

BAB I

PEMASANGAN DAN KONFIGURASI ANDROID STUDIO

A. TINJAUAN TEORI

Berkenalan dengan Android

Android adalah sistem operasi yang dikeluarkan oleh Google. Sistem operasi yang disimbolkan dengan robot hijau itu dibuat khusus untuk lingkungan pengembangan *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 bulannya, per Januari 2018.



Gambar 1.1. *Smartphone* Android

Android dibangun atas dasar sistem operasi Linux. Sistem operasi yang bebas virus komputer ini memiliki keamanan yang tinggi. Seluruh aplikasi android berjalan sesuai dengan proses-proses linux. Android memanfaatkan fitur-fitur yang ada di sistem operasi linux untuk memberi dukungan manajemen memori, manajemen sumber daya dan manajemen jaringan.

Sejarah Android

Android pada mulanya adalah sebuah perusahaan yang didirikan pada tahun 2003, dibawah naungan Android, Inc. Hingga pada saat ini di kelola oleh salah satu perusahaan IT terbesar di dunia yaitu google. Hingga saat tulisan ini dibuat, versi dari android sudah sampai 9 (code name : Pie). Perlu diketahui bahwa code name android urut dengan abjad dan namanya merupakan nama makanan-makanan manis. Lebih lengkap mengenai versi-versi android, perhatikan Tabel 1 berikut.

Tabel 1. Daftar Versi Android

Version	Code Name	Release Date	API Level	New Features	Icon
1.5	Cupcake	27 April 2009	3	-	
1.6	Donut	15 Sept 2009	4	<ul style="list-style-type: none"> • Quick Search Box • Screen Size Diversity • Android Market 	
2.0 – 2.1	Eclair	26 Okt 2009	5	<ul style="list-style-type: none"> • Google Maps Navigation • Home Screen Customization • Speech-to-Text 	
2.2 – 2.2.3	Froyo	20 Mei 2010	8	<ul style="list-style-type: none"> • Voice Action • Portable Hotspot • Dalvik JIT 	
2.3 – 2.3.7	Gingerbread	9 Februari 2011	9-10	<ul style="list-style-type: none"> • Gaming APIs • NFC • Battery Mangement 	
3.0 – 3.2.6	HoneyComb	22 Februari 2011	11-13	<ul style="list-style-type: none"> • Table-Friendly Design • System Bar • Quick Settings 	
4.0 – 4.6	Ice Cream Sandwich	19 Oktober 2011	15	<ul style="list-style-type: none"> • Custom Home Screen • Data Usage Control • Android Beam 	
4.1 – 4.3.1	Jelly Bean	9 Juli 2012	18	<ul style="list-style-type: none"> • Google Now • Actionable Notification • Account Switching 	
4.4 – 4.4.4	Kitkat	31 Okt 2013	19-20	<ul style="list-style-type: none"> • Voice : OK Google • Immersive Design • Smart Dialer 	

5.1 – 5.1.1	Lollipop	12 Nov 2014	21-22	<ul style="list-style-type: none"> • Material Design • Multiscreen • Notifications 	
6.0 – 6.0.1	Marshmallow	5 Okt 2015	23	<ul style="list-style-type: none"> • Now On Top • Permissions • Battery (Doze & App Standby) 	
7.1 – 7.1.2	Nougat	22 Agustus 2016	24-25	<ul style="list-style-type: none"> • Multi Window • GIF Keyboard 	
8.0 – 8.1	Oreo	25 Oktober 2017	26-27	<ul style="list-style-type: none"> • Picture-in-Picture 	
9	Pie	6 Agustus 2018	28	<ul style="list-style-type: none"> • Adaptive Battery • Adaptive Brightness 	

Mengapa Android

Android memanjakan penggunanya dengan fitur yang sangat canggih dan tampilan yang bagus. Sistem Android dapat digunakan sebagai alat multimedia seperti pemutar musik dan video. Ia juga memiliki perangkat keras seperti accelerometer, gyroscope dan sensor lainnya. Di samping itu ada beberapa hal yang membuat Android menjadi sistem operasi yang memang layak digunakan oleh pengguna atau dikembangkan para developer, seperti yang akan diuraikan berikut ini.

1) Sistem Operasi *Smartphone* Terpopuler

Sejak dirilis tahun 2008, Google telah mengeluarkan beberapa versi, dengan "Pie" sebagai versi yang terbaru. Pada tahun 2013, Android menjadi operation system (OS) terlaris pada tablet dan smartphone. Kini market share Android sedikitnya 80 % dari total penjualan smarphone di tingkat global (statista.com). Tercatat pada tahun 2016 Android *store* 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.

2) Adanya Store

Aplikasi Android bisa didistribusikan menggunakan web, copy APK, dan *store*. Android store, yaitu Google Play merupakan cara termudah bagi para developer untuk mendistribusikan aplikasinya ke pasar dengan 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.



Gambar 1.2. Logo Play Store

Google Play tak hanya menawarkan aplikasi. 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*) terbaik.

3) 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 menetapkan Kotlin sebagai tambahan bahasa resmi.

Berbicara tentang pemrograman tentunya tak 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.

B. TUJUAN

Pada pertemuan ini, tujuan yang harus dicapai antara lain

1. Mahasiswa memahami sistem operasi android, mulai dari sejarah hingga lingkungan pengembangannya
2. Mahasiswa mengetahui *tools* yang digunakan untuk membangun aplikasi android
3. Mahasiswa mempraktekan cara membuat proyek android
4. Mahasiswa dapat membuat aplikasi sederhana dengan android studio

C. PERALATAN YANG DIGUNAKAN

Pengembangan aplikasi android membutuhkan peralatan baik perangkat keras maupun perangkat lunak. Adapun perangkat keras yang digunakan untuk membangun aplikasi android adalah sebagai berikut.

1. Laptop / Personal Komputer dengan spesifikasi minimal seperti berikut.
 - ✓ Prosesor minimal Intel Core i3 Generasi kelima, atau prosesor AMD A8
 - ✓ RAM minimum 3 GB, RAM yang disarankan 8 GB,
 - ✓ Ruang disk minimum yang tersedia 2 GB,
 - ✓ Resolusi layar minimum 1280 x 800
2. *Smartphone* android yang memiliki versi minimal Ice Cream Sandwich (Android 4.0)
3. Kabel data

Sedangkan perangkat lunak yang harus terpasang di Laptop/PC antara lain

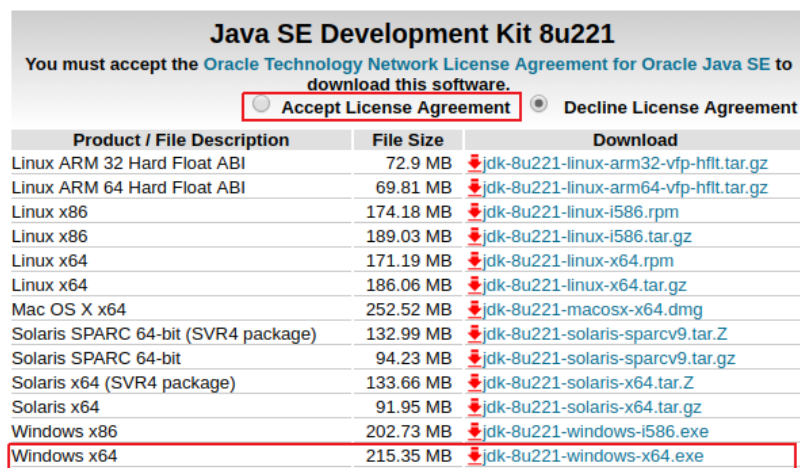
1. Sistem operasi windows (disarankan windows 10), Linux (disarankan Linux Mint), Macintosh (disarankan versi 10, yosemite)
2. Java Development Kit (JDK) 8 atau versi terbaru
3. Android Studio

D. LANGKAH PRAKTIKUM

1.1. PEMASANGAN JAVA DEVELOPMENT KIT (JDK)

Sebelum membangun aplikasi android, hal pertama yang harus dilakukan adalah memasang perangkat lunak Java Development Kit (JDK). Sebab aplikasi android yang akan dibuat menggunakan bahasa pemrograman java. JDK merupakan mesin virtual yang digunakan untuk meng-*compile* berkas java. JDK dapat berjalan di berbagai platform sistem operasi, seperti windows, linux, maupun macintosh.

JDK dapat diperoleh secara gratis dengan mengunduh pada situs resminya di <http://oracle.com>. Sesuaikan JDK yang akan diunduh dengan sistem operasi yang digunakan. Gambar 1.3 adalah langkah-langkah mengunduh JDK dari situs resminya.



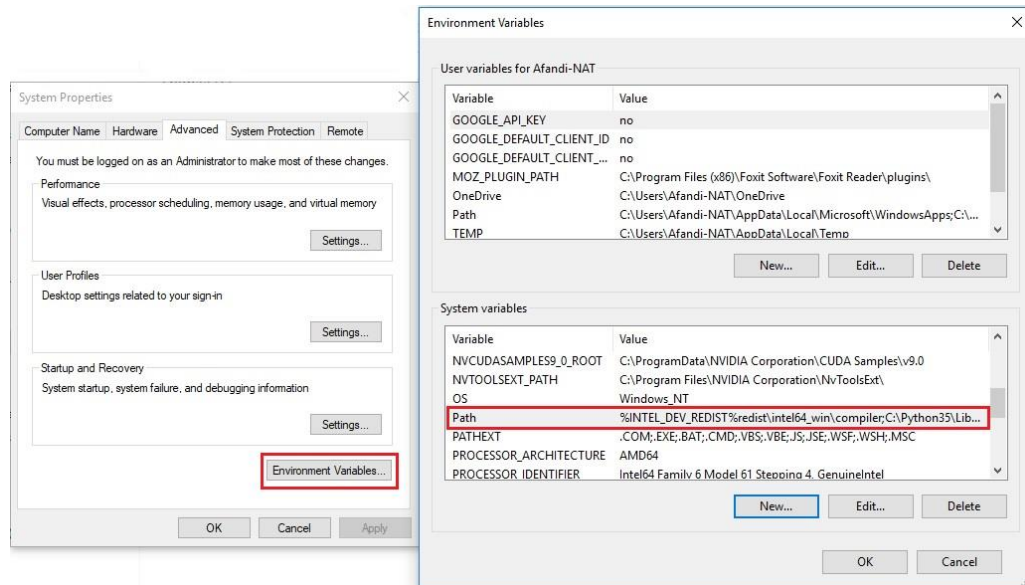
Java SE Development Kit 8u221		
You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software.		
<input checked="" type="radio"/> Accept License Agreement <input type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.9 MB	jdk-8u221-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.81 MB	jdk-8u221-linux-arm64-vfp-hflt.tar.gz
Linux x86	174.18 MB	jdk-8u221-linux-i586.rpm
Linux x86	189.03 MB	jdk-8u221-linux-i586.tar.gz
Linux x64	171.19 MB	jdk-8u221-linux-x64.rpm
Linux x64	186.06 MB	jdk-8u221-linux-x64.tar.gz
Mac OS X x64	252.52 MB	jdk-8u221-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	132.99 MB	jdk-8u221-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	94.23 MB	jdk-8u221-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	133.66 MB	jdk-8u221-solaris-x64.tar.Z
Solaris x64	91.95 MB	jdk-8u221-solaris-x64.tar.gz
Windows x86	202.73 MB	jdk-8u221-windows-i586.exe
Windows x64	215.35 MB	jdk-8u221-windows-x64.exe

Gambar 1.3. Cara mengunduh JDK

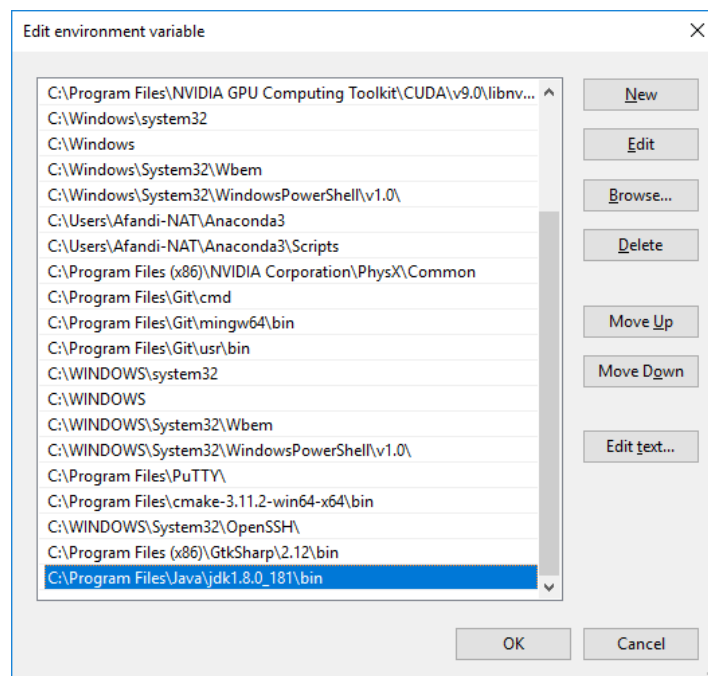
1. Pemasangan JDK di Sistem Operasi Windows

Setelah JDK selesai diunduh, langkah selanjutnya adalah memasang JDK dengan cara klik 2x pada berkas `jdk-8u221-windows-x64.exe`. Setelah itu ikuti langkah-langkah pemasangan JDK. Setelah selesai, atur PATH pada sistem windows. Caranya buka Control Panel, pilih System and Security, pilih System, klik Environment Variables.

Selanjutnya dalam Environment Variables, pada bagian System variables pilih path, kemudian klik new. Masukkan lokasi compiler JDK yang terpasang di sistem windows. Biasanya lokasi compiler tersebut ada `C:/Program Files/Java/[berkas_file_java]/bin`. Setelah itu klik OK dan restart PC/Laptop. Adapaun langkah-langkah dalam penambahan path java pada sistem windows, ditunjukkan pada Gambar 1.4 dan Gambar 1.5.

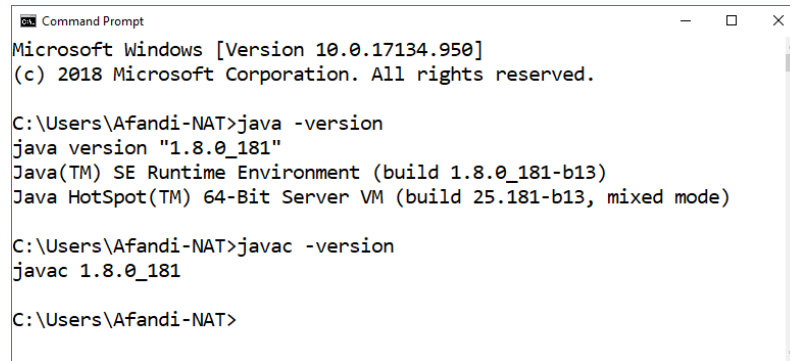


Gambar 1.4. Langkah Setting Path Java



Gambar 1.5. Menambahkan Path Java

Bukti jika JDK sudah terpasang pada komputer adalah dengan menjalankan perintah “java –version” untuk melihat versi java atau ketikkan perintah “javac –version” untuk melihat versi compiler. Apabila berhasil maka hasil dari perintah tersebut seperti ditunjukkan Gambar 1.6.



```
Microsoft Windows [Version 10.0.17134.950]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Afandi-NAT>java -version
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)

C:\Users\Afandi-NAT>javac -version
javac 1.8.0_181

C:\Users\Afandi-NAT>
```

Gambar 1.6. Cek Versi Java

2. Pemasangan JDK di Linux

Langkah-langkah pemasangan JDK pada linux (pada buku ini ubuntu 16.04) adalah sebagai berikut:

1. Buka terminal, lakukan update untuk memperbaharui Ubuntu Anda menjadi up-to-date, dengan perintah : **sudo apt-get update**
2. Kemudian, lakukan perintah di bawah ini untuk menginstal JDK : **sudo apt-get install default-jdk**
3. Jika ada pertanyaan “Do you want to continue ?” ketik “Y”
4. Selanjutnya install JRE dengan perintah : **sudo apt-get install default-jre**

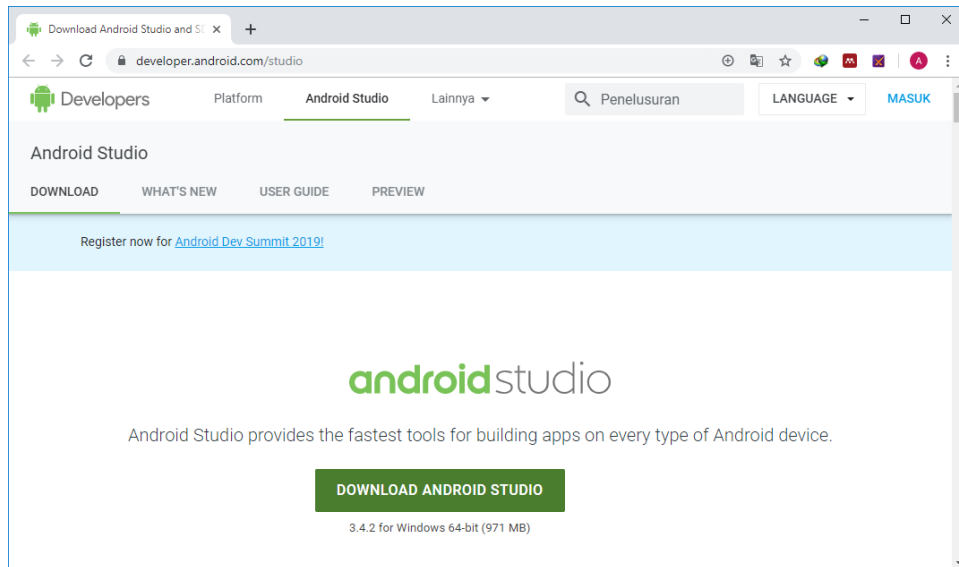
Pemasangan JRE bersifat optional, karena didalam JDK sudah terdapat JRE. JDK merupakan paket pemasangan Java yang biasanya dibutuhkan oleh programmer dengan memanfaatkan compiler yang terdapat di dalamnya untuk menjalankan *bytecode* dari program. Pada JDK juga terdapat Java Runtime untuk menjalankan aplikasi. Berbeda dengan JRE (*Java Runtime Environment*) yang hanya dapat dimanfaatkan untuk menjalankan aplikasi berbasis Java. JRE tidak terdapat compiler yang dapat menjalankan *bytecode* saat Anda menulis sebuah program.

1.2. PEMASANGAN ANDROID STUDIO

Sebelum memasang android studio, pastikan spesifikasi PC/Laptop anda sesuai dengan kebutuhan hardware minimal. Sebab jika spesifikasi kurang maka akan sangat berat pada saat menjalankan emulator. Bahkan dapat terjadi *crash* saat menggunakan android studio. Oleh karena itu spesifikasi *hardware* minimal dari android studio harus sesuai dengan persiapan peralatan yang ada pada poin C diatas.

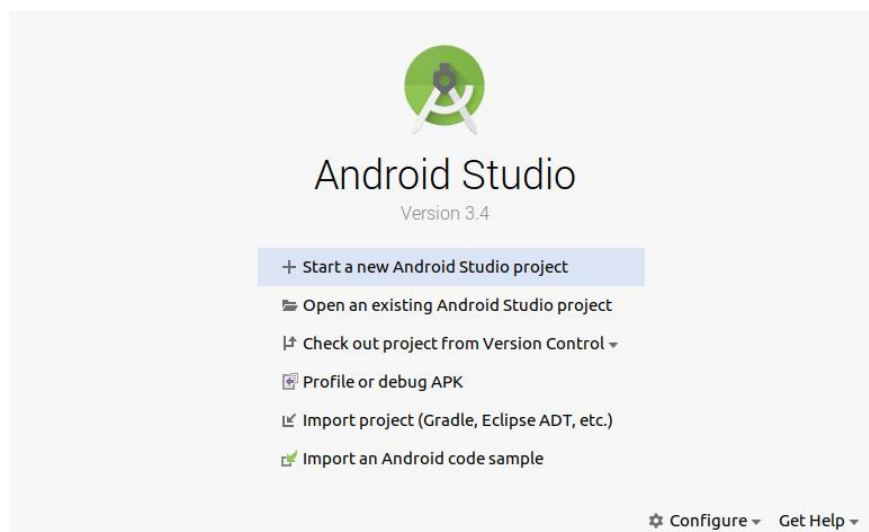
Android studio adalah *Integrated Development Environment* (IDE) untuk pengembangan aplikasi android. IDE buatan google ini dapat diperoleh secara gratis di situs <http://developer.android.com/studio>. Fitur-fitur yang terdapat di android studio

memungkinkan pengembang untuk membangun aplikasi android dengan mudah dan *compatible* dengan versi android yang saat ini berkembang di pasaran.



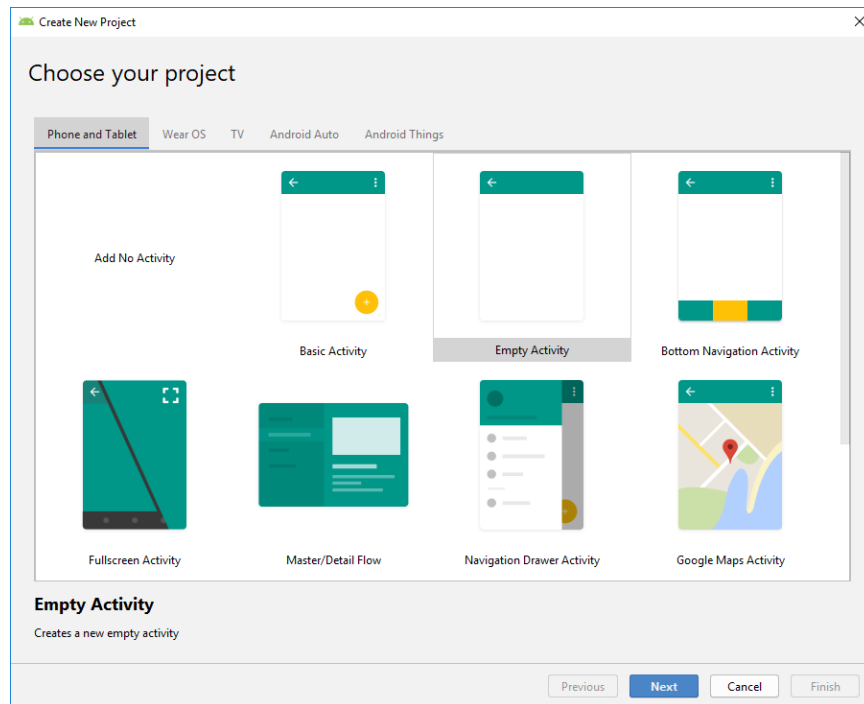
Gambar 1.7. Tampilan Lokasi Unduh Android Studio

Gambar 1.7 adalah situs yang digunakan untuk mengunduh android studio. Secara otomatis situs developer.android.com adalah mengidentifikasi sistem operasi yang digunakan pengunjung. Sehingga pengunjung hanya tinggal klik button download saja. Pemasangan android studio di windows maupun linux hampir sama, hanya saja di sistem operasi linux membutuhkan sebuah perintah untuk menjalankan berkas instalansi nya.



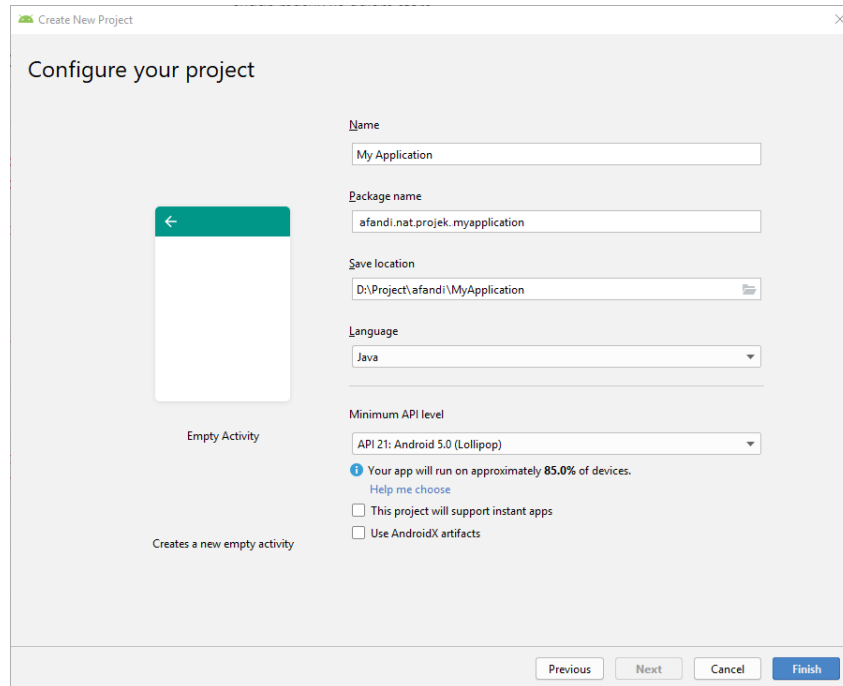
Gambar 1.8. Tampilan Awal Android Studio

Setelah android studio berhasil terpasang di komputer, maka tampilan awal dari android studio ditunjukkan seperti Gambar 1.8. Kemudian klik Start a new Android Studio Project untuk memulai membangun aplikasi android.



Gambar 1.9. Tampilan Layout Android

Saat klik New Android Project, kita akan dibawa ke halaman pilihan perangkat yang ingin kita bangun. Android studio sudah menyediakan beragam tipe perangkat android yang sesuai kebutuhan pasar. Gambar 1.9 juga menunjukkan beragam template yang dapat digunakan untuk membuat aplikasi android. Mulai dari Empty Activity, Basic Activity, Navigation Drawer Activity, dll.



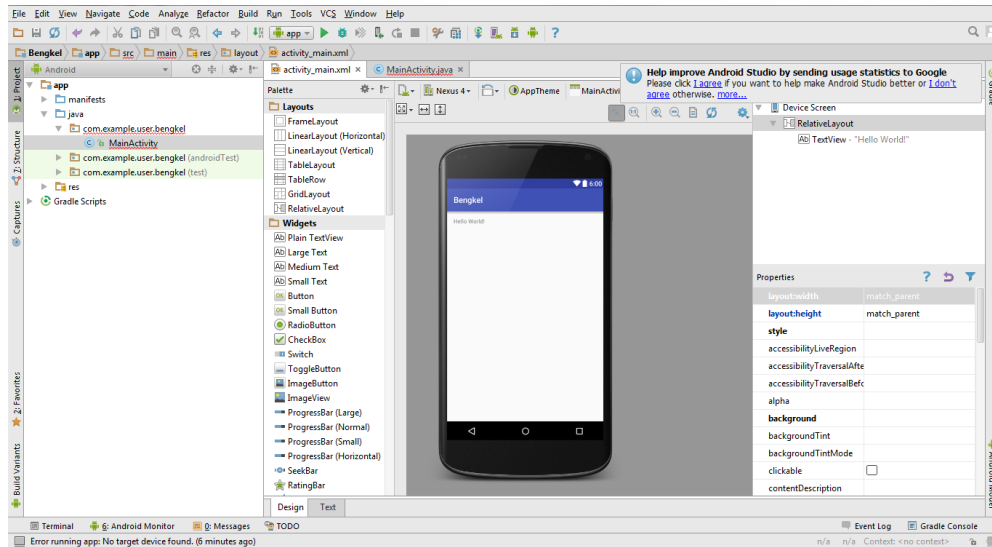
Gambar 1.10. Tampilan Konfigurasi Proyek Android

Setelah itu klik next untuk melanjutkan pembuatan proyek android. Gambar xx adalah tampilan akhir untuk mengkonfigurasi proyek android yang hendak dibuat. Pada Gambar 1.10 pengembang diharuskan mengisi nama dari aplikasi, lokasi proyek dan nama package. Pemberian nama package akan digunakan dalam identifikasi unik dari aplikasi ketika sudah di-publish.

Pengembang juga dapat mengganti nilai minimum SDK, yang berfungsi untuk membatasi penggunaan API pada sebuah aplikasi. Di dalam dialog ini pengembang juga bisa memberi nama dari Activity yang pertama kali yang dibuat. Selain itu juga dapat mengganti bahasa default pada aplikasi tersebut.

1.2.1. Antarmuka Android Studio

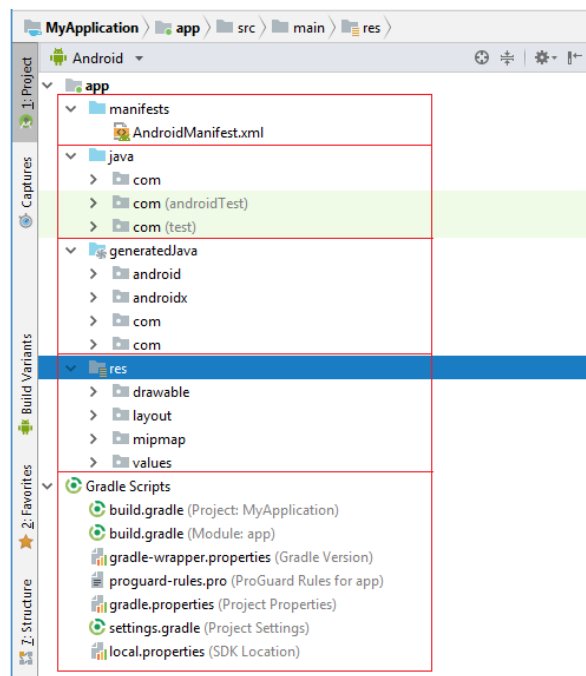
Tampilan antarmuka android studio sudah disesuaikan dengan lingkungan pemrograman *mobile*. Sehingga mendukung produktifitas pengembang dalam membangun aplikasi android. Direktori untuk bahasa pemrograman java dan berkas xml dipisahkan untuk menjembatani proses coding fungsi dan desain antarmuka aplikasi. Gambar 1.11 adalah *screenshot* tampilan penuh IDE Android Studio berbasis IntelliJ IDEA.



Gambar 1.11. Tampilan IDE Android Studio

1.2.2. Struktur Proyek

Setiap proyek di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file sumber daya. Secara default, Android Studio akan menampilkan berkas proyek Anda dalam tampilan proyek Android, seperti yang ditampilkan dalam Gambar 1.12.



Gambar 1.12. Stuktur Direktori Android Studio

1. Android Manifest

Manifest adalah salah satu berkas yang harus ada di dalam sebuah proyek Android. Manifest akan memberikan beragam informasi penting kepada sistem Android. Sistem perlu mengetahui apa yang akan digunakan oleh aplikasi sebelum dijalankan.

Beberapa fungsi yang ada di dalam manifest adalah sebagai berikut.

- Nama Package

Package name merupakan sebagai identitas unik dari sebuah aplikasi. Identitas ini juga yang akan digunakan di dalam *store* untuk membedakan suatu aplikasi dengan aplikasi lainnya. Jangan pernah mengganti *value* di dalam package karena nantinya akan dikenali sebagai aplikasi yang lain, jika sudah masuk ke dalam *store*.

```
1. <manifest
  xmlns:android="http://schemas.android.com/apk/res/android"
2.   package="afandi.nat.projek.myapplication">
```

- Komponen Aplikasi

Berfungsi untuk mendeskripsikan komponen dari aplikasi mulai dari activity, services, broadcast receiver, dan content provider.

```
1. <application
2.   android:allowBackup="true"
3.   android:icon="@mipmap/ic_launcher"
4.   android:label="@string/app_name"
5.   android:roundIcon="@mipmap/ic_launcher_round"
6.   android:supportsRtl="true"
7.   android:theme="@style/AppTheme">
8.   <activity android:name=".MainActivity">
9.     <intent-filter>
10.      <action android:name="android.intent.action.MAIN" />
11.
12.      <category
13.        android:name="android.intent.category.LAUNCHER" />
14.    </intent-filter>
15.  </activity>
16.  <service
17.    android:name=".MyIntentService"
18.    android:exported="false" />
19.
20.  <receiver
21.    android:name=".MyReceiver"
22.    android:enabled="true"
23.    android:exported="true"></receiver>
24.</application>
```

- Permission

Pada berkas manifest juga digunakan untuk memberikan hak akses (permission) kepada aplikasi untuk mengakses sebuah layanan tertentu. Sebagai contoh layanan

yang biasanya ditambahkan pada berkas manifest ini adalah internet, external storage, contact, dan juga komunikasi dengan aplikasi lainnya.

2. Java

Merupakan struktur direktori yang berisi berkas *source code* kita yang ditulis dalam bahasa Java, termasuk juga kode Unit Test dan androidTest (Instrumentation Test). Berkas yang terbentuk pada direktori java ini akan disesuaikan dengan package yang pada awal proyek kita isi.

3. GeneratedJava

Berisi file hasil dari generate library atau sebuah kelas dari proyek Android.

4. Res / Resources

Berisi kode xml yang digunakan untuk mengatur layout dari aplikasi android. Pada direktori ini juga terdapat gambar, ikon, hingga style yang diperlukan untuk mempercantik layout. Secara default direktori **res** akan membuat direktori drawable, layout, mipmap, dan launcher. Fungsi dari masing-masing direktori adalah sebagai berikut

- Direktori drawable digunakan untuk menyimpan gambar atau icon
- Direktori layout digunakan untuk menyimpan berkas xml yang akan dipanggil pada program utama java
- Direktori mipmap, fungsinya hampir sama seperti drawable tetapi khusus untuk menyimpan ikon/logo
- Direktori value, digunakan untuk menyimpan style yang berfungsi untuk mempercantik tampilan aplikasi

5. Gradle

Gradle merupakan *open source build automation system*. *Automation system* berguna untuk mengotomatisasi proses pembuatan dari *software build* dan proses-proses terkait lainnya termasuk *compile source code* menjadi *binary code*, *packaging binary code*, dan menjalankan *automated test*.

- **build.gradle (Project: MyApplication)**

Merupakan *software build* tingkat teratas yang meliputi keseluruhan dari proyek dari sebuah aplikasi. Di dalamnya berisi konfigurasi semua modul yang ada di dalam proyek.

- **build.gradle (Module: App)**

Merupakan *software build* yang ada pada setiap modul di dalam proyek sebuah aplikasi. Beberapa konfigurasi di antaranya adalah **android settings**, **defaultConfig** dan **productFlavors**, **buildTypes**, dan **dependencies**.

- ✓ Android Settings

Di dalam block android kita bisa menetapkan `compileSdkVersion` dan `buildToolsVersion`.

- ✓ Default Config

Di dalamnya terdapat properties seperti `applicationID`, `minSdkVersion`, `targetSdkVersion` dan test information.

- ✓ Build Types

Di dalamnya terdapat properties dari `debuggable`, `ProGuard enabling`, `debug signing`, `version name suffix` dan test information.


- ✓ Dependencies

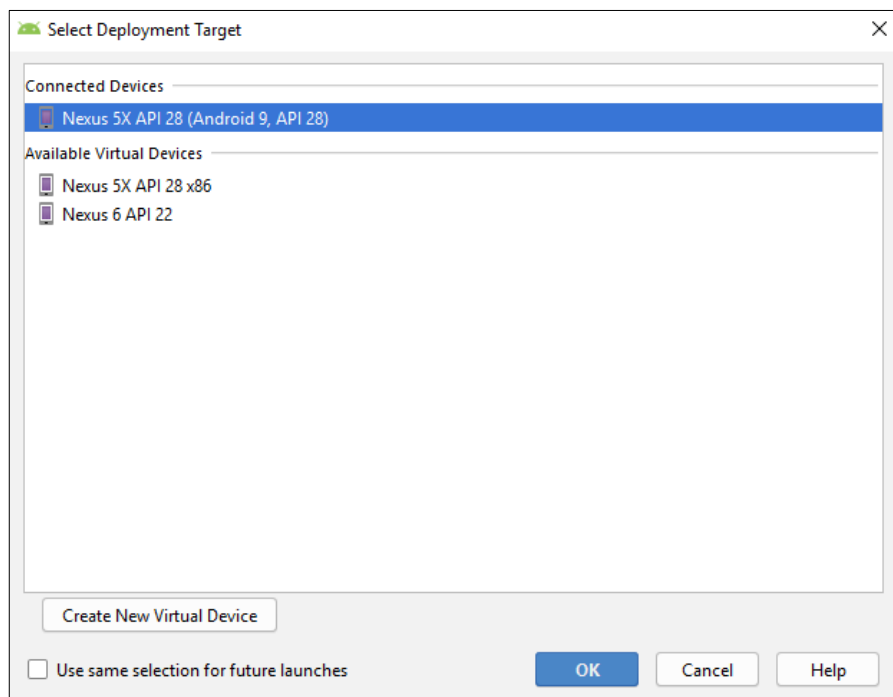
Di dalamnya terdapat informasi tentang *library* yang digunakan oleh aplikasi.

1.3.MENJALANKAN EMULATOR DAN DEVICE

1.3.1. Konfigurasi Android Virtual Device (AVD)

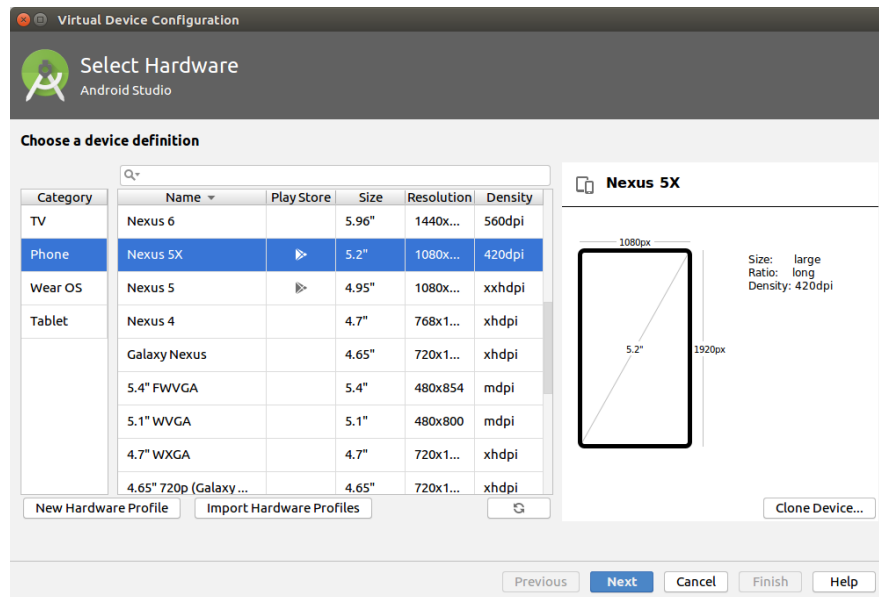
Salah satu cara menjalankan aplikasi (build & run) di komputer adalah dengan menggunakan emulator. Pada saat awal dalam menggunakan android studio, emulator belum ada. Oleh karena itu kita harus melakukan konfigurasi AVD agar emulator dapat berjalan dengan lancar.

Langkah pertama untuk menkonfigurasi AVD adalah dengan menekan tombol  yang ada di toolbar. Kemudian akan muncul dialog seperti Gambar 1.13, untuk membuat emulator baru klik “Create New Virtual Device”.



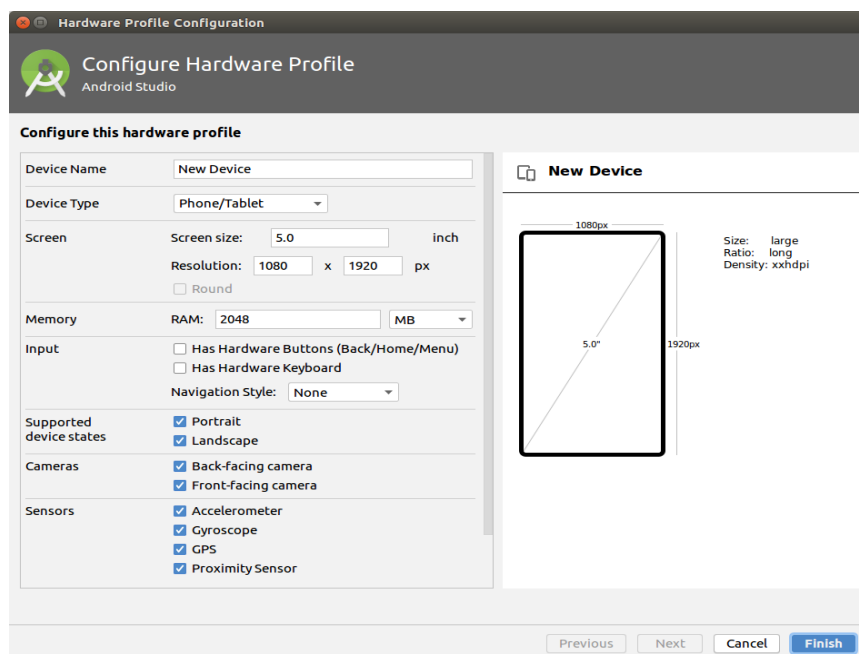
Gambar 1.13. Dialog *Select Deployment Target*

Selanjutnya akan muncul pilihan tampilan perangkat android yang akan dijadikan emulator seperti ditunjukkan Gambar 1.14. Kita dapat memilih kategori perangkat mulai dari phone, table, maupun TV. Pilihlah kategori phone dan pilih juga perangkat dengan ukuran layar 5 inchi. Pada tulisan ini dipilih perangkat Nexus 5X dengan ukuran layar 5,2 inchi



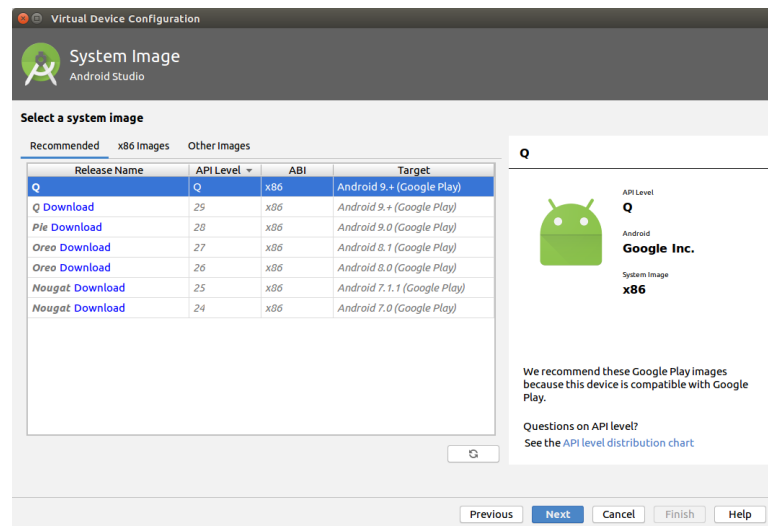
Gambar 1.14 Tampilan Jenis Emulator Android

Apabila hendak melakukan konfigurasi emulator secara custom, klik New Hardware Profile. Maka tampilan konfigurasi profil perangkat akan muncul seperti ditunjukkan pada Gambar 1.15. Pada saat mengkonfigurasi profile perangkat, kita dapat mengubah nama, ukuran layar, ukuran memori, dan sensor yang digunakan sesuai dengan keinginan kita.



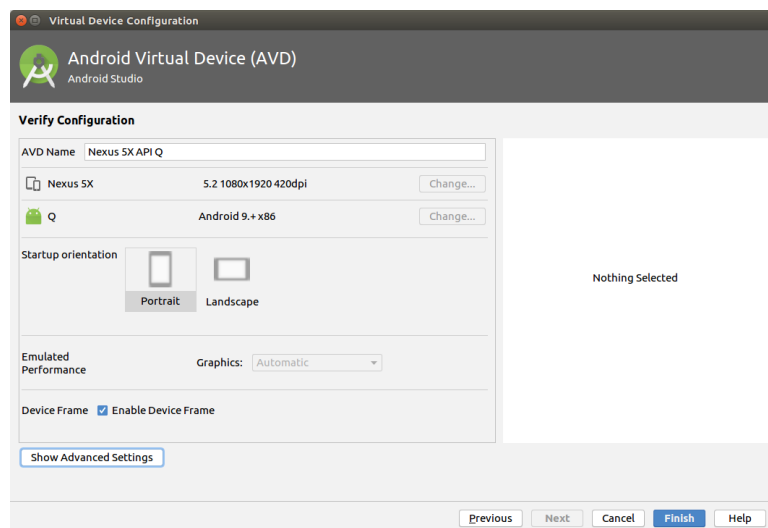
Gambar 1.15. Menambah Profil Hardware

Kemudian setelah memilih profil perangkat yang digunakan untuk emulator, langkah selanjutnya adalah memilih versi android yang dipasang ke perangkat. Tampilan versi android yang telah *release* di pasaran ditunjukkan pada Gambar 1.16. Pada saat tulisan ini dibuat versi yang terbaru adalah versi Q. Namun apabila ingin menggunakan versi terdahulu, maka dapat mengunduh versi lama tersebut. Pastinya laptop kita terhubung dengan internet yang stabil




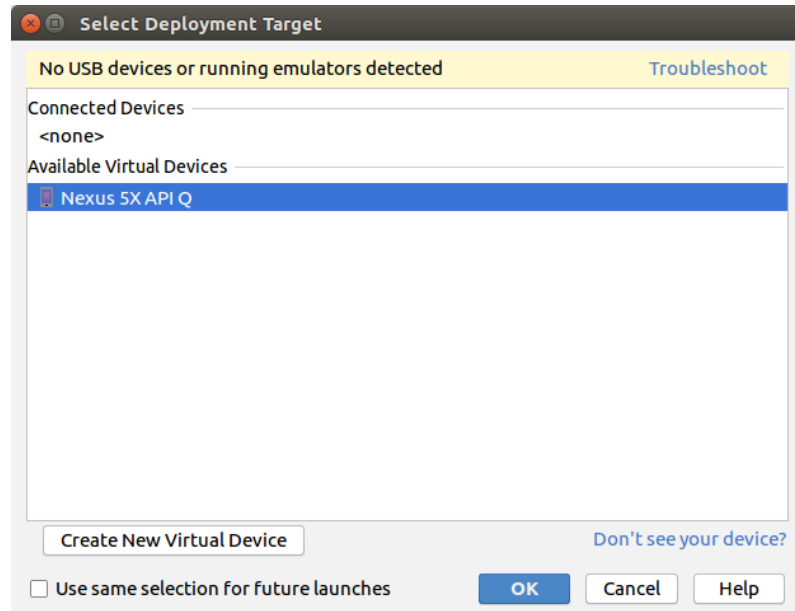
Gambar 1.16. Tampilan Pemilihan Versi Android

Langkah berikutnya adalah finishing pembuatan Android Virtual Device (AVD). Tampilan dialog untuk verifikasi AVD ditunjukkan pada Gambar 1.17. Pada langkah ini kita dapat mengganti nama AVD dan juga melakukan pengaturan lebih lanjut dengan klik “Show Advanced Settings”. Apabila telah sesuai maka klik Finish.



Gambar 1.17. Verifikasi Android Virtual Device (AVD)

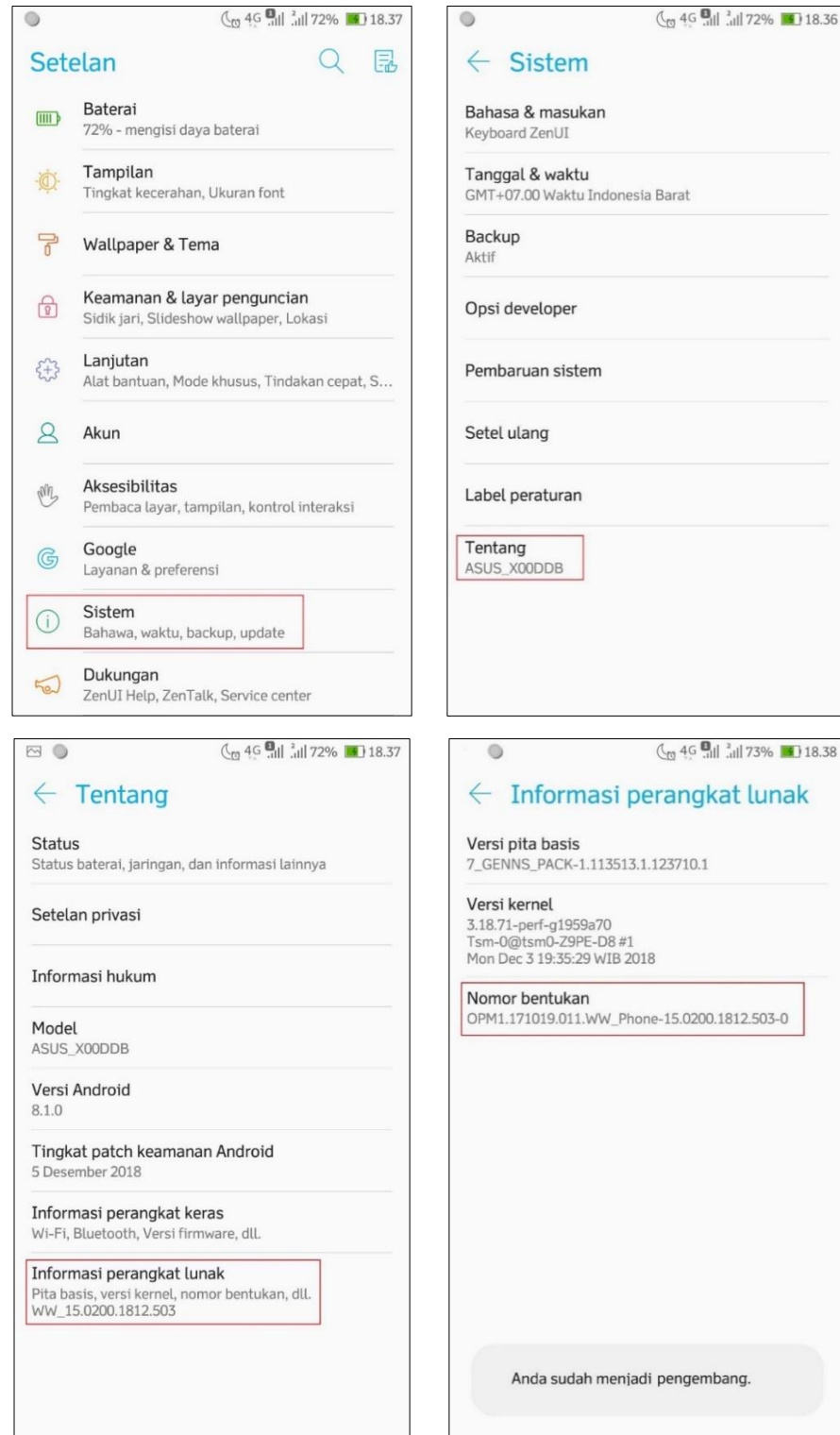
Apabila AVD berhasil dibuat maka ketika kita klik tombol  akan muncul tambahan perangkat pada Available Virtual Devices seperti ditunjukkan pada Gambar 1.18. Selanjutnya klik OK maka emulator yang kita tambahkan akan muncul untuk menjalankan aplikasi yang telah kita bangun.



Gambar 1.18. Berhasil menambahkan AVD pada target emulator

1.3.2. Konfigurasi Device Android

Selain menggunakan emulator, proses *running* aplikasi android juga dapat menggunakan perangkat android yang asli. Caranya seperti ditunjukkan pada Gambar 1.19. Langkah pertama adalah mengubah *smartphone* android kita menjadi mode developer dengan cara klik 7x pada build number *smartphone* kita. Langkah-langkah menuju build number dalam dengan pilih Setting → Sistem → About → About Software → Build Number. Apabila berhasil menambah mode developer, maka akan muncul pilihan Opsi Developer (Developer Option).



Gambar 1.19. Menjadikan Mode Pengembang

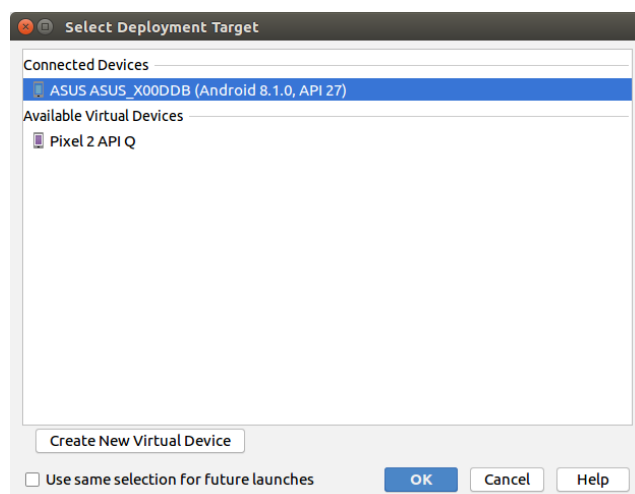
Aktifkan mode developer dengan menggeser ke slide aktif. Kemudian aktifkan juga Debugging USB pada perangkat android seperti ditunjukan Gambar 1.20. Langkah terakhir

adalah sambungkan perangkat android anda dengan komputer/laptop menggunakan kabel data.



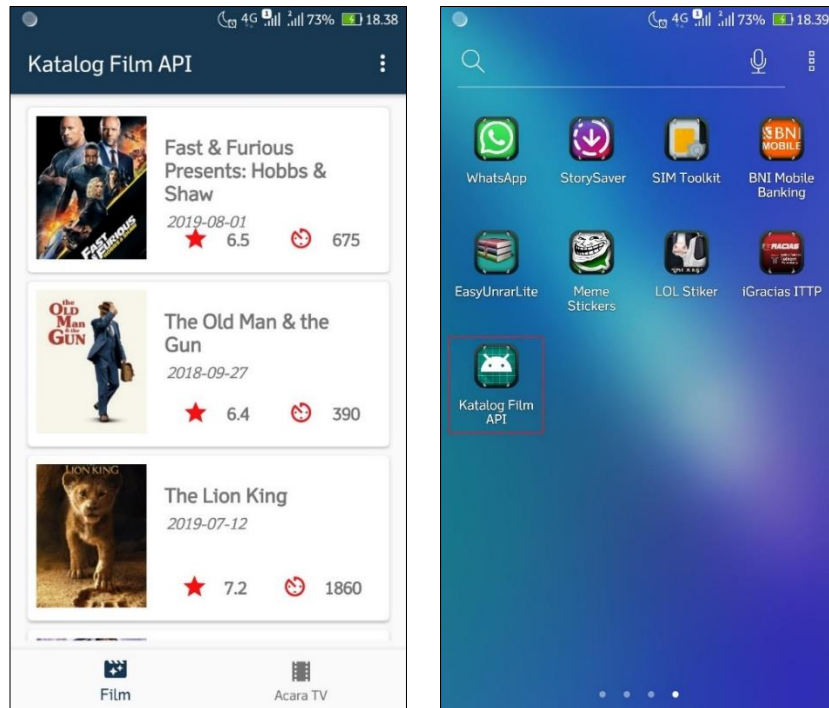
Gambar 1.20. Opsi Developer Android

Langkah terakhir melakukan pengecekan dengan menekan tombol play button warna hijau. Apabila sudah muncul nama perangkat handphone kita seperti Gambar 1.21, berarti perangkat sudah terintegrasi dengan android studio.



Gambar 1.21. Device Android telah diidentifikasi

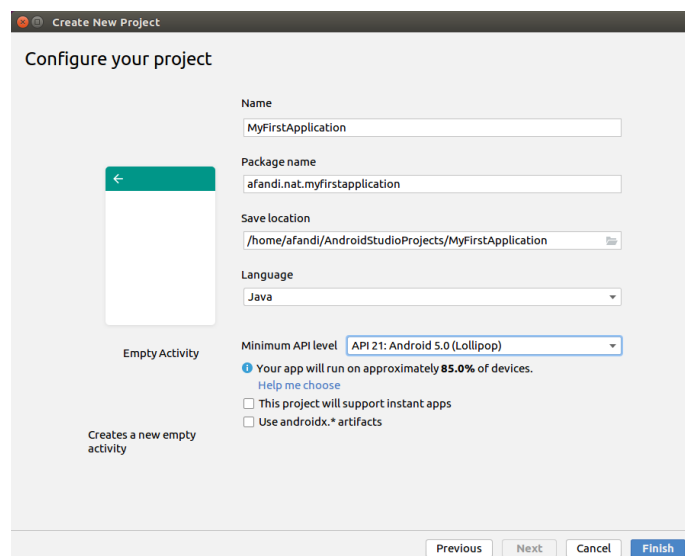
Jika berhasil koneksi dengan perangkat android, maka aplikasi akan langsung terpasang. Hasil instalasi aplikasi yang telah dibuat ditunjukkan pada Gambar 1.22.



Gambar 1.22. Hasil *Running* Aplikasi

1.3.3. MEMBUAT FILE ANDROID PACKAGE (APK)

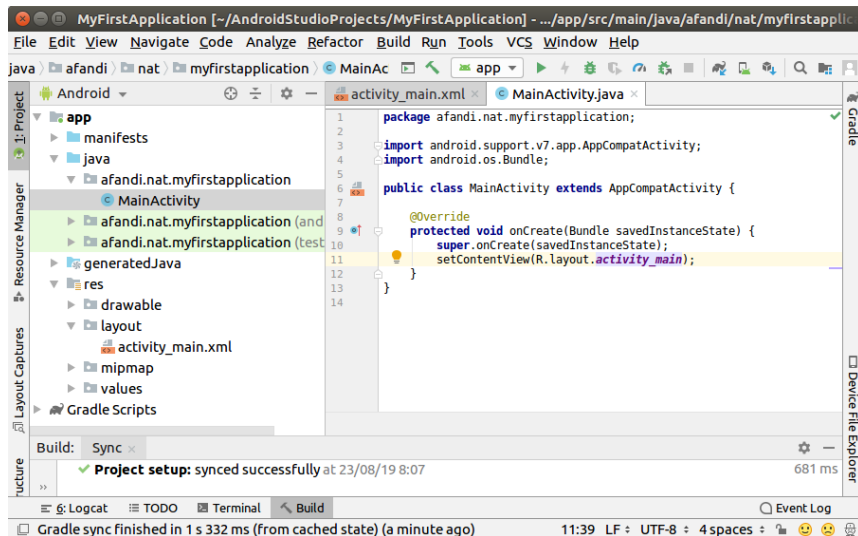
Berkas APK berisi aplikasi android yang sudah dibungkus dan siap dipasang dalam perangkat android. Untuk membuat berkas APK, hal pertama yang dilakukan adalah membuat proyek baru seperti ditunjukan Gambar 1.23.



Gambar 1.23. Konfigurasi Proyek Awal Android

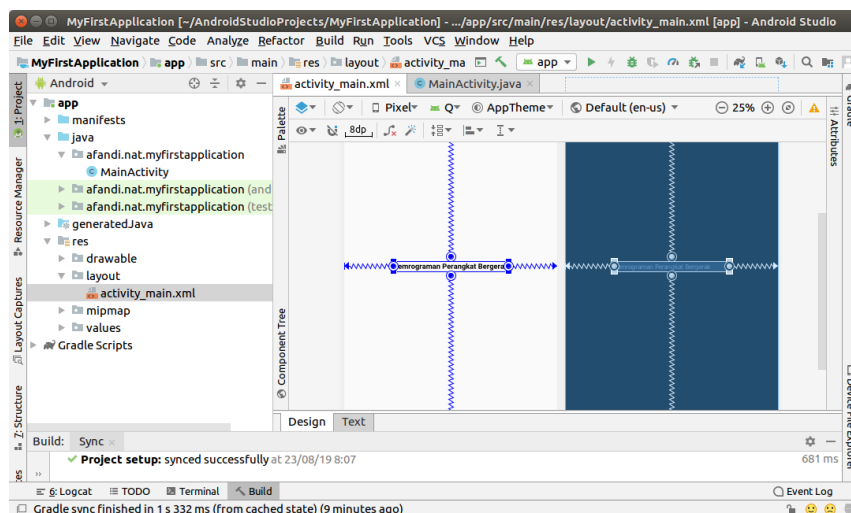
Buat proyek baru dengan nama MyFirstApplication menggunakan tipe Empty Activity dengan target minimum API adalah 21. Isinya nama package dengan dipisahkan tanda titik.

Secara otomatis akan muncul dua berkas yaitu MainActivity.java dan activity_main.xml. Isi dari MainActivity.java berisi *source code* java yang merujuk ke file activity_main.xml. Tampilan editor untuk menampilkan file java ditunjukkan pada Gambar 1.24.



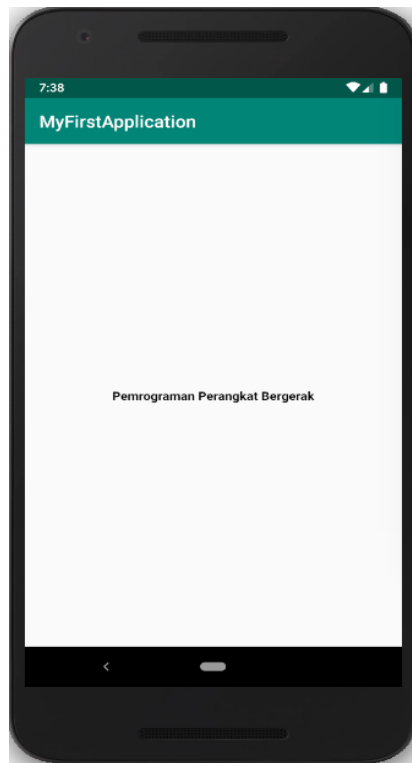
Gambar 1.24. Tampilan Editor di Android Studio untuk *File* Java

Tampilan editor XML pada android studio ditunjukkan pada Gambar 1.25. Pada editor XML terdapat pilihan antara design dan text. Pada tab design kita dapat mendesain tampilan android dengan cara drag and drop komponen-komponen GUI yang ada di pallette. Kemudian untuk tab text digunakan untuk mendesain tampilan dengan menuliskan kode xml. Ubah kode XML seperti pada kode berikut



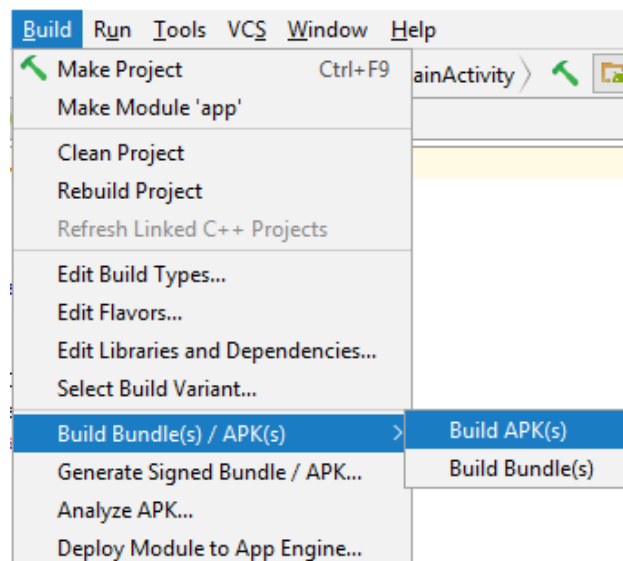
Gambar 1.25. Tampilan Editor di Android Studio untuk *File* XML

Kemudian jalankan program yang telah anda buat tersebut dengan cara menekan tombol play berwarna hijau. Pilih menggunakan emulator jika laptop anda sesuai spesifikasi yang telah dijelaskan. Hasil tampilan program ditunjukkan seperti pada Gambar 1.26.



Gambar 1.26. Tampilan Program MyFirstApplication

Langkah terakhir, lakukan build APK dengan cara klik menu Build → Build Bundles(s) / APK(s) → Build APK(s). Agar lebih jelas cara untuk build APK, amati Gambar 1.27.



Gambar 1.27. Langkah *Build* APK

Gradle akan membuat berkas APK secara otomatis. Lama proses ini bergantung pada seberapa kompleks Aplikasi yang Anda buat dan jumlah dependency yang digunakan serta

spesifikasi perangkat yang dipakai. Biasanya lokasinya mengikuti struktur project-name/module-name/build/outputs/apk/. Jika proyek bernama HelloWorld, maka lokasinya adalah HelloWorld/app/build/outputs/apk/apk-debug.apk.

Latihan 1

1. Buatlah sebuah program android menggunakan android studio dengan spesifikasi seperti berikut.

Nama Project	1. MyFirstExercise
Target & Minimum Target SDK	Phone and Table, API level 21
Tipe Activity	Empty Activity
Activity Name	MainActivity

Tampilan Program Android seperti berikut ini.

Biodata Mahasiswa

Nama : [Nama Mahasiswa]

NIM : [NIM Mahasiswa]

TTL : [TTL Mahasiswa]

Prodi : [Program Studi Mahasiswa]

Cita - Cita : [Cita-cita Mahasiswa]

2. Unggah proyek android yang sudah berhasil dibuat ke github masing-masing. Berikan screenshot tampilan program pada keterangan github. Struktur github yang dibuat adalah sebagai berikut

