pada **OriginService** akan dijalankan.

```
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    Log.d(ORIGIN_SERVICE, "OriginService dijalankan");
    ProcessAsync mProcessAsync = new ProcessAsync();
    mProcessAsync.execute();
    return START_STICKY;
}
```

Pada metode tersebut, kita menjalankan sebuah *thread* *asynctask* untuk melakukan proses yang menyerupai proses yang sulit. Dan ia berjalan secara *asynchronous*.

Kekurangan dari service tipe ini adalah tidak menyediakan *worker thread* diluar *ui thread* secara *default*. Jadi tidak ada cara lain, selain membuat *thread* secara sendiri.

**START\_STICKY** menandakan bahwa bila service tersebut dimatikan oleh sistem Android karena kekurangan memori, maka ia akan diciptakan kembali jika sudah ada memori yang bisa digunakan.

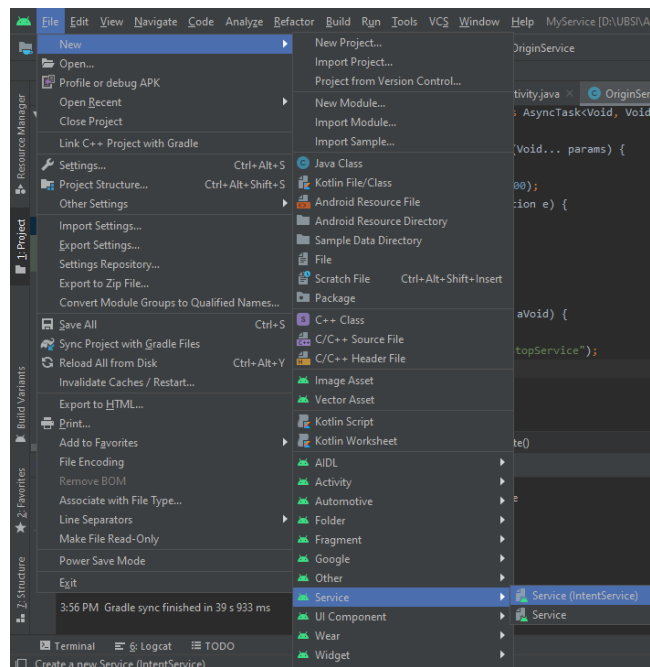
Metode `onStartCommand()` juga akan dijalankan kembali.

```
stopSelf();
```

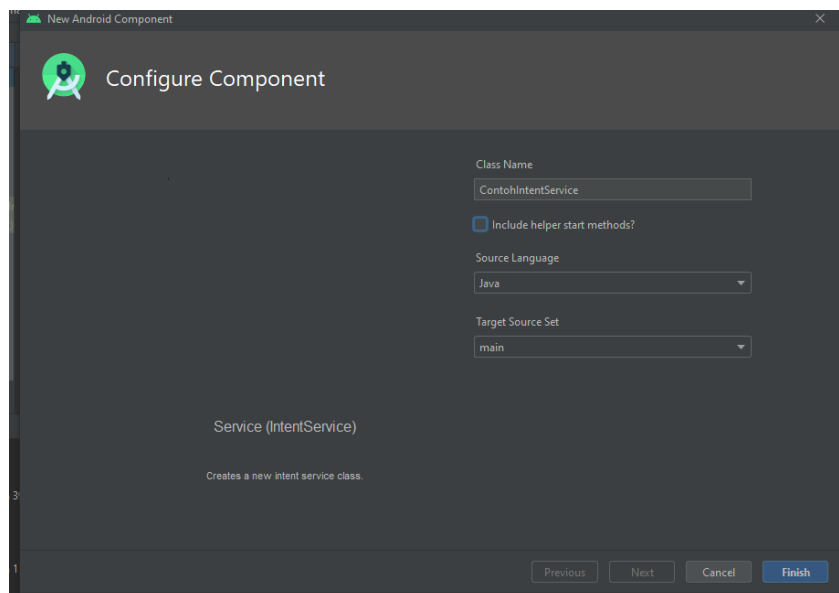
Masih ingat dengan pembahasan *asynctask* pada materi sebelumnya ? Pada method `onPostExecute()` akan memanggil `stopSelf()` yang berarti akan memberhentikan atau mematikan **OriginService** dari sistem Android.

Pada titik ini, Anda sudah bisa membuat sebuah service yang berjalan di background.

1. Sekarang, buat kembali sebuah service dengan nama **ContohIntentService** dengan cara klik kanan pada package utama project → New → Service → Service (Intent Service).



2. Pada dialog yang tampil, masukkan nama **ContohIntentService** seperti ini : Klik Finish untuk menyelesaikan proses. Pastikan untuk tidak mencentang *include helper*. *Include helper* akan menambahkan metode yang dapat membantu di dalam pembuatan services. Akan tetapi pada modul ini, kita tidak menggunakan metode tersebut.



3. Setelah kelas `ContohIntentService` diciptakan, terdapat banyak kode *auto-generated* yang telah disertakan. Kita tidak membutuhkan semua metode tersebut. Sehingga ubah kelas tersebut menjadi seperti contoh di bawah ini :

```
public class ContohIntentService extends IntentService {
    public ContohIntentService() {
        super("ContohIntentService");
    }
    @Override
    protected void onHandleIntent(Intent intent) {
        if (intent != null) {
        }
    }
}
```

4. Skenarionya adalah sebagai berikut :

- Kita menjalankan `IntentService` tersebut dengan sebuah obyek `Intent` dari `MainActivity` dengan membawa data, dalam konteks ini adalah nilai integer yang menentukan berapa lama *background* proses dijalankan.
- `ContohIntentService` dijalankan, dan kemudian melakukan pemrosesan obyek `Intent` yang dikirimkan untuk menjalankan *background*.
- Seperti sifatnya, `IntentService` tidak perlu mematikan dirinya sendiri, secara otomatis ketika proses yang dilakukan selesai, maka, `IntentService` berhenti dengan

5. Pada `ContohIntentService` lengkapi kodenya menjadi seperti berikut :

```
public class ContohIntentService extends IntentService {
    public static String EXTRA_DURATION = "extra_duration";
    public static final String TAG = "ContohIntentService";
    public ContohIntentService() {
        super("ContohIntentService");
    }
    @Override
    protected void onHandleIntent(Intent intent) {
        if (intent != null) {
            int duration = intent.getIntExtra(EXTRA_DURATION, 0);
            try {
                Thread.sleep(duration);
            } catch (InterruptedException e) {
                e.printStackTrace();
                Thread.currentThread().interrupt();
            }
        }
    }
    @Override
    public void onDestroy() {
        super.onDestroy();
        Log.d(TAG, "onDestroy()");
    }
}
```

6. Selanjutnya pada MainActivity.java di metode onClick() pada case R.id.btn\_start\_intent\_service kita tambahkan 3 baris berikut untuk menjalankan IntentService yang baru saja dibuat.

```
@Override
public void onClick(View v) {
    switch (v.getId()){
        case R.id.btn_start_service:
            Intent mStartServiceIntent = new Intent(MainActivity.this,
OriginService.class);
            startService(mStartServiceIntent);
            mStartServiceIntent.putExtra(ContohIntentService.EXTRA_DURATION,
5000);
            startService(mStartServiceIntent);
            break;
        case R.id.btn_start_intent_service:
            break;
    }
}
```

Kita tinjau kembali skenarionya. Kita menjalankan IntentService tersebut dengan sebuah obyek intent yang membawa data dari MainActivity. Kita dapat menentukan berapa lama service ini akan berjalan.

```
Intent mStartServiceIntent = new Intent(MainActivity.this, OriginService.class);
startService(mStartServiceIntent);
mStartServiceIntent.putExtra(ContohIntentService.EXTRA_DURATION, 5000);
startService(mStartServiceIntent);
```

ContohIntentService dijalankan. Service tersebut akan melakukan pemrosesan obyek Intent yang dikirimkan dan menjalankan suatu proses yang berjalan di background.

```
@Override
protected void onHandleIntent(Intent intent) {
    if (intent != null) {
        int duration = intent.getIntExtra(EXTRA_DURATION, 0);
        try {
            Thread.sleep(duration);
        } catch (InterruptedException e) {
            e.printStackTrace();
            Thread.currentThread().interrupt();
        }
    }
}
```

Kode diatas akan dijalankan pada *thread* terpisah secara *asynchronous*. Jadi kita tidak membutuhkan lagi obyek AsyncTask seperti pada service sebelumnya.

Terakhir, IntentService tidak perlu mematikan dirinya sendiri. Service ini akan berhenti dengan sendirinya ketika sudah menjalankan tugasnya.

#### D. Penggunaan Service di Alarm Manager

Adakalanya kita membutuhkan perlu suatu mekanisme untuk mengulang sebuah proses di waktu tertentu. Misalnya, kita ingin memiliki fungsi pengingat (*reminder*) pada aplikasi *event*, jadwal tayang film di bioskop, atau aplikasi alarm untuk mengingatkan kita untuk bangun di pagi hari. Disinilah alarm manager berperan penting.

Alarm manager akan memberikan kemudahan bagi aplikasi untuk melakukan operasi berbasis waktu diluar daur hidup aplikasi itu sendiri.

Contohnya seperti yang baru saja disebutkan diatas. Umumnya alarm memiliki beberapa karakteristik, antara lain :

1. Dapat menjalankan obyek intent berdasarkan waktu dan interval yang ditentukan.
2. Bisa bekerja dengan baik dengan broadcast receiver untuk menjalankan komponen lain seperti **service** untuk melakukan operasi tertentu.
3. Alarm berjalan diluar daur hidup aplikasi induknya sehingga kita bisa merancang fungsi alarm untuk melakukan sebuah aksi ketika aplikasi sedang tidak dijalankan bahkan ketika peranti dalam keadaan *idle* atau *sleep*.
4. Esensi utama dari alarm manager adalah untuk meminimalkan penggunaan *resource* dan menghindari penggunaan *timer* dan background **service** yang berkepanjangan untuk melakukan operasi atau aksi yang dibutuhkan.

Penggunaan dan perancangan alarm manager yang tepat akan membuat aplikasi kita menjadi lebih baik, serta tentunya memberikan pengalaman yang juga berkesan. Namun sebaliknya desain yang buruk akan menjadi bumerang bagi aplikasi kita. Akibat dari rancangan yang tidak sesuai akan menguras baterai di peranti kita.

Penggunaan alarm manager yang baik adalah dengan menerapkan cara berikut ini:

1. Lakukan hanya untuk proses di lokal
2. Jaga frekuensi alarm yang dijalankan dengan interval yang tidak pendek
3. Jika tidak terlalu penting, hindari penggunaan tipe alarm yang dapat membangunkan peranti seperti `RTC_WAKEUP`
4. Jika alarm manager digunakan untuk menjadwalkan *scheduling task* yang terhubung ke jaringan dan melakukan *request* ke *server*, misalnya unduh data terbaru atau sinkronisasi data, maka hindari penggunaan `setRepeating()` dan gunakan `setInExactRepeating()` dengan penggunaan tipe alarm berbasis `ELAPSED_REALTIME`. Pendekatan tersebut merupakan strategi untuk mengelompokkan proses alarm dari aplikasi yang berbeda dengan dijalankan secara bersamaan agar lebih menghemat daya baterai.

**Catatan :** Di atas API 19 KitKat semua pemanggilan *repeating alarm* adalah menggunakan *in exact repeating*.

Kali ini kita akan membuat sebuah proyek sederhana dengan skenario seperti berikut.

1. Aplikasi akan memiliki kemampuan untuk melakukan *one-time-execution alarm* dan alarm yang berulang berdasarkan waktu yang ditentukan.
2. Aplikasi akan menyimpan beberapa nilai untuk kedua tipe alarm tersebut.
3. Aplikasi akan menampilkan notifikasi dengan suara dan getar ketika alarm dijalankan.

## Petunjuk

1. Buat proyek baru dengan nama **MyAlarmManager** yang berjenis **Empty Activity** dan dengan konfigurasi *default* dan tambahkan kembali **dimens.xml** secara manual di dalam res → values. Isi berkasnya sehingga menjadi seperti berikut.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
</resources>
```

2. Pada activity\_main.xml sesuaikan kodenya menjadi seperti berikut.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:orientation="vertical"
    tools:context="bsi.ac.id.myalarmmanager.MainActivity">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="One Time Alarm"
        android:layout_marginBottom="@dimen/activity_vertical_margin"
        android:textSize="16sp"
        android:textStyle="bold"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <Button
            android:text="..."
            android:layout_width="50dp"
            android:layout_height="wrap_content"
            android:id="@+id/btn_one_time_alarm_date" />
        <TextView
            android:id="@+id/tv_one_time_alarm_date"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Tanggal Alarm"
            android:layout_marginBottom="@dimen/activity_vertical_margin"/>
```



```

</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <Button
        android:text="..."
        android:layout_width="50dp"
        android:layout_height="wrap_content"
        android:id="@+id/btn_one_time_alarm_time" />

    <TextView
        android:id="@+id/tv_one_time_alarm_time"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Waktu Alarm"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
</LinearLayout>
<EditText
    android:id="@+id/edt_one_time_alarm_message"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Pesanan Alarm"
    android:layout_marginBottom="@dimen/activity_vertical_margin"/>
<Button
    android:id="@+id/btn_set_one_time_alarm"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Set One Time Alarm"
    android:layout_marginBottom="@dimen/activity_vertical_margin"/>
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Repeating Alarm"
    android:layout_marginBottom="@dimen/activity_vertical_margin"
    android:textSize="16sp"
    android:textStyle="bold"/>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <Button
        android:text="..."
        android:layout_width="50dp"
        android:layout_height="wrap_content"
        android:id="@+id/btn_repeating_time_alarm_time" />

    <TextView
        android:id="@+id/tv_repeating_alarm_time"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Waktu Alarm"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
</LinearLayout>
<EditText
    android:id="@+id/edt_repeating_alarm_message"

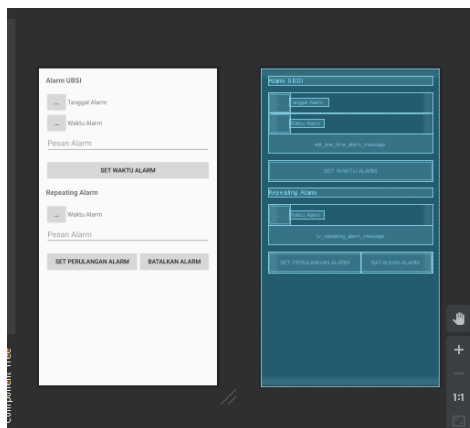
```

```

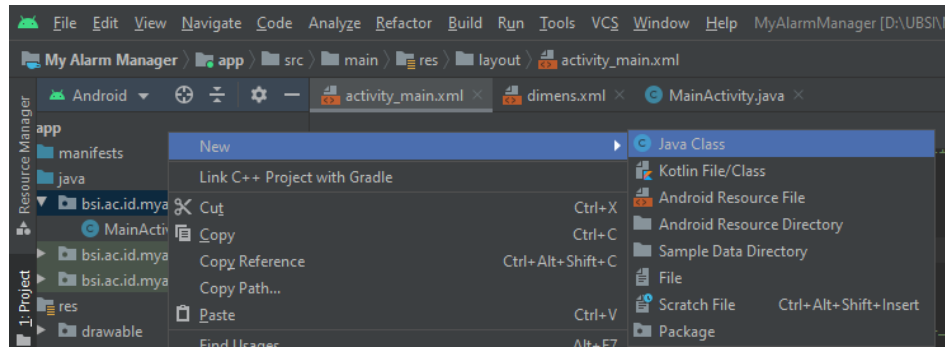
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Pesan Alarm"
        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <Button
            android:id="@+id/btn_repeating_time_alarm"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Set Repeating Alarm"
            android:layout_weight="0.5"
        />
        <Button
            android:id="@+id/btn_cancel_repeating_alarm"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Batalkan Alarm"
            android:layout_weight="0.5"
        />
    </LinearLayout>
</LinearLayout>

```

Sehingga tampilan aplikasi dari activity\_main.xml ketika dijalankan seperti berikut :



- Setelah itu kita akan membuat sebuah mekanisme penyimpanan sederhana dengan menggunakan **SharedPreferences**. Untuk itu, mari kita membuat sebuah kelas java baru dengan nama **AlarmPreference**. Caranya, klik kanan pada package project yang ada saat ini → new → Java Class. Ketika sudah selesai, lengkapi kodenya menjadi seperti berikut.



```
package bsi.ac.id.myalarmmanager;

import android.content.Context;
import android.content.SharedPreferences;

public class AlarmPreference {
    private final String PREF_NAME = "AlarmPreference";
    private final String KEY_ONE_TIME_DATE = "oneTimDate";
    private final String KEY_ONE_TIME_TIME = "oneTimeTime";
    private final String KEY_ONE_TIME_MESSAGE = "oneTimeMessage";
    private final String KEY_REPEATING_TIME = "repeatingTime";
    private final String KEY_REPEATING_MESSAGE = "repeatingMessage";
    private SharedPreferences mSharedPreferences;
    private SharedPreferences.Editor editor;
    public AlarmPreference(MainActivity context) {
        mSharedPreferences = context.getSharedPreferences(PREF_NAME,
Context.MODE_PRIVATE);
        editor = mSharedPreferences.edit();
    }
    public void setOneTimeDate(String date){
        editor.putString(KEY_ONE_TIME_DATE, date);
        editor.commit();
    }
    public String getOneTimeDate() {
        return mSharedPreferences.getString(KEY_ONE_TIME_DATE, null);
    }
    public void setOneTimeTime(String time){
        editor.putString(KEY_ONE_TIME_TIME, time);
        editor.commit();
    }
    public String getOneTimeTime() {
        return mSharedPreferences.getString(KEY_ONE_TIME_TIME, null);
    }
    public void setOneTimeMessage(String message) {
        editor.putString(KEY_ONE_TIME_MESSAGE, message);
        editor.commit();
    }
    public String getOneTimeMessage() {
        return mSharedPreferences.getString(KEY_ONE_TIME_MESSAGE, null);
    }
    public void setRepeatingTime(String time) {
        editor.putString(KEY_REPEATING_TIME, time);
        editor.commit();
    }
}
```

```

    public String getRepeatingTime() {
        return mSharedPreferences.getString(KEY_REPEATING_TIME, null);
    }
    public void setRepeatingMessage(String message) {
        editor.putString(KEY_REPEATING_MESSAGE, message);
        editor.commit();
    }
    public String getRepeatingMessage() {
        return mSharedPreferences.getString(KEY_REPEATING_MESSAGE, null);
    }
    public void clear() {
        editor.clear();
        editor.commit();
    }
}

```

- Setelah itu, buat kelas **BroadcastReceiver** bernama **AlarmReceiver** dan lengkapi kodenya seperti berikut.

```

package bsi.ac.id.myalarmmanager;

import android.annotation.SuppressLint;
import android.app.AlarmManager;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Build;
import android.widget.Toast;

import androidx.core.app.NotificationCompat;
import androidx.core.content.ContextCompat;

import java.util.Calendar;

public class AlarmReceiver extends BroadcastReceiver {
    public static final String TYPE_ONE_TIME = "OneTimeAlarm";
    public static final String TYPE_REPEATING = "RepeatingAlarm";
    public static final String EXTRA_MESSAGE = "message";
    public static final String EXTRA_TYPE = "type";
    private final int NOTIF_ID_ONETIME = 100;
    private final int NOTIF_ID_REPEATING = 101;

    public AlarmReceiver() {
    }

    @Override
    public void onReceive(Context context, Intent intent) {
        String type = intent.getStringExtra(EXTRA_TYPE);
        String message = intent.getStringExtra(EXTRA_MESSAGE);
    }
}

```

```

        String title = type.equalsIgnoreCase(TYPE_ONE_TIME) ? "One Time
Alarm" : "Repeating Alarm";
        int notifId = type.equalsIgnoreCase(TYPE_ONE_TIME) ? NOTIF_ID_ONETIME
: NOTIF_ID_REPEATING;
        showAlarmNotification(context, title, message, notifId);
    }
    private void showAlarmNotification(Context context, String title, String
message, int notifId) {
        NotificationManager notificationManagerCompat = (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);
        Uri alarmSound =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
        // NotificationCompat.Builder(Context context, String channelId)
        // NotificationCompat.Builder builder = new
NotificationCompat.Builder(context)

        String NOTIFICATION_CHANNEL_ID = "notifchannel_01";
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            @SuppressWarnings("WrongConstant") NotificationChannel
notificationChannel = new NotificationChannel(NOTIFICATION_CHANNEL_ID, "My
Notifications", NotificationManager.IMPORTANCE_MAX);
            // Configure the notification channel.
            notificationChannel.setDescription("Sample Channel description");
            notificationChannel.enableLights(true);
            notificationChannel.setLightColor(Color.RED);
            notificationChannel.setVibrationPattern(new long[]{0, 1000, 500,
1000});
            notificationChannel.enableVibration(true);

notificationManagerCompat.createNotificationChannel(notificationChannel);

            NotificationCompat.Builder builder = new
NotificationCompat.Builder(context, NOTIFICATION_CHANNEL_ID)
                .setSmallIcon(R.drawable.ic_baseline_access_alarms_24)
                .setContentTitle(title)
                .setContentText(message)
                .setColor(ContextCompat.getColor(context,
android.R.color.transparent))
                .setVibrate(new long[]{1000, 1000, 1000, 1000, 1000})
                .setSound(alarmSound);
            notificationManagerCompat.notify(notifId, builder.build());
        }
    }
    public void setOneTimeAlarm(MainActivity context, String type, String
date, String time, String message){
        AlarmManager alarmManager =
(AlarmManager) context.getSystemService(Context.ALARM_SERVICE);
        Intent intent = new Intent(context, AlarmReceiver.class);
        intent.putExtra(EXTRA_MESSAGE, message);
        intent.putExtra(EXTRA_TYPE, type);
        String dateArray[] = date.split("-");
        String timeArray[] = time.split(":");
        Calendar calendar = Calendar.getInstance();
        calendar.set(Calendar.YEAR, Integer.parseInt(dateArray[0]));
        calendar.set(Calendar.MONTH, Integer.parseInt(dateArray[1])-1);
        calendar.set(Calendar.DAY_OF_MONTH, Integer.parseInt(dateArray[2]));
        calendar.set(Calendar.HOUR_OF_DAY, Integer.parseInt(timeArray[0]));

```

```

        calendar.set(Calendar.MINUTE, Integer.parseInt(timeArray[1]));
        calendar.set(Calendar.SECOND, 0);
        int requestCode = NOTIF_ID_ONETIME;
        PendingIntent pendingIntent = PendingIntent.getBroadcast(context,
requestCode, intent, 0);
        alarmManager.set(AlarmManager.RTC_WAKEUP,
calendar.getTimeInMillis(), pendingIntent);
        Toast.makeText(context, "One time alarm set up",
Toast.LENGTH_SHORT).show();
    }
    public void cancelAlarm(Context context, String type){
        AlarmManager alarmManager =
(AlarmManager) context.getSystemService(Context.ALARM_SERVICE);
        Intent intent = new Intent(context, AlarmReceiver.class);
        int requestCode = type.equalsIgnoreCase(TYPE_ONE_TIME) ?
NOTIF_ID_ONETIME : NOTIF_ID_REPEATING;
        PendingIntent pendingIntent =
PendingIntent.getBroadcast(context, requestCode, intent, 0);
        alarmManager.cancel(pendingIntent);
    }
}

```

Untuk keperluan ikon alarm, kami menggunakan ikon dari material design spec yang dapat dicari di [halaman ini](#), dibagian kategori peranti. Setelah diunduh, masukkan ikon ke dalam folder res → drawable.

- Setelah kita membuat sebuah AlarmReceiver dan menyesuaikan kode pada activity\_main.xml, sekarang saatnya kita melengkapi MainActivity kodenya sehingga menjadi seperti ini :

```

6. package bsi.ac.id.myalarmmanager;

import androidx.appcompat.app.AppCompatActivity;

import android.app.DatePickerDialog;
import android.app.TimePickerDialog;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.TimePicker;

import java.text.SimpleDateFormat;
import java.util.Calendar;

public class MainActivity extends AppCompatActivity implements
View.OnClickListener{
    private TextView tvOneTimeDate, tvOneTimeTime ;
    private TextView tvRepeatingTime;
    private EditText edtOneTimeMessage , edtRepeatingMessage;
    private Button btnOneTimeDate, btnOneTimeTime, btnOneTime
,btnRepeatingTime ,btnRepeating, btnCancelRepeatingAlarm;

```

```

        private Calendar calOneTimeDate, calOneTimeTime ,
        calRepeatTimeTime;

        private AlarmReceiver alarmReceiver;
        private AlarmPreference alarmPreference;

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

            tvOneTimeDate =
            (TextView)findViewById(R.id.tv_one_time_alarm_date);
            tvOneTimeTime =
            (TextView)findViewById(R.id.tv_one_time_alarm_time);
            edtOneTimeMessage =
            (EditText)findViewById(R.id.edt_one_time_alarm_message);
            btnOneTimeDate =
            (Button)findViewById(R.id.btn_one_time_alarm_date);
            btnOneTimeTime =
            (Button)findViewById(R.id.btn_one_time_alarm_time);
            btnOneTime = (Button)findViewById(R.id.btn_set_one_time_alarm);
            tvRepeatingTime =
            (TextView)findViewById(R.id.tv_repeating_alarm_time);
            edtRepeatingMessage =
            (EditText)findViewById(R.id.edt_repeating_alarm_message);
            btnRepeatingTime =
            (Button)findViewById(R.id.btn_repeating_time_alarm_time);
            btnRepeating =
            (Button)findViewById(R.id.btn_repeating_time_alarm);
            btnCancelRepeatingAlarm =
            (Button)findViewById(R.id.btn_cancel_repeating_alarm);

            btnOneTimeDate.setOnClickListener(this);
            btnOneTimeTime.setOnClickListener(this);
            btnOneTime.setOnClickListener(this);
            btnRepeatingTime.setOnClickListener(this);
            btnRepeating.setOnClickListener(this);
            btnCancelRepeatingAlarm.setOnClickListener(this);

            calOneTimeDate = Calendar.getInstance();
            calOneTimeTime = Calendar.getInstance();
            calRepeatTimeTime = Calendar.getInstance();

            alarmPreference = new AlarmPreference(this);
            alarmReceiver = new AlarmReceiver();

            if (!TextUtils.isEmpty(alarmPreference.getOneTimeDate())) {
                setOneTimeText();
            }
        }
        @Override
        public void onClick(View v) {
            if (v.getId() == R.id.btn_one_time_alarm_date) {
                final Calendar currentDate = Calendar.getInstance();
                new DatePickerDialog(this, new
                DatePickerDialog.OnDateSetListener()

```

```

        {
            @Override
            public void onDateSet(DatePicker view, int year, int
monthOfYear, int dayOfMonth) {
                calOneTimeDate.set(year, monthOfYear, dayOfMonth);
                SimpleDateFormat dateFormat = new
SimpleDateFormat("yyyy-MM-dd");

                tvOneTimeDate.setText(dateFormat.format(calOneTimeDate.getTime()));
            }
        },
        currentDate.get(Calendar.YEAR), currentDate.get(Calendar.MONTH), currentD
ate.get(Calendar.DATE)).show();
    }
    else if (v.getId() == R.id.btn_one_time_alarm_time){
        final Calendar currentDate = Calendar.getInstance();
        new TimePickerDialog(this, new
TimePickerDialog.OnTimeSetListener() {
            @Override
            public void onTimeSet(TimePicker view, int hourOfDay,
int minute){
                calOneTimeTime.set(Calendar.HOUR_OF_DAY,
hourOfDay);
                calOneTimeTime.set(Calendar.MINUTE, minute);
                SimpleDateFormat dateFormat = new
SimpleDateFormat("HH:mm");

                tvOneTimeTime.setText(dateFormat.format(calOneTimeTime.getTime()));
                //Log.v(TAG, "The choosen one " + date.getTime());
            }
        }, currentDate.get(Calendar.HOUR_OF_DAY),
currentDate.get(Calendar.MINUTE), true).show();
    }
    else if (v.getId() == R.id.btn_set_one_time_alarm){
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-
MM-dd");
        String oneTimeDate =
dateFormat.format(calOneTimeDate.getTime());
        SimpleDateFormat timeFormat = new
SimpleDateFormat("HH:mm");
        String oneTimeTime =
timeFormat.format(calOneTimeTime.getTime());
        String oneTimeMessage =
edtOneTimeMessage.getText().toString();

        alarmPreference.setOneTimeDate(oneTimeDate);
        alarmPreference.setOneTimeMessage(oneTimeMessage);
        alarmPreference.setOneTimeTime(oneTimeTime);
        alarmReceiver.setOneTimeAlarm(this,
AlarmReceiver.TYPE_ONE_TIME,
alarmPreference.getOneTimeDate(),
alarmPreference.getOneTimeTime(),
alarmPreference.getOneTimeMessage());
    }
}

private void show() {

```



```

    }

    private void setOneTimeText() {
        tvOneTimeTime.setText(alarmPreference.getOneTimeTime());
        tvOneTimeDate.setText(alarmPreference.getOneTimeDate());
        edtOneTimeMessage.setText(alarmPreference.getOneTimeMessage());
    }
}

```

7. Jangan lupa untuk menambahkan permission pada **AndroidManifest.xml** beberapa baris berikut.

```

<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />

```

Ketika mengimport komponen **Calendar**, pastikan Anda menggunakan **java.util.calendar**. Jangan sampai keliru dengan komponen calendar lainnya yang baru ditambahkan pada API 24.

8. Jika sudah selesai, silakan jalankan aplikasi Anda. Dengan menekan tombol yang ada pada sebelah **TextView** tanggal dan waktu, Anda dapat mengganti tanggal dan waktu yang diinginkan. Isikan juga pesan alarm yang diinginkan. Setelah tanggal, waktu dan pesan telah Anda isi, lanjutkan dengan klik tombol SET ONE TIME ALARM dan silakan tunggu sampai waktu yang ditentukan, sehingga alarm akan dijalankan. Jika sukses, maka waktu yang kita set di awal akan tampil pada obyek **TextView** yang telah disediakan.

Ketika waktunya sudah sesuai, maka alarm akan dijalankan dan akan menampilkan notifikasi dengan getaran dan suara seperti berikut.

## Bedah Kode

Kita mulai dengan skenario nya terlebih dahulu.

Kode pada aplikasi diatas berfungsi untuk menjalankan sebuah alarm yang bersifat *one-time-execution*. Logika dasarnya adalah, kita memanfaatkan fasilitas alarm manager yang terdapat didalam framework Android untuk menjalankan tugas atau proses tertentu berdasarkan waktu yang ditentukan.

Di sini kita akan melakukan proses penampilan notifikasi berdasarkan nilai yang dimasukkan oleh pengguna. Baik itu tanggal, waktu dan pesannya.

Ketika waktu yang ditentukan cocok dengan waktu yang berjalan pada sistem, maka metode onReceive pada BroadcastReceiver akan dijalankan.

Logika yang sederhana bukan?

Ada beberapa istilah baru yang bisa jadi teman-teman baru menemukannya pada modul ini. Untuk memahaminya, mari kita lanjut pembedahan kodenya.

Kita awali dengan pembuatan kelas **AlarmPreference**. Idenya adalah kita dapat menyimpan nilai-nilai terakhir yang kita set sewaktu membuat obyek alarm. Nilai-nilai seperti tanggal, jam, dan pesan akan disimpan ke dalam sebuah mekanisme penyimpanan berbasis pasangan kunci dan nilai (*key value pair*). Mekanisme tersebut memanfaatkan **SharedPreferences**. Semua nilai kita simpan ke dalam tipe data string seperti ini.

```
alarmPreference.setOneTimeDate(oneTimeDate);
alarmPreference.setOneTimeMessage(oneTimeMessage);
alarmPreference.setOneTimeTime(oneTimeTime);
```

Ketika pengguna membuka kembali aplikasi, nilai-nilai di atas akan ditampilkan ke dalam obyek TextView.

```
private void setOneTimeText() {
    tvOneTimeTime.setText(alarmPreference.getOneTimeTime());
    tvOneTimeDate.setText(alarmPreference.getOneTimeDate());
    edtOneTimeMessage.setText(alarmPreference.getOneTimeMessage());
}
```

Anda mempelajari sesuatu yang beda saat ini. Sebuah mekanisme penyimpanan yang paling sederhana dan paling mudah digunakan.

Esensi dari kelas AlarmPreference adalah untuk menyimpan nilai terakhir yang dimasukkan pengguna sewaktu memasang alarm. Anda dapat menyimpan beragam tipe data selain string. Misalnya data bertipekan double, int, boolean, float, long dan stringset.

Anda akan mendapatkan detail tentang SharedPreferences lebih mendalam lagi pada modul ke 3 nanti. Selanjutnya pada metode onclick, terdapat beberapa kondisi if-else yang berguna untuk membedakan button mana yang disentuh. Button dengan id btn\_one\_time\_alarm\_date dan btn\_one\_time\_alarm\_time memiliki fungsi menampilkan dialog yang menampilkan date picker dan time picker.

```
public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {
    calOneTimeDate.set(year, monthOfYear, dayOfMonth);
    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
    tvOneTimeDate.setText(dateFormat.format(calOneTimeDate.getTime()));
}

},
currentDate.get(Calendar.YEAR), currentDate.get(Calendar.MONTH), currentDate.get(Calendar.DATE)).show();
}
else if (v.getId() == R.id.btn_one_time_alarm_time){
    final Calendar currentDate = Calendar.getInstance();
    new TimePickerDialog(this, new TimePickerDialog.OnTimeSetListener() {
        @Override
        public void onTimeSet(TimePicker view, int hourOfDay, int minute){
            calOneTimeTime.set(Calendar.HOUR_OF_DAY, hourOfDay);
            calOneTimeTime.set(Calendar.MINUTE, minute);
            SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm");
            tvOneTimeTime.setText(dateFormat.format(calOneTimeTime.getTime()));
            //Log.v(TAG, "The choosen one " + date.getTime());
        }
    }, currentDate.get(Calendar.HOUR_OF_DAY), currentDate.get(Calendar.MINUTE),
true).show();
}
```

Selanjutnya, perhatikan pada baris berikut ini.

```
public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {
    calOneTimeDate.set(year, monthOfYear, dayOfMonth);
}
```

```
SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
tvOneTimeDate.setText(dateFormat.format(calOneTimeDate.getTime()));
}
```

Fungsi `onDateSet` akan dipanggil ketika kita memilih tanggal yang kita inginkan. Kemudian kita pasang variable `calOneTimeDate` dengan tanggal yang kita pilih. Variable tersebut dapat digunakan untuk mengisi `TextView` tanggal alarm.

Untuk mendapatkan string dari variable `calOneTimeDate`, kita menggunakan komponen `SimpleDateFormat`. Perhatikan kode “yyyy-MM-dd”, ini adalah format untuk menampilkan tanggal dengan format seperti 2016-09-29.

Terdapat beberapa format yang dapat kita gunakan dengan komponen `SimpleDateFormat`. Untuk mengetahui lebih dalam mengenai `SimpleDateFormat`, Anda dapat langsung menuju tautan berikut :

- SimpleDateFormat

Selanjutnya, pada `AlarmReceiver` kita membuat beberapa variabel konstanta penanda (*flag*) yang akan digunakan di keseluruhan bagian kode, diantaranya :

```
public static final String TYPE_ONE_TIME = "OneTimeAlarm";
public static final String TYPE_REPEATING = "RepeatingAlarm";
```

Dua baris diatas adalah konstanta untuk menentukan tipe alarm. Dan selanjutnya, dua baris dibawah ini adalah konstanta untuk intent key.

```
public static final String EXTRA_MESSAGE = "message";
public static final String EXTRA_TYPE = "type";
```

Di sini kita menggunakan dua konstanta bertipe data integer untuk menentukan notif ID sebagai ID untuk menampilkan notifikasi kepada pengguna.

```
private final int NOTIF_ID_ONETIME = 100;
private final int NOTIF_ID_REPEATING = 101;
```

Selanjutnya, bagian penting pada metode `setOneTimeAlarm()` adalah :

```
public void setOneTimeAlarm(Context context, String type, String date, String time,
String message){
    AlarmManager alarmManager =
(AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
    Intent intent = new Intent(context, AlarmReceiver.class);
    intent.putExtra(EXTRA_MESSAGE, message);
    intent.putExtra(EXTRA_TYPE, type);
    String dateArray[] = date.split("-");
    String timeArray[] = time.split(":");
    Calendar calendar = Calendar.getInstance();
    calendar.set(Calendar.YEAR, Integer.parseInt(dateArray[0]));
    calendar.set(Calendar.MONTH, Integer.parseInt(dateArray[1])-1);
    calendar.set(Calendar.DAY_OF_MONTH, Integer.parseInt(dateArray[2]));
    calendar.set(Calendar.HOUR_OF_DAY, Integer.parseInt(timeArray[0]));
    calendar.set(Calendar.MINUTE, Integer.parseInt(timeArray[1]));
    calendar.set(Calendar.SECOND, 0);
    int requestCode = NOTIF_ID_ONETIME;
    PendingIntent pendingIntent = PendingIntent.getBroadcast(context, requestCode,
```

```
intent, 0);
    alarmManager.set(AlarmManager.RTC_WAKEUP,    calendar.getTimeInMillis(),
pendingIntent);
    Toast.makeText(context, "One time alarm set up", Toast.LENGTH_SHORT).show();
}
```

Pada kode di atas, kita membuat sebuah obyek dari kelas AlarmManager. Kita menyiapkan sebuah Intent yang akan menjalankan AlarmReceiver dan membawa data alarm dan pesan.

```
String dateArray[] = date.split("-");
String timeArray[] = time.split(":");
Calendar calendar = Calendar.getInstance();
calendar.set(Calendar.YEAR, Integer.parseInt(dateArray[0]));
calendar.set(Calendar.MONTH, Integer.parseInt(dateArray[1])-1);
calendar.set(Calendar.DAY_OF_MONTH, Integer.parseInt(dateArray[2]));
calendar.set(Calendar.HOUR_OF_DAY, Integer.parseInt(timeArray[0]));
calendar.set(Calendar.MINUTE, Integer.parseInt(timeArray[1]));
calendar.set(Calendar.SECOND, 0);
```

Pada kode di atas, kita memecah data date dan time untuk mengambil nilai tahun, bulan, hari, jam dan menit.

```
calendar.set(Calendar.MONTH, Integer.parseInt(dateArray[1])-1);
```

Perlu menjadi perhatian mengapa kode di atas, data kita kurang 1.

Misal tanggal yang kita masukkan adalah 2016-09-27. Jika kita pecah, maka kita akan memperoleh nilai 2016 (tahun), 9 (bulan), dan 27 (hari).

Masalahnya adalah, nilai bulan ke 9 pada kelas Calendar bukanlah bulan September. Ini karena indeksny dimulai dari 0. Jadi, untuk memperoleh bulan September, maka nilai 9 tadi harus kita kurangi 1.

Intent yang dibuat akan dieksekusi ketika waktu alarm sama dengan waktu pada sistem Android. Di sini komponen **PendingIntent** akan diberikan kepada **BroadcastReceiver**.

```
int requestCode = NOTIF_ID_ONETIME;
PendingIntent pendingIntent = PendingIntent.getBroadcast(context, requestCode,
intent, 0);
alarmManager.set(AlarmManager.RTC_WAKEUP,    calendar.getTimeInMillis(),
pendingIntent);
```

Yang membedakan satu alarm dengan alarm lain adalah pada nilai **requestCode**.

```
int requestCode = NOTIF_ID_ONETIME;
PendingIntent pendingIntent = PendingIntent.getBroadcast(context, requestCode,
intent, 0);
```

Jika kita merubah nilai waktu dan menjalankan ulang alarm dengan requestCode yang sama, maka akan merubah yang sudah diset sebelumnya.

Maksud dari baris ini **alarmManager.set(AlarmManager.RTC\_WAKEUP, calendar.getTimeInMillis(), pendingIntent);** adalah kita memasang alarm yang dibuat dengan tipe RTC\_WAKEUP. Tipe alarm ini dapat membangunkan peranti (jika dalam posisi sleep) untuk menjalankan obyek PendingIntent. Ketika kondisi sesuai, maka akan menjalankan **BroadcastReceiver** dengan semua proses yang terdapat didalam method **onReceive()**.

```
String type = intent.getStringExtra(EXTRA_TYPE);
String message = intent.getStringExtra(EXTRA_MESSAGE);
```

```
String title = type.equalsIgnoreCase(TYPE_ONE_TIME) ? "One Time Alarm" : "Repeating Alarm";
int notifId = type.equalsIgnoreCase(TYPE_ONE_TIME) ? NOTIF_ID_ONETIME : NOTIF_ID_REPEATING;
showAlarmNotification(context, title, message, notifId);
```

Sebuah notifikasi sederhana akan tampil pada panel notifikasi pengguna dengan bunyi notifikasi umum dan getaran. Di sini metode `showAlarmNotification(context, title, message, notifId);` dijalankan.

Kita lanjutkan ngoding untuk latihan alarm manager. Kali ini kita akan membuat latihan sederhana tentang *repeating-alarm*.

1. Pada AlarmReceiver silakan buat metode `setRepeatingAlarm()` pada baris berikut. Sehingga kode AlarmManager kita saat ini menjadi seperti ini.

```
package bsi.ac.id.myalarmmanager;

import android.annotation.SuppressLint;
import android.app.AlarmManager;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Build;
import android.widget.Toast;

import androidx.core.app.NotificationCompat;
import androidx.core.content.ContextCompat;

import java.util.Calendar;

public class AlarmReceiver extends BroadcastReceiver {
    public static final String TYPE_ONE_TIME = "OneTimeAlarm";
    public static final String TYPE_REPEATING = "RepeatingAlarm";
    public static final String EXTRA_MESSAGE = "message";
    public static final String EXTRA_TYPE = "type";
    private final int NOTIF_ID_ONETIME = 100;
    private final int NOTIF_ID_REPEATING = 101;

    public AlarmReceiver() {
    }

    @Override
    public void onReceive(Context context, Intent intent) {
        String type = intent.getStringExtra(EXTRA_TYPE);
        String message = intent.getStringExtra(EXTRA_MESSAGE);
        String title = type.equalsIgnoreCase(TYPE_ONE_TIME) ? "One Time Alarm" :
```

```

"Repeating Alarm";
    int notifId = type.equalsIgnoreCase(TYPE_ONE_TIME) ? NOTIF_ID_ONETIME :
NOTIF_ID_REPEATING;
    showAlarmNotification(context, title, message, notifId);
}
private void showAlarmNotification(Context context, String title, String message,
int notifId) {
    NotificationManager notificationManagerCompat = (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);
    Uri alarmSound =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
    // NotificationCompat.Builder(Context context, String channelId)
    // NotificationCompat.Builder builder = new
NotificationCompat.Builder(context)

    String NOTIFICATION_CHANNEL_ID = "notifchannel_01";
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        @SuppressWarnings("WrongConstant") NotificationChannel notificationChannel =
new NotificationChannel(NOTIFICATION_CHANNEL_ID, "My Notifications",
NotificationManager.IMPORTANCE_MAX);
        // Configure the notification channel.
        notificationChannel.setDescription("Sample Channel description");
        notificationChannel.enableLights(true);
        notificationChannel.setLightColor(Color.RED);
        notificationChannel.setVibrationPattern(new long[]{0, 1000, 500, 1000});
        notificationChannel.enableVibration(true);
        notificationManagerCompat.createNotificationChannel(notificationChannel);

        NotificationCompat.Builder builder = new
NotificationCompat.Builder(context, NOTIFICATION_CHANNEL_ID)
            .setSmallIcon(R.drawable.ic_access_alarm_black_36dp)
            .setContentTitle(title)
            .setContentText(message)
            .setColor(ContextCompat.getColor(context,
android.R.color.transparent))
            .setVibrate(new long[]{1000, 1000, 1000, 1000, 1000})
            .setSound(alarmSound);
        notificationManagerCompat.notify(notifId, builder.build());
    }
}
public void setOneTimeAlarm(Context context, String type, String date,String
time, String message){
    AlarmManager alarmManager =
(AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
    Intent intent = new Intent(context, AlarmReceiver.class);
    intent.putExtra(EXTRA_MESSAGE, message);
    intent.putExtra(EXTRA_TYPE, type);
    String dateArray[] = date.split("-");
    String timeArray[] = time.split(":");
    Calendar calendar = Calendar.getInstance();
    calendar.set(Calendar.YEAR, Integer.parseInt(dateArray[0]));
    calendar.set(Calendar.MONTH, Integer.parseInt(dateArray[1])-1);
    calendar.set(Calendar.DAY_OF_MONTH, Integer.parseInt(dateArray[2]));
    calendar.set(Calendar.HOUR_OF_DAY, Integer.parseInt(timeArray[0]));
    calendar.set(Calendar.MINUTE, Integer.parseInt(timeArray[1]));
}

```

```

        calendar.set(Calendar.SECOND, 0);
        int requestCode = NOTIF_ID_ONETIME;
        PendingIntent pendingIntent = PendingIntent.getBroadcast(context,
requestCode, intent, 0);
        alarmManager.set(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(),
pendingIntent);
        Toast.makeText(context, "One time alarm set up", Toast.LENGTH_SHORT).show();
    }

    public void setRepeatingAlarm(Context context, String type, String time, String
message){
        AlarmManager alarmManager =
(AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
        Intent intent = new Intent(context, AlarmReciver.class);
        intent.putExtra(EXTRA_MESSAGE, message);
        intent.putExtra(EXTRA_TYPE, type);
        String timeArray[] = time.split(":");
        Calendar calendar = Calendar.getInstance();
        calendar.set(Calendar.HOUR_OF_DAY, Integer.parseInt(timeArray[0]));
        calendar.set(Calendar.MINUTE, Integer.parseInt(timeArray[1]));
        calendar.set(Calendar.SECOND, 0);
        int requestCode = NOTIF_ID_ONETIME;
        PendingIntent pendingIntent = PendingIntent.getBroadcast(context,
requestCode, intent, 0);
        alarmManager.setInexactRepeating(AlarmManager.RTC_WAKEUP,
calendar.getTimeInMillis(), AlarmManager.INTERVAL_DAY, pendingIntent);
        Toast.makeText(context, "Repeating alarm set up", Toast.LENGTH_SHORT).show();
    }

    public void cancelAlarm (Context context, String type){
        AlarmManager alarmManager =
(AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
        Intent intent = new Intent(context, AlarmReciver.class);
        int requestCode = type.equalsIgnoreCase(TYPE_ONE_TIME) ? NOTIF_ID_ONETIME :
NOTIF_ID_REPEATING;
        PendingIntent pendingIntent = PendingIntent.getBroadcast(context,requestCode,
intent, 0);
        alarmManager.cancel(pendingIntent);
    }
}

```

2. Selanjutnya, pada method onClick() di MainActivity silakan tambahkan baris berikut :

```

3. public class MainActivity extends AppCompatActivity implements
View.OnClickListener{
    private TextView tvOneTimeDate, tvOneTimeTime;
    private TextView tvRepeatingTime;
    private EditText edtOneTimeMessage , edtRepeatingMessage;
    private Button btnOneTimeDate, btnOneTimeTime, btnOneTime
,btnRepeatingTime ,btnRepeating, btnCancelRepeatingAlarm;
    private Calendar calOneTimeDate, calOneTimeTime , calRepeatTimeTime;
    private AlarmReceiver alarmReceiver;
    private AlarmPreference alarmPreference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```



```

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        tvOneTimeDate = (TextView)findViewById(R.id.tv_one_time_alarm_date);
        tvOneTimeTime = (TextView)findViewById(R.id.tv_one_time_alarm_time);
        edtOneTimeMessage =
        (EditText)findViewById(R.id.edt_one_time_alarm_message);
        btnOneTimeDate = (Button)findViewById(R.id.btn_one_time_alarm_date);
        btnOneTimeTime = (Button)findViewById(R.id.btn_one_time_alarm_time);
        btnOneTime = (Button)findViewById(R.id.btn_set_one_time_alarm);
        tvRepeatingTime =
        (TextView)findViewById(R.id.tv_repeating_alarm_time);
        edtRepeatingMessage =
        (EditText)findViewById(R.id.edt_repeating_alarm_message);
        btnRepeatingTime =
        (Button)findViewById(R.id.btn_repeating_time_alarm_time);
        btnRepeating = (Button)findViewById(R.id.btn_repeating_time_alarm);
        btnCancelRepeatingAlarm =
        (Button)findViewById(R.id.btn_cancel_repeating_alarm);

        btnOneTimeDate.setOnClickListener(this);
        btnOneTimeTime.setOnClickListener(this);
        btnOneTime.setOnClickListener(this);
        btnRepeatingTime.setOnClickListener(this);
        btnRepeating.setOnClickListener(this);
        btnCancelRepeatingAlarm.setOnClickListener(this);

        calOneTimeDate = Calendar.getInstance();
        calOneTimeTime = Calendar.getInstance();
        calRepeatTimeTime = Calendar.getInstance();

        alarmPreference = new AlarmPreference(this);
        alarmReceiver = new AlarmReceiver();

        if (!TextUtils.isEmpty(alarmPreference.getOneTimeDate())){
            setOneTimeText();
        }
    }
    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_one_time_alarm_date){
            final Calendar currentDate = Calendar.getInstance();
            new DatePickerDialog(this, new
            DatePickerDialog.OnDateSetListener()
            {
                @Override
                public void onDateSet(DatePicker view, int year, int
                monthOfYear, int dayOfMonth) {
                    calOneTimeDate.set(year, monthOfYear, dayOfMonth);
                    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-
                    MM-dd");

                    tvOneTimeDate.setText(dateFormat.format(calOneTimeDate.getTime()));
                }
            },
        },

```



```

currentDate.get(Calendar.YEAR),currentDate.get(Calendar.MONTH),currentDate.get(
(Calendar.DATE)).show();
    }
    else if (v.getId() == R.id.btn_one_time_alarm_time){
        final Calendar currentDate = Calendar.getInstance();
        new TimePickerDialog(this, new
TimePickerDialog.OnTimeSetListener() {
            @Override
            public void onTimeSet(TimePicker view, int hourOfDay, int
minute){
                calOneTimeTime.set(Calendar.HOUR_OF_DAY, hourOfDay);
                calOneTimeTime.set(Calendar.MINUTE, minute);
                SimpleDateFormat dateFormat = new
SimpleDateFormat("HH:mm");

                tvOneTimeTime.setText(dateFormat.format(calOneTimeTime.getTime()));
                //Log.v(TAG, "The choosen one " + date.getTime());
            }
        }, currentDate.get(Calendar.HOUR_OF_DAY),
currentDate.get(Calendar.MINUTE), true).show();
    }

    else if (v.getId() == R.id.btn_set_one_time_alarm){
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
        String oneTimeDate = dateFormat.format(calOneTimeDate.getTime());
        SimpleDateFormat timeFormat = new SimpleDateFormat("HH:mm");
        String oneTimeTime = timeFormat.format(calOneTimeTime.getTime());
        String oneTimeMessage = edtOneTimeMessage.getText().toString();

        alarmPreference.setOneTimeDate(oneTimeDate);
        alarmPreference.setOneTimeMessage(oneTimeMessage);
        alarmPreference.setOneTimeTime(oneTimeTime);
        alarmReceiver.setOneTimeAlarm(this, AlarmReciver.TYPE_ONE_TIME,
            alarmPreference.getOneTimeDate(),
            alarmPreference.getOneTimeTime(),
            alarmPreference.getOneTimeMessage());
    }
    else if (v.getId() == R.id.btn_repeating_time_alarm_time){
        final Calendar currentDate = Calendar.getInstance();
        new TimePickerDialog(this, new
TimePickerDialog.OnTimeSetListener() {
            @Override
            public void onTimeSet(TimePicker view, int hourOfDay, int
minute) {
                calRepeatTimeTime.set(Calendar.HOUR_OF_DAY, hourOfDay);
                calRepeatTimeTime.set(Calendar.MINUTE, minute);
                SimpleDateFormat dateFormat = new
SimpleDateFormat("HH:mm");

                tvRepeatingTime.setText(dateFormat.format(calRepeatTimeTime.getTime()));
            }
        },
            currentDate.get(Calendar.HOUR_OF_DAY),
currentDate.get(Calendar.MINUTE), true).show();
    }
}

```

```

        else if (v.getId() == R.id.btn_repeating_time_alarm){
            SimpleDateFormat timeFormat = new SimpleDateFormat("HH:mm");
            String repeatTimeTime =
timeFormat.format(calRepeatTimeTime.getTime());
            String repeatTimeMessage =
edtRepeatingMessage.getText().toString();

            alarmPreference.setRepeatingTime(repeatTimeTime);
            alarmPreference.setRepeatingMessage(repeatTimeMessage);

            alarmReceiver.setRepeatingAlarm(this, AlarmReciver.TYPE_REPEATING,
alarmPreference.getRepeatingTime(),
alarmPreference.getRepeatingMessage());

        }

    }

    private void setOneTimeText() {
        tvOneTimeTime.setText(alarmPreference.getOneTimeTime());
        tvOneTimeDate.setText(alarmPreference.getOneTimeDate());
        edtOneTimeMessage.setText(alarmPreference.getOneTimeMessage());
    }

}

```

4. Kemudian lengkapi dengan membuat method setRepeatingText() seperti berikut :

```

private void SetRepeatingText(){
    tvRepeatingTime.setText(alarmPreference.getRepeatingTime());
    edtRepeatingMessage.setText(alarmPreference.getRepeatingMessage());
}

```

5. Silakan tambahkan baris berikut di method onCreate() yang dapat memberikan nilai default yang pernah kita gunakan ketika aplikasi kita buka

```

if (!TextUtils.isEmpty(alarmPreference.getRepeatingTime())){
    setRepeatingText();
}

```

6. Jika berhasil seharusnya notifikasi akan muncul setiap hari pada waktu yang kamu tentukan seperti ini.

## Bedah Code

Tidak ada perubahan yang signifikan ketika kita membuat metode **setRepeatingAlarm()** dengan method setOneTimeAlarm(). Yang membedakan adalah 115

pada pemanggilan metode terkait. Pada **setOneTimeAlarm()** kita menggunakan metode **set()** sedangkan pada **setRepeatingAlarm()** kita menggunakan **setInexactRepeating()**.

```

public void setRepeatingAlarm(Context context, String type, String time, String
message){
    AlarmManager alarmManager =
(AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
    Intent intent = new Intent(context, AlarmReceiver.class);
    intent.putExtra(EXTRA_MESSAGE, message);
    intent.putExtra(EXTRA_TYPE, type);
    String timeArray[] = time.split(":");
    Calendar calendar = Calendar.getInstance();
    calendar.set(Calendar.HOUR_OF_DAY, Integer.parseInt(timeArray[0]));
    calendar.set(Calendar.MINUTE, Integer.parseInt(timeArray[1]));
    calendar.set(Calendar.SECOND, 0);
    int requestCode = NOTIF_ID_ONETIME;
    PendingIntent pendingIntent = PendingIntent.getBroadcast(context, requestCode,
intent, 0);
    alarmManager.setInexactRepeating(AlarmManager.RTC_WAKEUP,
calendar.getTimeInMillis(), AlarmManager.INTERVAL_DAY, pendingIntent);
    Toast.makeText(context, "Repeating alarm set up", Toast.LENGTH_SHORT).show();
}

```

Secara umum proses yang disusun pada metode diatas adalah sama. Yang membedakan hanya ide dan skenarionya. Disini kita akan mendemonstrasikan sebuah proses yang berulang dengan satuan waktu yang ditentukan dengan memanfaatkan fasilitas **AlarmManager**.

```

alarmManager.setInexactRepeating(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(),
AlarmManager.INTERVAL_DAY, pendingIntent);

```

Baris diatas akan menjalankan obyek PendingIntent di setiap waktu yang ditentukan dalam millisecond dengan interval per hari. Penggunaan metode setInexactRepeating() menjadi pilihan yang lebih tepat karena Android akan menjalankan alarm ini secara bersamaan dengan alarm lain. Meskipun waktu antara tiap alarm tersebut tidak sama persis.

Ini penting dilakukan agar baterai device lebih hemat dan tidak cepat habis. Ingat! Penggunaan alarm manager yang tidak tepat akan memakan daya baterai yang cukup besar.

### Topik Alarm Manager - Latihan: Alarm Manager Sederhana 3

1. Berkas activity\_main.xml sekarang seharusnya berisi kode berikut :

Ubah kode di atas sehingga menjadi :

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <Button
        android:id="@+id/btn_repeating_time_alarm"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Set Repeating Alarm"
        android:layout_weight="0.5" />
    <Button
        android:id="@+id/btn_cancel_repeating_alarm"
        android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content"
        android:text="Batalan Alarm"
        android:layout_weight="0.5" />

```

2. Selanjutnya pada MainActivity tambahkan btnCancelRepeating pada deklarasi obyek Button seperti ini

```

public class MainActivity extends AppCompatActivity implements View.OnClickListener{
    private TextView tvOneTimeDate, tvOneTimeTime;
    private TextView tvRepeatingTime;
    private EditText edtOneTimeMessage , edtRepeatingMessage;
    private Button btnOneTimeDate, btnOneTimeTime, btnOneTime ,btnRepeatingTime
    ,btnRepeating, btnCancelRepeatingAlarm;
    private Calendar calOneTimeDate, calOneTimeTime , calRepeatTimeTime;
    private AlarmReciver alarmReceiver;
    private AlarmPreference alarmPreference;

```

Seperti biasa, lakukan casting obyek view btnCancelRepeatingAlarm agar view tersebut dapat kita manipulasi seperti ini :

```

btnCancelRepeatingAlarm = (Button)findViewById(R.id.btn_cancel_repeating_alarm);
btnCancelRepeatingAlarm.setOnClickListener(this);

```

3. Selanjutnya kita tambahkan kode seleksi baru pada method onClick() yang akan menjalankan proses pembatalan repeating alarm ketika tombol Batalan Alarm diklik. Sehingga kode MainActivity kita secara keseluruhan menjadi seperti berikut.

```

else if (v.getId() == R.id.btn_cancel_repeating_alarm){
    alarmReceiver.cancelAlarm(this, AlarmReciver.TYPE_REPEATING);
}

```

```

public class MainActivity extends AppCompatActivity implements View.OnClickListener{
    private TextView tvOneTimeDate, tvOneTimeTime;
    private TextView tvRepeatingTime;
    private EditText edtOneTimeMessage , edtRepeatingMessage;
    private Button btnOneTimeDate, btnOneTimeTime, btnOneTime ,btnRepeatingTime
    ,btnRepeating, btnCancelRepeatingAlarm;
    private Calendar calOneTimeDate, calOneTimeTime , calRepeatTimeTime;
    private AlarmReciver alarmReceiver;
    private AlarmPreference alarmPreference;

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

        tvOneTimeDate = (TextView)findViewById(R.id.tv_one_time_alarm_date);
        tvOneTimeTime = (TextView)findViewById(R.id.tv_one_time_alarm_time);
        edtOneTimeMessage = (EditText)findViewById(R.id.edt_one_time_alarm_message);
        btnOneTimeDate = (Button)findViewById(R.id.btn_one_time_alarm_date);
        btnOneTimeTime = (Button)findViewById(R.id.btn_one_time_alarm_time);
        btnOneTime = (Button)findViewById(R.id.btn_set_one_time_alarm);
        tvRepeatingTime = (TextView)findViewById(R.id.tv_repeating_alarm_time);
        edtRepeatingMessage =
        (EditText)findViewById(R.id.edt_repeating_alarm_message);
        btnRepeatingTime = (Button)findViewById(R.id.btn_repeating_time_alarm_time);

```

```

        btnRepeating = (Button)findViewById(R.id.btn_repeating_time_alarm);
        btnCancelRepeatingAlarm =
(Button)findViewById(R.id.btn_cancel_repeating_alarm);

        btnOneTimeDate.setOnClickListener(this);
        btnOneTimeTime.setOnClickListener(this);
        btnOneTime.setOnClickListener(this);
        btnRepeatingTime.setOnClickListener(this);
        btnRepeating.setOnClickListener(this);
        btnCancelRepeatingAlarm.setOnClickListener(this);

        calOneTimeDate = Calendar.getInstance();
        calOneTimeTime = Calendar.getInstance();
        calRepeatTimeTime = Calendar.getInstance();

        alarmPreference = new AlarmPreference(this);
        alarmReceiver = new AlarmReciver();

        if (!TextUtils.isEmpty(alarmPreference.getOneTimeDate())){
            setOneTimeText();
        }
        if (!TextUtils.isEmpty(alarmPreference.getRepeatingTime())){
            setRepeatingText();
        }
    }
    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_one_time_alarm_date){
            final Calendar currentDate = Calendar.getInstance();
            new DatePickerDialog(this, new DatePickerDialog.OnDateSetListener()
            {
                @Override
                public void onDateSet(DatePicker view, int year, int monthOfYear, int
dayOfMonth) {
                    calOneTimeDate.set(year, monthOfYear, dayOfMonth);
                    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

                    tvOneTimeDate.setText(dateFormat.format(calOneTimeDate.getTime()));
                }
            },
            currentDate.get(Calendar.YEAR),currentDate.get(Calendar.MONTH),currentDate.get(Calend
ar.DATE)).show();
        }
        else if (v.getId() == R.id.btn_one_time_alarm_time){
            final Calendar currentDate = Calendar.getInstance();
            new TimePickerDialog(this, new TimePickerDialog.OnTimeSetListener() {
                @Override
                public void onTimeSet(TimePicker view, int hourOfDay, int minute){
                    calOneTimeTime.set(Calendar.HOUR_OF_DAY, hourOfDay);
                    calOneTimeTime.set(Calendar.MINUTE, minute);
                    SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm");

                    tvOneTimeTime.setText(dateFormat.format(calOneTimeTime.getTime()));
                    //Log.v(TAG, "The choosen one " + date.getTime());
                }
            }

```

```

        }, currentDate.get(Calendar.HOUR_OF_DAY),
currentDate.get(Calendar.MINUTE), true).show();
    }

    else if (v.getId() == R.id.btn_set_one_time_alarm){
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
        String oneTimeDate = dateFormat.format(calOneTimeDate.getTime());
        SimpleDateFormat timeFormat = new SimpleDateFormat("HH:mm");
        String oneTimeTime = timeFormat.format(calOneTimeTime.getTime());
        String oneTimeMessage = edtOneTimeMessage.getText().toString();

        alarmPreference.setOneTimeDate(oneTimeDate);
        alarmPreference.setOneTimeMessage(oneTimeMessage);
        alarmPreference.setOneTimeTime(oneTimeTime);
        alarmReceiver.setOneTimeAlarm(this, AlarmReciver.TYPE_ONE_TIME,
            alarmPreference.getOneTimeDate(),
            alarmPreference.getOneTimeTime(),
            alarmPreference.getOneTimeMessage());
    }
    else if (v.getId() == R.id.btn_repeating_time_alarm_time){
        final Calendar currentDate = Calendar.getInstance();
        new TimePickerDialog(this, new TimePickerDialog.OnTimeSetListener() {
            @Override
            public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
                calRepeatTimeTime.set(Calendar.HOUR_OF_DAY, hourOfDay);
                calRepeatTimeTime.set(Calendar.MINUTE, minute);
                SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm");
tvRepeatingTime.setText(dateFormat.format(calRepeatTimeTime.getTime()));
            }
        },
            currentDate.get(Calendar.HOUR_OF_DAY),
currentDate.get(Calendar.MINUTE), true).show();
    }
    else if (v.getId() == R.id.btn_repeating_time_alarm){
        SimpleDateFormat timeFormat = new SimpleDateFormat("HH:mm");
        String repeatTimeTime = timeFormat.format(calRepeatTimeTime.getTime());
        String repeatTimeMessage = edtRepeatingMessage.getText().toString();

        alarmPreference.setRepeatingTime(repeatTimeTime);
        alarmPreference.setRepeatingMessage(repeatTimeMessage);

        alarmReceiver.setRepeatingAlarm(this, AlarmReciver.TYPE_REPEATING,
            alarmPreference.getRepeatingTime(),
alarmPreference.getRepeatingMessage());
    }
    else if (v.getId() == R.id.btn_cancel_repeating_alarm){
        alarmReceiver.cancelAlarm(this, AlarmReciver.TYPE_REPEATING);
    }
}

private void setOneTimeText() {
    tvOneTimeTime.setText(alarmPreference.getOneTimeTime());
    tvOneTimeDate.setText(alarmPreference.getOneTimeDate());
}

```

```

        edtOneTimeMessage.setText(alarmPreference.getOneTimeMessage());
    }

    private void setRepeatingText(){
        tvRepeatingTime.setText(alarmPreference.getRepeatingTime());
        edtRepeatingMessage.setText(alarmPreference.getRepeatingMessage());
    }
}

```

4. Ketika aplikasi dijalankan, maka tampilannya kurang lebih menjadi seperti ini.
5. Terakhir, tambahkan sebuah Toast seperti berikut :
6. Metode cancelAlarm() akan berubah menjadi :

```

public void cancelAlarm (Context context, String type){
    AlarmManager alarmManager =
    (AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
    Intent intent = new Intent(context, AlarmReceiver.class);
    int requestCode = type.equalsIgnoreCase(TYPE_ONE_TIME) ? NOTIF_ID_ONETIME :
    NOTIF_ID_REPEATING;
    PendingIntent pendingIntent = PendingIntent.getBroadcast(context, requestCode,
    intent, 0);
    alarmManager.cancel(pendingIntent);
    Toast.makeText(context, "Repeating alarm dibatalkan", Toast.LENGTH_SHORT).show();
}

```

Selesai! Sekarang mari kita uji kodenya. Skenarionya seperti berikut :

1. Ubah nilai pada variabel **repeatingTime** di **MainActivity** sesuai dengan waktu yang mendekati saat ini (bedakan sekitar 5 menit dari sekarang). Sebagai contoh, saat ini waktu dipasang pada pukul 01.45
2. Klik tombol SET REPEATING ALARM untuk menjalankan alarm untuk waktu yang ditentukan
3. Klik tombol BATALKAN ALARM untuk membatalkan alarm yang baru saja dijalankan.
4. Jika sukses seharusnya pada waktu yang diberikan tidak tampil notifikasi alarm seperti sebelumnya.

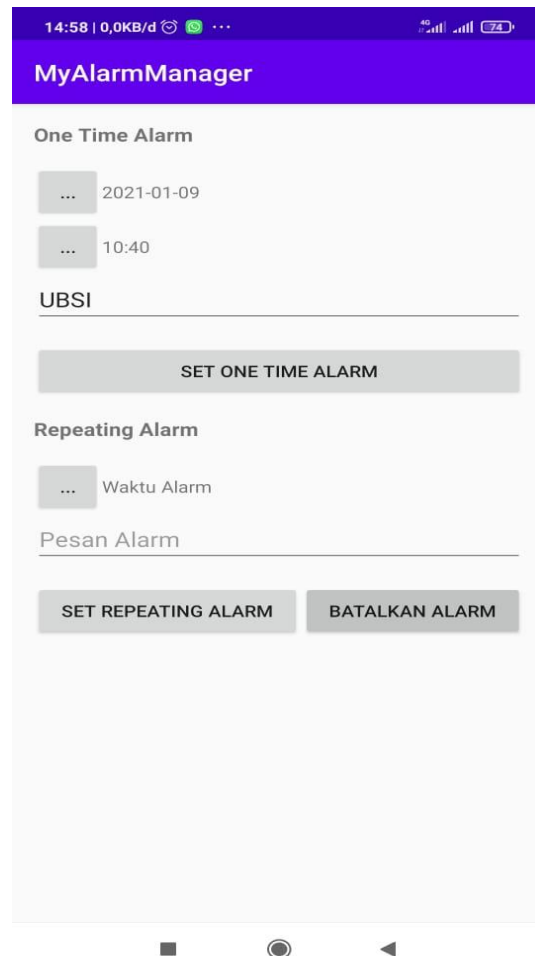
### Bedah Kode

```

public void cancelAlarm (Context context, String type){
    AlarmManager alarmManager =
    (AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
    Intent intent = new Intent(context, AlarmReceiver.class);
    int requestCode = type.equalsIgnoreCase(TYPE_ONE_TIME) ? NOTIF_ID_ONETIME :
    NOTIF_ID_REPEATING;
    PendingIntent pendingIntent = PendingIntent.getBroadcast(context, requestCode,
    intent, 0);
    alarmManager.cancel(pendingIntent);
    Toast.makeText(context, "Repeating alarm dibatalkan", Toast.LENGTH_SHORT).show();
}

```

Karena tipe alarm yang dikirimkan adalah **TYPE\_REPEATING**, maka nilai dari **requestCode** adalah **NOTIF\_ID\_REPEATING** yang bernilai 101. Setelah memperoleh **requestCode** tersebut, maka kita dapat mengetahui Intentnya. Dan kemudian kita dapat membatalkan alarmnya.





## **BAB VIII**

### **CONTOH APLIKASI BERBASIS ANDROID**

Dalam bab ini diberikan beberapa contoh aplikasi yang menggunakan sistem Activity Inten dan Service.

#### **A. Aplikasi Kalkulator Resistor SMD**

Bagaimana cara kerja aplikasi resistor SMD yang bertujuan untuk membantu teknisi handphone yang mengalami kendala dalam pembacaan suatu nilai resistor SMD yang ukuran komponennya sangat kecil. Ini dikarenakan resistor SMD memiliki kode-kode unik untuk menentukan nilai suatu komponen yang tidak mudah dihafal bagi para teknisi reparasi handphone dan komputer sehingga berpengaruh pada keefektifan dan kecepatan kerja para teknisi handphone.

##### **1. Menu Utama**

Interface dimana terdapat 4 pilihan button yaitu button kalkulator resistor SMD, but-ton bantuan, button tentang dan button keluar. Dalam menu utama ini menerapkan konsep activity pada setiap buttonnya, dimana artinya jika button tersebut di klik maka akan berpindah ke tampilan lain.



##### **2. Menu Kalkulator SMD**

Dalam bagian perhitungan digunakan konsep dasar intent dan service, dimana setelah mengisi nilai resistor SMD maka selanjutnya tekan proses, maka data yang sudah diinputkan akan dihitung dan serviceakan aktif. Service akan berhenti dengan sendirinya saat hasil perhitungan sudah muncul.



### 3. Menu Tentang dan Menu Bantuan

Pada bagian ini hanya menggunakan konsep dasar activity yaitu terletak pada button back.

Hal tersebut memberikan Activity dimana tombol back diklik maka selanjutnya akan kembali ke menu utama.

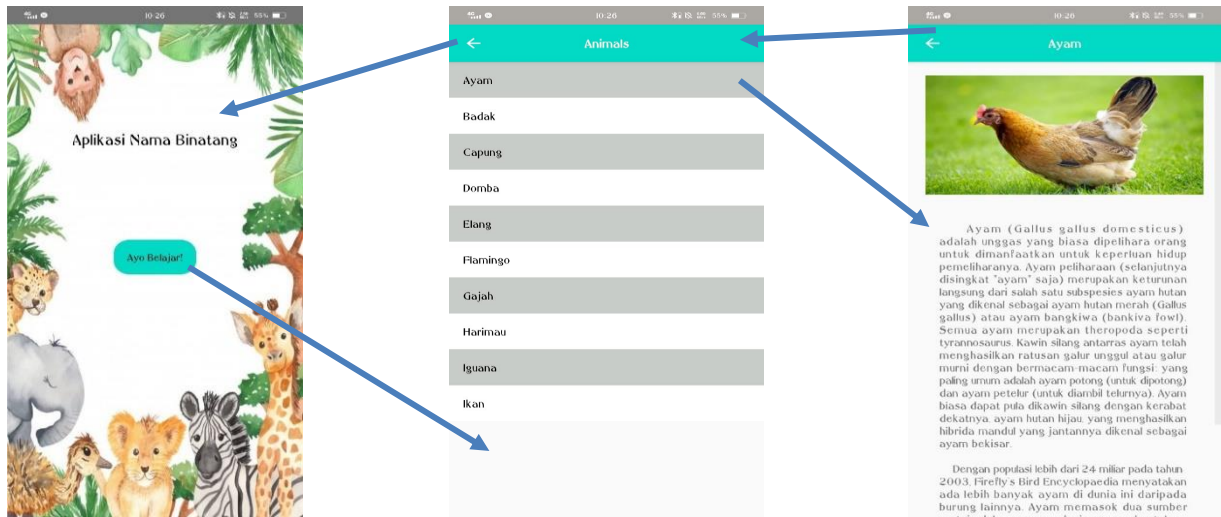


## B. Aplikasi Pengenalan Nama Binatang

Aplikasi Pengenalan Nama Binatang terdiri dari tiga activity. Antarmuka adalah tampilan dari suatu aplikasi yang dapat di jalankan oleh pengguna/user sebagai media komunikasi antara pengguna dan aplikasi. Maka dari itu tampilan antar muka harus dibuat sedemikian rupa untuk dapat lebih mudah dipahami oleh pengguna aplikasi.

Pada antar muka ini user akan disambut dengan sampul pembuka minimalis. Dimana saat aplikasi dibuka, user akan disambut dengan cover bintang-binatang dengan animasi yang unik sehingga semakin menarik anak-anak untuk belajar. Pada antar muka ini user akan diberikan button untuk memulai proses belajar, dimana hal tersebut memanfaatkan proses intent untuk berpindah halaman.

Pada activity daftar binatang pengguna akan di sambut dengan list button atau pilihan nama-nama binatang dari A sampai Z. Sehingga memudahkan user untuk mencari maupun belajar pengenalan nama-nama binatang dan title bar, user akan diberikan tampilan interface berupa tombol untuk kembali ke activity tampilan utama. Selanjutnya setelah user menekan tombol dari nama binatang yang ingin dipelajari, user akan disambut dengan gambar dan deskripsi hewan. Kemudian pada title bar, user akan diberikan tampilan interface berupa tombol untuk kembali ke activity daftar hewan.



## C. Aplikasi Menghitung Waktu Belajar

### 1. MainActivity (Timer)

Halaman utama atau disini adalah MainActivity merupakan activity utama yang akan berisikan Timer, control timer, dan navigasi lainnya (menuju activity lainnya).

### 2. SetTimerActivity (Atur Timer)

Sama seperti namanya, Activity ini digunakan untuk mengatur Timer, baik itu durasi Timer, Break dan Label Timer.

### 3. AboutActivity (Tentang kami)

Activity hanya berisi mengenai about atau informasi tentang kami sebagai pembuat aplikasi

### 4. Alur Aplikasi

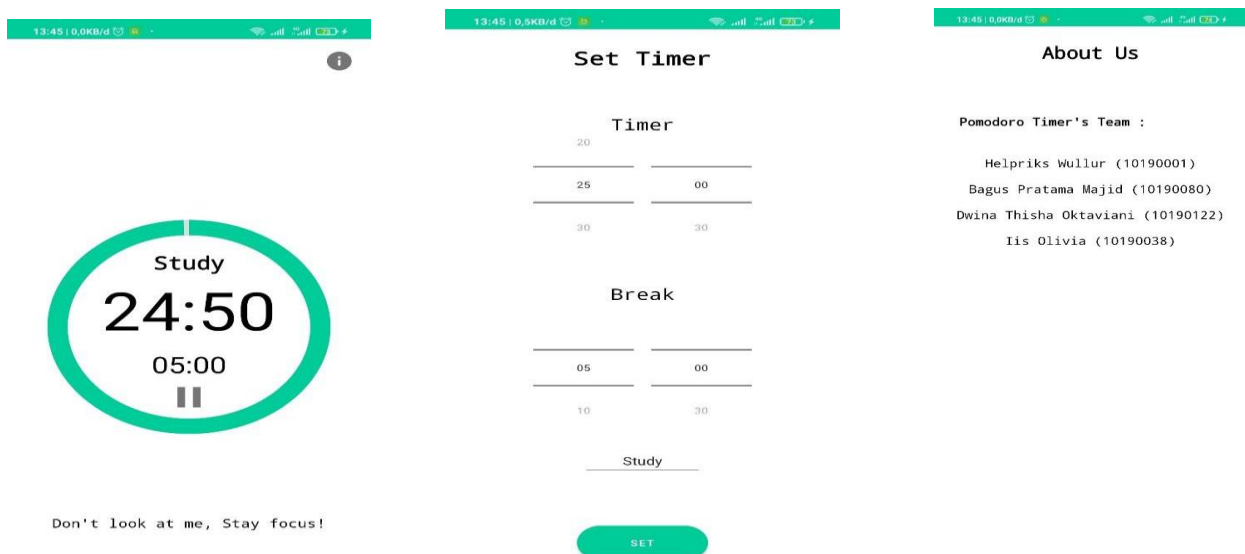
- Ketika membuka aplikasi, langsung diarahkan ke tampilan MainActivity
- Aplikasi akan menampilkan Timer (settingan default), tombol Play dan About
- Jika user memilih :

Play button : Timer akan berjalan dan button akan berubah menjadi pause button

Pause button : Timer akan berhenti dan button akan berubah menjadi play button

About button : Aplikasi akan berpindah ke About Activity, berisi informasi pembuat aplikasi

Timer : Aplikasi akan berpindah ke SetTimerActivity, Activity untuk setel timer



## REFERENSI

### Utama :

Hansun, Seng, Kristanda, dkk. 2018. *Pemrograman Android dengan Android Studio IDE*. Yogyakarta: Andi Publisher.

Imaduddin, Ahmad., Permana, Sidiq. 2018. *Menjadi Android Developer Expert – Dicoding*. Bandung: Android Associate.

<https://developer.android.com>

<http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>

Hendriyani, Yeka., Karmila, Suryani. 2020. *Pemrograman Android : Teori dan Aplikasi*. Pasuruan: Qiara media

Setiawan, Ade. Dkk. 2019. *Aplikasi Perhitungan Resistor SMD Berbasis Android*. Pekanbaru: Jurnal AKRAB JUARA

### Pendukung :

<https://www.dicoding.com/>

<http://egi.degenius.id/mengenal-istilah-jdk-jre-dan-ide-pada-java/>

<http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>

<https://inixindojogja.co.id/akselerasi-android-emulator-dengan-intel-haxm/>