

1. INTRODUCTION

Apa itu Android

Android adalah sistem operasi yang dikeluarkan oleh Google. Android dibuat khusus untuk *smartphone* dan *tablet*. Berbagai macam produsen telah menggunakan Android sebagai sistem operasi untuk peranti (device) yang mereka produksi. Android juga mempunyai *store* dengan lebih dari 2 miliar pengguna aktif, per Januari 2018 ketika tulisan ini dibuat.



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.

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.

Store

Aplikasi Android bisa di distribusikan menggunakan *web*, *copy APK*, dan *store*. *Store* Android, yaitu Google Play, merupakan cara termudah bagi para *developer* untuk medistribusikan 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.

Bagaimana para developer memonetisasi aplikasi yang ada di dalam Google Play?

Strategi monetisasi aplikasi yang ditawarkan Google Play ada bermacam-macam. Dimulai dari app berbayar (*paid distribution*), pembelian dalam aplikasi (*in-app purchase*), langganan (*subscriptions*), dan iklan (*ads*). Tentunya *developer* harus mengikuti aturan yang ada untuk memastikan bahwa pengguna mendapatkan pengalaman (*user experience*) yang paling baik.

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.

Bahasa pemrograman yang sering digunakan untuk mengembangkan aplikasi Android adalah Java. Namun ada beberapa bahasa lainnya yang dapat digunakan, seperti C++, dan Go. Pada IO 2017, Google juga meresmikan penggunaan Kotlin sebagai tambahan bahasa resmi.



Bercerita tentang pemrograman tentunya tidak lepas dari *Integrated Development Environment* (IDE). Pada 2014 Google mengeluarkan IDE yang bernama Android Studio yang berbasiskan IntelliJ IDEA.

Dengan menggunakan Android Studio, para *developer* dapat membuat aplikasi dari nol hingga dipublikasikan ke dalam *store*. Android Studio juga mempunyai beberapa fitur *built-in* yang sangat membantu para *developer* untuk memaksimalkan proses pembuatan aplikasi. Fitur-fitur ini misalnya Gradle, Code Completion, dan beragam integrasi dengan layanan dari Google, seperti Firebase.

Sejarah Perkembangan Android

Berikut adalah rangkaian sejarah perkembangan Android yang resmi diluncurkan oleh Google, dari waktu ke waktu

Version	Code name	Release date	API level	DVM/ART	New features	Icon
9	Android P	-	-	-	-	-
8.1	Oreo	October 25, 2017	26 - 27	ART	<ul style="list-style-type: none"> Picture-in-Picture 	
7.1 - 7.1.2	Nougat	August 22, 2016	24 - 25	ART	<ul style="list-style-type: none"> Multi window GIF Keyboard 	
6.0 - 6.0.1	Marshmallow	October 5, 2015	23	ART	<ul style="list-style-type: none"> Now On Tap Permissions Battery (Doze & App Standby) 	
5.1 - 5.1.1	Lollipop	November 12, 2014	21 - 22	ART	<ul style="list-style-type: none"> Material Design Multiscreen Notifications 	
4.4 - 4.4.4	KitKat	October 31, 2013	19 - 20	DVM (and ART 1.6.0)	<ul style="list-style-type: none"> Voice : Ok Google Immersive Design Smart Dialer 	
4.1 - 4.3.1	Jelly Bean	July 9, 2012	18	DVM	<ul style="list-style-type: none"> Google Now Actionable Notifications Account Switching 	

4.0 - 4.6	Ice Cream Sandwich	October 19, 2011	15	DVM	<ul style="list-style-type: none"> • Custom Home Screen • Data Usage Control • Android Beam 	
3.0 - 3.2.6	HoneyComb	February 22, 2011	11 - 13	DVM	<ul style="list-style-type: none"> • Tablet-Friendly Design • System Bar • Quick Settings 	
2.3 - 2.3.7	Gingerbread	February 9, 2011	9 - 10	DVM	<ul style="list-style-type: none"> • Gaming APIs • NFC • Battery Management 	
2.2 - 2.23	Froyo	May 20, 2010	8	DVM	<ul style="list-style-type: none"> • Voice Action • Portable Hotspot • Dalvik JIT 	
1.6	Donut	September 15, 2009	4	-	<ul style="list-style-type: none"> • Quick Search Box • Screen Size Diversity • Android Market 	
1.5	Cupcake	April 27, 2009	3	-	-	

ART dan DVM

Dari tabel sejarah perkembangan di atas dapat kita lihat ada kolom DVM / ART. Kolom ini menunjukkan eksekusi kompilasi ketika menjalankan aplikasi Android. Pada API KitKat dan sebelumnya Android menggunakan DVM (Dalvik Virtual Machine). DVM menerapkan pendekatan JIT (Just-In-Time), di mana kompilasi dijalankan ketika ada permintaan untuk menjalankan aplikasi.

Sedangkan ART (Android Runtime) menerapkan pendekatan berbeda yaitu AOT (Ahead-Of-Time). AOT melakukan kompilasi pada saat proses instalasi aplikasi.

Dari versi Lollipop hingga Oreo, Android sepenuhnya mengadopsi ART. Kenapa demikian ?

DVM menggunakan JIT yang berarti kompilasi dilakukan setiap kali aplikasi dijalankan. Hal ini sangat mempengaruhi kecepatan respon aplikasi. Setiap kali kita menyentuh ikon aplikasi maka kompilasi akan dilakukan. Tentu proses ini menghabiskan CPU dan berimbang pada relatif lebih borosnya penggunaan baterai.

Beda dengan DVM, ART melakukan proses kompilasi pada saat proses instalasi. Jadi setiap kali aplikasi dijalankan, sudah tidak ada lagi proses kompilasi. Hal ini meningkatkan performa dalam menjalankan aplikasi. Selain itu, karena penggunaan sumber daya CPU bisa dikurangi, pemakaian baterai jadi lebih hemat. Akan tetapi ART membutuhkan *space* (ukuran berkas) yang lebih besar jika dibandingkan dengan DVM.

Jika ingin mendalami proses run-time yang ada di Android, silahkan Anda klik tautan berikut ini:

- <https://source.android.com/devices/tech/dalvik/>

Beberapa bacaan dasar yang dapat menambah wawasan Anda, antara lain:

- <https://www.android.com/history/>
- https://en.wikipedia.org/wiki/Android_version_history

2. HOW TO INSTALL

Java

Salah satu bahasa yang bisa digunakan untuk mengembangkan Android adalah Java. Selain Java, ada beberapa Bahasa lain yang bisa digunakan seperti C/C++, Go. Pada Mei 2017, Google resmi mendukung bahasa pemrograman Kotlin.

Pada akademi ini kita hanya akan fokus menggunakan Java. Dan sebelum memulai menuliskan kode, ada beberapa hal yang perlu disiapkan.

Pertama, instal Java Development Kit. Anda dapat memperolehnya pada tautan berikut:

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

Biasanya muncul pertanyaan? Apakah JRE cukup? Tidak! JRE adalah Java Runtime Environment yang berfungsi sebagai *virtual machine* untuk menjalankan program Java.

Sementara itu, JDK adalah singkatan dari Java SE Development Kit. Ketika kita menggunakan JDK, maka kita dapat menemukan JRE di dalamnya. Yang lebih penting lagi, JDK memiliki *compiler* dan beragam alat untuk mendukung proses kompilasi.

Sederhananya JRE digunakan untuk menjalankan program. Sedangkan JDK digunakan untuk membuatnya.

Mari kita mulai dengan proses instalasi dari JDK.

1. Langsung saja buka tautan di atas menggunakan *browser* Anda, dan pilih Java Platform (JDK). Berikut adalah tampilan halaman unduh dari

JDK.



- Java SE
- Java EE
- Java ME
- Java SE Support
- Java SE Advanced & Suite
- Java Embedded
- Java DB
- Web Tier
- Java Card
- Java TV
- New to Java
- Community
- Java Magazine

[Overview](#)[Downloads](#)[Documentation](#)[Community](#)

Java SE Downloads



Java Platform (JDK) 8u131

Java Platform, Standard Edition

Java SE 8u131

Java SE 8u131 includes important security fixes and bug fixes. We strongly recommend all Java SE 8 users upgrade to this release.

[Learn more →](#)

Important planned change for MD5-signing

Starting with the April Critical Patch Update releases, Oracle Java SE products will treat JARs signed with MD5 as unsigned. [Learn more →](#) For more information on cryptographic algorithm support, see the [Java Cryptographic Algorithm Policy](#) and the [Roadmap](#).

- [Installation Instructions](#)
- [Release Notes](#)
- [Oracle License](#)
- [Java SE Products](#)
- [Third Party Licenses](#)
- [Certified System Configurations](#)
- [Readme Files](#)
 - [JDK ReadMe](#)
 - [JRE ReadMe](#)

2. Lalu pilihlah yang sesuai dengan peranti dan sistem operasi yang Anda pakai.

Technology, and subscribe to Java.

- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u131 checksum

Java SE Development Kit 8u131

You must accept the Oracle Binary Code License Agreement for this software.

Accept License Agreement Decline License

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.87 MB	jdk-8u131-linux-arm32abi
Linux ARM 64 Hard Float ABI	74.81 MB	jdk-8u131-linux-arm64abi
Linux x86	164.66 MB	jdk-8u131-linux-i386
Linux x86	179.39 MB	jdk-8u131-linux-i386
Linux x64	162.11 MB	jdk-8u131-linux-x64
Linux x64	176.95 MB	jdk-8u131-linux-x64
Mac OS X	226.57 MB	jdk-8u131-macosx
Solaris SPARC 64-bit	139.79 MB	jdk-8u131-solaris-sparc
Solaris SPARC 64-bit	99.13 MB	jdk-8u131-solaris-sparc
Solaris x64	140.51 MB	jdk-8u131-solaris-x64
Solaris x64	96.96 MB	jdk-8u131-solaris-x64
Windows x86	191.22 MB	jdk-8u131-windows-i386
Windows x64	198.03 MB	jdk-8u131-windows-x64

Java SE Development Kit 8u131 Demos Downloads

You must accept the Oracle BSD License. to download

Accept License Agreement Decline License

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	9.94 MB	jdk-8u131-linux-arm32abi
Linux ARM 64 Hard Float ABI	9.97 MB	jdk-8u131-linux-arm64abi
Linux x86	52.66 MB	jdk-8u131-linux-i386
Linux x86	52.51 MB	jdk-8u131-linux-i386
Linux x64	52.72 MB	jdk-8u131-linux-x64
Linux x64	52.58 MB	jdk-8u131-linux-x64
Mac OS X	53.09 MB	jdk-8u131-macosx-x86
Solaris SPARC 64-bit	13.54 MB	jdk-8u131-solaris-sparc
Solaris SPARC 64-bit	9.33 MB	jdk-8u131-solaris-sparc
Solaris x64	13.59 MB	jdk-8u131-solaris-x64
Solaris x64	9.29 MB	jdk-8u131-solaris-x64
Windows x86	53.8 MB	jdk-8u131-windows-i386
Windows x64	53.82 MB	jdk-8u131-windows-x64

3. Setelah proses mengunduh selesai, langsung instal ke peranti Anda dan ikuti petunjuknya sampai selesai.

Setelah instalasi JDK selesai, selanjutnya adalah IDE Android Studio.

Android Studio

Pada akademi ini kita akan menggunakan Android Studio versi 3.1 sebagai IDE (*Integrated Development Environment*). Android Studio dirilis 16 Mei 2013 pada Google IO. Android Studio berbasiskan JetBrains IntelliJ IDEA, dan dikhususkan untuk mengembangkan aplikasi berplatform Android.

Semua versi dari Android Studio dapat Anda unduh pada tautan ini:

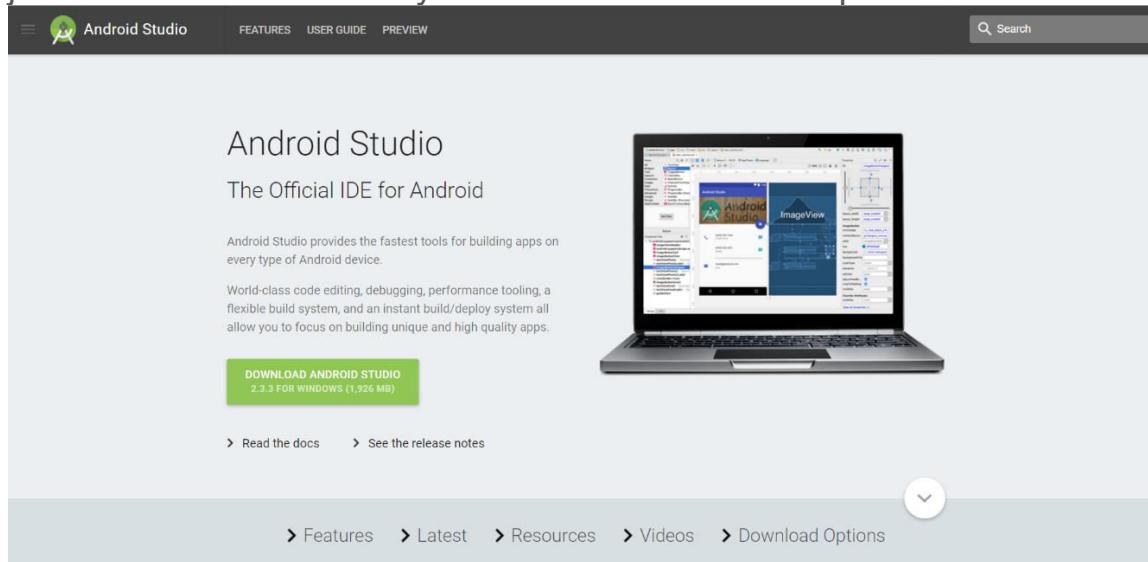
- <https://developer.android.com/studio/archive.html>

Atau versi terbaru bisa Anda unduh pada tautan ini:

- <https://developer.android.com/studio/index.html#downloads>

Mari langsung saja kita mulai instalasi dari Android Studio.

1. Tampilan dari laman unduh Android Studio. Silakan unduh, kemudian jalankan berkasnya di komputer Anda.



2. Ikuti petunjuknya sampai selesai.

Catatan : Untuk migrasi ke Android Studio 3.0 terkadang perlu menyesuaikan beberapa kode terutama pada berkas *gradle*-nya. Silakan kunjungi tautan ini untuk keterangan lebih lanjut <https://developer.android.com/studio/build/gradle-plugin-3-0-0-migration.html>

Persiapan Running Menggunakan Emulator

Sebelum menggunakan emulator, Anda perlu memastikan beberapa hal berikut ini:

Virtualization

Untuk menjalankan emulator di dalam Android Studio, pastikan aspek virtualization. Sistem Anda harus memenuhi persyaratannya, yakni ketentuan prosesor dan sistem operasi dari laptop / PC yang Anda gunakan.

Processor

- Prosesor Intel: Jika laptop/pc Anda menggunakan prosesor Intel, maka pastikan ia mendukung Intel VT-x, Intel EM64T (Intel 64), dan Execute Disable (XD) Bit functionality.
- Prosesor AMD: Jika laptop/pc Anda menggunakan AMD, maka pastikan bahwa ia support dengan AMD Virtualization (AMD-V) dan Supplemental Streaming SIMD Extensions 3 (SSSE3).

Sistem Operasi

- Intel : Jika menggunakan processor Intel maka Anda dapat menjalankannya di sistem operasi Windows, Linux, maupun Mac
- AMD : Untuk prosesor AMD maka hanya bisa menjalankannya di sistem operasi Linux.

Menginstal Hardware Accelerated Execution Manager (HAXM)

Setelah memenuhi persyaratan di atas, langkah selanjutnya adalah menginstal HAXM. HAXM adalah *hardware-assisted virtualization engine* yang menggunakan teknologi VT dari Intel untuk mempercepat

aplikasi Android yang diemulasi di mesin host. HAXM diperlukan untuk menjalankan emulator di Android Studio.

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

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

Menginstal Kernel-based Virtual Machine (KVM) untuk Pengguna Linux

Karena HAXM hanya untuk Windows dan Mac, bagaimana dengan sistem operasi Linux? Untuk Linux, Anda harus menginstal KVM. Sistem operasi Linux dapat support accelerated virtual machine dengan menggunakan KVM. Untuk instal KVM, Anda bisa menggunakan perintah berikut ini.

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

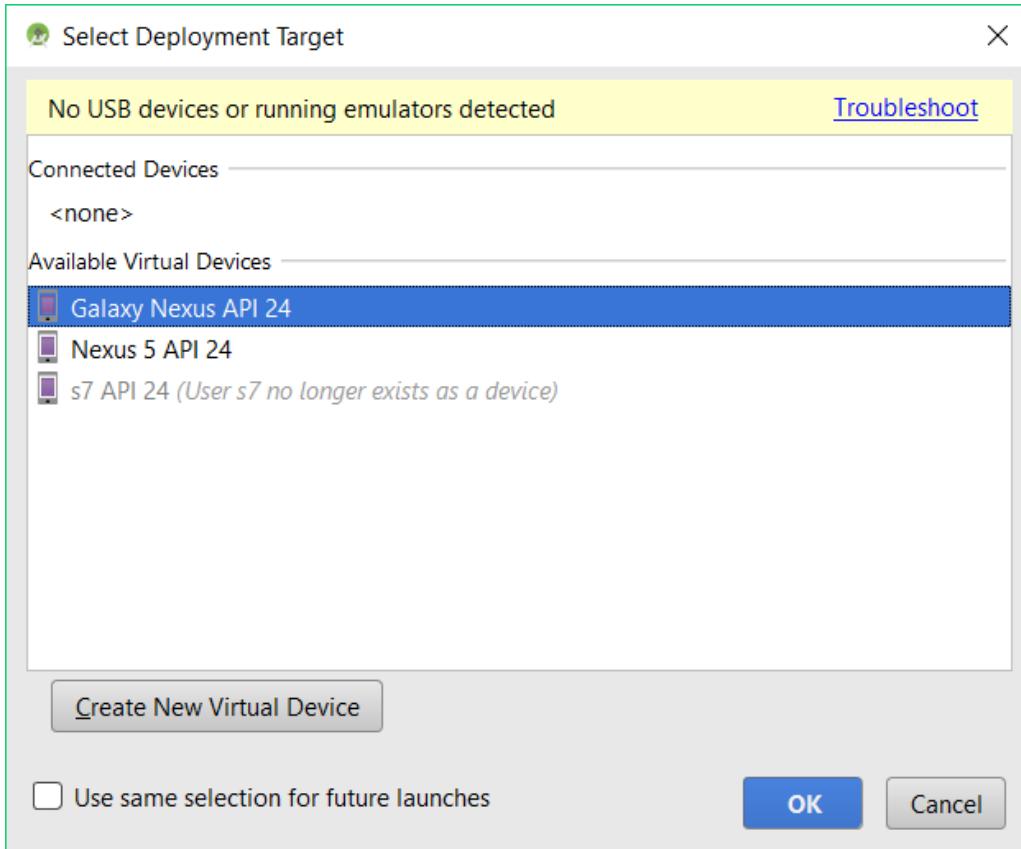
Selengkapnya dapat Anda baca pada halaman ini <https://developer.android.com/studio/run/emulator.html>

<https://developer.android.com/studio/run/emulator-acceleration.html>

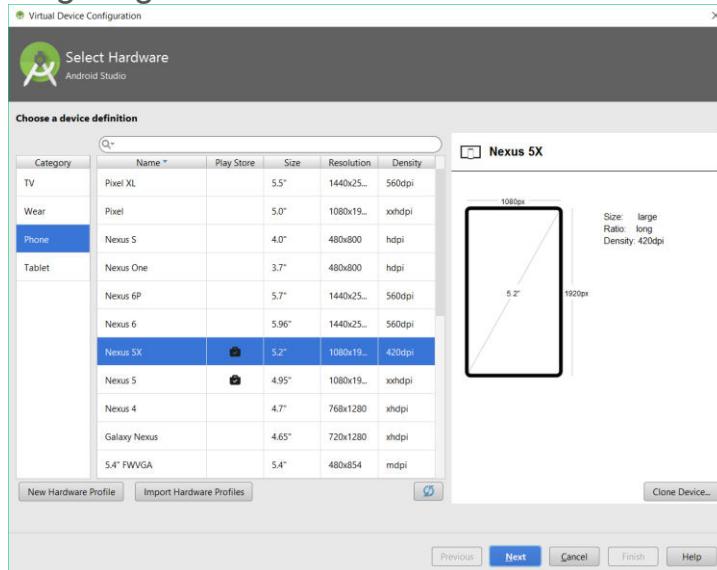
Menggunakan Emulator

Setelah Anda memastikan bahwa virtualization bisa berjalan di komputer Anda, ikuti langkah-langkah berikut untuk menjalankan aplikasi kita dengan menggunakan emulator *built-in* dari Android Studio.

1. Jalankan ikon , kemudian akan muncul dialog seperti ini. Mari kita coba buat emulator baru dengan memilih **Create New Virtual Device**.

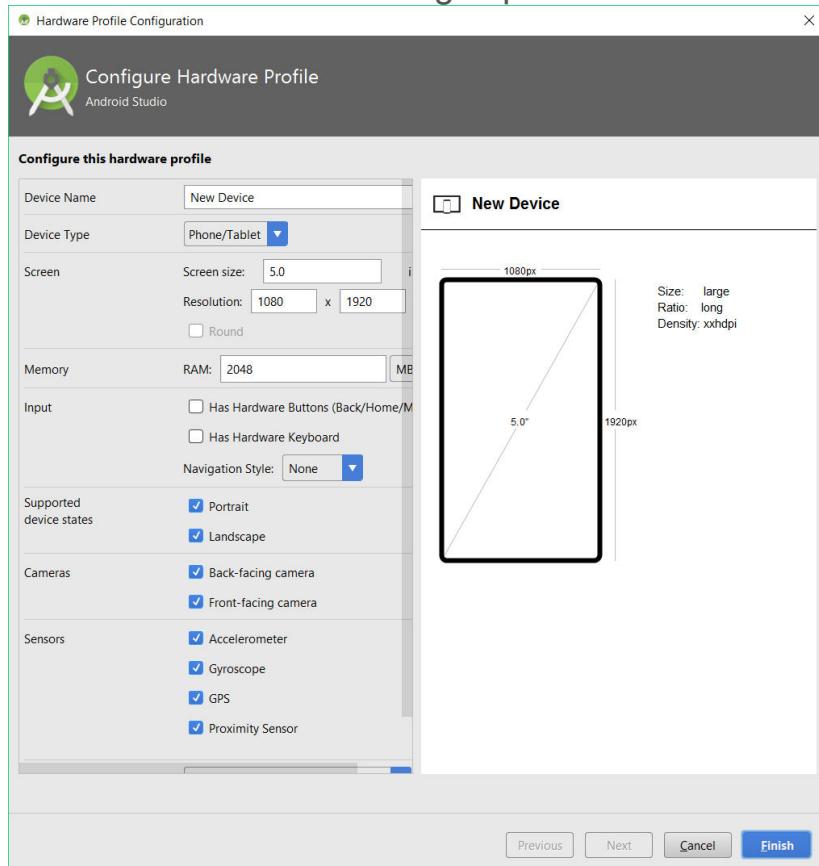


2. Akan muncul dialog dengan pilihan beberapa emulator yang bisa langsung Anda



3. Jika anda ingin membuat spesifikasi *hardware* (perangkat keras) sendiri, Anda bisa memilihnya pada pilihan **New Hardware**

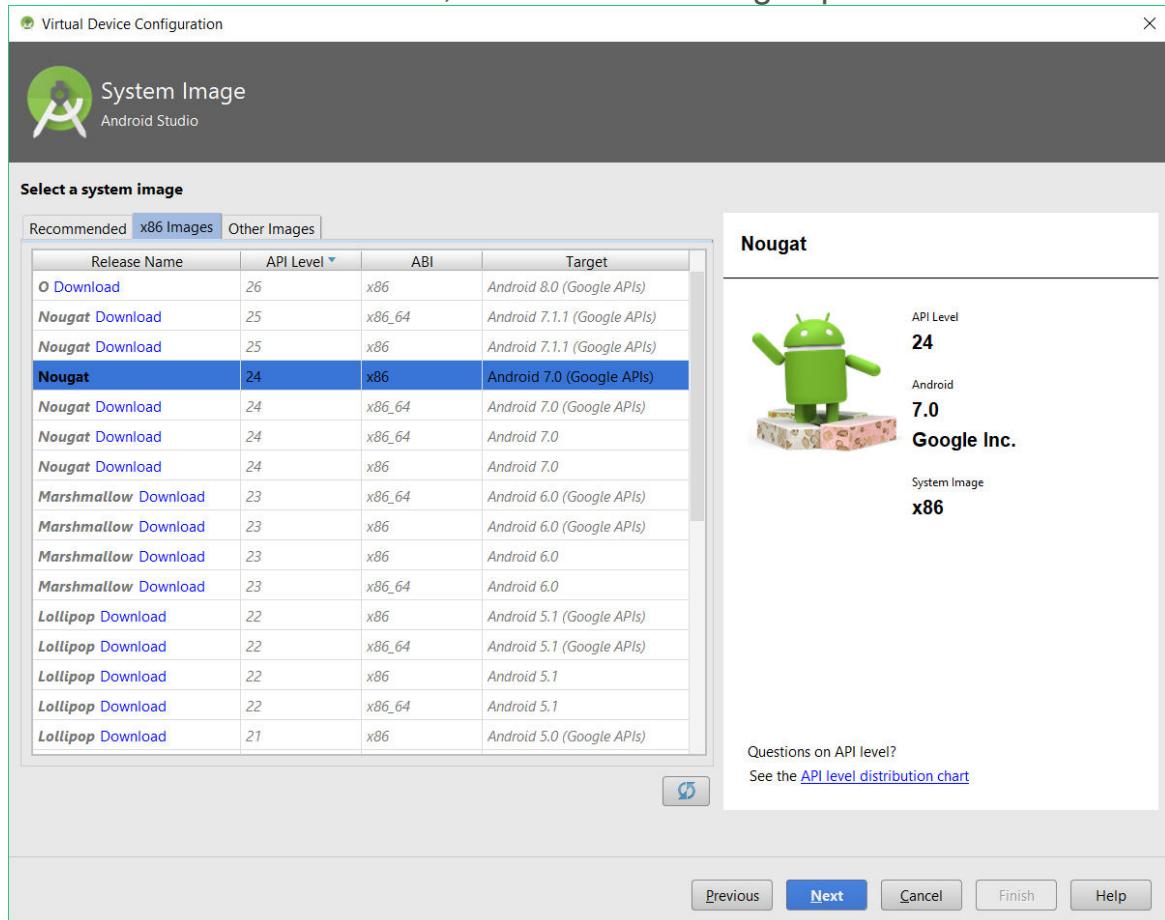
Profile. Akan muncul dialog seperti di bawah ini



Anda dapat menentukan konfigurasi *hardware* sesuai dengan kebutuhan Anda. Yang perlu diingat adalah untuk menggunakan konfigurasi emulator yang sesuai dengan kemampuan laptop atau komputer yang Anda gunakan.

4. Anda dapat membuat *hardware* emulator baru atau memilih *hardware* emulator yang sudah ada. Setelah

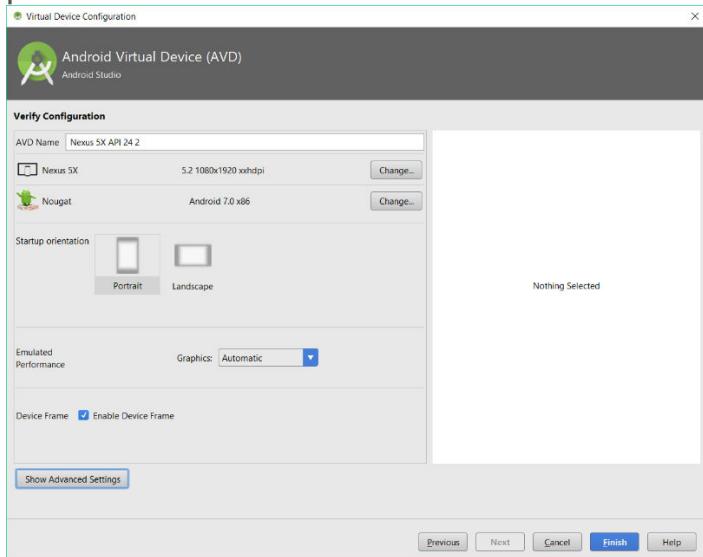
memilih *hardware emulator*, akan muncul dialog seperti ini.



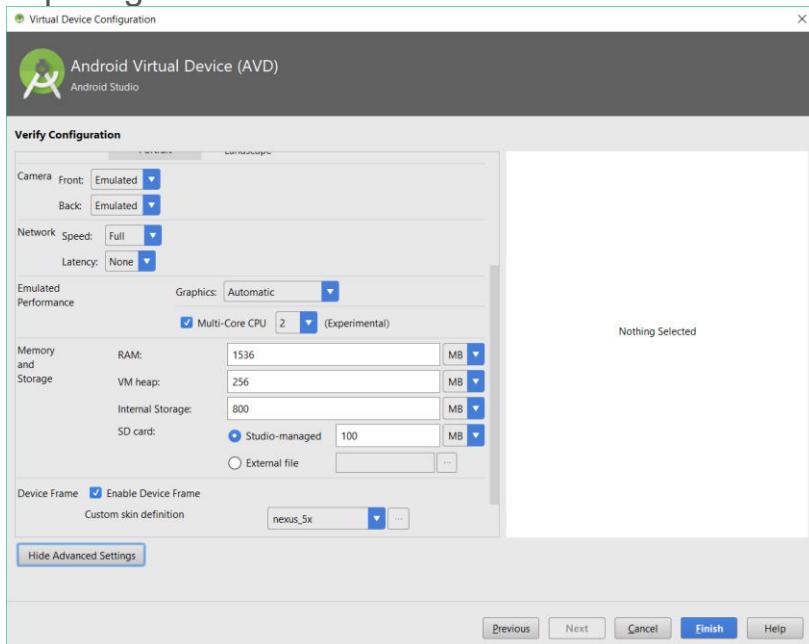
Pada dialog ini Anda akan memilih versi android dari emulator yang akan anda buat. Pada dialog tersebut, Anda perlu memilih versi yang sudah diunduh yaitu Nougat. Tombol **download** di sebelah kanan versi menunjukkan bahwa anda perlu mengunduhnya terlebih dahulu jika ingin menggunakannya.

5. Selanjutnya klik **Next**. Akan muncul dialog *verify configuration*. Pada dialog ini, Anda bisa memeriksa konfigurasi dari emulator yang Anda

pilih.

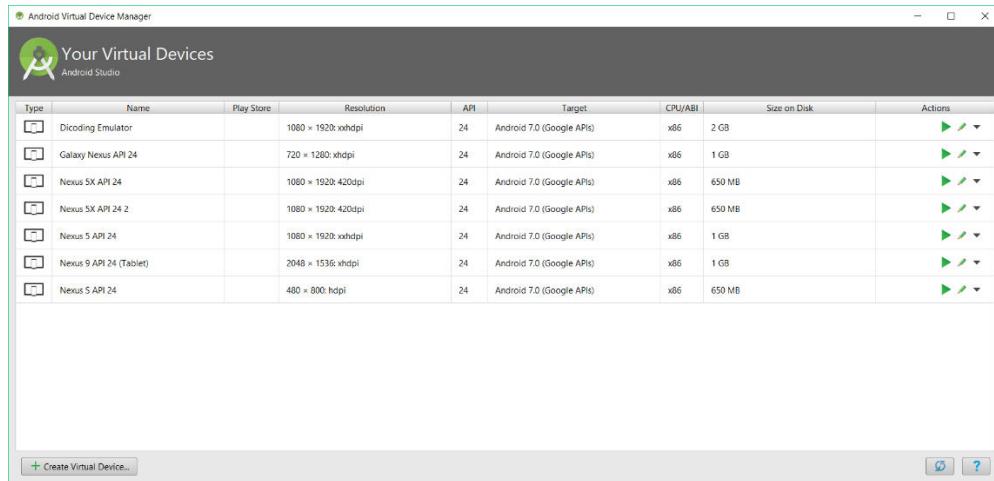


6. Pada bagian kiri bawah, terdapat tombol **Show Advanced Settings**. Bila Anda menekan tombol ini, maka akan muncul tampilan dialog baru seperti gambar di bawah ini.



7. Pada bagian advanced setting, Anda bisa mengubah konfigurasi *hardware* yang telah ditentukan sebelumnya.
8. Jika sudah selesai, Anda dapat menekan tombol **Finish**. Anda dapat membuka emulatorenya dengan menekan tombol hijau yang ada di

sebelah kanan.



9. Pengaturan emulator sudah selesai dan bisa langsung dijalankan.

Run dengan device

Bila Anda hendak melakukan *run* atau *debugging*, lebih baik Anda menjalankannya pada peranti *smartphone* asli. *Running* dengan menggunakan peranti memiliki beberapa kelebihan jika dibandingkan dengan emulator yaitu :

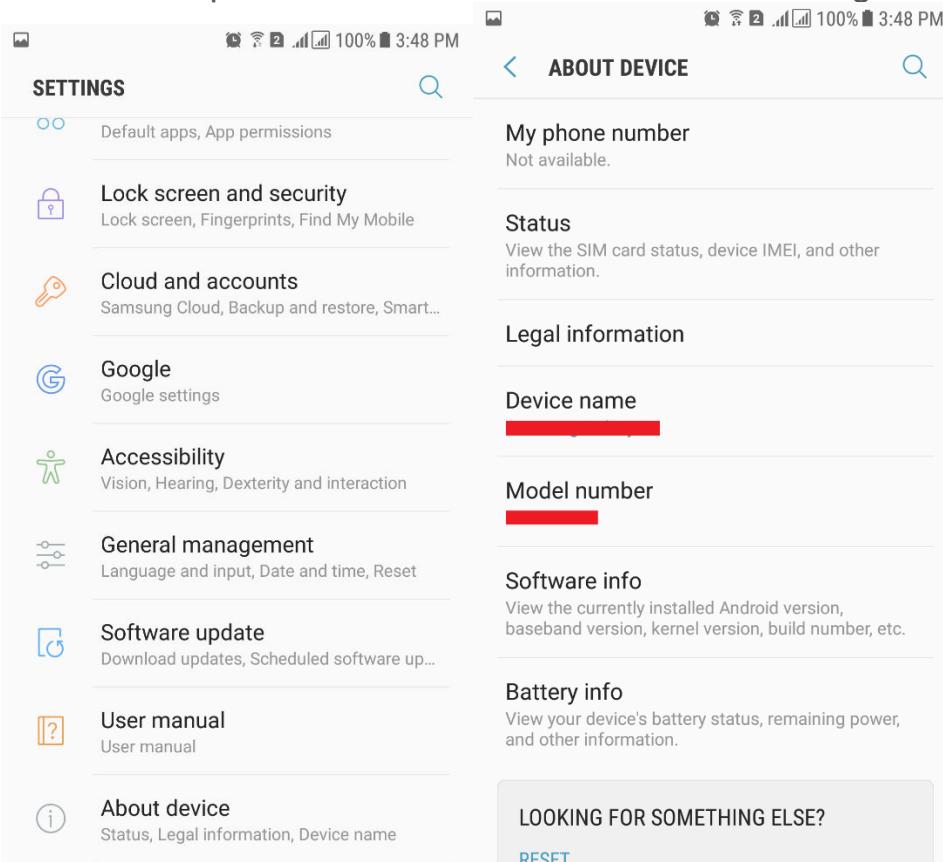
- Lebih cepat;
- Fitur seperti geo-location, push notif bisa digunakan;
- Bisa mengetahui daya serap baterai terhadap aplikasi.
- Lebih mudah

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.

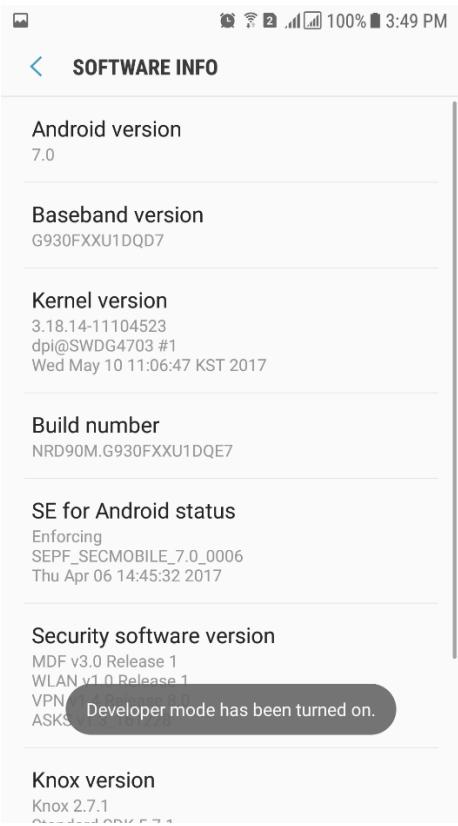
2. Buka **setting** dan masuk ke dalam menu **About**. Pada halaman menu ini, Anda perlu menemukan informasi tentang **Build number**.



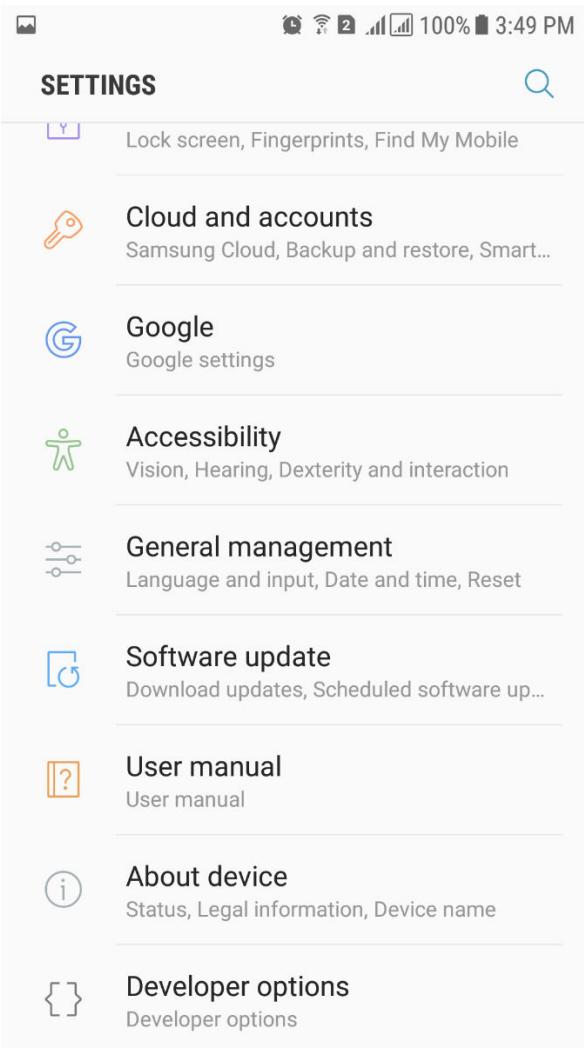


3. Kemudian tekan **Build number** sebanyak 7 kali.

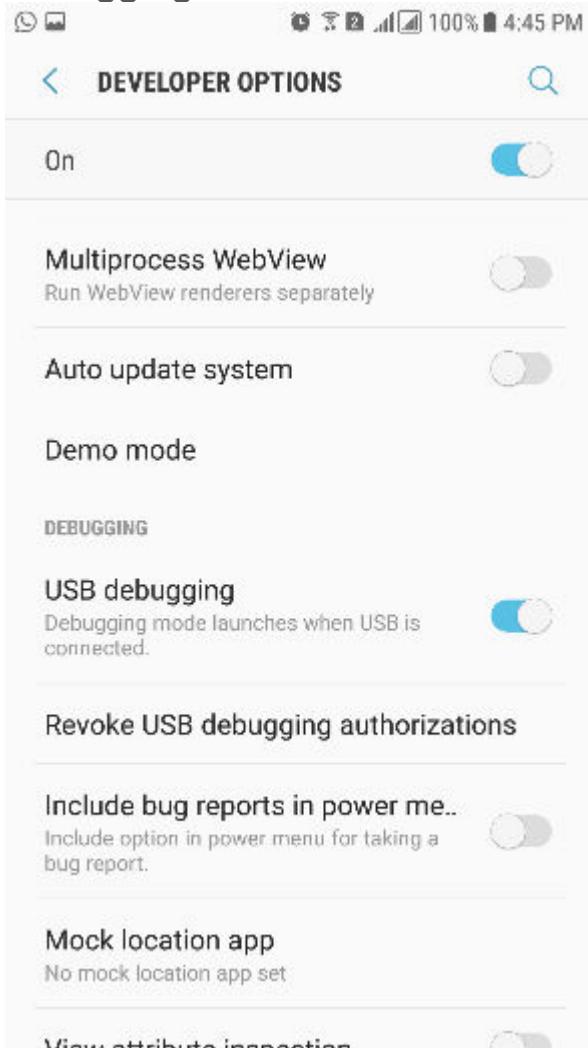
SOFTWARE INFO	
Android version	7.0
Baseband version	G930FXXU1DQD7
Kernel version	3.18.14-11104523 dpl@SWDG4703 #1 Wed May 10 11:06:47 KST 2017
Build number	NRD90M.G930FXXU1DQE7
SE for Android status	Enforcing SEPF_SECMOBILE_7.0_0006 Thu Apr 06 14:45:32 2017
Security software version	MDE v3.0 Release 1 W... V... A... You are now 4 steps away from being a developer.
Knox version	Knox 2.7.1 Standard OSK 2.7.1
SOFTWARE INFO	
Android version	7.0
Baseband version	G930FXXU1DQD7
Kernel version	3.18.14-11104523 dpl@SWDG4703 #1 Wed May 10 11:06:47 KST 2017
Build number	NRD90M.G930FXXU1DQE7
SE for Android status	Enforcing SEPF_SECMOBILE_7.0_0006 Thu Apr 06 14:45:32 2017
Security software version	MDE v3.0 Release 1 W... V... A... You are now 2 steps away from being a developer.
Knox version	Knox 2.7.1 Standard OSK 2.7.1



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



5. Masuk ke dalam menu **Developer Options** dan pastikan opsi **USB Debugging Mode** sudah nyala.



6. Setelah menyelesaikan pengaturan pada peranti, maka peranti dapat dihubungkan dengan laptop atau komputer yang Anda pakai.

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 Anda.

3. TEORI ACTIVITY

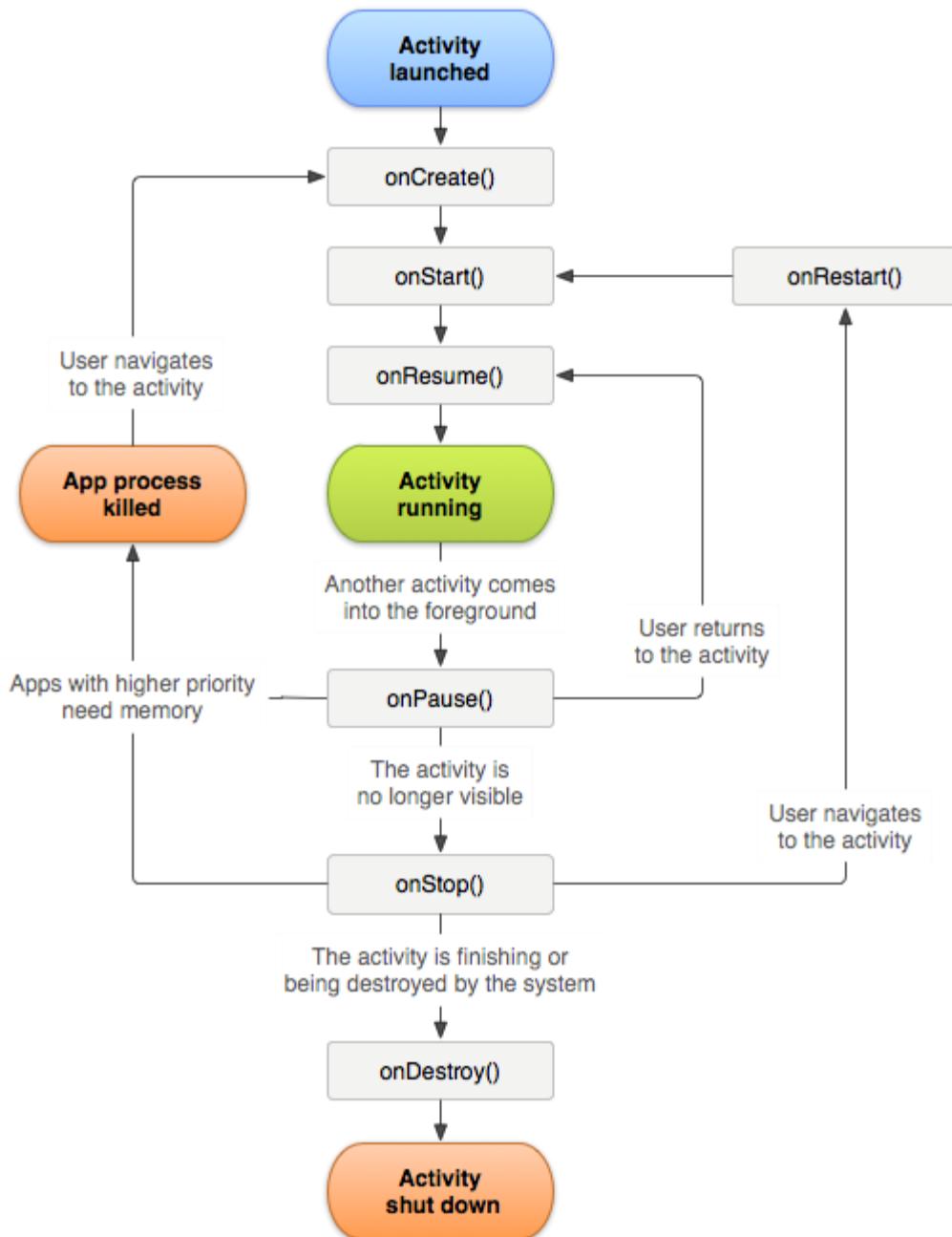
Teori

1. Activity merupakan sebuah komponen di Android yang berfungsi untuk menampilkan user *interface* ke layar handset Android pengguna. Ini seperti pada saat Anda melihat daftar percakapan pada aplikasi *chat* atau daftar *email* pada aplikasi Gmail di ponsel Android Anda.
2. Umumnya dalam sebuah aplikasi terdapat lebih dari satu activity yang saling terhubung dengan tugas yang berbeda-beda.
3. 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*.
4. Pada implementasinya, activity selalu memiliki satu layout *user interface* dalam bentuk berkas xml.
5. Suatu aplikasi Android bisa memiliki lebih dari satu activity dan harus terdaftar di berkas **AndroidManifest.xml** sebagai sub aplikasi.
6. Sebuah class Java dinyatakan sebuah activity jika mewarisi (*extends*) superclass **Activity** atau turunannya seperti **AppCompatActivity** dan **FragmentActivity**.

Untuk lebih mendalami activity, kami menyarankan Anda untuk membaca referensi berikut :

<https://developer.android.com/guide/components/activities/>

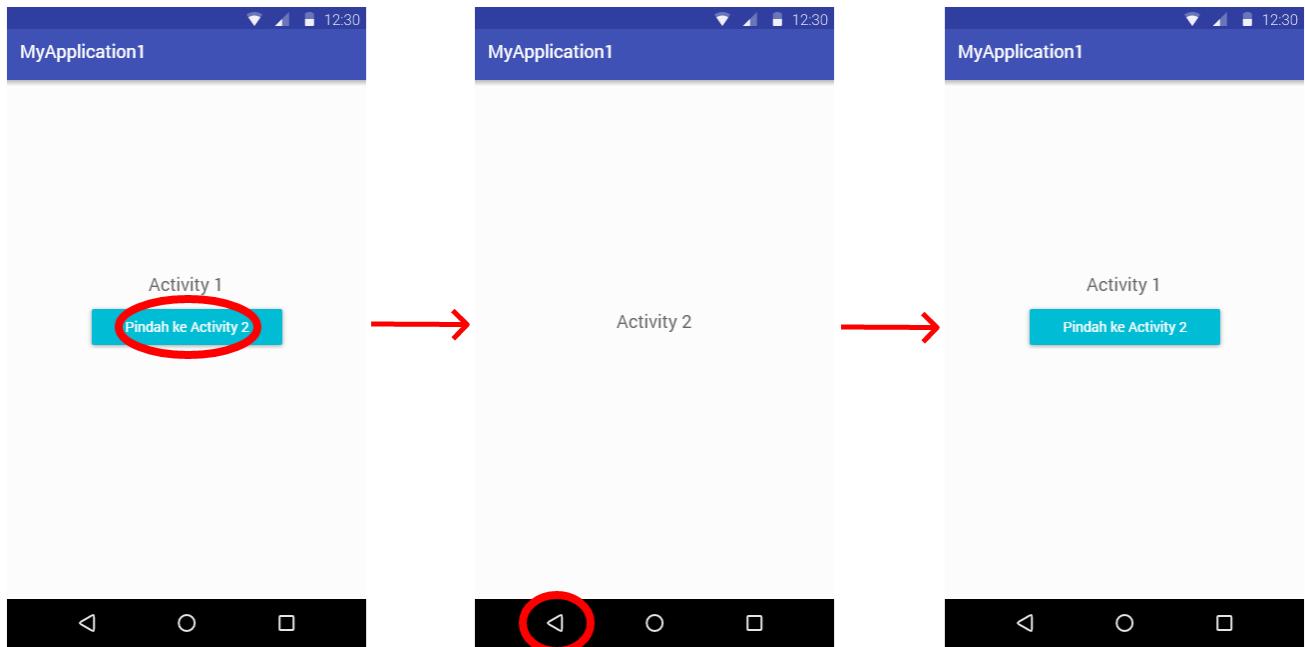
Activity Lifecycle



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

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`.

Detailnya dapat Anda baca disini :

<https://developer.android.com/guide/components/activities/#Lifecycle>

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.

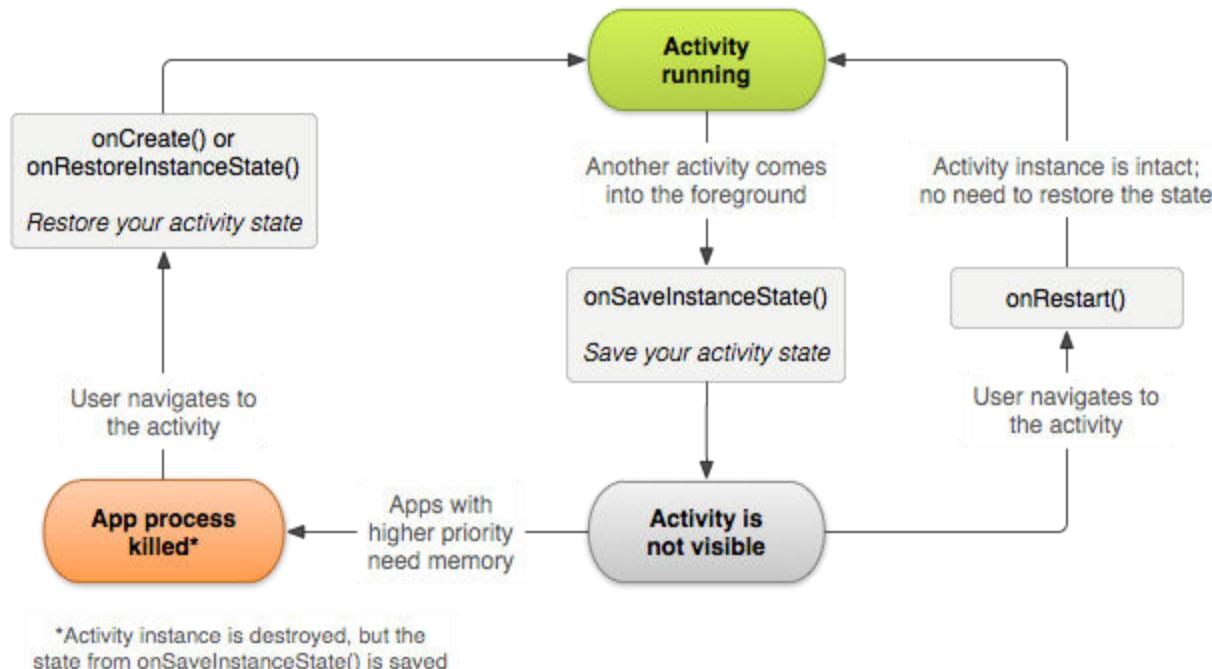
Dengan menjaga *state* dari activity, maka ketika activity tersebut ditampilkan, kondisinya akan tetap sama dengan kondisi sebelumnya.

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.

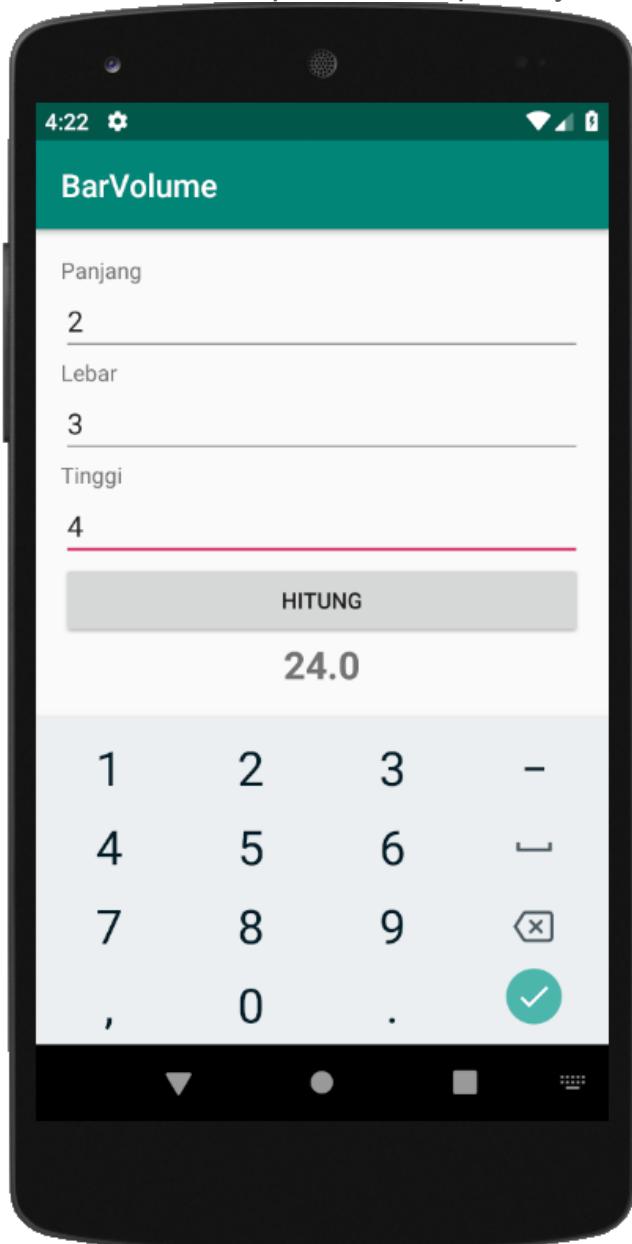
Hal yang perlu diingat ketika menggunakan `onSaveInstanceState` adalah untuk tidak menyimpan data yang besar pada bundle. Contohnya, hindari penyimpanan data *bitmap* pada bundle. Bila data pada bundle berukuran besar, proses serialisasi dan deserialisasi akan memakan banyak memori.

Activity - Latihan : Aplikasi Sederhana

Tujuan

Codelab ini bertujuan untuk mengimplementasikan komponen activity pada aplikasi pertama yang Anda bangun. Harapannya aktifitas ini dapat memberi gambaran yang jelas tentang cara kerja activity.

Codelab pertama adalah dengan membuat aplikasi yang dapat menghitung volume balok. Seperti ini tampilannya.

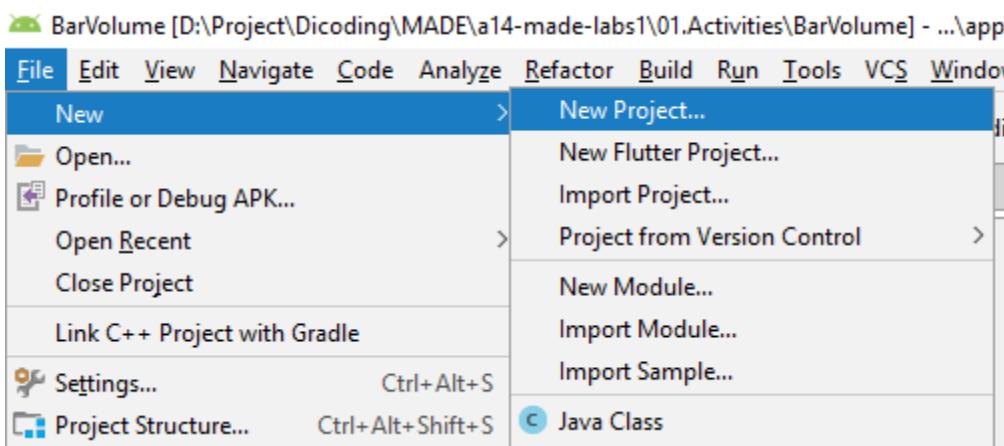


Logika Dasar

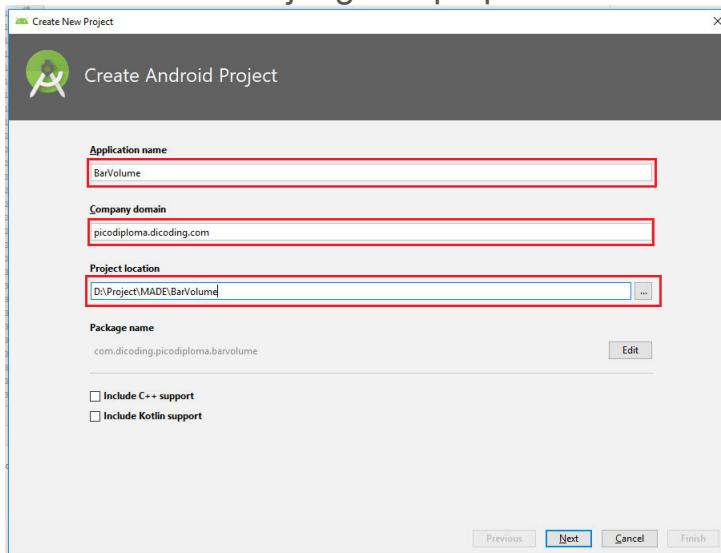
Melakukan input ke dalam obyek **EditText** → melakukan validasi input → melakukan perhitungan volume balok ketika tombol hitung diklik

Codelab Perhitungan Volume

1. Buat proyek baru dengan klik **File** → **New** → **New Project** pada Android Studio Anda.



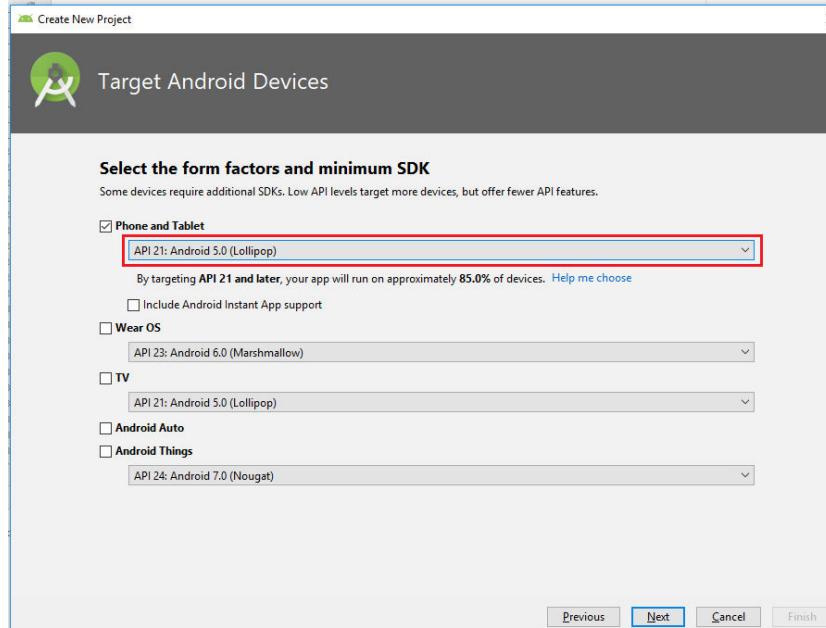
2. Setelah muncul jendela **Create New Project**, atur nama aplikasi dan domain perusahaan/website Anda. Sebaiknya jangan sama dengan apa yang ada di contoh. Dan jangan lupa pula untuk menentukan lokasi



proyek.

3. Kemudian pilih tipe peranti (*device*) untuk aplikasi beserta target minimum SDK yang akan digunakan. Pilihan target Android SDK akan

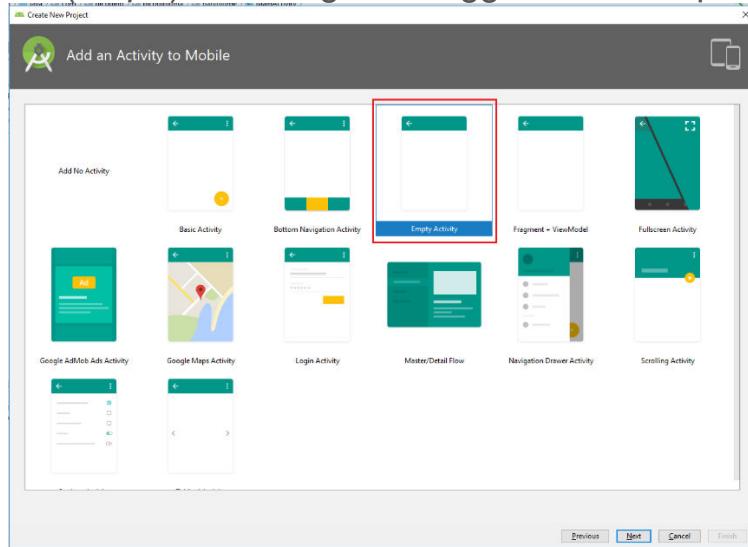
mempengaruhi banyaknya peranti yang dapat menggunakan aplikasi. Di sini kita memilih tipe peranti **Phone and Tablet** dan nilai minimum SDK kita pasang ke **Level 21 (Lollipop)**. Klik **Next** untuk melanjutkan.



4. Pada bagian ini kita akan memilih tipe activity awal dari *template* yang telah disediakan. Saat ini Android Studio sudah menyediakan berbagai macam *template* activity dari yang paling sederhana hingga yang paling kompleks seperti :
 - Add No Activity** : Tidak ada activity yang ditambahkan
 - Basic Activity** : Activity dengan template komponen material design seperti FloatingActionButton
 - Bottom Navigation Activity** : Activity dengan tampilan side bar menu di bagian bawah
 - Empty Activity** : Activity dalam bentuk yang paling dasar
 - Fragment + ViewModel** : Activity dengan menerapkan *architecture component*
 - Fullscreen Activity** : Activity *fullscreen* tanpa status bar
 - Google AdMob Ads Activity** : Activity dengan konfigurasi *default* iklan Admob
 - Google Maps Activity** : Activity dengan menyediakan konfigurasi dasar Google Maps
 - Login Activity** : Activity untuk halaman login
 - Master / Detail Flow** : Activity yang diperuntukan untuk alur aplikasi *master detail* pada peranti tablet
 - Navigation Drawer Activity** : Activity dengan tampilan side bar menu
 - Scrolling Activity** : Activity dengan kemampuan scroll

konten didalamnya secara vertikal

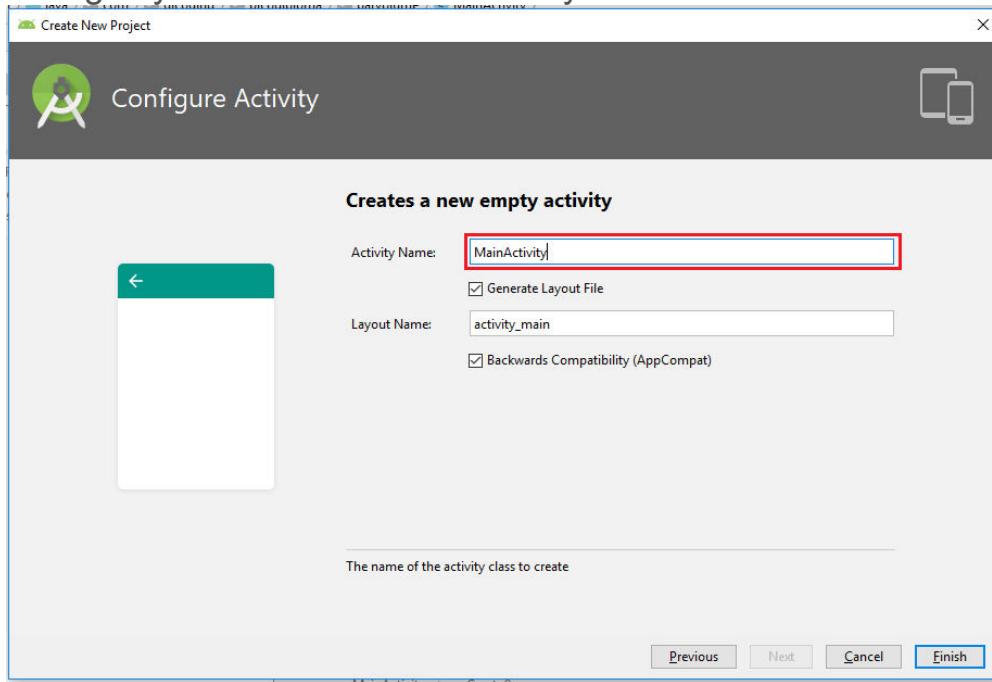
- Settings Activity** : Activity yang diperuntukan untuk konfigurasi aplikasi
- Tabbed Activity** : Activity yang diperuntukan untuk menampilkan lebih dari satu tampilan, dapat digeser ke kanan dan ke kiri (*swipe*) dan dengan menggunakan komponen viewpager



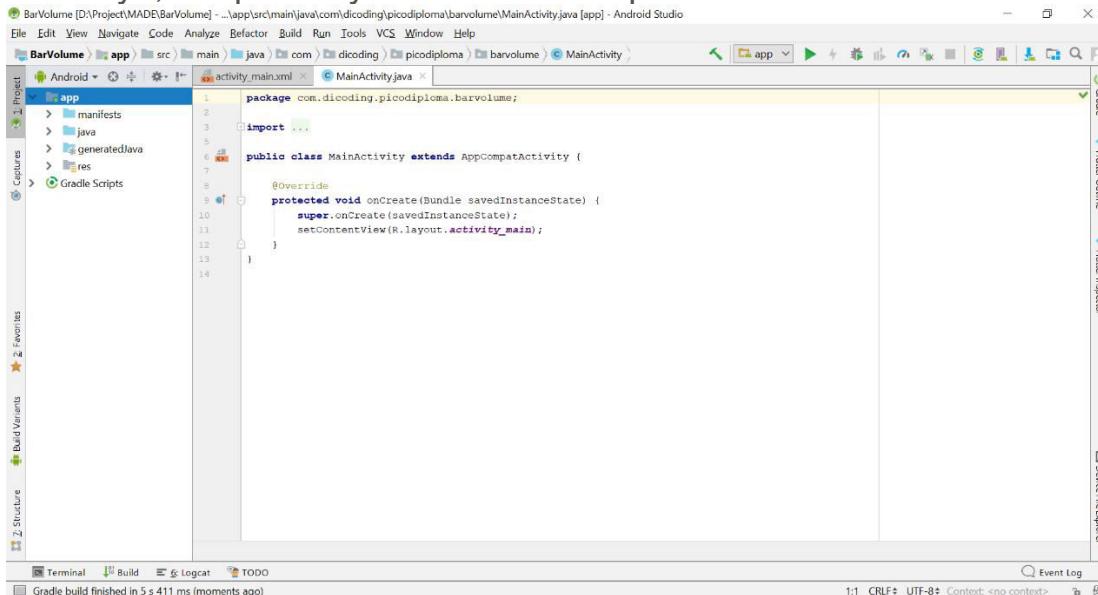
Saat ini kita pilih tipe **Empty Activity**, klik **Next** untuk melanjutkan.

5. Selanjutnya, tentukan nama **Activity** pertama kita, saat ini kita biarkan pada kondisi apa adanya. Ingat, jika suatu saat nanti kita ingin melakukan penambahan **Activity** , maka sebaiknya nama untuk **Activity** tersebut disamakan dengan nama kelasnya dan diakhiri dengan activity . Misal: **ProfileActivity**, **SettingsActivity** dan lain

sebagainya. Klik **Finish** untuk menyelesaikan.



6. Biasanya, tampilan layar Anda akan seperti contoh di bawah ini:



7. Di sebelah kanan anda adalah *workspace* dimana Activity anda berada dan bernama **MainActivity.java** dengan layoutnya **activity_main.xml**. Di sebelah kiri Anda terdapat struktur proyek, di mana nanti kita akan banyak menambahkan berbagai komponen baru, *asset* dan *library*. Untuk lebih mengenal Android Studio lebih dalam silakan baca materi ini <https://developer.android.com/studio/intro/index.html>

Selanjutnya kita akan mulai melakukan pengkodean aplikasi atau lebih enaknya disebut ngoding.

Berikut flow umumnya.

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

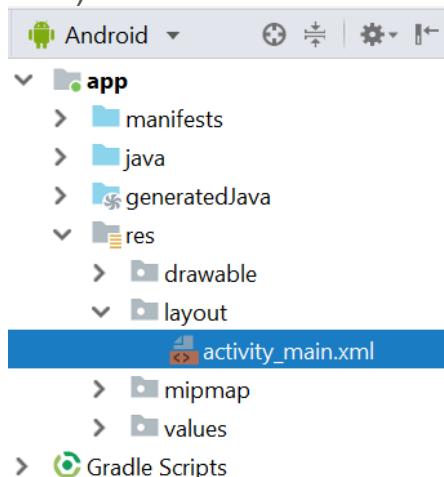
Untuk mengoptimalkan proses pengetikan, Anda dapat memanfaatkan *code completion* dengan menekan **ctrl + space**. Android Studio juga akan meng-import package dari komponen yang digunakan.

Dilarang Keras untuk copy - paste! Ngoding pelan-pelan akan membuat Anda lebih jago di masa depan.

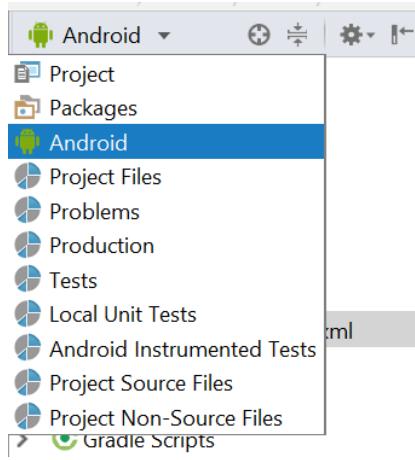
Selamat ngoding!

Menambahkan Code Sederhana pada Layout Activity

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

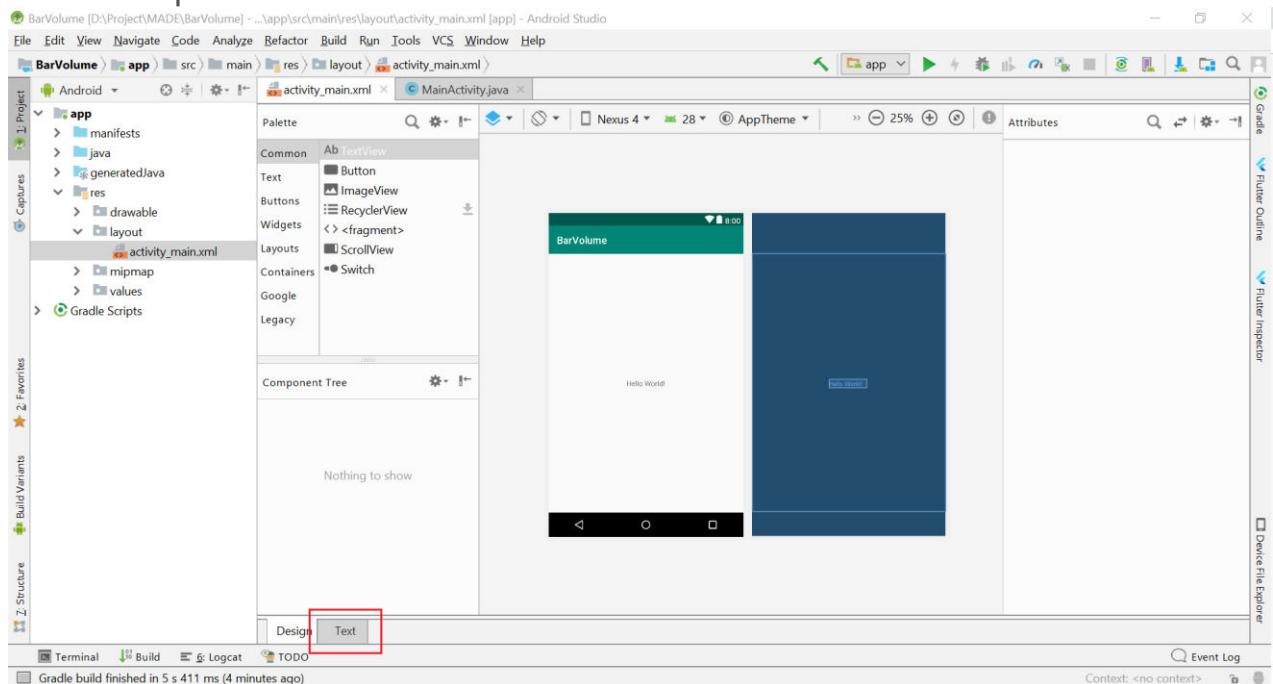


Pastikan **project window** pada pilihan **Android**, seperti di bawah ini.



Maka akan ada tampilan seperti ini,

kemudian pilih metode text.



Dan tambahkan baris-baris berikut:

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:tools="http://schemas.android.com/tools"
4.     android:layout_width="match_parent"
5.     android:layout_height="match_parent"
6.     android:paddingBottom="@dimen/activity_vertical_margin"
7.     android:paddingLeft="@dimen/activity_horizontal_margin"
8.     android:paddingRight="@dimen/activity_horizontal_margin"
9.     android:paddingTop="@dimen/activity_vertical_margin"
10.    android:orientation="vertical"
11.    tools:context="com.dicoding.picodiploma.barvolume.MainActivity">
12.
13.    <TextView
14.        android:layout_width="match_parent"
15.        android:layout_height="wrap_content"

```

```
16.        android:text="Panjang" />
17.
18.    <EditText
19.        android:id="@+id/edt_length"
20.        android:layout_width="match_parent"
21.        android:layout_height="wrap_content"
22.        android:inputType="numberDecimal"
23.        android:lines="1" />
24.
25.    <TextView
26.        android:layout_width="match_parent"
27.        android:layout_height="wrap_content"
28.        android:text="Lebar" />
29.
30.    <EditText
31.        android:id="@+id/edt_width"
32.        android:layout_width="match_parent"
33.        android:layout_height="wrap_content"
34.        android:inputType="numberDecimal"
35.        android:lines="1" />
36.
37.    <TextView
38.        android:layout_width="match_parent"
39.        android:layout_height="wrap_content"
40.        android:text="Tinggi" />
41.
42.    <EditText
43.        android:id="@+id/edt_height"
44.        android:layout_width="match_parent"
45.        android:layout_height="wrap_content"
46.        android:inputType="numberDecimal"
47.        android:lines="1" />
48.
49.    <Button
50.        android:id="@+id/btn_calculate"
51.        android:layout_width="match_parent"
52.        android:layout_height="wrap_content"
53.        android:text="Hitung" />
54.
55.    <TextView
56.        android:id="@+id/tv_result"
57.        android:layout_width="match_parent"
58.        android:layout_height="wrap_content"
59.        android:gravity="center"
60.        android:text="Hasil"
61.        android:textSize="24sp"
62.        android:textStyle="bold" />
63.</LinearLayout>
```

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

Untuk layout akan dibahas nanti pada modul yang berbeda.

2. Akan muncul *warning* pada atribut android:text pada layout tersebut.

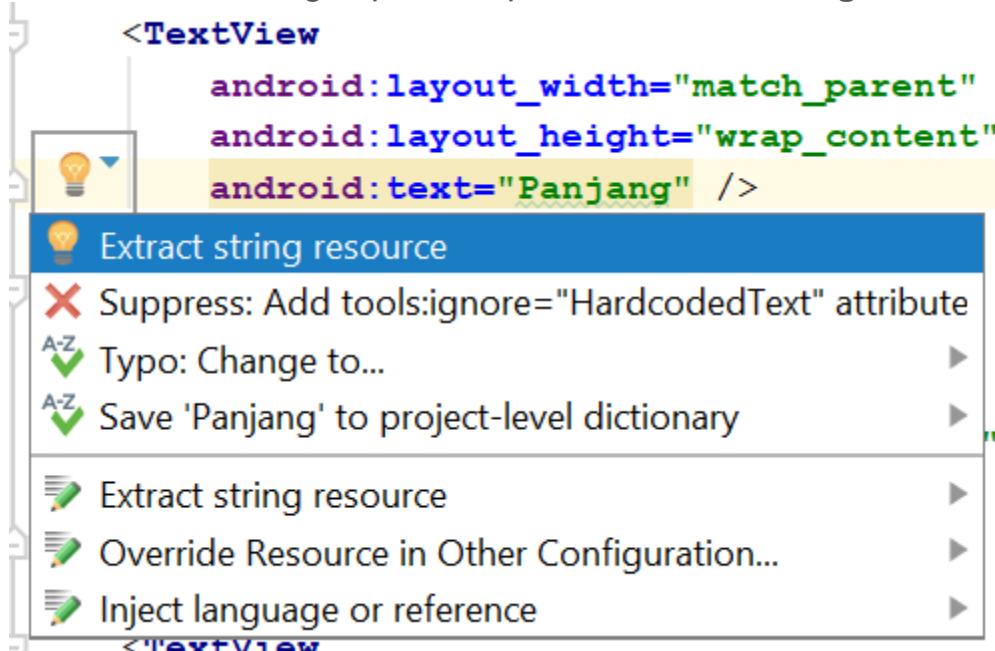
The screenshot shows a portion of an XML layout file in the Android Studio code editor. The code is:

```
10 <TextView  
11     android:layout_width="match_parent"  
12     android:layout_height="wrap_content"  
13     android:text="Panjang" />
```

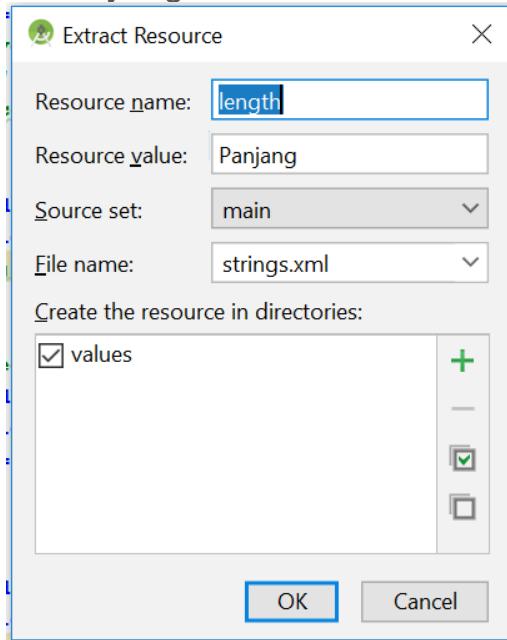
A yellow warning icon is positioned next to the line "android:text="Panjang"" at line 13. A tooltip below the editor window reads: "Hardcoded string 'Panjang', should use @string resource more... (Ctrl+F1)".

Ini karena kita melakukan *hardcoding* pada nilai *string*-nya. Mari kita hilangkan *code warning* tersebut dengan menekan **Alt+Enter** (**option + return** pada Mac) atau menekan lampu kuning yang muncul pada attribut android:text.

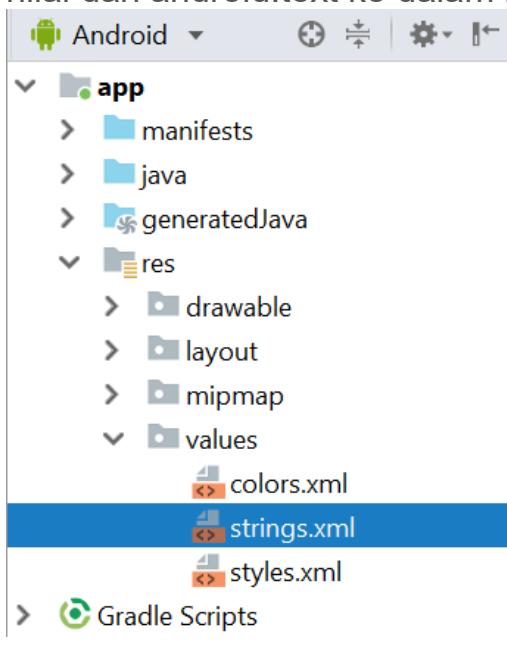
Akan muncul dialog seperti ini, pilihlah **extract string resource**.



3. Kemudian akan muncul dialog seperti di bawah ini. Sesuaikan dengan nama yang ada.



4. 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 berkas **strings.xml**, maka isinya akan menjadi seperti ini:

```
1. <resources>
2.   <string name="app_name">BarVolume</string>
3.   <string name="width">Lebar</string>
4.   <string name="height">Tinggi</string>
```

```

5.      <string name="calculate">Hitung</string>
6.      <string name="result">Hasil</string>
7.      <string name="length">Panjang</string>
8.  </resources>

```

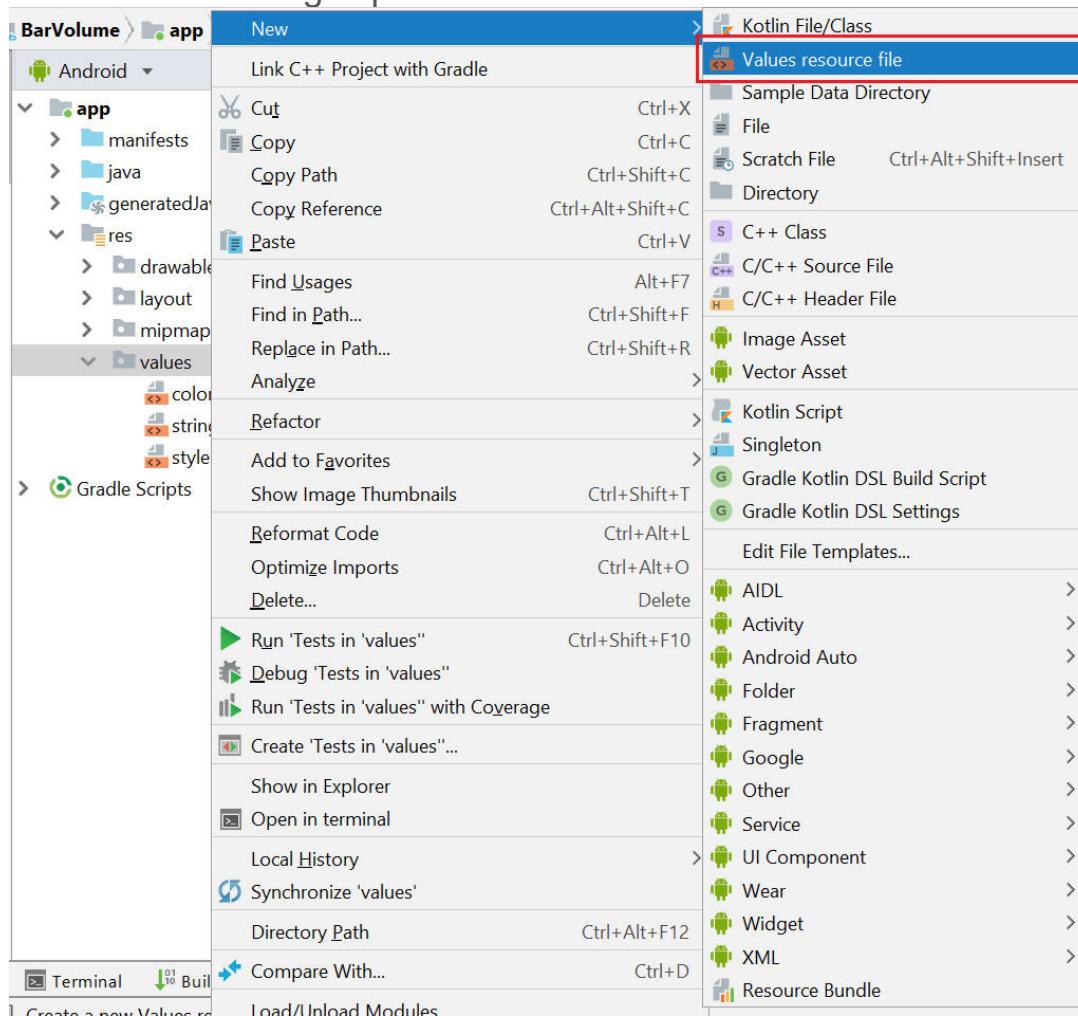
5. Jika terjadi kesalahan pada atribut dimens, maka kita perlu menambahkan berkas dimens.xml di dalam **res → values → dimens.xml**. Kesalahan ini disebabkan karena pada Android Studio 3, berkas **dimens.xml** sudah tidak dibuat secara otomatis.

```

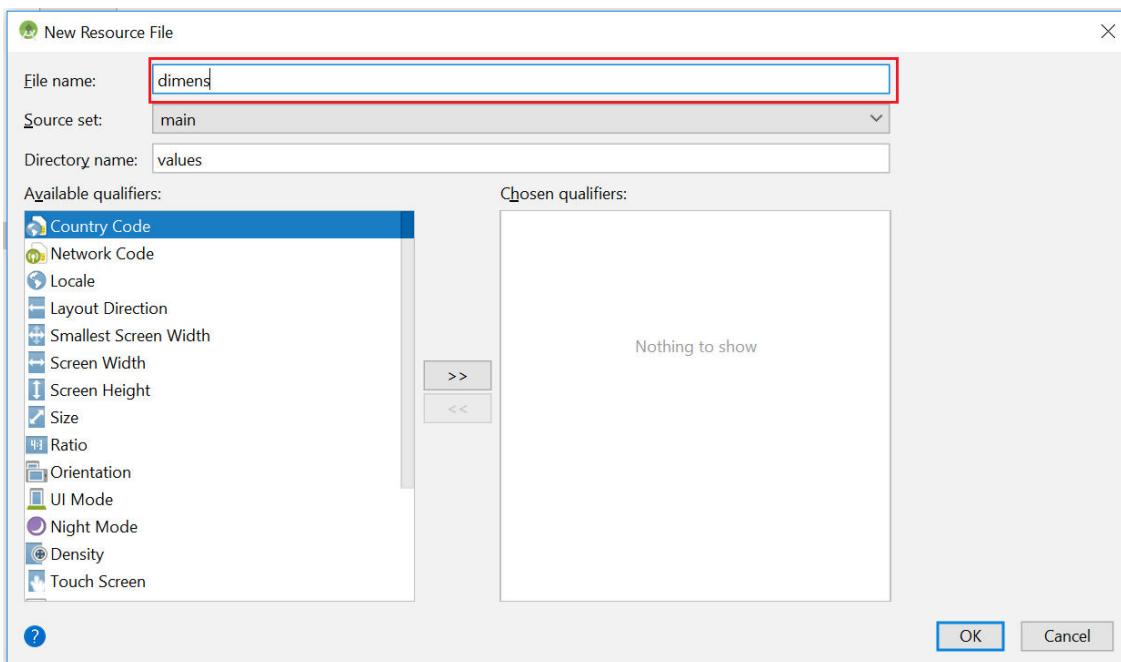
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"

```

Langsung saja tambahkan dengan cara **klik kanan** pada directory res. Akan muncul dialog seperti ini.



6. Kemudian isi sesuai di bawah ini.



7. Jika berkas **dimens.xml** sudah dibuat, sesuaikan isinya menjadi seperti berikut.

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <resources>
3.   <dimen name="activity_horizontal_margin">16dp</dimen>
4.   <dimen name="activity_vertical_margin">8dp</dimen>
5. </resources>
```

Menambahkan Kode Logika Sederhana pada MainActivity

1. Selanjutnya setelah selesai, lanjutkan dengan membuka berkas **MainActivity.java** dan lanjutkan ngoding baris-baris dibawah ini. Tambahkan beberapa variabel yang akan digunakan untuk menampung view.

```
1. EditText edtWidth, edtHeight, edtLength;
2. Button btnCalculate;
3. TextView tvResult;
```

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

```
4. @Override
5. protected void onCreate(Bundle savedInstanceState) {
6.     super.onCreate(savedInstanceState);
7.     setContentView(R.layout.activity_main);
8.
9.     edtWidth = findViewById(R.id.edt_width);
10.    edtHeight = findViewById(R.id.edt_height);
11.    edtLength = findViewById(R.id.edt_length);
12.    btnCalculate = findViewById(R.id.btn_calculate);
13.    tvResult = findViewById(R.id.tv_result);
14.
15. }
```

Kemudian tambahkan implementasi interface **onClickListener** pada kelas **MainActivity**.

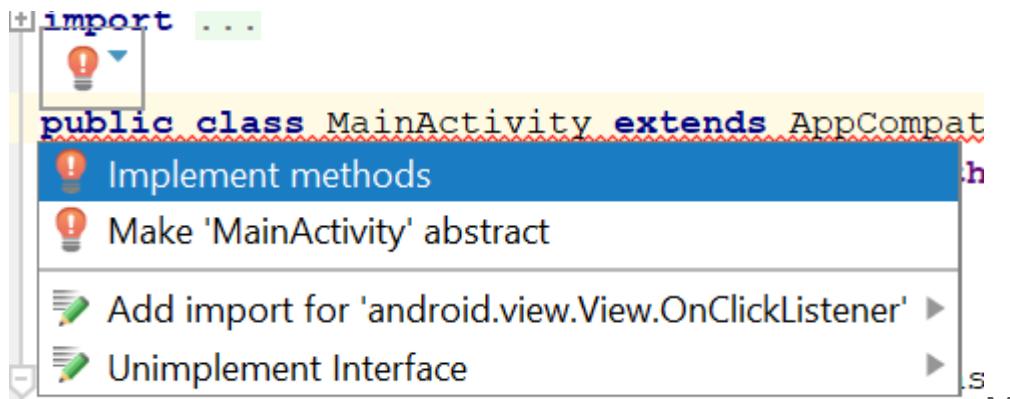
```
16. public class MainActivity extends AppCompatActivity
17.     implements View.OnClickListener {
18.
19. ...
20.
21. }
```

Jika terdapat baris merah seperti ini :

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener{
    EditText edtWidth, edtHeight, edtLength;
    Button btnCalculate;
    TextView tvResult;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        edtWidth = findViewById(R.id.edt_width);
        edtHeight = findViewById(R.id.edt_height);
        edtLength = findViewById(R.id.edt_length);
        btnCalculate = findViewById(R.id.btn_calculate);
        tvResult = findViewById(R.id.tv_result);
    }
}
```

Jangan khawatir! Silakan klik di atas baris merah tersebut dan klik pada ikon bola lampu. Atau dengan tekan tombol **Alt + Enter (option + return pada Mac)** atau menekan lampu merah yang muncul lalu pilih **implement methods**.



Maka, secara otomatis akan ditambahkan metode `onClick` di kelas `MainActivity`. Tambahkan kode berikut ini.

```
22. @Override
23. protected void onCreate(Bundle savedInstanceState) {
24.     super.onCreate(savedInstanceState);
25.     setContentView(R.layout.activity_main);
26.
27.     ...
28.
29.     btnCalculate.setOnClickListener(this);
30. }
31.
32. @Override
33. public void onClick(View v) {
34.     if (v.getId() == R.id.btn_calculate) {
35.         String inputLength = edtLength.getText().toString().trim();
36.         String inputWidth = edtWidth.getText().toString().trim();
37.         String inputHeight = edtHeight.getText().toString().trim();
38.
39.         boolean isEmptyFields = false;
40.         boolean isInvalidDouble = false;
41.
42.         if (TextUtils.isEmpty(inputLength)) {
43.             isEmptyFields = true;
44.             edtLength.setError("Field ini tidak boleh kosong");
45.         }
46.
47.         if (TextUtils.isEmpty(inputWidth)) {
48.             isEmptyFields = true;
49.             edtWidth.setError("Field ini tidak boleh kosong");
50.         }
51.
52.         if (TextUtils.isEmpty(inputHeight)) {
53.             isEmptyFields = true;
54.             edtHeight.setError("Field ini tidak boleh kosong");
55.         }
56.
57.         Double length =toDouble(inputLength);
58.         Double width =toDouble(inputWidth);
```

```

59.         Double height = toDouble(inputHeight);
60.
61.         if (length == null) {
62.             isValidDouble = true;
63.             edtLength.setError("Field ini harus berupa nomer yang valid"
64. );
65.         }
66.         if (width == null) {
67.             isValidDouble = true;
68.             edtWidth.setError("Field ini harus berupa nomer yang valid")
69. ;
70.         }
71.         if (height == null) {
72.             isValidDouble = true;
73.             edtHeight.setError("Field ini harus berupa nomer yang valid"
74. );
75.         }
76.         if (!isEmptyFields && !isValidDouble) {
77.
78.             double volume = length * width * height;
79.
80.             tvResult.setText(String.valueOf(volume));
81.         }
82.     }
83. }
```

Terakhir tambahkan kode untuk validasi nilai double.

```

84. Double toDouble(String str) {
85.     try {
86.         return Double.valueOf(str);
87.     } catch (NumberFormatException e) {
88.         return null;
89.     }
90. }
```

Akhirnya kelas **MainActivity** akan memiliki kode seperti berikut ini.

```

91. public class MainActivity extends AppCompatActivity implements View.OnClickListener {
92.     EditText edtWidth, edtHeight, edtLength;
93.     Button btnCalculate;
94.     TextView tvResult;
95.
96.     @Override
97.     protected void onCreate(Bundle savedInstanceState) {
```

```
98.        super.onCreate(savedInstanceState);
99.        setContentView(R.layout.activity_main);
100.
101.        edtWidth = findViewById(R.id.edt_width);
102.        edtHeight = findViewById(R.id.edt_height);
103.        edtLength = findViewById(R.id.edt_length);
104.        btnCalculate = findViewById(R.id.btn_calculate);
105.        tvResult = findViewById(R.id.tv_result);
106.
107.        btnCalculate.setOnClickListener(this);
108.    }
109.
110.    @Override
111.    public void onClick(View v) {
112.        if (v.getId() == R.id.btn_calculate) {
113.            String inputLength = edtLength.getText().toString().trim()
114.                ();
115.            String inputWidth = edtWidth.getText().toString().trim()
116.                ;
117.            String inputHeight = edtHeight.getText().toString().trim()
118.                ();
119.
120.            boolean isEmptyFields = false;
121.            boolean isValidDouble = false;
122.
123.            if (TextUtils.isEmpty(inputLength)) {
124.                isEmptyFields = true;
125.                edtLength.setError("Field ini tidak boleh kosong");
126.            }
127.
128.            if (TextUtils.isEmpty(inputWidth)) {
129.                isEmptyFields = true;
130.                edtWidth.setError("Field ini tidak boleh kosong");
131.            }
132.
133.            if (TextUtils.isEmpty(inputHeight)) {
134.                isEmptyFields = true;
135.                edtHeight.setError("Field ini tidak boleh kosong");
136.            }
137.
138.            Double length = toDouble(inputLength);
139.            Double width = toDouble(inputWidth);
140.            Double height = toDouble(inputHeight);
141.
142.            if (length == null) {
143.                isValidDouble = true;
144.                edtLength.setError("Field ini harus berupa nomer yang
145.                    valid");
146.            }
147.        }
148.
```

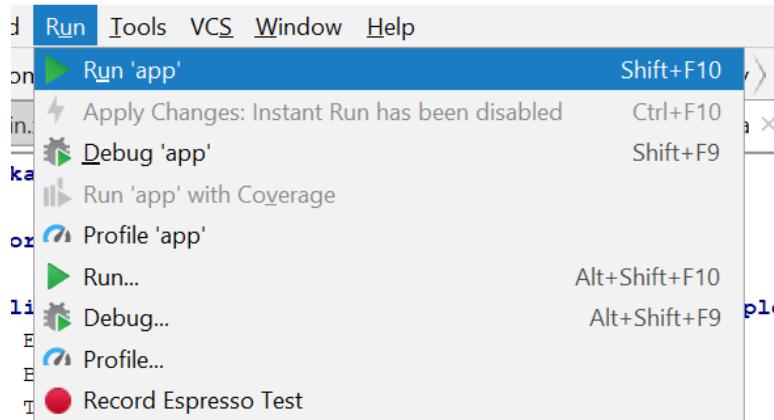
```

148.
149.             if (height == null) {
150.                 isValidDouble = true;
151.                 edtHeight.setError("Field ini harus berupa nomer yang valid");
152.             }
153.
154.             if (!isEmptyFields && !isValidDouble) {
155.                 double volume = length * width * height;
156.                 tvResult.setText(String.valueOf(volume));
157.             }
158.         }
159.     }
160.
161.     Double toDouble(String str) {
162.         try {
163.             return Double.valueOf(str);
164.         } catch (NumberFormatException e) {
165.             return null;
166.         }
167.     }
168. }

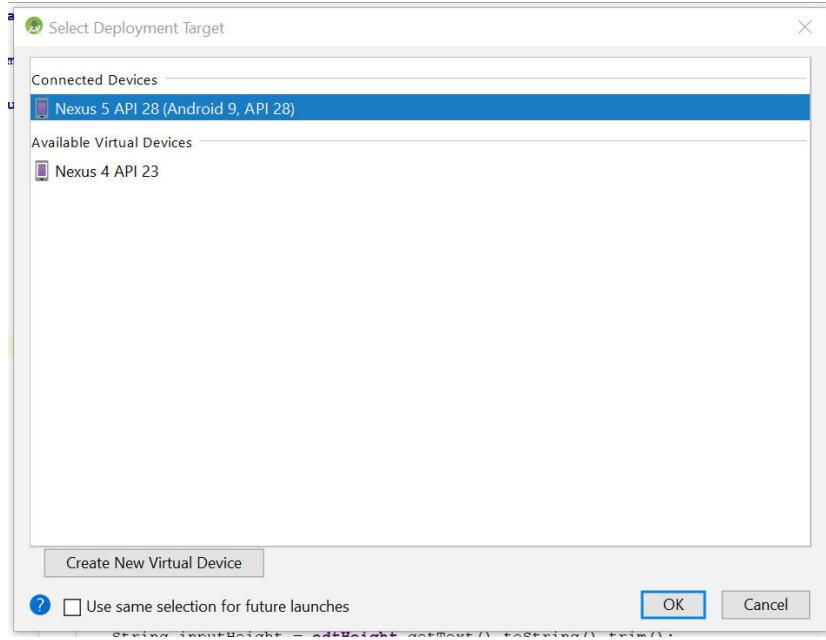
```

2. Setelah selesai, silakan jalankan aplikasi dengan memilih menu **Run → Run 'app'** dari menu bar.

java\com\dicoding\picodiploma\barvolume\MainActivity.java [app] - And



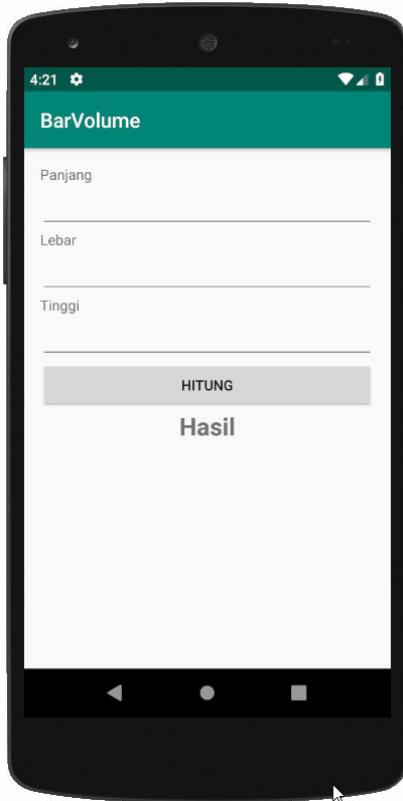
Kemudian ada pilihan seperti ini.



Itu tandanya adb (*Android Debugger*) pada peranti yang Anda punya telah terhubung dengan Android Studio. Jika Anda tidak memiliki peranti, maka Anda dapat menggunakan emulator. Ikuti materinya di modul sebelumnya <https://www.dicoding.com/academies/14/tutorials/1132> atau di sini <https://developer.android.com/studio/run/managing-avds.html>.

Kami merekomendasikan Anda menggunakan peranti Android sewaktu mengembangkan aplikasi. Selain karena beban memori pada peranti Anda akan jadi lebih rendah, pendekatan ini juga akan memungkinkan Anda untuk merasakan bagaimana aplikasi berjalan di device sebenarnya.

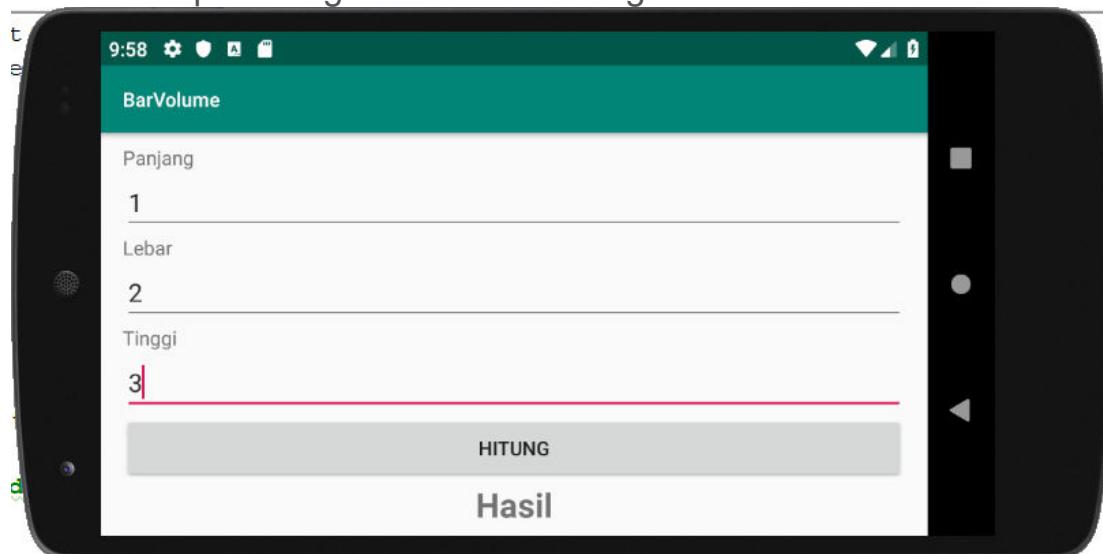
Pilih **OK** untuk menjalankan dan tunggu hingga proses *building* dan instalasi APK selesai. Jika sudah, seharusnya hasilnya akan seperti ini.



Silakan masukan nilai panjang, lebar dan tinggi kemudian tekan tombol **Hitung** dan hasilnya akan ditampilkan di obyek textView **tvHasil**.

3. Apakah kita sudah selesai?

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



Untuk mengatasinya tambahkan beberapa baris berikut

pada **MainActivity.java**.

```
1. private static final String STATE_RESULT = "state_result";
2.
3. ...
4.
5. @Override
6. protected void onSaveInstanceState(Bundle outState) {
7.     super.onSaveInstanceState(outState);
8.     outState.putString(STATE_RESULT, tvResult.getText().toString());
9. }
```

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

```
10. @Override
11. protected void onCreate(Bundle savedInstanceState) {
12.     super.onCreate(savedInstanceState);
13.     setContentView(R.layout.activity_main);
14.
15.     ...
16.
17.     if (savedInstanceState != null) {
18.         String result = savedInstanceState.getString(STATE_RESULT);
19.         tvResult.setText(result);
20.     }
21. }
```

4. Silakan jalankan kembali aplikasinya. Ulangi proses perhitungan seperti sebelumnya. Kemudian ganti orientasi peranti Anda. Jika sudah benar maka hasil perhitungan tidak akan hilang.

Bedah Kode

Pembahasan tentang layout xml

Layout merupakan user interface dari suatu activity. Layout dituliskan dalam format xml (extensible markup language).

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

Baris ini mengidentifikasi bahwa berkas ini berformat xml.

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

Kode di atas menandakan namespace yang digunakan dalam keseluruhan berkas xml ini.

Macam Views

Di sini kita menggunakan 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 :

```
1. <TextView  
2.     android:layout_width="match_parent"  
3.     android:layout_height="wrap_content"  
4.     android:text="@string/calculate"  
5.     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

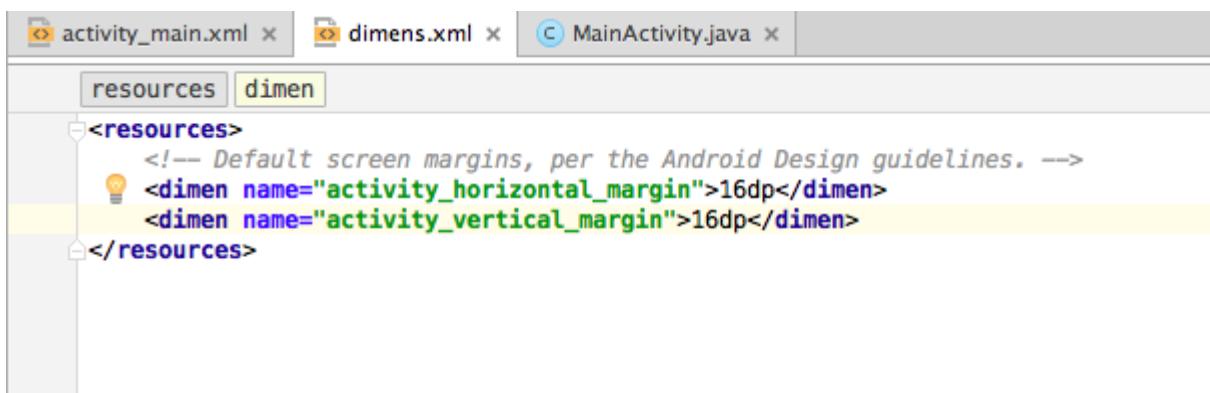
Attribute

Value

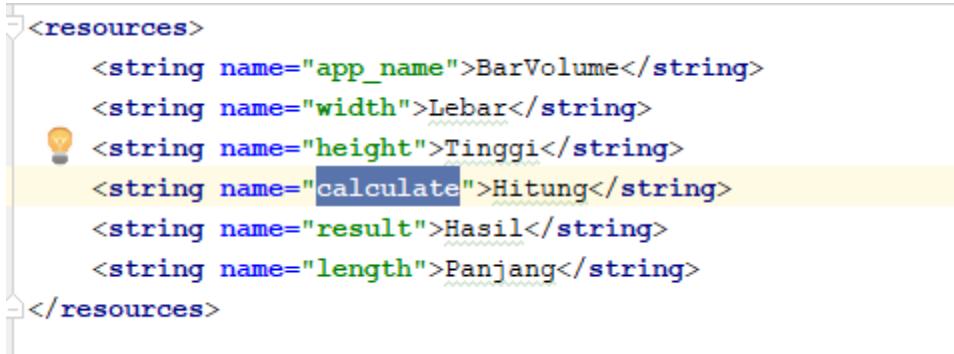
match_parent: Ini berarti bahwa ukuran dimensi sebuah View disesuaikan dengan ukuran layar secara horizontal jika pada **layout_width** dan vertikal jika pada **layout_height**.

wrap_content : Ini berarti bahwa ukuran dimensi sebuah View disesuaikan dengan ukuran konten di dalamnya 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. .



@string/calculate: value calculate berasal dari berkas strings.xml.



Penggunaan *centralize resource value* akan memudahkan Anda sewaktu mengembangkan aplikasi Android. Cara tersebut digunakan agar Anda tidak menulis nilai yang sama berulang-ulang.

Apa itu '@+id/' ?

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

1. <Button

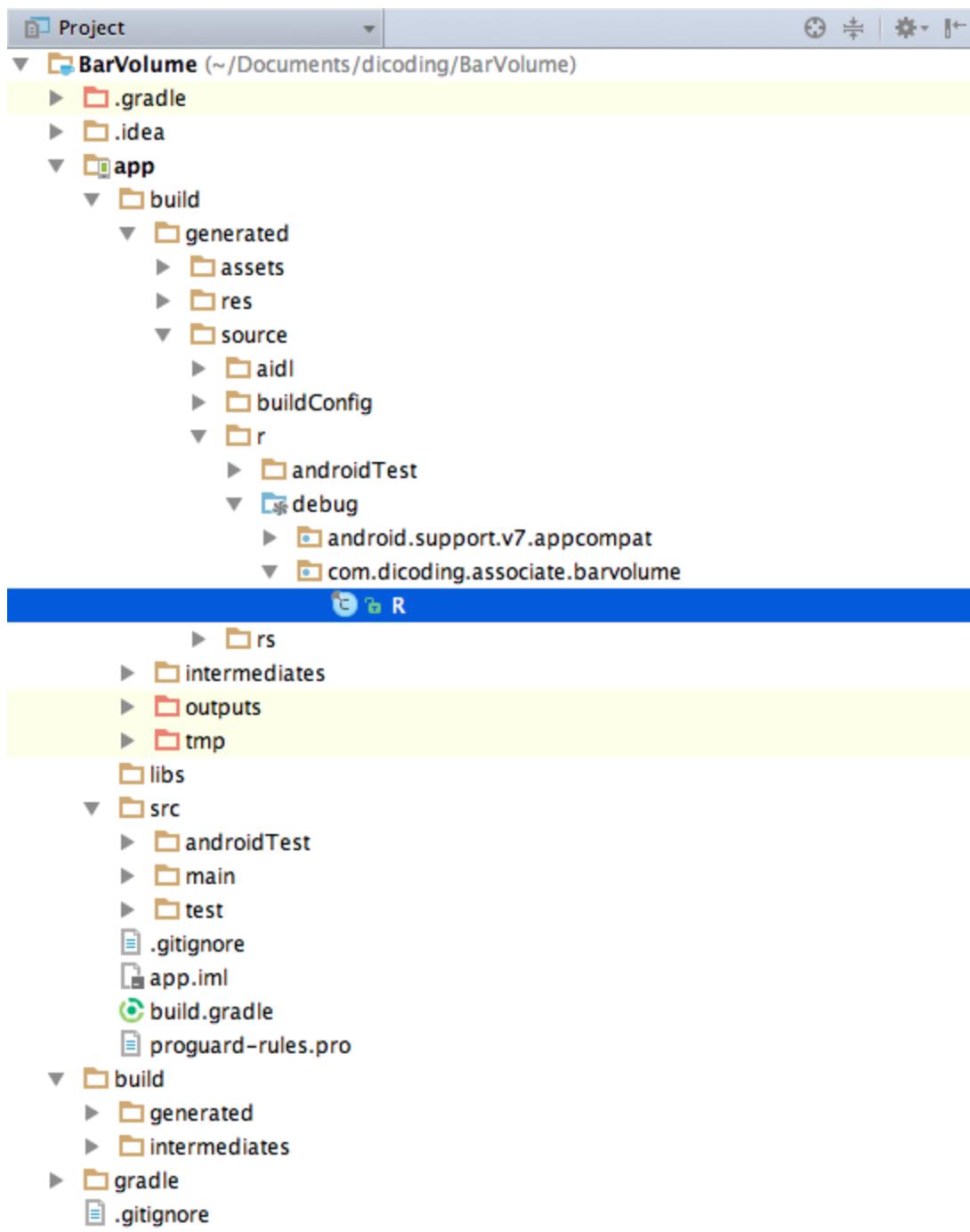
```
2.     android:id="@+id	btn_calculate"
3.     android:layout_width="match_parent"
4.     android:layout_height="wrap_content"
5.     android:text="@string/calculate"
6.     android:layout_marginBottom="@dimen/activity_vertical_margin"/>
```

Penjelasannya sebagai berikut:

```
1. 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**.



Acuan untuk menyusun tampilan pada relativelayout akan dibahas pada modul selanjutnya.

Pembahasan tentang Logika Kode

Kode logika dituliskan ke dalam kelas Java. Di sini lah semua aktifitas dari suatu aplikasi berjalan.

Activity

```
1. public class MainActivity extends AppCompatActivity
```

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

OnClickListener

```
1. implements View.OnClickListener
```

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

Views

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

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

OnCreate

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

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

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

Maksud baris diatas adalah kelas **MainActivity** akan menampilkan tampilan yang berasal dari layout **activity_main.xml**.

Casting View

```
1. edtWidth = findViewById(R.id.edt_width);
```

Maksud dari baris diatas adalah obyek **edittext** **edtWidth** disesuaikan (cast) dengan komponen **edittext** ber-ID **edt_width** di layout xml melalui metode **findViewById()**.

SetOnClickListener

```
1. btnCalculate.setOnClickListener(this);
```

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

```
1. @Override
2. protected void onSaveInstanceState(Bundle outState) {
3.     outState.putString(STATE_RESULT, tvResult.getText().toString());
4.
5.     super.onSaveInstanceState(outState);
6. }
```

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.

```
1. if (savedInstanceState != null){
2.     String hasil = savedInstanceState.getString(STATE_RESULT);
```

```
3.      tvResult.setText(hasil);  
4. }
```

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

INTENT TEORI

Teori

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 : com.dicoding.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.

Tujuan

Codelab ini menitik beratkan pada implementasi intent untuk melakukan perpindahan dari activity ke activity lain, dengan atau tidak membawa data. Beberapa bagian dari codelab 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.

Codelab selanjutnya adalah dengan membuat aplikasi yang dapat menerapkan kegunaan intent. Seperti ini tampilannya.

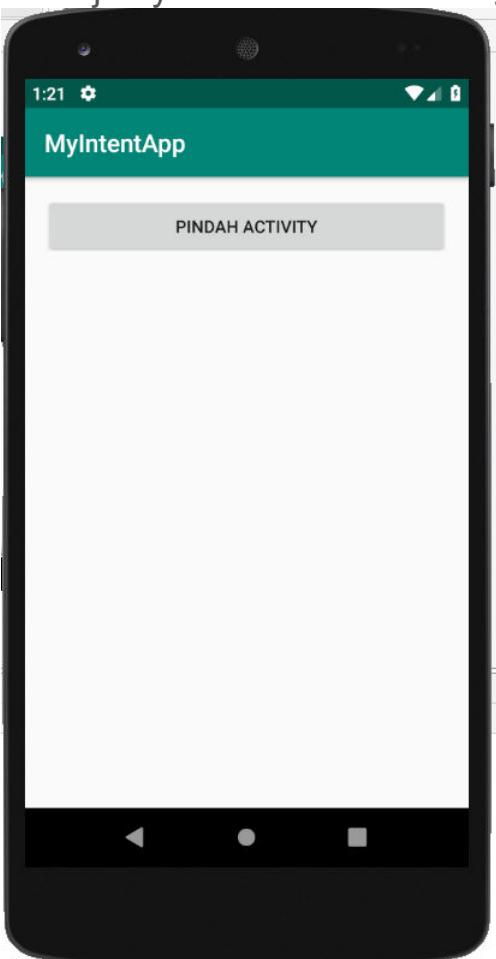
Logika Dasar

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.

Codelab Intent Sederhana

1. Buat Project baru di Android Studio dengan kriteria sebagai berikut:
 - Nama Project : **MyIntentApp**
 - Target & Minimum Target SDK : **Phone and Tablet, Api level 21**
 - Tipe Activity : **Empty Activity**
 - Activity Name : **MainActivity**

2. Selanjutnya kita akan membangun antarmuka (*interface*) seperti ini:



3. Kondisikan `activity_main.xml` menjadi seperti ini :

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

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

```
20. <resources>
21.     <!-- Default screen margins, per the Android Design guidelines. -->
22.     <dimen name="activity_horizontal_margin">16dp</dimen>
23.     <dimen name="activity_vertical_margin">16dp</dimen>
24. </resources>
```

Untuk resource stringnya juga perlu kita tambahkan. Kita tambahkan semua string yang akan digunakan di project ini, buka berkas **strings.xml** dan tambahkan kode berikut ini.

```
25. <resources>
26.     <string name="app_name">MyIntentApp</string>
27.     <string name="move_activity">Pindah Activity</string>
28.     <string name="move_with_data">Pindah Activity dengan Data</string>
29.     <string name="move_with_object">Pindah Activity dengan Object</strin
g>
30.     <string name="dial_number">Dial a Number</string>
31.     <string name="move_with_result">Pindah Activity untuk Result</string>
    >
32.     <string name="result_from_activity">Hasil dari Activity</string>
33.     <string name="choose_number">Pilih angka yang kamu suka</string>
34.     <string name="_50">50</string>
35.     <string name="_100">100</string>
36.     <string name="_150">150</string>
37.     <string name="_200">200</string>
38.     <string name="choose">Pilih</string>
39.     <string name="data_received">Data Diterima</string>
40.     <string name="object_received">Object Diterima</string>
41.     <string name="this_is_moveactivity">Ini MoveActivity</string>
42. </resources>
```

4. Kemudian untuk **MainActivity.java** tambahkan beberapa baris yang berfungsi untuk meperkenalkan button yang sudah ditambahkan di layout **activity_main.xml** seperti ini :

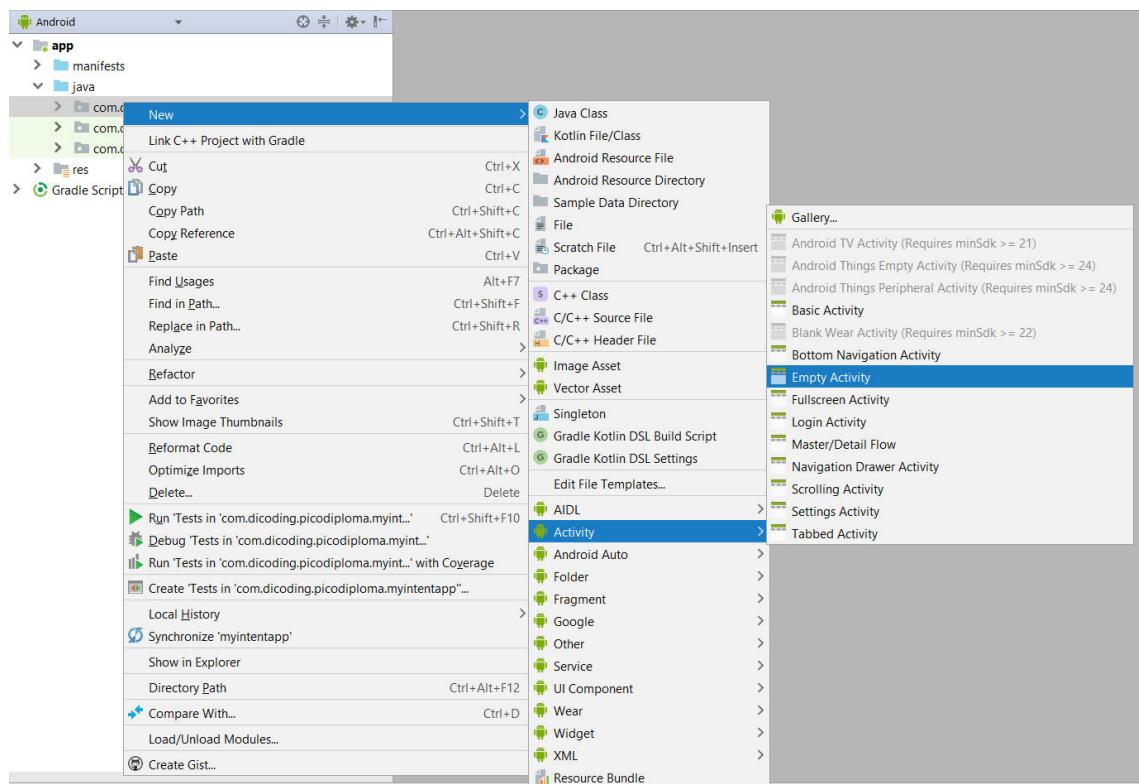
```
0. Button btnMoveActivity;
1.
2. @Override
3. protected void onCreate(Bundle savedInstanceState) {
```

```
4.     super.onCreate(savedInstanceState);
5.     setContentView(R.layout.activity_main);
6.
7.     btnMoveActivity = findViewById(R.id.btn_move_activity);
8. }
```

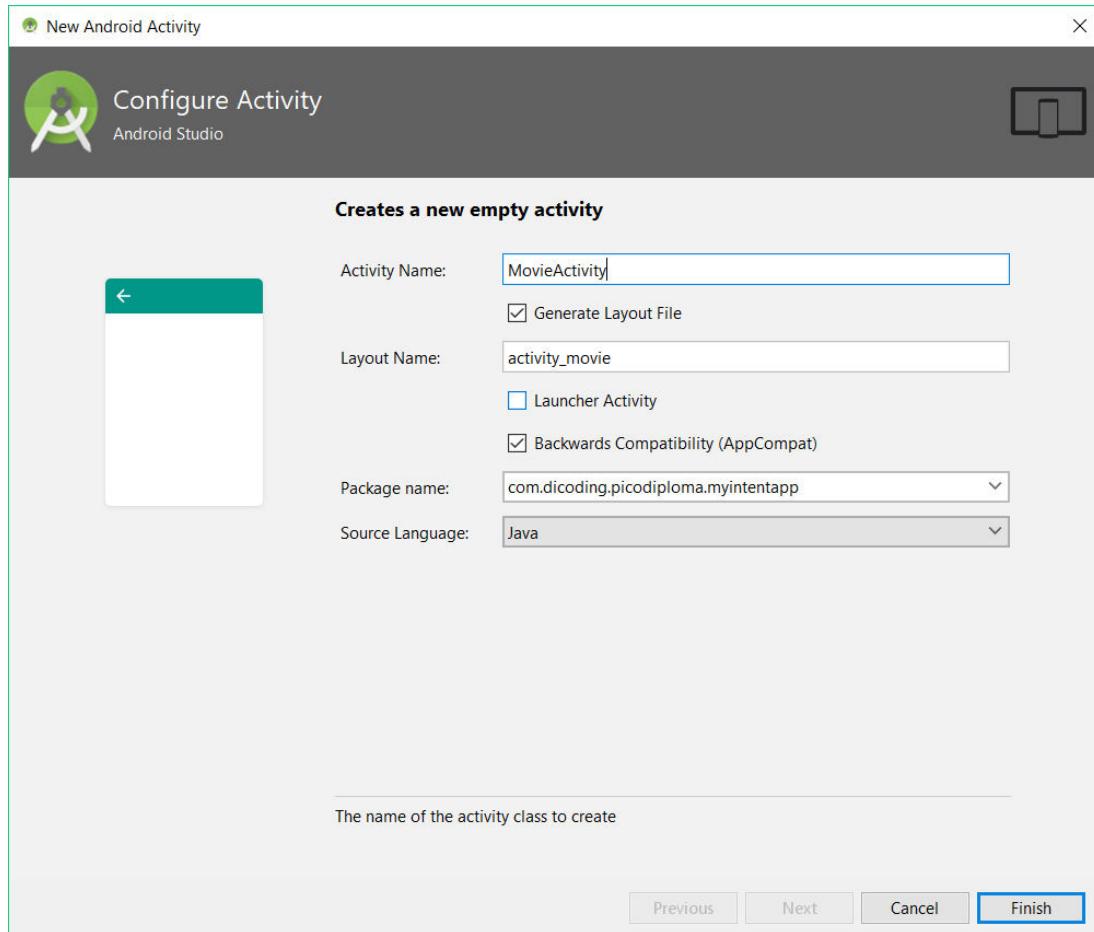
5. Lalu tambahkan beberapa baris yang berfungsi untuk menambahkan event onClick pada button **btnMoveActivity** seperti ini.

```
0. public class MainActivity extends AppCompatActivity implements View.OnClickListener {
1.     Button btnMoveActivity;
2.
3.     @Override
4.     protected void onCreate(Bundle savedInstanceState) {
5.         super.onCreate(savedInstanceState);
6.         setContentView(R.layout.activity_main);
7.
8.         btnMoveActivity = findViewById(R.id.btn_move_activity);
9.         btnMoveActivity.setOnClickListener(this);
10.    }
11.
12.    @Override
13.    public void onClick(View v) {
14.        switch (v.getId()) {
15.            case R.id.btn_move_activity:
16.                break;
17.        }
18.    }
19. }
```

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**.



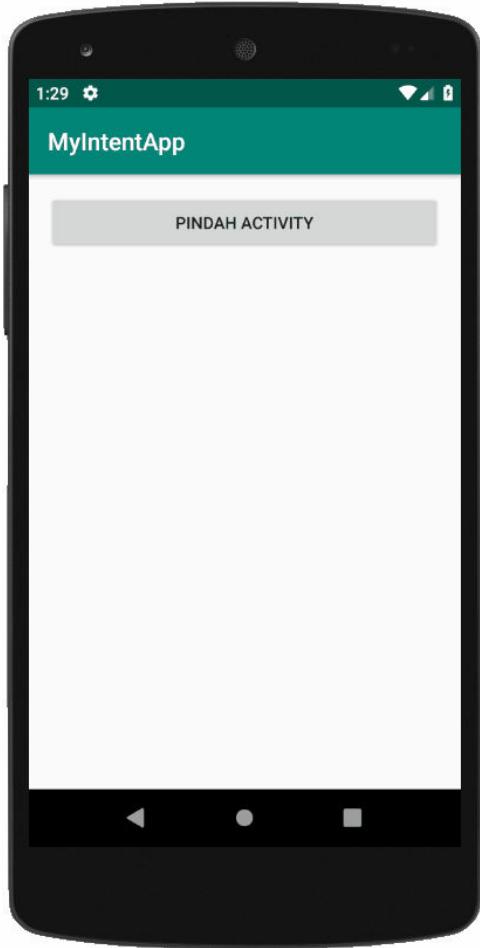
7. Untuk menandakan bahwa perpindahan activity berhasil, silakan tambahkan satu **TextView** dan kondisikan **activity_move.xml** menjadi seperti berikut.

```
0. <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
1.     xmlns:tools="http://schemas.android.com/tools"
2.     android:layout_width="match_parent"
3.     android:layout_height="match_parent"
4.     android:paddingBottom="@dimen/activity_vertical_margin"
5.     android:paddingLeft="@dimen/activity_horizontal_margin"
6.     android:paddingRight="@dimen/activity_horizontal_margin"
7.     android:paddingTop="@dimen/activity_vertical_margin"
8.     tools:context="com.dicoding.picodiploma.myintentapp.MoveActivity">
9.
10.    <TextView
11.        android:layout_width="match_parent"
12.        android:layout_height="wrap_content"
13.        android:text="@string/this_is_moveactivity" />
14. </RelativeLayout>
```

8. Setelah activity tujuan sudah berhasil diciptakan, sekarang saatnya menambahkan suatu intent pada method `onClick()` di `MainActivity.java` menjadi sebagai berikut.

```
0. @Override
1. public void onClick(View v) {
2.     switch (v.getId()){
3.         case R.id.btn_move_activity:
4.             Intent moveIntent = new Intent(MainActivity.this, MoveActivity.class);
5.             startActivity(moveIntent);
6.             break;
7.     }
8. }
```

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



Bedah Kode

Register Activity

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.

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3.   package="com.dicoding.picodiploma.myintentapp">
4.
5.   <application
6.     android:allowBackup="true"
7.     android:icon="@mipmap/ic_launcher"
8.     android:label="@string/app_name"
9.     android:supportsRtl="true"
10.    android:theme="@style/AppTheme">
```

```
11.      <activity android:name=".MainActivity">
12.          <intent-filter>
13.              <action android:name="android.intent.action.MAIN" />
14.
15.              <category android:name="android.intent.category.LAUNCHER" />
16.          </intent-filter>
17.      </activity>
18.      <activity android:name=".MoveActivity" />
19.  </application>
20.
21.</manifest>
```

Perhatikan bahwa **MoveActivity** sudah teregistrasi di **AndroidManifest.xml** dan sekarang sudah aman jika kita melakukan perpindahan activity dari **MainActivity** ke **MoveActivity**.

Jika kita lupa meregistrasikan Activity baru ke dalam berkas **AndroidManifest.xml**, maka akan terjadi *error* seperti ini "android.content.ActivityNotFoundException:Unable to find explicit activity class".

Intent Filter

Berikutnya, perhatikan pada kode dengan tag intent-filter yang ada di dalam **MainActivity**.

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

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

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

Baris di atas bermakna bahwa **MainActivity** menjadi entry point ke aplikasi.

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

Kode di atas menandakan bahwa **MainActivity** akan dikategorikan sebagai *activity launcher*. Ini menandakan bahwa activity ini akan muncul di halaman home screen dalam bentuk launcher.

Pindah Activity

Selanjutnya, perhatikan kode berikut.

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

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.

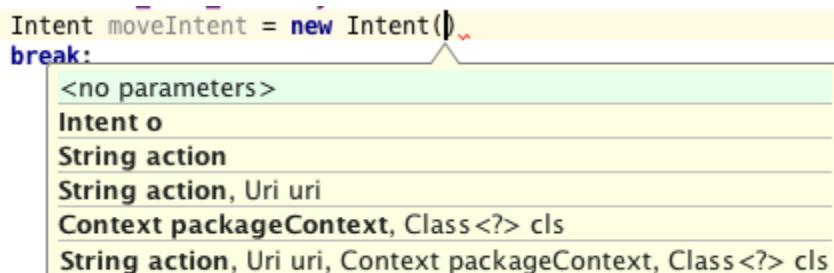
Intent(`MainActivity.this`, `MoveActivity.class`);

Kelas Asal

Kelas Tujuan

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

Jika kita menggunakan *code assistant* (tekan `ctrl + spasi`) dari Android Studio, maka akan tampil rekomendasi (*code hint*) sebagai berikut:



The screenshot shows the Android Studio code editor with the following code:

```
Intent moveIntent = new Intent()  
break:
```

A code completion dropdown menu is open at the cursor position, listing the following parameters:

- <no parameters>
- Intent o
- String action
- String action, Uri uri
- Context packageContext, Class<?> cls
- String action, Uri uri, Context packageContext, Class<?> cls

Pada konteks di atas kita memilih `Context packageContext, Class cls` sebagai inputan untuk nilai *constructor* intent.

Kemudian perhatikan baris kode berikutnya.

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

`startActivity(moveIntent)` metode ini akan menjalankan activity baru tanpa membawa data. Obyek intent yang diinputkan adalah obyek `moveIntent` yang akan ketika kode ini dijalankan maka akan membuka 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.

INTENT Explisit

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 di bawah tombol sebelumnya, sehingga kode `activity_main.xml` kita menjadi seperti ini.

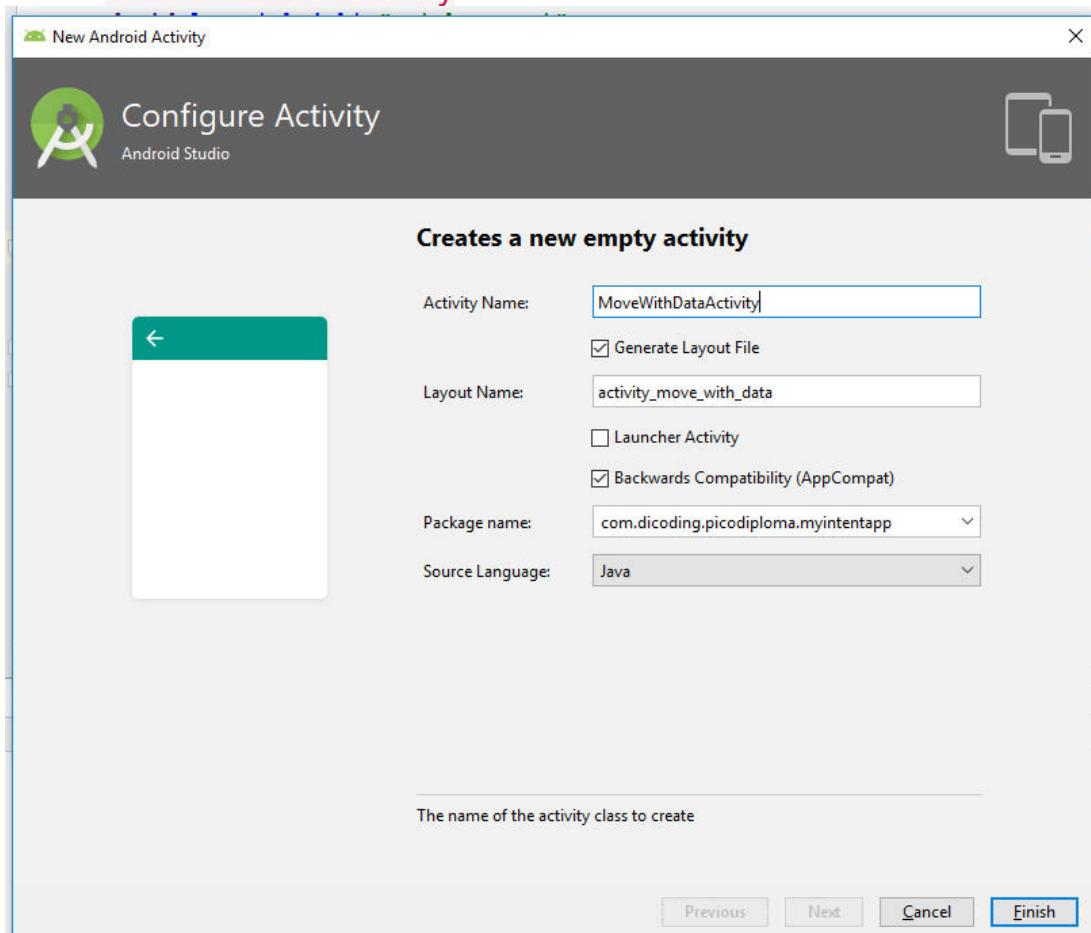
```
1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.   xmlns:tools="http://schemas.android.com/tools"
4.   android:layout_width="match_parent"
5.   android:layout_height="match_parent"
6.   android:orientation="vertical"
7.   android:paddingBottom="@dimen/activity_vertical_margin"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

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

di **MainActivity.java**.

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

3. Selanjutnya, buat **Activity** baru lagi seperti cara sebelumnya dan beri nama **MoveWithDataActivity**.



4. Lalu, pada layout **activity_move_with_data.xml** kita tambahkan sebuah **TextView** ber-ID untuk menampilkan data yang dikirimkan dari **Activity** asal.

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

```
15.      android:layout_height="wrap_content"
16.      android:text="@string/data_received" />
17. </RelativeLayout>
```

5. Selanjutnya pada **MoveWithDataActivity.java** kita kondisikan kita kenalkan **TextView** seperti berikut ini :

```
1. public class MoveWithDataActivity extends AppCompatActivity {
2.     TextView tvDataReceived;
3.
4.     @Override
5.     protected void onCreate(Bundle savedInstanceState) {
6.         super.onCreate(savedInstanceState);
7.         setContentView(R.layout.activity_move_with_data);
8.
9.         tvDataReceived = findViewById(R.id.tv_data_received);
10.
11.    }
12. }
```

6. Selanjutnya untuk menerima data dari **Activity** asal, kondisikan **MoveWithDataActivity** menjadi seperti berikut :

```
1. public class MoveWithDataActivity extends AppCompatActivity {
2.     public static final String EXTRA_NAME = "extra_name";
3.     public static final String EXTRA_AGE = "extra_age";
4.
5.     TextView tvDataReceived;
6.
7.     @Override
8.     protected void onCreate(Bundle savedInstanceState) {
9.         super.onCreate(savedInstanceState);
10.        setContentView(R.layout.activity_move_with_data);
11.
12.        tvDataReceived = findViewById(R.id.tv_data_received);
13.
14.        String name = getIntent().getStringExtra(EXTRA_NAME);
15.        int age = getIntent().getIntExtra(EXTRA_AGE, 0);
16.
17.        String text = "Name : " + name + ",\nYour Age : " + age;
18.        tvDataReceived.setText(text);
19.    }
20. }
```

7. Dan sekarang saatnya kita menambahkan obyek Intent pada `MainActivity.java` seperti baris yang ditebalkan.

```
1.  @Override
2.  public void onClick(View v) {
3.      switch (v.getId()){
4.          case R.id.btn_move_activity:
5.              Intent moveIntent = new Intent(MainActivity.this, MoveActivity.class);
6.              startActivity(moveIntent);
7.              break;
8.          case R.id.btn_move_activity_data:
9.              Intent moveWithDataIntent = new Intent(MainActivity.this,
10.                  MoveWithDataActivity.class);
11.              moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_NAME,
12.                  "DicodingAcademy Boy");
13.              moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_AGE,
14.                  5);
15.              startActivity(moveWithDataIntent);
16.              break;
17.      }
18.  }
```

8. Maka `MainActivity.java` akan jadi seperti ini :

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

```
23.             break;
24.         case R.id.btn_move_activity_data:
25.             Intent moveWithDataIntent = new Intent(MainActivity.this
26. , MoveWithDataActivity.class);
27.             moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_N
28. AME, "DicodingAcademy Boy");
29.             moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_A
30. GE, 5);
31.         }
32.     }
```

9. 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 activity lain dengan membawa data. Dan itu sangat penting karena ketika kita mengembangkan suatu aplikasi Android yang kompleks, akan ada banyak activity yang terlibat. Lancarnya perpindahan activity dengan data menjadi hal yang penting untuk memberikan pengalaman yang baik kepada pengguna.

Put Extra

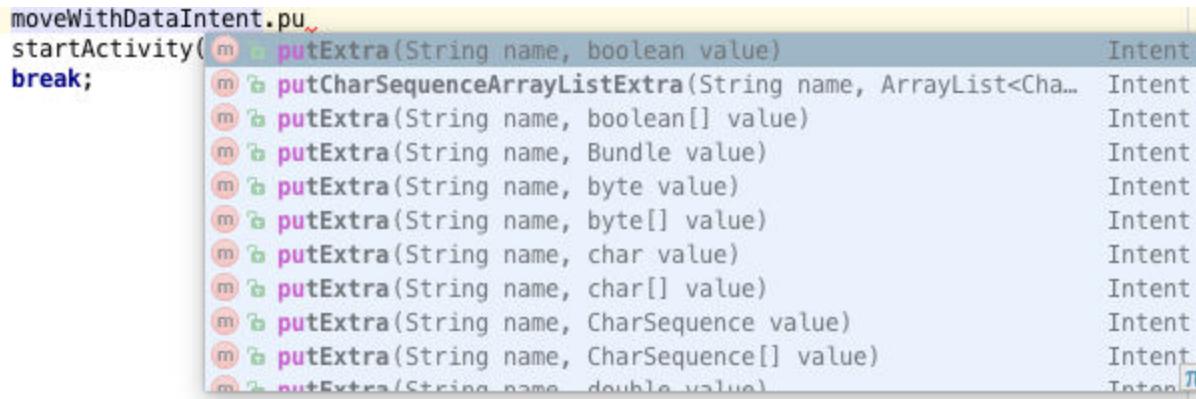
Perhatikan kode di bawah ini.

```
1. Intent moveWithDataIntent = new Intent(MainActivity.this, MoveWithDataActivity.class);
2. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_NAME, "DicodingAcademy Boy");
3. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_AGE, 5);
4. startActivity(moveWithDataIntent);
```

Perbedaan mendasar antara memindahkan activity dengan membawa data atau tidak, adalah dengan menempatkan data ke obyek intent pada baris ini.

```
1. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_NAME, "DicodingAcademy Boy");
2. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_AGE, 5);
```

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()`.

```
1. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_NAME, "DicodingAcademy Boy");
```

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 : DicodingAcademy Boy dengan tipe data string.

Get Data

Setelah data dikirimkan, selanjutnya adalah mengambil data tersebut.

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:

```
1. 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 variable age.

```
1. 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”). Nilai default di sini merupakan nilai yang akan digunakan jika

ternyata datanya kosong. Data kosong atau nilainya null bisa terjadi ketika datanya memang tidak ada, atau key-nya tidak sama.

Selamat! Anda telah mempelajari dua intent explicit dengan atau tidak membawa data.

Sebelumnya, mengirimkan data bernilai tunggal dari satu activity ke activity lain adalah hal yang mudah. Bernilai tunggal karena data yang dikirimkan berasal dari satu tipe data.

Misalnya pada contoh di atas, pengiriman nilai data `name` dan `age` dilakukan secara individu. Yang satu bertipe data string dan yang lainnya bertipe data integer.

Sekarang pertanyaanya bagaimana Anda bisa mengirimkan data dalam bentuk Plain Old Java Object (POJO) dari satu activity ke activity lain melalui intent? Caranya adalah dengan mengubah obyek POJO yang Anda miliki ke dalam bentuk obyek *parcelable*. Di Android Studio sudah ada fasilitas plugin bernama Android Parcelable Generator untuk memudahkan proses tersebut.

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

Intent - Latihan : Intent Explicit dengan Parcelable

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

```
1. public class MyBean {  
2.     private String someProperty;  
3.  
4.     public String getSomeProperty() {  
5.         return someProperty;  
6.     }  
}
```

```

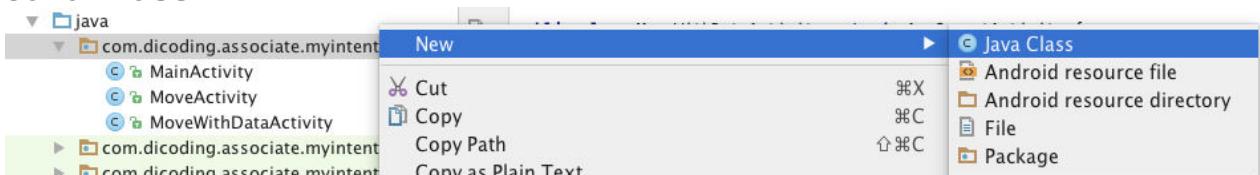
7. }
8.
9. public void setSomeProperty(String someProperty) {
10.     this.someProperty = someProperty;
11. }
12.

```

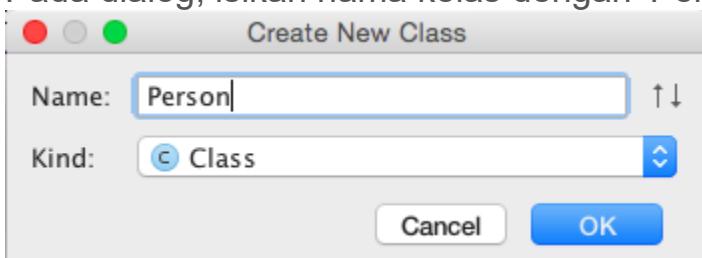
Lalu apa kegunaan obyek POJO? POJO akan membantu kita saat aplikasi semakin kompleks. Contohnya, POJO bisa kita gunakan untuk melakukan koneksi ke server untuk *request API* atau akses ke *database* lokal dengan SQLite.

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

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



2. Pada dialog, isikan nama kelas dengan 'Person'. Lalu klik **OK**.



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

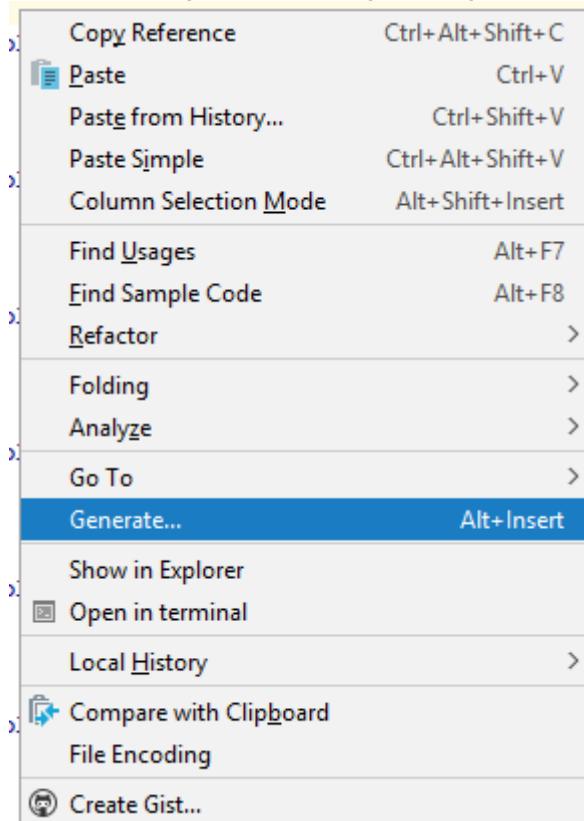
```

1. public class Person {
2.     private String name;
3.     private int age;
4.     private String email;
5.     private String city;
6.

```

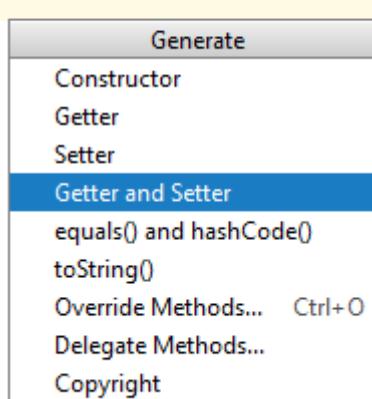
Anda tidak perlu menulis satu per satu metode *setter-getter* nya, di Android Studio tersedia fasilitas *generator* untuk hal tersebut. Caranya, setelah Anda menentukan variabel apa saja yang akan digunakan lakukan, lakukan beberapa langkah sebagai berikut :

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

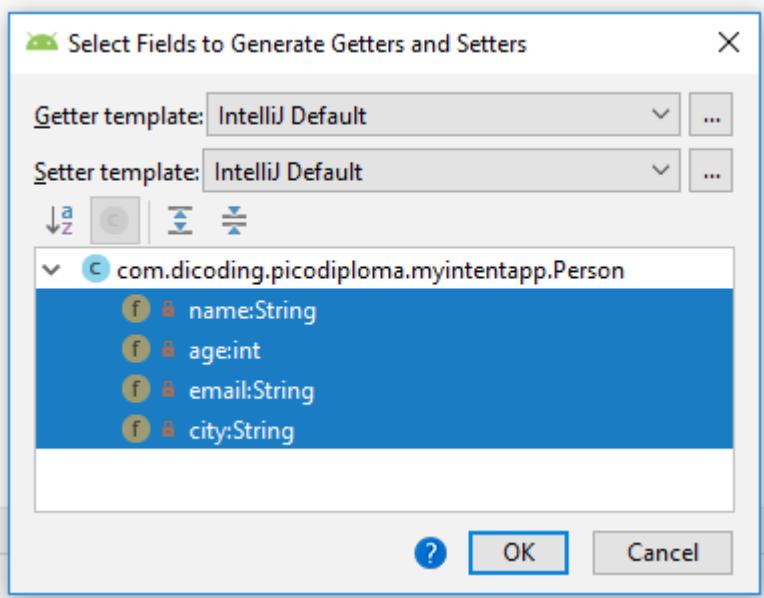


4. Pilih **Getter and Setter**, atau bisa kombinasikan dengan menekan alt+insert.

```
public class Person {  
    private String name;  
    private int age;  
    private String email;  
    private String city;
```



5. Pilih semua variabel dengan cara **Ctrl+A** pada windows dan **Command+A** pada mac.



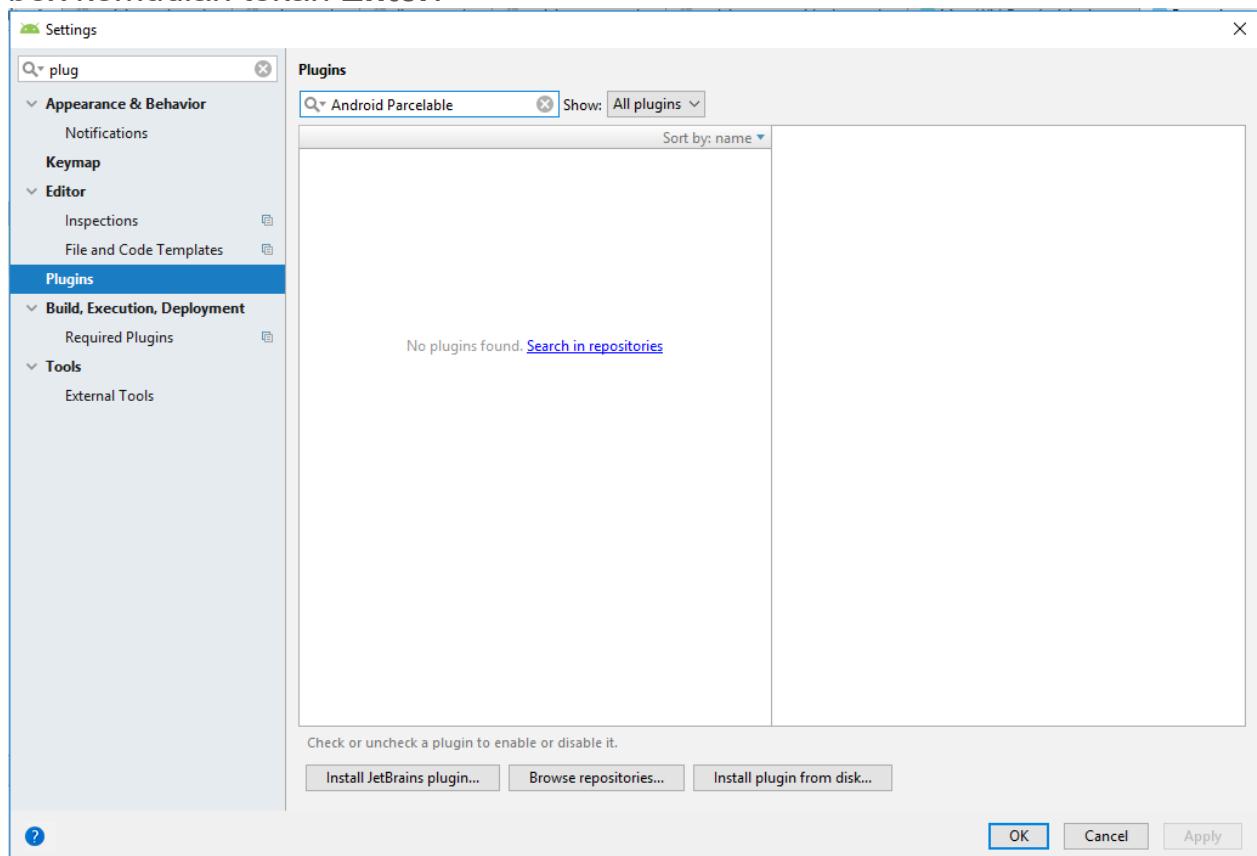
6. Bila sudah, maka klik **OK**. Secara otomatis metode *setter* dan *getter* masing-masing variabel akan dibuatkan.

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

```
30.
31.     public String getCity() {
32.         return city;
33.     }
34.
35.     public void setCity(String city) {
36.         this.city = city;
37.     }
38. }
```

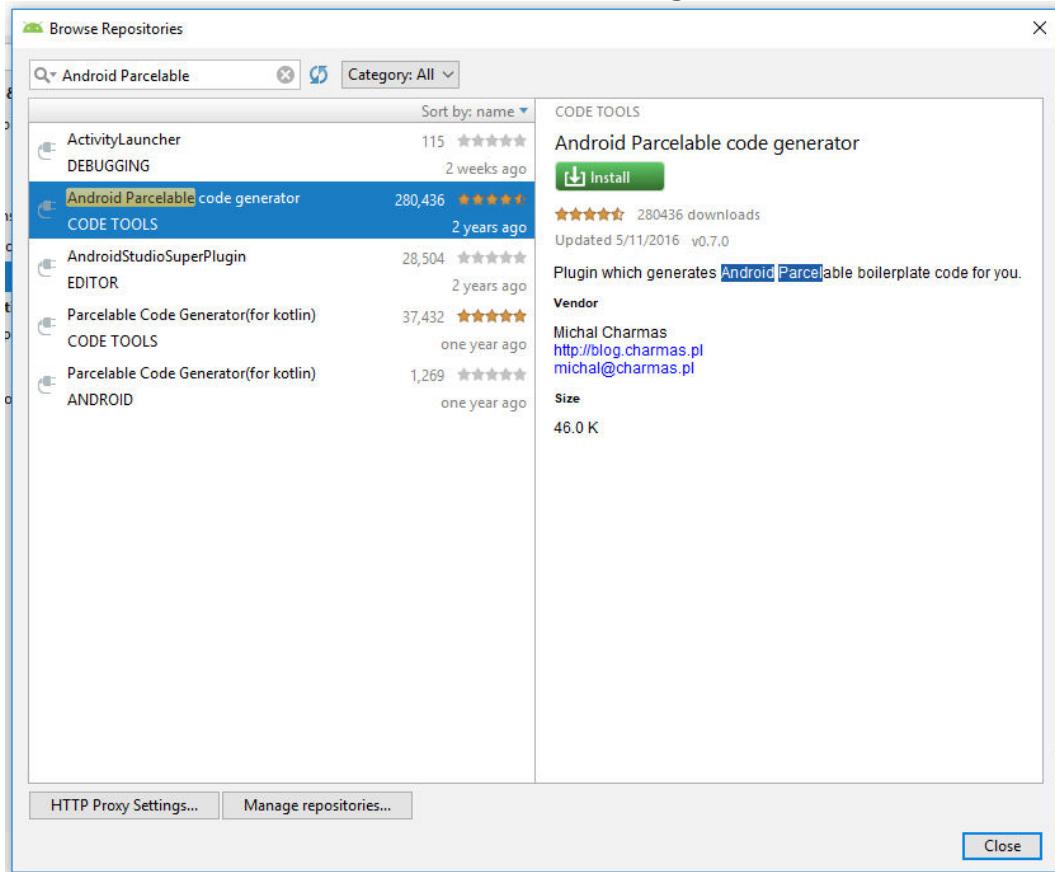
Selanjutnya kita akan menambahkan *plugin* Android Parcelable Generator pada Android Studio. *Plugin* ini berguna untuk mengubah kelas POJO menjadi obyek **Parcelable**. Langkah-langkahnya sebagai berikut :

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

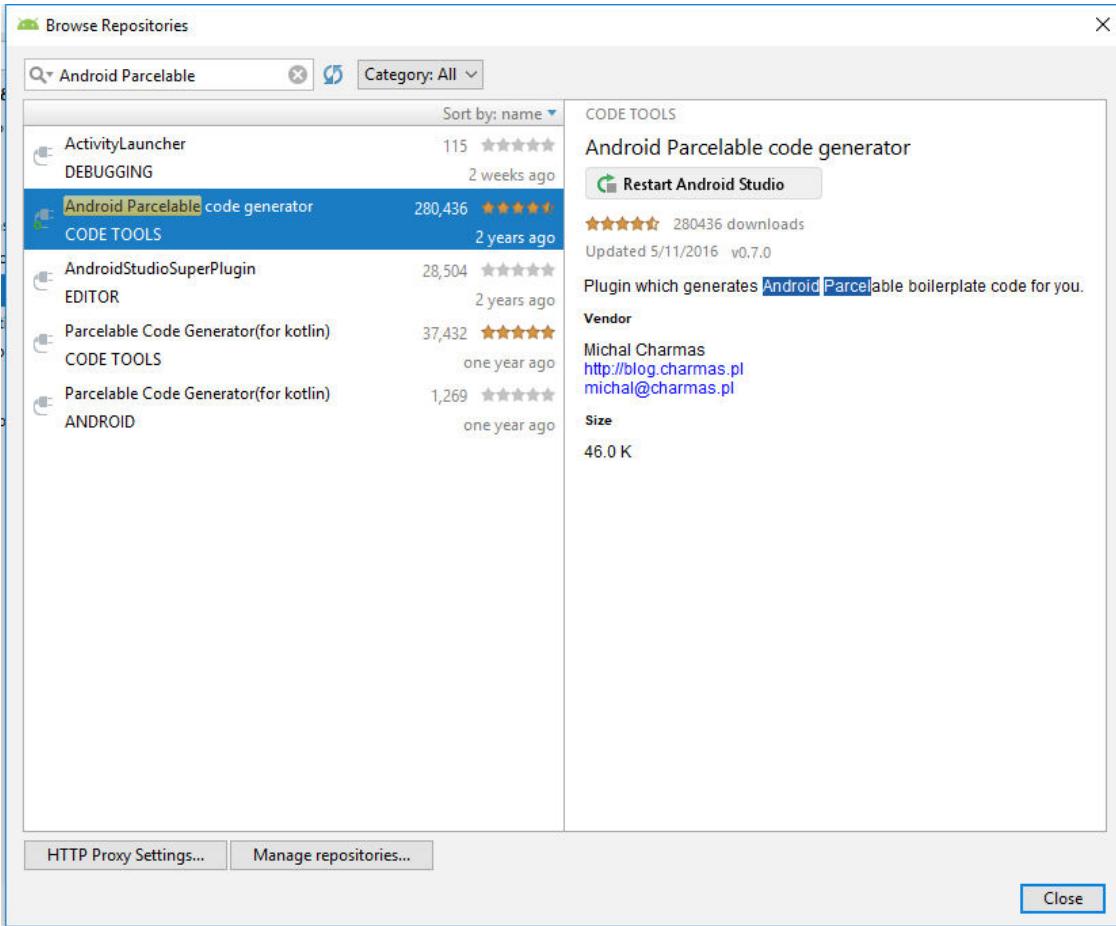


3. Klik **Search in Repositories**, dengan ini kita memerintahkan Android Studio untuk melakukan pencarian *plugin* di server.

Setelah ketemu Android Parcelable code generator, klik **Install**.



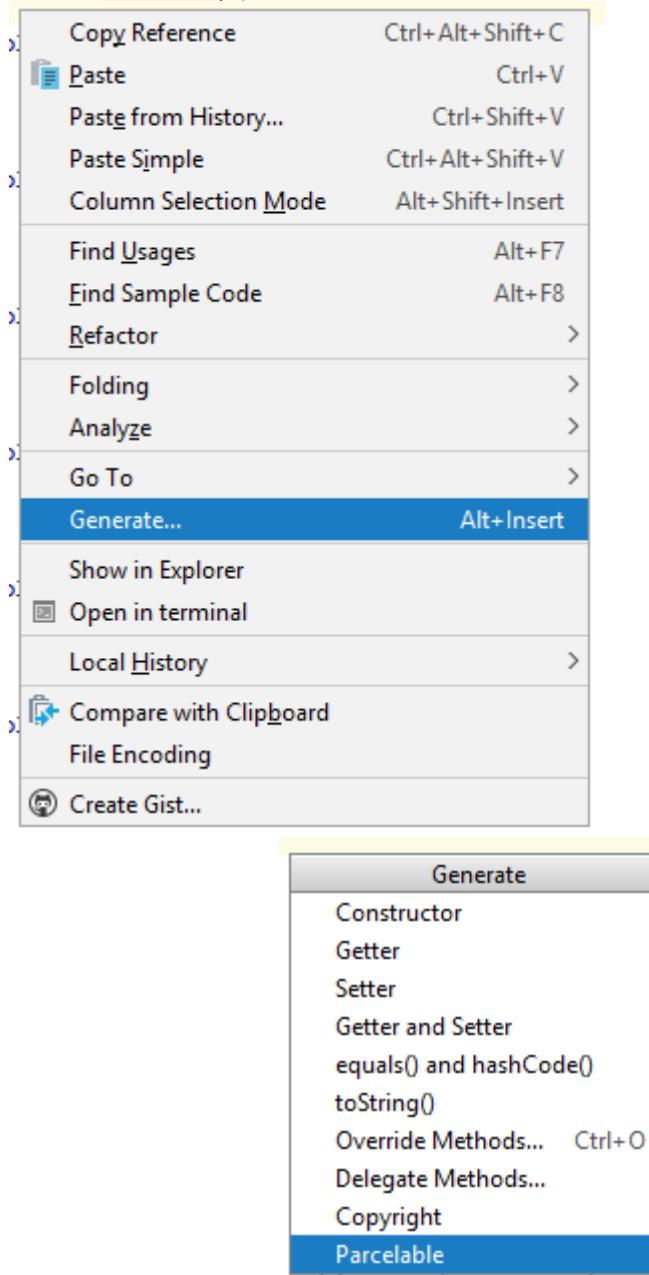
4. Biarkan Android Studio melakukan proses instalasi *plugin*. Bila sukses, maka Anda akan diminta untuk melakukan *restart* Android Studio yang sedang berjalan.



Klik **Restart Android Studio** → 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.

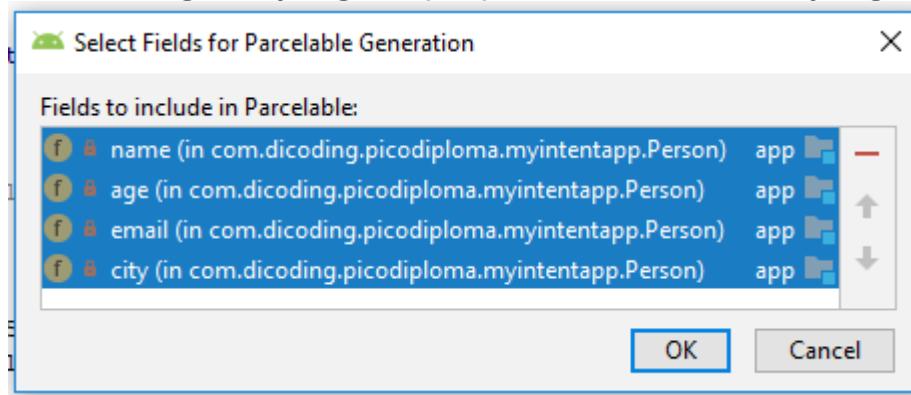
5. Selanjutnya, setelah plugin Parcelable generator terinstal saatnya kita implementasikan pada kelas **Person**. Dengan cara :
Klik kanan pada *workspace* (pastikan anda berada pada tab

kelas Person) pilih menu **Generate**.



6. Pilih **Parcelable**.

7. 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.

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

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

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

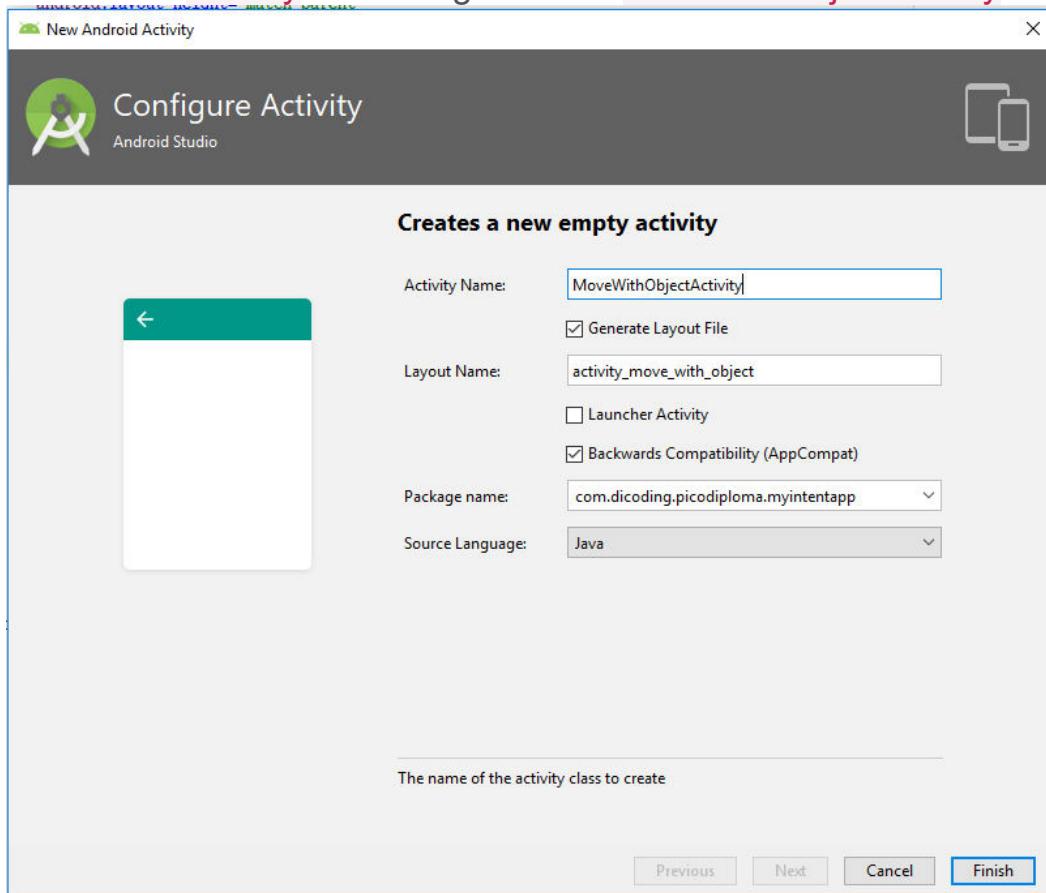
```
7. <?xml version="1.0" encoding="utf-8"?>  
8. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
9.     xmlns:tools="http://schemas.android.com/tools"  
10.    android:layout_width="match_parent"  
11.    android:layout_height="match_parent"  
12.    android:orientation="vertical"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

```
24.        android:text="@string/move_activity" />
25.
26.    <Button
27.        android:id="@+id	btn_move_activity_data"
28.        android:layout_width="match_parent"
29.        android:layout_height="wrap_content"
30.        android:layout_marginBottom="@dimen/activity_vertical_margin"
31.        android:text="@string/move_with_data" />
32.
33.    <Button
34.        android:id="@+id	btn_move_activity_object"
35.        android:layout_width="match_parent"
36.        android:layout_height="wrap_content"
37.        android:layout_marginBottom="@dimen/activity_vertical_margin"
38.        android:text="@string/move_with_object" />
39.
40. </LinearLayout>
```

9. Setelah selesai dengan layout xml, sekarang kita kenalkan button tersebut di kelas MainActivity.java.

```
1. ...
2. Button btnMoveWithObject;
3.
4. @Override
5. protected void onCreate(Bundle savedInstanceState) {
6.     super.onCreate(savedInstanceState);
7.     setContentView(R.layout.activity_main);
8.
9.     ...
10.
11.    btnMoveWithObject = findViewById(R.id.btn_move_activity_object);
12.    btnMoveWithObject.setOnClickListener(this);
13. }
14. ...
```

10. Buat **Activity** baru dengan nama **MoveWithObjectActivity**.



11. Kemudian, pada layout **activity_move_with_object.xml** tambahkan satu **TextView** untuk menampilkan data yang dikirimkan.

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

12. Sekarang kita kenalkan TextView pada **MoveWithObjectActivity.java** tambahkan kodennya sehingga menjadi sebagai berikut:

```
1. public class MoveWithObjectActivity extends AppCompatActivity {  
2.     TextView tvObject;  
3.  
4.     @Override  
5.     protected void onCreate(Bundle savedInstanceState) {  
6.         super.onCreate(savedInstanceState);  
7.         setContentView(R.layout.activity_move_with_object);  
8.  
9.         tvObject = findViewById(R.id.tv_object_received);  
10.    }  
11. }
```

13. Selanjutnya pada kelas **MoveWithObjectActivity.java** kita tambahkan kode untuk menerima obyek dari Activity Asal.

```
1. public class MoveWithObjectActivity extends AppCompatActivity {  
2.     public static final String EXTRA_PERSON = "extra_person";  
3.     TextView tvObject;  
4.  
5.     @Override  
6.     protected void onCreate(Bundle savedInstanceState) {  
7.         super.onCreate(savedInstanceState);  
8.         setContentView(R.layout.activity_move_with_object);  
9.         tvObject = findViewById(R.id.tv_object_received);  
10.  
11.         Person person = getIntent().getParcelableExtra(EXTRA_PERSON);  
12.         String text = "Name : " + person.getName() + ",\nEmail : " + per  
son.getEmail() + ",\nAge : " + person.getAge() + ",\nLocation : " + pers  
on.getCity();  
13.         tvObject.setText(text);  
14.     }  
15. }
```

14. Setelah kita membuat activity tujuan, sekarang kita tinggal kita tambahkan beberapa baris kode pada bagian **switch... case** di mainactivity.

```
1. @Override
```

```

2. public void onClick(View v) {
3.     switch (v.getId()){
4.         ...
5.         case R.id.btn_move_activity_object:
6.             Person person = new Person();
7.             person.setName("DicodingAcademy");
8.             person.setAge(5);
9.             person.setEmail("academy@dicoding.com");
10.            person.setCity("Bandung");
11.
12.            Intent moveWithObjectIntent = new Intent(MainActivity.this,
13.                MoveWithObjectActivity.class);
14.            moveWithObjectIntent.putExtra(MoveWithObjectActivity.EXTRA_P
15.                ERSON, person);
16.            startActivity(moveWithObjectIntent);
17.            break;
17.    }
17. }

```

15. Sehingga kode **MainActivity.java** kita menjadi sebagai berikut :

```

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

```

```
27.             Intent moveWithDataIntent = new Intent(MainActivity.this,
28.             MoveWithDataActivity.class);
29.             moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_NA-
30. ME, "DicodingAcademy Boy");
31.             moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_AG-
32. E, 5);
33.             startActivity(moveWithDataIntent);
34.             break;
35.         case R.id.btn_move_activity_object:
36.             Person person = new Person();
37.             person.setName("DicodingAcademy");
38.             person.setAge(5);
39.             person.setEmail("academy@dicoding.com");
40.             person.setCity("Bandung");
41.             Intent moveWithObjectIntent = new Intent(MainActivity.thi-
42. s, MoveWithObjectActivity.class);
43.             moveWithObjectIntent.putExtra(MoveWithObjectActivity.EXTR-
44. A_PERSON, person);
45.             startActivity(moveWithObjectIntent);
46.             break;
47.     }
48. }
```

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

secara utuh, Selamat!



Bedah Kode

Akhirnya, Anda sudah berhasil menerapkan 3 tipe perpindahan Activity secara eksplisit melalui Intent. Sekarang, mari kita dalami penjelasannya satu per satu.

Put Parcelable

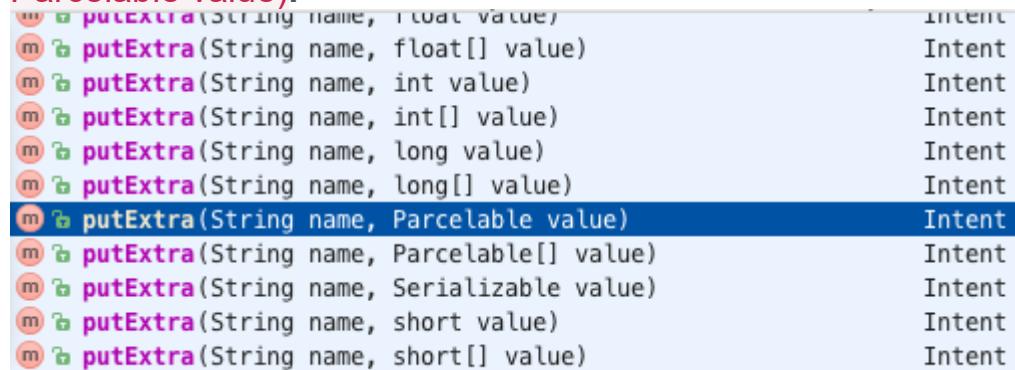
```
1. Person person = new Person();
2. person.setName("DicodingAcademy");
3. person.setAge(5);
4. person.setEmail("academy@dicoding.com");
```

```
5. person.setCity("Bandung");
```

Di atas kita menciptakan sebuah obyek Person bernama `person` yang mana kelas tersebut adalah Parcelable. Kita atur semua data sesuai dengan propertinya. Setelah selesai kita akan mengirimkan obyek tersebut ke `MoveWithObjectActivity` melalui sebuah intent di bawah ini.

```
1. Intent moveWithObjectIntent = new Intent(MainActivity.this, MoveWithObjectActivity.class);
2. moveWithObjectIntent.putExtra(MoveWithObjectActivity.EXTRA_PERSON, person);
3. startActivity(moveWithObjectIntent);
```

Metode `putExtra()` yang kita pilih saat ini adalah `putExtra(String name, Parcelable value)`.



The screenshot shows a code completion dropdown for the `Intent` class. It lists several overloads of the `putExtra` method, each taking a string key and a specific type of value. The method `m putExtra(String name, Parcelable value)` is highlighted with a blue background, indicating it is the one being used in the current context. Other listed methods include `putExtra(String name, float value)`, `putExtra(String name, float[] value)`, `putExtra(String name, int value)`, `putExtra(String name, int[] value)`, `putExtra(String name, long value)`, `putExtra(String name, long[] value)`, `putExtra(String name, Serializable value)`, `putExtra(String name, short value)`, and `putExtra(String name, short[] value)`. The `Intent` class name is also visible at the top of the dropdown.

Did you know that Quick Definition View (`Space`) works in completion lookups as well? [» π](#)

Get Parcelable

Dengan `EXTRA_PERSON` merupakan variabel *static* bertipe data string dan bernilai “`extra_person`”.

Selanjutnya pada `MoveWithObjectActivity` akan dilakukan pengambilan nilai seperti berikut :

```
1. Person person = getIntent().getParcelableExtra(EXTRA_PERSON);
```

Dan ini yang seru, karena obyek ini terdiri dari beragam tipe data pada atribut atau propertinya, kita hanya cukup membungkus itu semua kedalam obyek parcelable. Melalui `getIntent().getParcelableExtra(Key)` Anda dapat mengambil nilai obyek `Person` yang sebelumnya telah dikirim dan menampilkannya seperti ini.

```
1. String text = "Name : "+person.getName()+" , Email : "+person.getEmail()+" , Age : "+person.getAge()
```

```
2.           + ", Location : "+person.getCity());
3. tvObject.setText(text);
```

Rows of Data

Lalu, bagaimana jika kita ingin mengirimkan kumpulan obyek parcelable ke activity lain? Untuk mengirimkan kumpulan data, kita bisa memanfaatkan arraylist dan metode putParcelableArrayListExtra.

Contoh kode ketika put extra adalah seperti ini.

```
1. ArrayList<Person> persons = new ArrayList<>();
2.
3. ...
4.
5. moveWithObjectIntent.putParcelableArrayListExtra(KEY, persons);
```

Dan ketika kita mengambil kumpulan datanya bisa dengan menggunakan kode ini.

```
1. ArrayList<Person> persons = getIntent().getParcelableArrayListExtra(KEY);
```

Selanjutnya, kita akan membuat sebuah contoh aplikasi dengan menggunakan intent implicit.

Intent - 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 :

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

```
6.      android:text="@string/dial_number" />
```

Sehingga file `activity_main.xml` kita sekarang menjadi:

```
7. <?xml version="1.0" encoding="utf-8"?>
8. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
9.     xmlns:tools="http://schemas.android.com/tools"
10.    android:layout_width="match_parent"
11.    android:layout_height="match_parent"
12.    android:orientation="vertical"
13.    android:paddingBottom="@dimen/activity_vertical_margin"
14.    android:paddingLeft="@dimen/activity_horizontal_margin"
15.    android:paddingRight="@dimen/activity_horizontal_margin"
16.    android:paddingTop="@dimen/activity_vertical_margin"
17.    tools:context="com.dicoding.picodiploma.myintentapp.MainActivity">
18.
19.    <Button
20.        android:id="@+id	btn_move_activity"
21.        android:layout_width="match_parent"
22.        android:layout_height="wrap_content"
23.        android:layout_marginBottom="@dimen/activity_vertical_margin"
24.        android:text="@string/move_activity" />
25.
26.    <Button
27.        android:id="@+id	btn_move_activity_data"
28.        android:layout_width="match_parent"
29.        android:layout_height="wrap_content"
30.        android:layout_marginBottom="@dimen/activity_vertical_margin"
31.        android:text="@string/move_with_data" />
32.
33.    <Button
34.        android:id="@+id	btn_move_activity_object"
35.        android:layout_width="match_parent"
36.        android:layout_height="wrap_content"
37.        android:layout_marginBottom="@dimen/activity_vertical_margin"
38.        android:text="@string/move_with_object" />
39.
40.    <Button
41.        android:id="@+id	btn_dial_number"
42.        android:layout_width="match_parent"
43.        android:layout_height="wrap_content"
44.        android:layout_marginBottom="@dimen/activity_vertical_margin"
45.        android:text="@string/dial_number" />
46.
47. </LinearLayout>
```

2. Sekarang, buka kembali `MainActivity.java` dan lanjutkan kenalkan button yang sudah ditambahkan

```
1. Button btnDialPhone;
2.
3. @Override
4. protected void onCreate(Bundle savedInstanceState) {
5.     super.onCreate(savedInstanceState);
6.     setContentView(R.layout.activity_main);
7.
8.     ...
9.
10.    btnDialPhone = findViewById(R.id.btn_dial_number);
11.    btnDialPhone.setOnClickListener(this);
12. }
```

3. Kemudian dengan melakukan casting tombol Dial Number, set listener dan menambahkan *action* ketika tombol diklik.

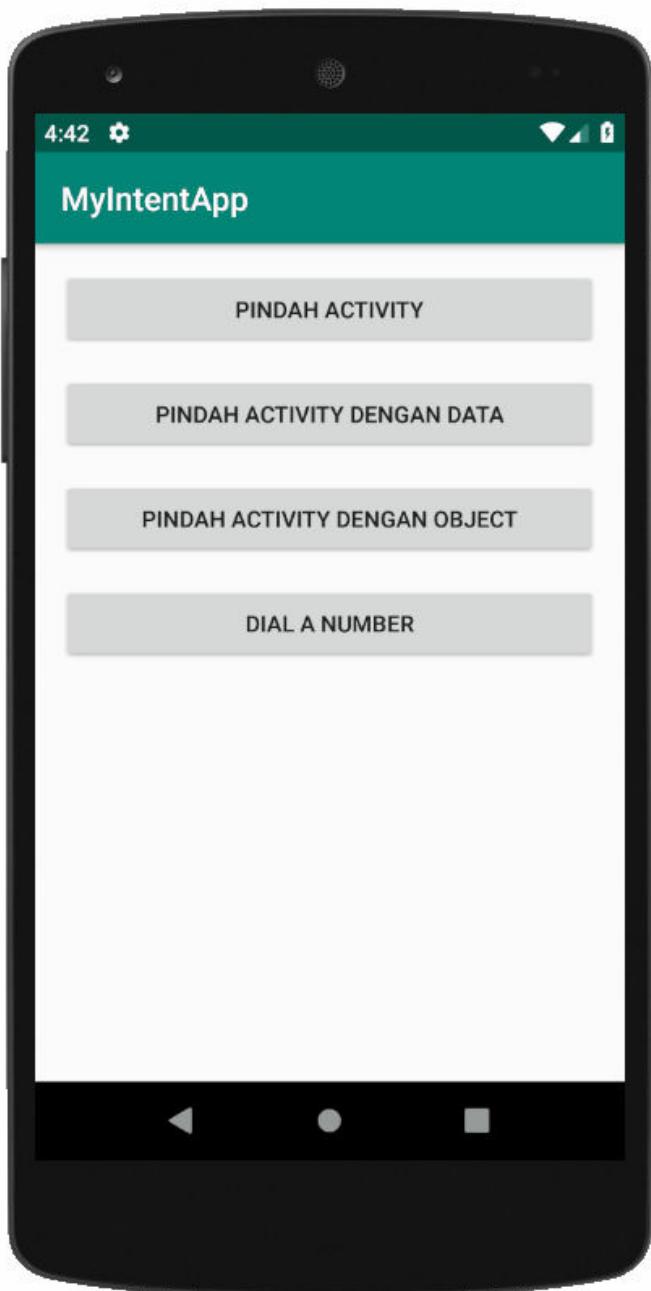
```
1. @Override
2. public void onClick(View v) {
3.     switch (v.getId()){
4.
5.         ...
6.
7.         case R.id.btn_dial_number:
8.             String phoneNumber = "081210841382";
9.             Intent dialPhoneIntent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:"+phoneNumber));
10.            startActivity(dialPhoneIntent);
11.            break;
12.        }
13.    }
```

4. Sehingga kode keseluruhan untuk kelas `MainActivity.java` saat ini adalah.

```
1. public class MainActivity extends AppCompatActivity implements View.OnClickListener{
2.     Button btnMoveActivity;
3.     Button btnMoveWithDataActivity;
4.     Button btnMoveWithObject;
5.     Button btnDialPhone;
6.     @Override
7.     protected void onCreate(Bundle savedInstanceState) {
```

```
8.         super.onCreate(savedInstanceState);
9.         setContentView(R.layout.activity_main);
10.
11.        btnMoveActivity = findViewById(R.id.btn_move_activity);
12.        btnMoveActivity.setOnClickListener(this);
13.
14.        btnMoveWithDataActivity = findViewById(R.id.btn_move_activity_da-
ta);
15.        btnMoveWithDataActivity.setOnClickListener(this);
16.
17.        btnMoveWithObject = findViewById(R.id.btn_move_activity_object);
18.        btnMoveWithObject.setOnClickListener(this);
19.
20.        btnDialPhone = findViewById(R.id.btn_dial_number);
21.        btnDialPhone.setOnClickListener(this);
22.    }
23.
24.    @Override
25.    public void onClick(View v) {
26.        switch (v.getId()){
27.            case R.id.btn_move_activity:
28.                Intent moveIntent = new Intent(MainActivity.this, MoveAc-
tivity.class);
29.                startActivity(moveIntent);
30.                break;
31.            case R.id.btn_move_activity_data:
32.                Intent moveWithDataIntent = new Intent(MainActivity.this
, MoveWithDataActivity.class);
33.                moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_N-
AME, "DicodingAcademy Boy");
34.                moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA_A-
GE, 5);
35.                startActivity(moveWithDataIntent);
36.                break;
37.            case R.id.btn_move_activity_object:
38.                Person person = new Person();
39.                person.setName("DicodingAcademy");
40.                person.setAge(5);
41.                person.setEmail("academy@dicoding.com");
42.                person.setCity("Bandung");
43.
44.                Intent moveWithObjectIntent = new Intent(MainActivity.th-
is, MoveWithObjectActivity.class);
45.                moveWithObjectIntent.putExtra(MoveWithObjectActivity.EXT-
RA_PERSON, person);
46.                startActivity(moveWithObjectIntent);
47.                break;
48.            case R.id.btn_dial_number:
49.                String phoneNumber = "081210841382";
50.                Intent dialPhoneIntent = new Intent(Intent.ACTION_DIAL,
Uri.parse("tel:"+phoneNumber));
51.                startActivity(dialPhoneIntent);
52.                break;
53.        }
54.    }
}
```

- Setelah selesai, silakan jalankan lagi aplikasi anda. Yeay, Anda sudah bisa men-dial sebuah nomor telepon melalui aplikasi bawaan dari peranti anda. Selamat!



Bedah Kode

Intent Implicit

```
1. String phoneNumber = "081210841382";
2.         Intent dialPhoneIntent = new Intent(Intent.ACTION_DIAL, Uri.par
se("tel:"+phoneNumber));
3.         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 konstruktur 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](#)

Apa itu URI (Uniform Resource Identifier)

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

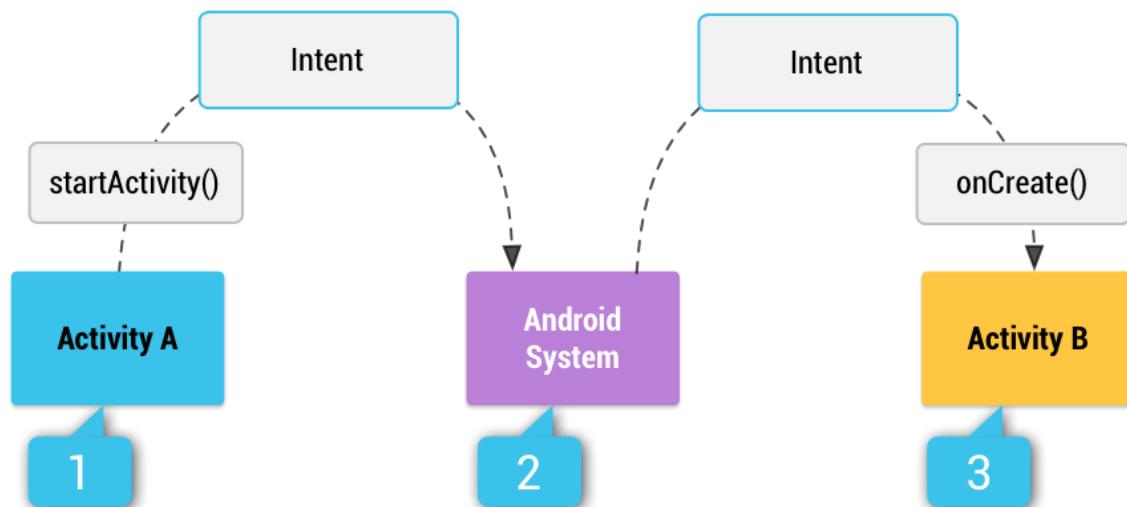
teks

Dimana “tel” adalah sebuah skema yang disepakati untuk sumber daya telepon dan **phoneNumber** adalah variabel string yang bernilai “081210841382”. Skema lain dari Uri seperti “geo” untuk peta, “http” untuk browser sisanya bisa dilihat di halaman ini :

- [Common Intents](#)

Proses Intent Implicit

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!

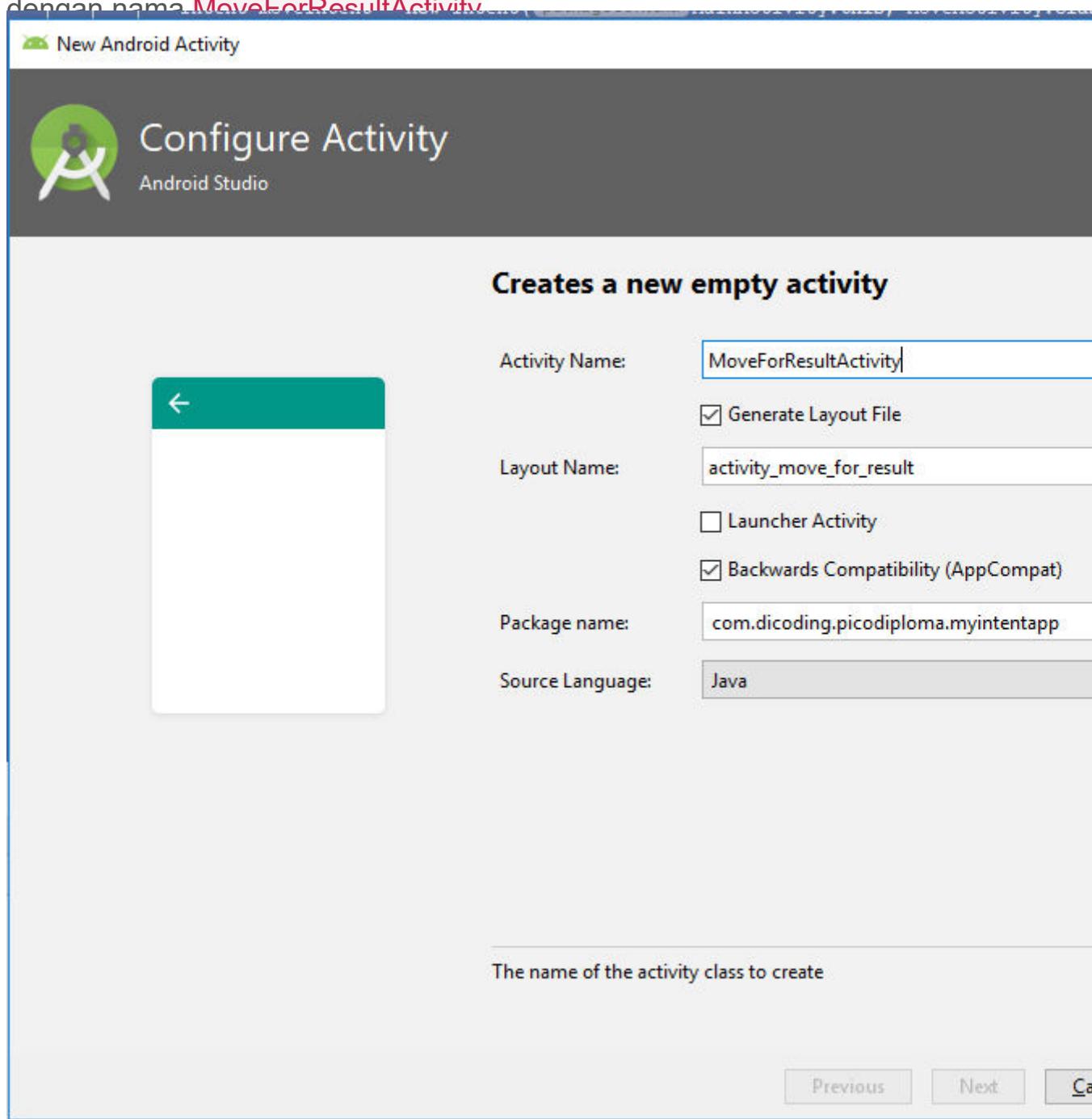
Selanjutnya, kita akan membuat sebuah Intent yang didalamnya akan membawa data dengan **ResultActivity**.

Intent - Latihan : Intent 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 dia jalankan. Untuk memperjelas, ayo kita lanjutkan lagi proses ngodingnya.

1. Sekarang untuk menerapkan konsep diatas, buat sebuah activity baru dengan nama **MoveForResultActivity**



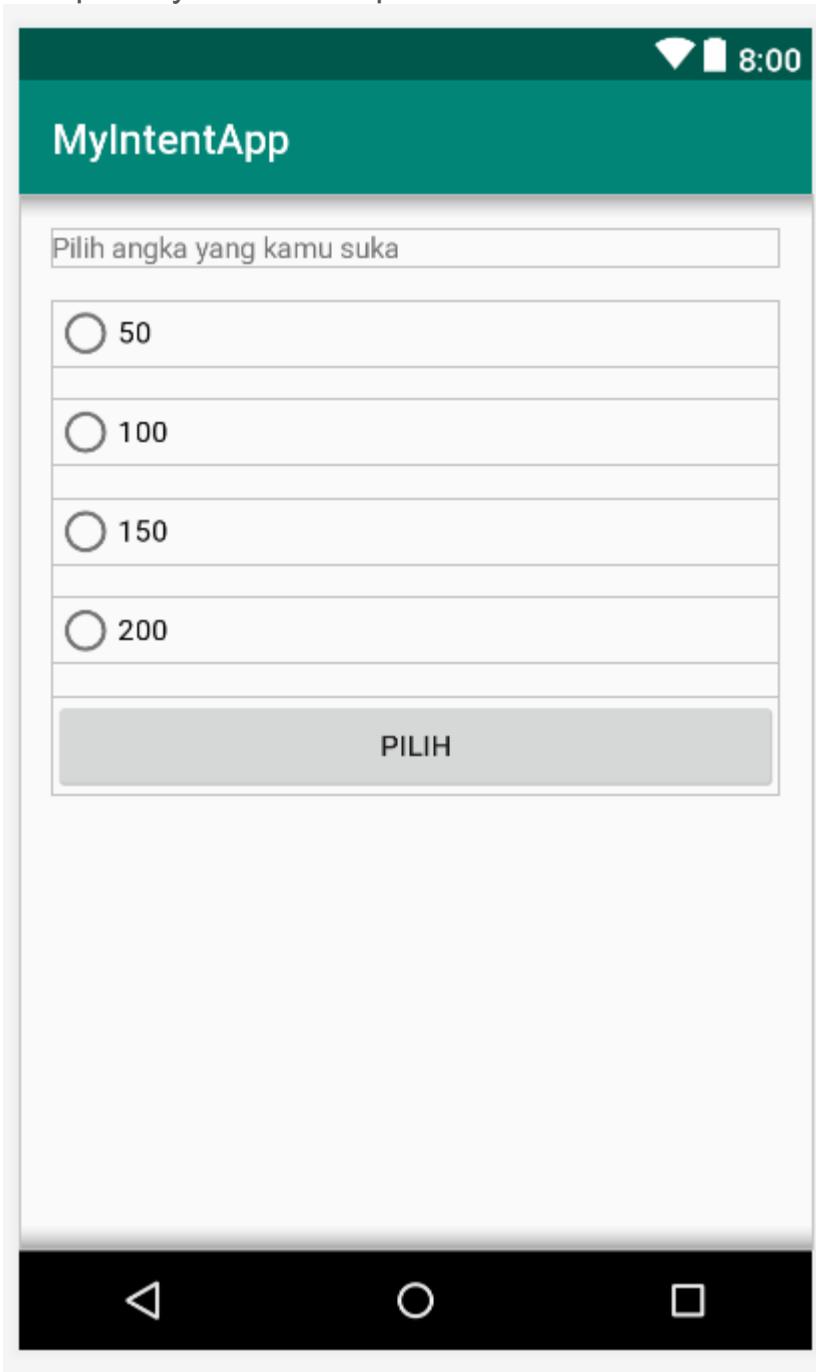
2. Setelah itu, pada **activity_move_for_result.xml** silakan kondisikan layout kita dengan **1 buah TextView, 1 buah Radio Group, 3 buah Radio Button dan 2 buah Button** sebagai berikut :

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

```
5.      android:layout_height="match_parent"
6.      android:orientation="vertical"
7.      android:paddingBottom="@dimen/activity_vertical_margin"
8.      android:paddingLeft="@dimen/activity_horizontal_margin"
9.      android:paddingRight="@dimen/activity_horizontal_margin"
10.     android:paddingTop="@dimen/activity_vertical_margin"
11.     tools:context="com.dicoding.picodiploma.myintentapp.MoveForResultActivity">
12.
13.     <TextView
14.         android:layout_width="match_parent"
15.         android:layout_height="wrap_content"
16.         android:layout_marginBottom="@dimen/activity_vertical_margin"
17.         android:text="@string/choose_number" />
18.
19.     <RadioGroup
20.         android:id="@+id/rg_number"
21.         android:layout_width="match_parent"
22.         android:layout_height="wrap_content"
23.         android:orientation="vertical">
24.
25.         <RadioButton
26.             android:id="@+id/rb_50"
27.             android:layout_width="match_parent"
28.             android:layout_height="wrap_content"
29.             android:layout_marginBottom="@dimen/activity_vertical_margin"
30.             android:text="@string/_50" />
31.
32.         <RadioButton
33.             android:id="@+id/rb_100"
34.             android:layout_width="match_parent"
35.             android:layout_height="wrap_content"
36.             android:layout_marginBottom="@dimen/activity_vertical_margin"
37.             android:text="@string/_100" />
38.
39.         <RadioButton
40.             android:id="@+id/rb_150"
41.             android:layout_width="match_parent"
42.             android:layout_height="wrap_content"
43.             android:layout_marginBottom="@dimen/activity_vertical_margin"
44.             android:text="@string/_150" />
45.
46.         <RadioButton
47.             android:id="@+id/rb_200"
48.             android:layout_width="match_parent"
49.             android:layout_height="wrap_content"
50.             android:layout_marginBottom="@dimen/activity_vertical_margin"
51.             android:text="@string/_200" />
52.     </RadioGroup>
53.
54.     <Button
```

```
55.        android:id="@+id	btn_choose"
56.        android:layout_width="match_parent"
57.        android:layout_height="wrap_content"
58.        android:text="@string/choose" />
59.    </LinearLayout>
```

Tampilannya adalah seperti ini.



3. Setelah selesai, lanjutkan ke berkas **MoveForResultActivity.java** dengan mengenalkan komponen yang sudah di tambahkan

```
1. public class MoveForResultActivity extends AppCompatActivity {  
2.     Button btnChoose;  
3.     RadioGroup rgNumber;  
4.  
5.     public static String EXTRA_SELECTED_VALUE = "extra_selected_value";  
6.     public static int RESULT_CODE = 110;  
7.  
8.     @Override  
9.     protected void onCreate(Bundle savedInstanceState) {  
10.         super.onCreate(savedInstanceState);  
11.         setContentView(R.layout.activity_move_for_result);  
12.  
13.         btnChoose = findViewById(R.id.btn_choose);  
14.         rgNumber = findViewById(R.id.rg_number);  
15.     }  
16. }
```

4. Kemudian kita beri aksi pada button dengan menuliskan kode sebagai berikut :

```
1. public class MoveForResultActivity extends AppCompatActivity implements  
View.OnClickListener {  
2.     ...  
3.     @Override  
4.     protected void onCreate(Bundle savedInstanceState) {  
5.         super.onCreate(savedInstanceState);  
6.         setContentView(R.layout.activity_move_for_result);  
7.         ...  
8.         btnChoose.setOnClickListener(this);  
9.     }  
10.  
11.    @Override  
12.    public void onClick(View v) {  
13.        if (v.getId() == R.id.btn_choose) {  
14.            if (rgNumber.getCheckedRadioButtonId() != 0) {  
15.                int value = 0;  
16.                switch (rgNumber.getCheckedRadioButtonId()) {  
17.                    case R.id.rb_50:  
18.                        value = 50;  
19.                        break;  
20.  
21.                    case R.id.rb_100:  
22.                        value = 100;  
23.                        break;  
24.  
25.                    case R.id.rb_150:
```

```

26.                     value = 150;
27.                 break;
28.
29.             case R.id.rb_200:
30.                 value = 200;
31.                 break;
32.             }
33.
34.             Intent resultIntent = new Intent();
35.             resultIntent.putExtra(EXTRA_SELECTED_VALUE, value);
36.             setResult(RESULT_CODE, resultIntent);
37.             finish();
38.         }
39.     }
40. }
41. }
```

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

```

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

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

```

15. <?xml version="1.0" encoding="utf-8"?>
16. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
17.     xmlns:tools="http://schemas.android.com/tools"
18.     android:layout_width="match_parent"
19.     android:layout_height="match_parent"
20.     android:orientation="vertical"
21.     android:paddingBottom="@dimen/activity_vertical_margin"
22.     android:paddingLeft="@dimen/activity_horizontal_margin"
23.     android:paddingRight="@dimen/activity_horizontal_margin"
24.     android:paddingTop="@dimen/activity_vertical_margin"
```

```
25.     tools:context="com.dicoding.picodiploma.myintentapp.MainActivity">
26.
27.     <Button
28.         android:id="@+id	btn_move_activity"
29.         android:layout_width="match_parent"
30.         android:layout_height="wrap_content"
31.         android:layout_marginBottom="@dimen/activity_vertical_margin"
32.         android:text="@string/move_activity" />
33.
34.     <Button
35.         android:id="@+id	btn_move_activity_data"
36.         android:layout_width="match_parent"
37.         android:layout_height="wrap_content"
38.         android:layout_marginBottom="@dimen/activity_vertical_margin"
39.         android:text="@string/move_with_data" />
40.
41.     <Button
42.         android:id="@+id	btn_move_activity_object"
43.         android:layout_width="match_parent"
44.         android:layout_height="wrap_content"
45.         android:layout_marginBottom="@dimen/activity_vertical_margin"
46.         android:text="@string/move_with_object" />
47.
48.     <Button
49.         android:id="@+id	btn_dial_number"
50.         android:layout_width="match_parent"
51.         android:layout_height="wrap_content"
52.         android:layout_marginBottom="@dimen/activity_vertical_margin"
53.         android:text="@string/dial_number" />
54.
55.     <Button
56.         android:id="@+id	btn_move_for_result"
57.         android:layout_width="match_parent"
58.         android:layout_height="wrap_content"
59.         android:layout_marginBottom="@dimen/activity_vertical_margin"
60.         android:text="@string/move_with_result" />
61.
62.     <TextView
63.         android:id="@+id/tv_result"
64.         android:layout_width="match_parent"
65.         android:layout_height="wrap_content"
66.         android:gravity="center"
67.         android:text="@string/result_from_activity"
68.         android:textSize="24sp" />
69. </LinearLayout>
```

6. Setelah selesai, kita lanjut mengenalkan **TextView** dan **Button** kedalam berkas **MainActivity.java**.

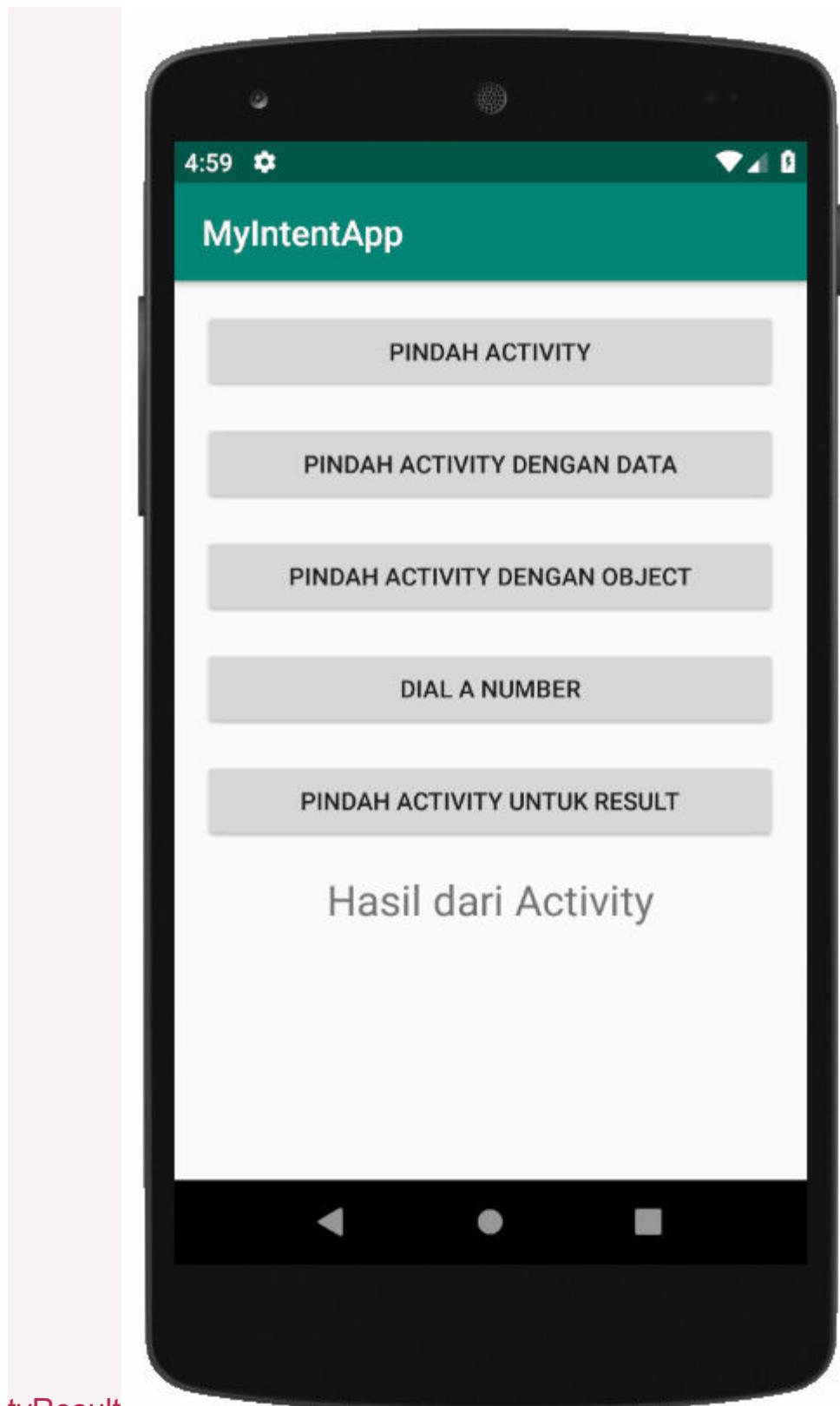
```
1. public class MainActivity extends AppCompatActivity implements View.OnClickListener{
2.     Button btnMoveActivity;
3.     Button btnMoveWithDataActivity;
4.     Button btnMoveWithObject;
5.     Button btnDialPhone;
6.     Button btnMoveForResult;
7.     TextView tvResult;
8.
9.     private int REQUEST_CODE = 100;
10.
11.    @Override
12.    protected void onCreate(Bundle savedInstanceState) {
13.        super.onCreate(savedInstanceState);
14.        setContentView(R.layout.activity_main);
15.
16.        ...
17.
18.        btnMoveForResult = findViewById(R.id.btn_move_for_result);
19.        btnMoveForResult.setOnClickListener(this);
20.
21.        tvResult = findViewById(R.id.tv_result);
22.    }
23.    ...
24. }
```

7. Selanjutnya kita beri aksi pada **button** dengan menyesuaikan kodennya sebagai berikut.

```
1. ...
2.
3. @Override
4. public void onClick(View v) {
5.     switch (v.getId()) {
6.         ...
7.         case R.id.btn_move_for_result:
8.             Intent moveForResultIntent = new Intent(MainActivity.this,
9.                 MoveForResultActivity.class);
10.            startActivityForResult(moveForResultIntent, REQUEST_CODE);
11.        }
12.    }
13.
14.
15. @Override
16. protected void onActivityResult(int requestCode, int resultCode, Intent data) {
17.     super.onActivityResult(requestCode, resultCode, data);
18. }
```

```
19.         if (requestCode == REQUEST_CODE) {
20.             if (resultCode == MoveForResultActivity.RESULT_CODE) {
21.                 int selectedValue = data.getIntExtra(MoveForResultActivi
ty.EXTRA_SELECTED_VALUE, 0);
22.                 tvResult.setText(String.format("Hasil : %s", selectedVal
ue));
23.             }
24.         }
25.     }
```

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



textview **tvResult**.

Bedah Kode

startActivity dan startActivityForResult

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)**.

```
1. Intent moveForResultIntent = new Intent(MainActivity.this, MoveForResultActivity.class);
2. 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.

Result Activity

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.

```
1. @Override
2. public void onClick(View v) {
3.     if (v.getId() == R.id.btn_choose){
4.         if (rgNumber.getCheckedRadioButtonId() != 0){
5.             int value = 0;
6.             switch (rgNumber.getCheckedRadioButtonId()){
7.                 case R.id.rb_50:
8.                     value = 50;
9.                     break;
10.                case R.id.rb_100:
11.                    value = 100;
12.                    break;
13.                case R.id.rb_150:
14.                    value = 150;
15.                    break;
16.                case R.id.rb_200:
17.                    value = 200;
18.                    break;
19.            }
20.            Intent resultIntent = new Intent();
21.            resultIntent.putExtra(EXTRA_SELECTED_VALUE, value);
22.            setResult(RESULT_CODE, resultIntent);
23.            finish();
```

```
24.     }
25. }
26. }
```

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()`.

```
1. <RadioGroup
2.     android:id="@+id/rg_number"
3.     android:layout_width="match_parent"
4.     android:layout_height="wrap_content"
5.     android:orientation="vertical">
6.     <RadioButton
7.         android:id="@+id/rb_50"
8.         android:layout_width="match_parent"
9.         android:layout_height="wrap_content"
10.        android:text="50"
11.        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
12.     <RadioButton
13.        android:id="@+id/rb_100"
14.        android:layout_width="match_parent"
15.        android:layout_height="wrap_content"
16.        android:text="100"
17.        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
18.     <RadioButton
19.        android:id="@+id/rb_150"
20.        android:layout_width="match_parent"
21.        android:layout_height="wrap_content"
22.        android:text="150"
23.        android:layout_marginBottom="@dimen/activity_vertical_margin"/>
24.     <RadioButton
25.         android:id="@+id/rb_200"
26.         android:layout_width="match_parent"
27.         android:layout_height="wrap_content"
28.         android:text="200"
29.         android:layout_marginBottom="@dimen/activity_vertical_margin"/>
30. </RadioGroup>
```

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

```
1. Intent resultIntent = new Intent();
2. resultIntent.putExtra(EXTRA_SELECTED_VALUE, value);
3. setResult(RESULT_CODE, resultIntent);
```

```
4. finish();
```

Kita membuat sebuah intent tanpa ada inputan apapun di konstruktornya. Kemudian kita meletakkan variabel `value` ke dalam 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.

```
1. @Override
2. protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
3.     super.onActivityResult(requestCode, resultCode, data);
4.     if (requestCode == REQUEST_CODE){
5.         if (resultCode == MoveForResultActivity.RESULT_CODE){
6.             int selectedValue = data.getIntExtra(MoveForResultActivity.EXTRA_SE
LECTED_VALUE, 0);
7.             tvResult.setText("Hasil : "+selectedValue);
8.         }
9.     }
10. }
```

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`.

Kombinasi antara `requestCode` dan `resultCode` akan sangat berguna ketika kita ingin membuat suatu activity yang dapat merespon banyak skenario . Perhatikan contoh kode di bawah ini, di mana terjadi banyak skenario yang bisa terjadi.

```
1. protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
2.     super.onActivityResult(requestCode, resultCode, data);
3.
4.     if (requestCode == REQUEST_1) {
5.         if (resultCode == RESULT_YES) {
6.
7.             } else if (resultCode == RESULT_NO) {
```

```
8.
9.        }
10.    } else if (requestCode == REQUEST_2) {
11.        if (resultCode == RESULT_OK) {
12.
13.            } else if (resultCode == RESULT_CANCELED) {
14.
15.                } else {
16.                    // do nothing
17.                }
18.            } else {
19.                // do something
20.            }
21.        }
```

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.

Untuk lebih dalam lagi silakan pelajari materi pada link berikut:

- [Intent Result](#)

Keren! Anda sudah belajar dasar-dasar penggunaan intent secara umum di proyek aplikasi Android. 4 intent secara eksplisit dan 1 secara implisit dengan satu di antaranya mencakup nilai balik dari activity yang dijalankan.

Untuk mendalami materi intent, silakan meluncur ke link di bawah ini.

- [Intent Filters](#)

Untuk source code materi bisa Anda unduh di tautan berikut ini :

<https://github.com/dicodingacademy/a14-made-labs1/tree/master/02.Intent/MyIntentApp>

