

# **DEBUGER PADA APLIKASI INVENTORY BARANG MASUK DAN KELUAR DENGAN SCANNER**

Laporan ini dibuat untuk memenuhi persyaratan kelulusan  
matakuliah Proyek 2



**Oleh:**

**1.18.4.065 Etika Khusnul Laeli**

**1.18.4.109 Jenly Ramdan**

**PROGRAM STUDI DIPLOMA 1V TEKNIK INFORMATIKA**

**POLITEKNIK POS INDONESIA**

**BANDUNG**

**2019**

## PERKENALAN ANDROID STUDIO

### MENGENAL ANDROID STUDIO

Android Studio adalah IDE pemrograman Android resmi dari Google yang dikembangkan dari IntelliJ. Sebelum ada Android Studio, programmer Android telah menggunakan Eclipse. Eclipse adalah IDE pemrograman Android sebelum munculnya Android Studio. Bisa dibilang Google telah berpaling dari Eclipse dan menjadikan Android Studio sebagai IDE resminya. Dikarenakan sudah meresmikan Android Studio pada tanggal 16 Mei 2013, Google menghentikan support ADT ke Eclipse tak lama kemudian dan ADT resmi hanya didapatkan oleh Android Studio.

Dikembangkan di atas IntelliJ IDEA besutan JetBrains, Android Studio dirancang khusus untuk pengembangan Android. IDE Ini tersedia untuk digunakan pada sistem operasi Windows, Mac OS X dan Linux.

Android Studio dipilih karena memiliki banyak fitur yang memudahkan para pembuat program terutama programmer level dasar yang ingin belajar lebih tentang android. Walaupun pada proses penggunaannya Android Studio cukup banyak menghabiskan RAM pada perangkat PC kita, teman-teman tidak perlu khawatir karena untuk menutupi kekurangan tersebut Android Studio memiliki beberapa kelebihan lainnya seperti :

1. Instant Run

Saat kita akan menjalankan program yang telah dibuat melalui perintah **run** (compile), kita akan meng-compile program Android untuk membuat APK lalu mengirimkannya ke perangkat yang dipakai untuk menjalankannya (bisa emulator atau perangkat asli). Fitur *instant run* dapat membuat program berjalan dengan cepat tanpa harus meng-*compile* ulang aplikasi atau membuat kembali APK saat melakukan perubahan kode sehingga proses yang didapatkan menjadi lebih cepat. Instant Run dapat berjalan apabila pembuat program menggunakan versi Gradle 2.0.0 atau lebih dan versi SDK 15 atau lebih. Editor kode yang cerdas

2. Emulator yang cepat dan kaya fitur

Android Studio membantu kita menulis kode dengan lebih baik, lebih cepat, dan lebih produktif. Android Studio memiliki fitur *intelligent code editor* yang memberikan kemudahan dalam analisis kode dan menyediakan saran kode yang akan digunakan dengan sistem auto complete. Saat kita mengetik suatu kode, Android Studio memberikan saran secara otomatis bila ada kelas yang telah terpasang dan kita dapat menekan tombol TAB untuk memasukkan kode tersebut bila sesuai dengan apa yang kita butuhkan. Dengan fitur - fitur ini tentunya mempercepat waktu pembuatan program sehingga kinerja pembuat program menjadi lebih produktif.

3. Sistem versi yang fleksibel

Android Studio menawarkan otomatisasi versi, manajemen dependensi, dan konfigurasi versi yang bisa disesuaikan. Anda bisa mengonfigurasi proyek untuk menyertakan library lokal dan di-host, serta mendefinisikan varian versi yang menyertakan kode berbeda. Teman-teman dibebaskan untuk melakukan konfigurasi memasang library yang memudahkan teman-teman dalam membuat aplikasi android fitur ini merupakan bagian dari fleksibilitas dari android studio.

4. Dioptimalkan untuk semua perangkat Android

Android Studio memberikan tempat bagi Anda untuk membuat aplikasi dengan berbagai macam perangkat namun berbasis android seperti tablet Android, Android Wear, Android TV, dan Android Auto. Fungsi terstruktur ini memungkinkan Anda untuk membagi proyek menjadi unit-unit fungsi yang bisa Anda buat, uji, dan debug sesuai dengan keinginan anda.

5. Didesain untuk tim

Android Studio memiliki integrasi dengan beberapa version control populer seperti Git dan Subversion. Bahkan untuk membantu kolaborasi, kita juga dapat menggunakan layanan Github langsung dari Android Studio. Dengan demikian pembaca dan tim tetap bekerja efektif dengan proyek yang dapat diakses dengan mudah antara satu dengan yang lainnya.

6. Membuat Aplikasi yang Lengkap dan Terkoneksi

Android Studio mendukung sepenuhnya pengeditan file proyek yang menggunakan bahasa C/C++ sehingga teman-teman dapat dengan cepat membuat

komponen-komponen JNI dalam aplikasi. IDE ini pun menyediakan sintaks dan pemfaktoran ulang untuk C/C++, dan debugger berbasis LLDB sehingga Anda bisa men-*debug* kode Java dan C/C++ secara bersamaan.

#### 7. Integrasi Firebase dan Cloud

Firebase Assistant membantu menghubungkan aplikasi Anda dengan Firebase dan menambahkan layanan seperti Analytics, Autentikasi, Notifikasi, dan lainnya dengan prosedur sesuai dengan urutan di dalam Android Studio. Selain itu tools bawaan untuk Google Cloud Platform juga membantu Anda membuat dan menerapkan backend untuk aplikasi Android, seperti penggunaan layanan Google Cloud Endpoints dan modul proyek yang dirancang khusus untuk Google App Engine.

#### 8. Memberikan Efektivitas dalam membuat aplikasi

Android Studio menyediakan alat bantu GUI yang menyederhanakan bagian yang kurang menarik dari pengembangan aplikasi sehingga pihaknya memberikan beberapa fitur diantaranya:

- Layout Editor
- Penganalisis APK
- Vector Asset Studio
- Editor Terjemahan

Saat ini Android Studio telah merilis versi **Android Studio 2.3** dengan beberapa perubahan dari versi sebelumnya. Pada versi Android Studio 2.3 ini berfokus pada beberapa perbaikan IDE.

Diantara fitur baru yang bisa teman-teman nikmati Google menyebutkan bahwa adanya dukungan WebP untuk gambar pada aplikasi juga dukungan Constraint Layout Library dan Palet widget di Layout editor dan menambahkan App Link Assistant baru dimana fitur ini akan membantu para pengembang aplikasi untuk membangun dan mendapatkan tampilan yang solid dari URL didalam aplikasi yang akan mereka buat.

Kabar menariknya dalam **belajar Android Studio**, para pengembang aplikasi bisa

memanfaatkan tombol *run compiler* yang sudah diperbarui untuk mendapatkan

pengalaman *instant run* yang lebih cepat.

Selain fitur tersebut, Google juga ikut menambahkan dukungan copy dan paste teks yang cocok bagi para pengembang yang akan menguji aplikasi buatan mereka di Emulator Android. Fitur ini

telah mereka tambahkan berdasarkan vote permintaan fitur teratas. Sekarang fitur ini akan bisa diterapkan pada sistem berarsitektur x86 Google API Emulator image API Level 19 (Android 4.4 KitKat) dan atau versi keatasnya.

Tidak hanya sebatas itu, banyaknya perubahan yang terjadi pada rilis terbaru ini meliputi adanya pilihan untuk mencari material ikon wizard, instan run untuk perubahan UI, build cache dan tentu saja masih banyak tools menarik yang bisa anda eksplorasi lebih dalam pada versi terbaru ini. Bagi anda yang ingin mencoba beberapa fitur tersebut, siapkan koneksi internet dan kuota internet yang lebih agar tidak terputus ketika melakukan update atau mendownload Android Studio terbaru 2.3 melalui situs resmi mereka.

Android memiliki empat karakteristik sebagai berikut :

#### 1. Terbuka

Android dibangun untuk benar-benar terbuka sehingga sebuah aplikasi dapat memanggil salah satu fungsi inti ponsel seperti membuat panggilan, mengirim pesan teks, menggunakan kamera dan lain-lain. Android merupakan sebuah mesin virtual yang dirancang khusus untuk mengoptimalkan sumber daya memori dan perangkat keras yang terdapat di dalam perangkat. Android merupakan open source, dapat secara bebas diperluas untuk memasukkan teknologi baru yang lebih maju pada saat teknologi tersebut muncul. Platform ini akan terus berkembang untuk membangun aplikasi mobile yang inovatif.

#### 2. Semua aplikasi dibuat sama

Android tidak memberikan perbedaan terhadap aplikasi utama dari telepon dan aplikasi pihak ketiga (third-party application). Semua aplikasi dapat dibangun untuk memiliki akses yang sama terhadap kemampuan sebuah telepon dalam menyediakan layanan dan aplikasi yang luas terhadap para pengguna.

#### 3. Memecahkan hambatan pada aplikasi

Android memecah hambatan untuk membangun aplikasi yang baru dan inovatif. Misalnya, pengembang dapat menggabungkan informasi yang diperoleh dari web dengan data pada ponsel seseorang seperti kontak pengguna, kalender atau lokasi geografis.

#### 4. Pengembangan aplikasi yang cepat dan mudah

Android menyediakan akses yang sangat luas kepada pengguna untuk menggunakan aplikasi yang semakin baik. Android memiliki sekumpulan tools yang dapat digunakan sehingga membantu para pengembang dalam meningkatkan produktivitas pada saat membangun aplikasi yang dibuat.

#### Android Software Development Kit (SDK)

Android SDK adalah tool API (application Programming Interface) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang release oleh Google. Saat ini disediakan Android SDK (Software Development Kit) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman java (Safaat H 2011 : 5).

#### Android Development Tools (ADT)

Android Development Tools (ADT) yang lebih dikenal plugin Eclipse. plugin ini yang membuat eclipse dapat membuat project yang berbasis android. ADT adalah plugins di eclipse yang harus kita instal sehingga Android SDK dapat dihubungkan dengan IDE Eclipse yang digunakan sebagai tempat coding aplikasi android nantinya. ADT adalah kepanjangan dari Android Development Tools yang menghubungkan antara IDE Eclipse dengan Android SDK (Safaat H 2011 : 17).

#### Android Virtual Device (AVD)

AVD yang merupakan emulator untuk menjalankan program aplikasi android yang kita buat, AVD ini nantinya yang kita jadikan sebagai tempat test dan menjalankan aplikasi android yang kita buat, AVD berjalan di virtual Machine (Safaat H 2011 : 19).

## 7. Bahasa Pemrograman Android Paling Populer

- Bahasa Pemrograman Java



Tahun lalu tepatnya bulan Juni 2017 bahasa pemrograman **Java** telah menjadi bahasa pemrograman paling populer (data Tiobex Index). Namun sekarang sudah disalip oleh bahasa pemrograman **JavaScript** dan **Phyton**.

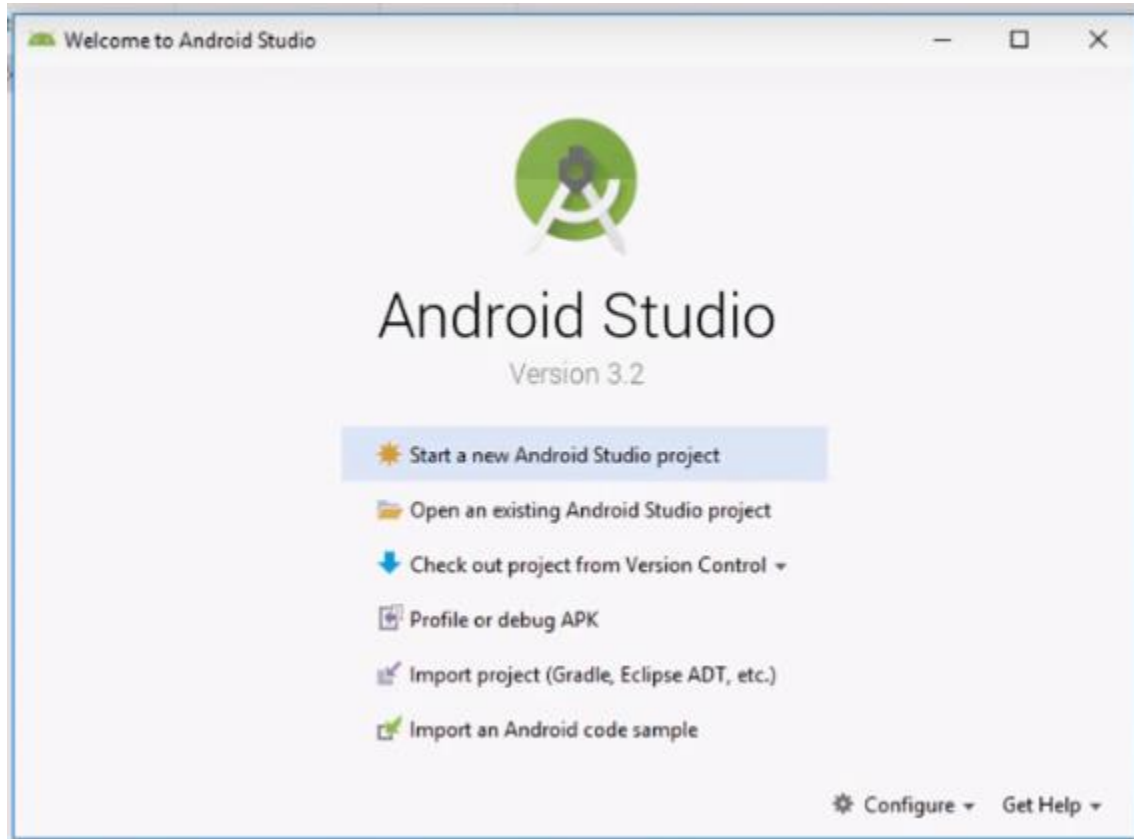
Namun tetap saja walaupun tidak menjadi pertama, java tetap berada di urutan 10 besar bahasa pemrograman yang paling populer di dunia saat ini.

Salah satu penyebabnya yaitu karena jutaan aplikasi android dibuat menggunakan bahasa pemrograman android. Untuk membuat aplikasi android menggunakan bahasa pemrograman java kita bisa menggunakan tools atau IDE :

- Android Studio (IDE resmi yang didukung penuh oleh google)
- Eclipse (IDE lain yang sebelumnya didukung penuh oleh google sebelum adanya android studio)

Untuk pemula yang baru ingin belajar membuat aplikasi android disarankan menggunakan bahasa pemrograman java.

## 2. Bahasa Pemrograman C++



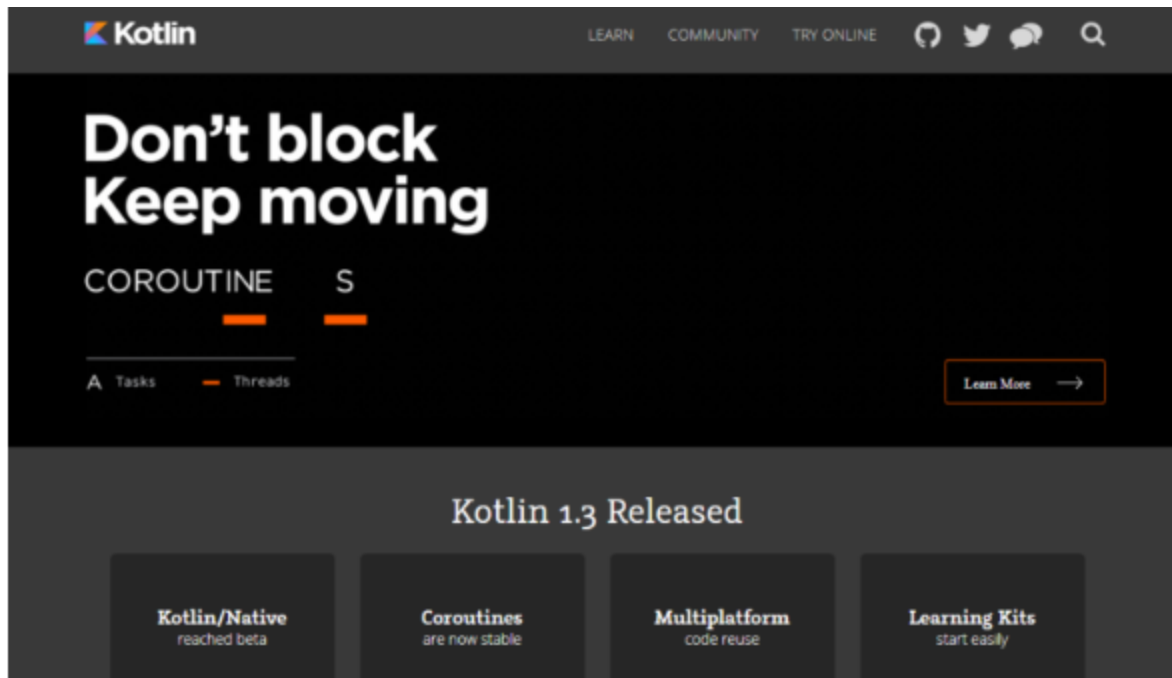
Sebelum kotlin diresmikan oleh Google menjadi salah satu bahasa pemrograman yang didukung penuh untuk membuat aplikasi android, bahasa pemrograman C++ sudah lebih dulu setelah Java.

Java, C++, dan kotlinpun saat ini menjadi bahasa pemrograman populer yang digunakan banyak developer di seluruh dunia.

Lagi-lagi kita bisa menggunakan IDE Android Studio untuk mengembangkan aplikasi android dengan bahasa pemrograman C++.

## 3. Bahasa Pemrograman Kotlin





Kotlin baru mendapatkan dukungan dari google sebagai salah satu bahasa pemrograman resmi untuk membuat aplikasi android sekitar bulan ..... .

Kotlin diciptakan oleh JetBrains yaitu perusahaan yang terkenal membuat IDE seperti : Android Studio, RubyMine, PhpStorm, dll.

Kotlin sengaja diciptakan oleh JetBrains untuk melengkapi segala kekurangan dari bahasa pemrograman java. Memang benar bahasa pemrograman kotlin lebih simple dibandingkan java.

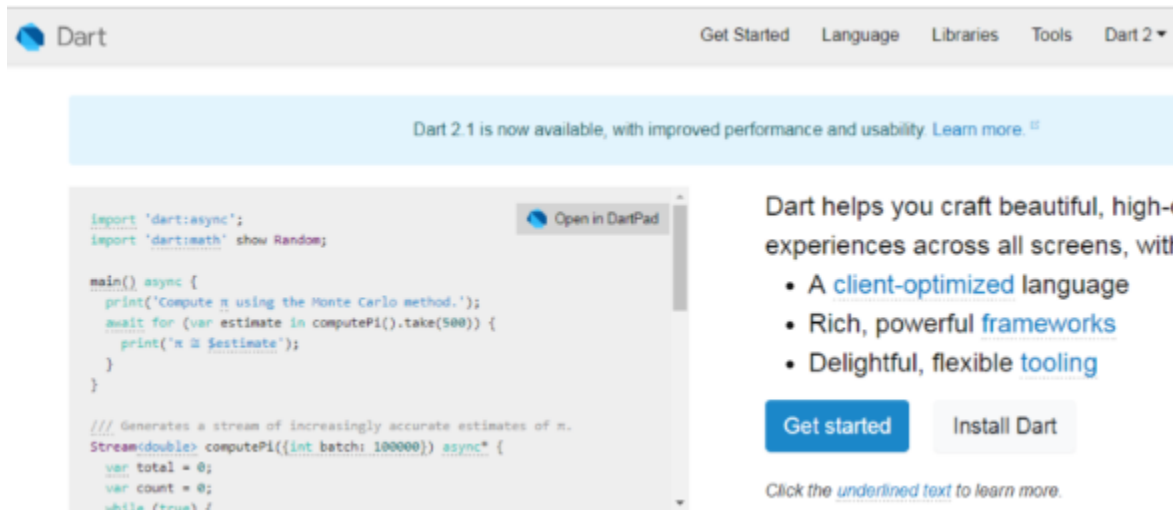
Keunggulan lainnya bahasa ini bisa berjalan beriringan dengan bahasa pemrograman java. Dan bisa menggunakan library dari java.

Jika kamu sudah mempelajari bahasa pemrograman java dan sudah mencapai level menengah. Kamu bisa mencoba mempraktekan bahasa pemrograman kotlin untuk membuat aplikasi android.

Pembuatan aplikasi android menggunakan saat ini bisa menggunakan IDE :

- IntelliJ IDEA
- Android Studio
- Eclipse

## 4. Bahasa Pemrograman Dart

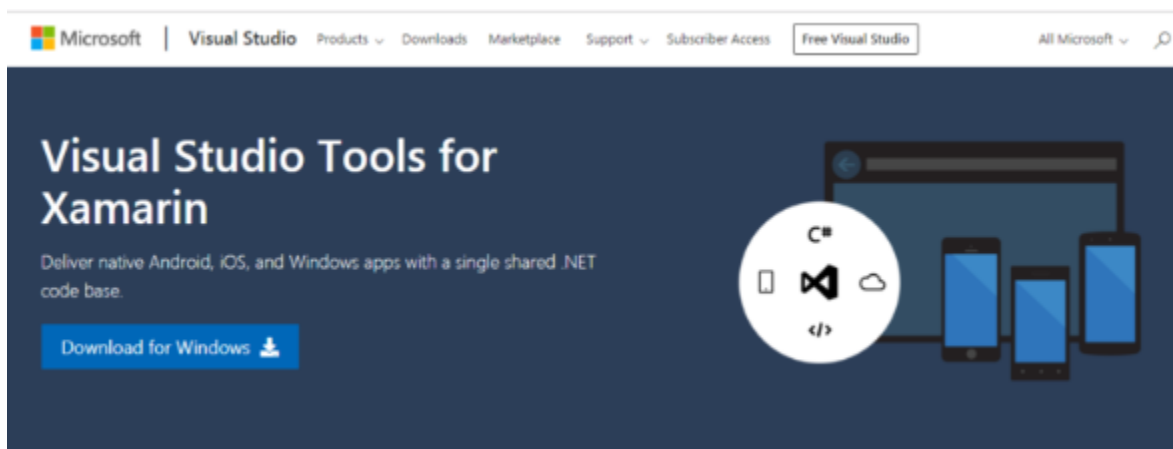


Bahasa pemrograman Dart dikembangkan oleh perusahaan raksasa nomor 1 di dunia yaitu google. Selain dart google juga telah mengembangkan golang dan bahasa pemrograman lainnya.

Bahasa pemrograman ini bisa digunakan untuk membuat aplikasi di berbagai platform seperti : website, mobile, Internet of Things (IoT), dan server. Dart sendiri dirilis oleh google secara open source.

Untuk mengembangkan aplikasi android menggunakan bahasa pemrograman Dart kita harus menggunakan Tools Flutter.

## 5. Bahasa Pemrograman C#



Bahasa pemrograman lain yang bisa kita gunakan untuk membuat aplikasi android yaitu bahasa pemrograman C# (C Sharp). Jika kamu sudah memiliki dasar pemrograman C#, maka kamu

tidak akan menemukan kesulitan yang berarti saat belajar membuat aplikasi android menggunakan bahasa pemrograman C#.

Tools yang kita perlukan untuk mengembangkan aplikasi android menggunakan bahasa pemrograman C# adalah Xamarin. Terdapat dua IDE pada xamarin yaitu :

- Xamarin yang terintegrasi dengan Microsoft Visual Studio
- Xamarin Studio

Perbedaannya :

Jika komputer yang kita gunakan memakai sistem operasi windows maka Xamarin Microsoft Visual Studio cocok untuk digunakan. Namun bila sistem operasinya Mac maka disarankan memakai Xamarin Studio.

## 6. Bahasa Pemrograman JavaScript



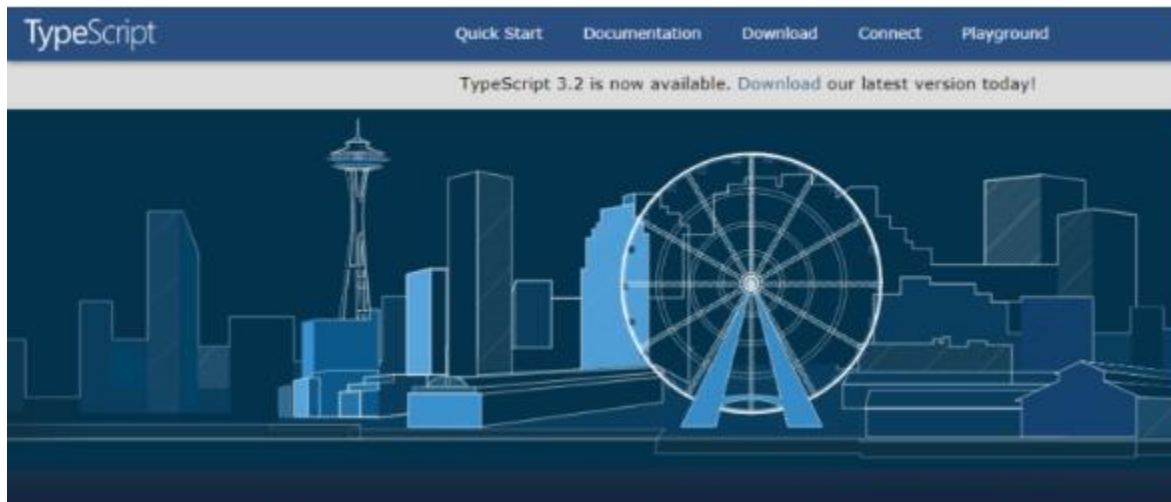
Bahasa pemrograman website terpopuler saat ini yaitu JavaScript ternyata bisa juga digunakan untuk membuat aplikasi android.

Kita membutuhkan Framework untuk membuat aplikasi berbasis android. Beberapa framework yang bisa kita gunakan untuk mengembangkan aplikasi android yaitu : React Native, Ionic, dan PhoneGap.

Jika kita sudah punya basic HTML5, CSS, dan JavaScript Modern kita akan bisa dengan mudah belajar membuat aplikasi dengan framework-framework di atas.

Badoystudio.com sendiri sudah merilis tutorial membuat aplikasi android menggunakan React Native.

## 7. Bahasa Pemrograman TypeScript



TypeScript sendiri adalah bahasa pemrogramna yang dikembangkan oleh microsoft. TypeScript diciptakan oleh microsoft untuk menutupi kekurangan bahasa pemrograman JavaScript yaitu membuat aplikasi yang besar dan kompleks dengan mudah.

Selain digunakan untuk membuat aplikasi web bahasa ini bisa juga digunakan untuk membuat aplikasi mobile baik android ataupun Ios.

Sebagian besar tutorial membuat aplikasi android yang dirilis oleh badoystudio.com saat ini menggunakan bahasa pemrograman java.

### DEFINISI ANDROID STUDIO

Android Studio adalah Lingkungan Pengembangan Terpadu (Integrated Development Environment/IDE) resmi untuk pengembangan aplikasi Android, yang didasarkan pada IntelliJ IDEA. Selain sebagai editor kode dan fitur developer IntelliJ yang andal, Android Studio menawarkan banyak fitur yang meningkatkan produktivitas Anda dalam membuat aplikasi Android, seperti:

- Sistem build berbasis Gradle yang fleksibel
- Emulator yang cepat dan kaya fitur

- Lingkungan terpadu tempat Anda bisa mengembangkan aplikasi untuk semua perangkat Android
- Terapkan Perubahan untuk melakukan push pada perubahan kode dan resource ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi
- Template kode dan integrasi GitHub untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel
- Framework dan fitur pengujian yang lengkap
- Fitur lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya
- Dukungan C++ dan NDK
- Dukungan bawaan untuk [Google Cloud Platform](#), yang memudahkan integrasi Google Cloud Messaging dan App Engine

## SEJARAH ANDROID STUDIO

Pertama kali muncul Android Inc merupakan sebuah perusahaan software kecil yang didirikan pada bulan Oktober 2003 di Palo Alto, California, USA. Perusahaan ini dibangun oleh beberapa senior di beberapa perusahaan yang berbasis IT & Communication, Andy Rubin, Rich Miner, Nick Sears dan Chris White.

Rubin menyatakan bahwa, Android Inc Didirikan untuk mewujudkan mobile device yang lebih fleksibel terhadap lokasi dan preferensi pemilik. Sehingga, Android Inc ingin mewujudkan mobile device yang lebih mengerti pemiliknya selain karena OS nya yang *open source*.

Berawal dari konsep inilah Android Inc ternyata menarik minat Google untuk memilikinya. Maka, pada bulan Agustus 2005, Akhirnya Android Inc diakuisisi oleh Google Inc. dan seluruh sahamnya dibeli oleh Google.

Perusahaan milik Andy Rubin, Rich Miner, Nick Sears dan Chris White tetap di Android Inc yang dibeli Google, sehingga akhirnya mereka pun ikut menjadi bagian dari raksasa Google dan sejarah Android. Disini mereka mulai menggunakan platform Linux untuk membuat sistem operasi bagi mobile phone.

Dari sinilah akhirnya banyak pengembang sistem maupun software yang mengembangkan maupun merancang sistem Android menggunakan software – software yang support dengan Android, Contohnya ialah : Android Studio.





## **MACAM-MACAM ANDROID STUDIO :**

### **1. Android 1.0 (Apple Pie)**

Android versi pertama ini dirilis pada 23 September 2008 dan hanya dilengkapi fitur-fitur seperti Play Store, Web Browser, Kamera, Sinkronisasi antara Gmail, Contacts dan Google Agenda. Selain itu, diawal perilisannya, Android juga sudah dilengkapi aplikasi Google Maps serta dukungan streaming Youtube.

### **2. Android 1.1 (Banana Bread)**

Sistem Operasi android yang rilis selanjutnya adalah Banana Bread, rilis pada bulan Februari 2009. Dan fitur ini juga tidak jauh berbeda dengan versi sebelumnya.

HTC adalah salah satu ponsel Android pertama yang menggunakan versi ini.

### **3. Android 1.5 (Cupcake)**

Rilis pada awal bulan April 2009 dan juga tidak jauh berbeda dengan versi Android sebelumnya. Hanya saja ada fitur tambahan seperti Support Bluetooth A2DP, AVRCP, Soft-keyboard dengan prediksi text dan record/watch videos.

### **4. Android 1.6 (Donut)**

Android Donut rilis pada 15 September 2009, dan mendapat fitur tambahan seperti Gesture Framework hingga Turn-by-turn navigation. Selain itu, Android ini juga terlihat lebih sempurna pada waktu itu. Dengan minimnya bug, ditambah lebih lengkapnya fitur-fitur yang disediakan Google.

### **5. Android 2.0 (Éclair)**

Android versi 2.0 bernama Eclair dan rilis pada 26 Oktober 2009 silam. Yang selain bluetooth, Android versi ini juga mendapatkan fitur multi-touch, Live Wallpaper dan juga flash kamera.

Selain itu, adapun beberapa fitur yang dapat anda nikmati dalam Android versi ini adalah yakni, HTML, Digital zoom, Support Microsoft Exchange, dan Updated UI.

### **6. Android 2.2 9 (Froyo)**

Pada bulan Mei 2010 lalu, Google telah merilis Android versi terbaru pada waktu itu. Yakni adalah Android 2.2 9 (Froyo). Versi ini merupakan salah satu sistem operasi



Android yang juga telah disempurnakan, utamanya tentu untuk meningkatkan kecepatan kinerja suatu Android.

Dan berikut ini adalah fitur dan perbaikan yang disediakan oleh Android versi 2.2.9 :

- Peningkatan Speed
- Implementasi JIT
- USB Tethering
- Aplikasi instalasi untuk perluasan memori
- Support file upload pada the browser
- Animated GIFs

## **7. Android 2.3 (Gingerbread)**

Pada bulan Desember 2010 lalu, Google secara resmi merilis Android versi terbaru, Gingerbread. Yang secara fitur jelas sudah sangat sempurna. Ditambah lagi, Android versi 2.3 ini juga diadopsi oleh salah satu perusahaan Smartphone paling terkenal, yaitu Samsung dengan menanamkan sistem operasi ini dalam ponsel seri Nexus-nya.

## **8. Android 3.0 – 3.2.6 (Honeycomb)**

Honeycomb merupakan salah satu sistem operasi Android versi terbaru yang rilis pada bulan Februari 2011 silam. Namun, versi ini lebih ditujukan untuk Tablet yang mana pada tahun itu sangat laris dipasaran.

Fitur dan perbaikan pada Android versi ini:

- Support Multi core
- Support Tablet lebih baik
- Updated 3D UI
- Layar Utama (homescreens) yang bisa diatur
- Melihat aplikasi yang barusan dibuka
- Menyempurnakan layout keyboard
- Transport protocol untuk Media/Picture

- video chat Google Talk
- Google eBooks
- “Private browsing”
- System-wide Clipboard
- HTTP Live streaming

### **Update 3.1**

- Peningkatan UI
- Open Accessory API
- USB host API
- Support mouse, joysticks dan gamepad
- Widget Home screen yang bisa di atur ukurannya
- Notifikasi MTP
- RTP API untuk audio

### **Update 3.2**

- Optimise untuk berbagai tablets
- Mode kompatibilitas display (zoom for fixed-sized apps)
- Sinkronisasi Media dari SD card

#### **Update 3.2.1**

- Update Android Market termasuk automatic updates yang lebih mudah
- Update Google Books
- Peningkatan kinerja Wi-Fi
- Perbaikan prediksi tulisan tangan huruf Chinese

#### **Update 3.2.2**

- Perbaikan kecil

#### **Update 3.2.4**

- Update tambahan ‘Pay as you go’ untuk tablet

#### **Update 3.2.6**

- Perbaikan kecil

## **9. Android 4.0 (Ice Cream Sandwich )**

Puncak kematangan Android yakni ketika pada versi ini, yang mana Ice Cream Sandwich rilis pada bulan Oktober 2011 silam. Dan operasi sistem ini mulai bekerja di semua jenis smartphone apapun. Selain bertambahnya fitur-fitur menarik, Ice Cream Sandwich juga merupakan versi Android paling banyak disukai pada waktu itu. Bahkan, Android Ice Cream Sandwich juga dilengkapi dengan fitur ekstra multitasking dan notifikasi yang lebih banyak.

#### **10. Android 4.1.2 (Jelly Bena)**

Jelly Bean rilis pada 9 Juli 2012 lewat konferensi I/O Google. Versi ini merupakan salah satu versi Android yang kerap mendapatkan update fitur-fitur yang berguna dan menarik, beberapa halnya adalah seperti memperbaiki rotasi layar, seperti Support resolusi video 4K, Support penulisan huruf Hebrew and Arabic dari kanan ke kiri, dan peningkatan kinerja, sistem keamanan dan masih banyak lainnya.

#### **11. Android 4.4 (Kitkat)**

Android versi inilah yang saat ini banyak digunakan oleh mayoritas masyarakat Indonesia. Kitkat adalah versi Android yang rilis pada 2013 lalu. pada versi ini, Android banyak mendapatkan pembaharuan fitur. Seperti, terdapat fitur Screen recording, untuk merekam kegiatan yang terjadi pada layar smartphone anda, New Translucent system UI, Peningkatan akses notifikasi, System-wide settings untuk closed captioning, Peningkatan kinerja dan masih banyak yang lainnya.

#### **12. Android 5.0 (Lollipop)**

Rilis pada tahun 2014, Android yang satu ini lebih banyak menawarkan fitur tambahan untuk menyempurnakan fitur-fitur yang sudah ada. Dan Nexus 6 adalah salah satu ponsel yang paling pertama mencicipi Android versi ini. Selain itu, Google juga lebih menyempurnakan kinerja dari Android Lollipop sendiri.

#### **13. Android 6.0 (Marshmallow)**

Android versi 6.0 merupakan salah satu sistem operasi Android yang rilis pada tahun 2015 silam, yang mana banyak membawa pembaharuan. Salah satunya adalah support USB Type-C. Tidak hanya itu saja, Android versi 6 ini serta memberikan fasilitas autentikasi sidik jari dan daya baterai yang lebih meningkat.

#### **14. Android 7.0 (Nougat)**

Android Nougat versi 7.0 rilis pada bulan Agustus 2016 silam yang lebih meningkatkan kinerja versi Android sebelumnya. Selain itu, Android Nougat juga mendapatkan banyak fitur-fitur baru yang diantaranya seperti dapat multitasking, meningkatkan fitur Doze yang dulu telah rilis di Android versi sebelumnya.

Dan inilah beberapa fitur terbaru yang terdapat pada Nougat.

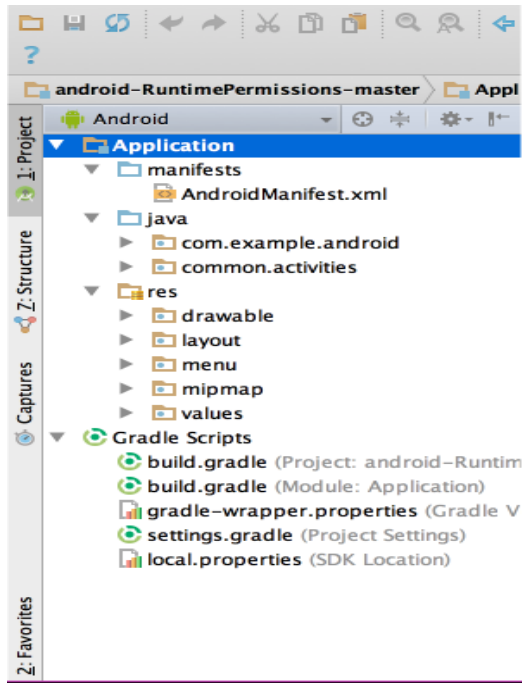
- Support Multi window
- Dapat langsung membalas pesan dari jendela atau menu notifikasi.
- Tampilan panel notifikasi dan quick settings yang baru.
- Mode Doze yang ditingkatkan, (Doze Mode 2.0)
- Menu di antara system settings.

### **15. Android 8.0 ( Oreo)**

Android versi Oreo rilis pada bulan Agustus 2017 lalu. Tentu saja Android versi ini adalah versi final untuk sekarang ini. Beberapa fitur juga turut diluncurkan Google selaku pihak pengelola. Adapun fitur-fitur tersebut antara lain adalah

- Android O lebih fokus pada kecepatan dan efisiensi
- Kecepatan Boot up 2X lebih cepat
- Mode Picture in picture lebih flexibel dari Android N
- Aplikasi yang berjalan di latarbelakang lebih diperketat untuk menghemat battery
- Battery lebih tahan lama
- Emoji yang diperbaharui dan lebih banyak

### **Struktur Project Android Studio**



Gambar 1.1 File Project dalam tampilan Android

Setiap project di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file resource. Jenis modul meliputi :

- Modul aplikasi Android
- Modul Library
- Modul Google App Engine

Secara default, Android Studio menampilkan file project Anda dalam tampilan project Android, seperti yang ditunjukkan gambar 1. Tampilan ini disusun menurut modul untuk memberikan akses cepat ke file sumber utama project Anda.

Semua file build terlihat pada gambar di atas di bagian **Gradle Script** dan setiap modul aplikasi berisi folder berikut :

- **Manifest:** Berisi file **AndroidManifest.xml**
- **Java:** Berisi file kode sumber Java, termasuk kode pengujian JUnit.
- **Res:** Berisi semua resource non-kode, seperti tata letak XML, String UI, dan gambar bitmap







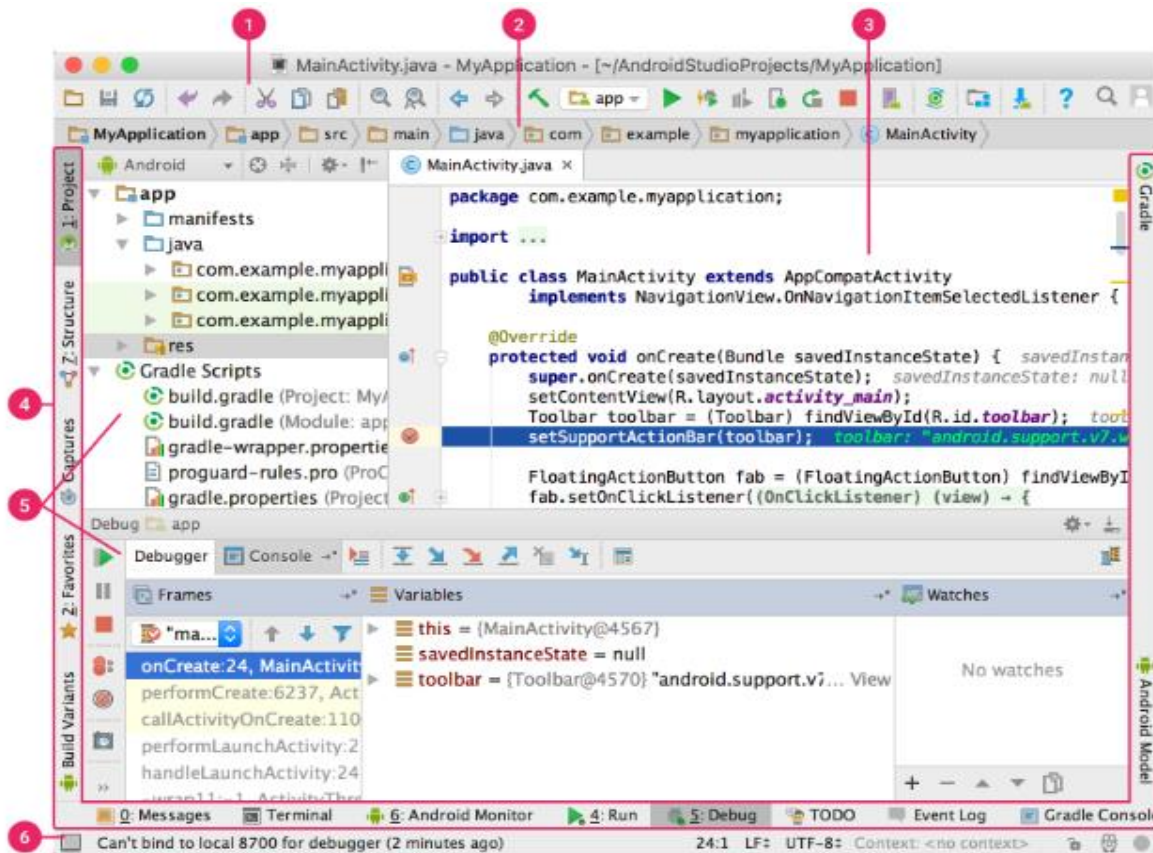






## Antarmuka Pengguna

Jendela utama Android Studio terdiri dari beberapa area logis yang diidentifikasi dalam gambar diawah ini.




Gambar 1.2 Jendela Utama Android Studio

1. **Toolbar** memungkinkan Anda melakukan berbagai tindakan, termasuk menjalankan aplikasi dan meluncurkan fitur Android.
2. **Menu Navigasi** membantu Anda menjelajah project dan membuka file untuk diedit. Menu ini memberikan tampilan struktur yang lebih ringkas yang terlihat di jendela **Project**.
3. **Jendela editor** adalah tempat Anda membuat dan memodifikasi kode. Tergantung jenis file yang ada, editor ini dapat berubah. Misalnya, saat menampilkan file tata letak, editor akan menampilkan Layout Editor.
4. **Panel jendela fitur** berada di sisi luar jendela IDE dan berisi tombol-tombol yang memungkinkan Anda memperluas atau menciutkan setiap jendela fitur.

5. **Jendela fitur** memberi Anda akses ke tugas tertentu seperti pengelolaan project, penelusuran, kontrol versi, dan banyak lagi. Anda dapat memperluas dan menciutkan jendela ini.
6. **Status bar** menampilkan status project Anda dan IDE itu sendiri, serta semua peringatan atau pesan.

## Jendela Fitur

Sebagai ganti menggunakan perspektif prest, Android Studio mengikuti konteks Anda dan Otomatis menampilkan jendela fitur yang relevan saat Anda bekerja. Secara default, jendela fitur yang paling umum digunakan disematkan ke panel fitur di tepi jendela.

- Untuk memperluas atau menciutkan jendela fitur, klik nama fitur di panel jendela fitur. Anda juga dapat menarik, menyematkan, melepaskan sematan, memasang, dan melepas jendela fitur.
- Untuk kembali ke tata letak jendela fitur default saat ini, klik **Window > Restore Default Layout** atau sesuaikan tata letak default dengan mengklik **Window > Store Current Layout as Default**.
- Untuk menampilkan atau menyembunyikan seluruh panel jendela fitur, klik ikon jendela  di pojok kiri bawah jendela Android Studio.
- Untuk menemukan jendela fitur tertentu, arahkan kursor ke atas ikon jendela dan pilih jendela fitur tersebut dari menu.

Anda juga bisa menggunakan pintasan keyboard untuk membuka jendela fitur. Tabel 1 mencantumkan pintasan jendela paling umum.

**Tabel 1.** Pintasan keyboard untuk beberapa jendela fitur yang bermanfaat .

Jenis	Deskripsi	Windows dan Linux	Mac
Basic Completion	Menampilkan saran dasar untuk variabel, jenis, metode, ekspresi, dan sebagainya. Jika memanggil Basic Completion dua kali berturut-turut, Anda akan melihat lebih banyak hasil, termasuk anggota pribadi dan anggota statis yang tidak diimpor.	Control+Space	Control+Space
Smart Completion	Menampilkan opsi yang relevan berdasarkan konteks. Smart Completion mengetahui jenis dan alur data yang diharapkan. Jika Anda memanggil Smart Completion dua kali berturut-turut, Anda akan melihat lebih banyak hasil, termasuk chain.	Control+Shift+Space	Control+Shift+Space
Statement Completion	Melengkapi pernyataan saat ini secara otomatis, seperti menambahkan tanda kurung, tanda kurung siku, tanda kurung kurawal, pemformatan, dll. yang tidak lengkap.	Control+Shift+Enter	Shift+Command+Enter

**Tabel 2.** Pintasan keyboard untuk pelengkapan kode.

Jenis	Deskripsi	Windows dan Linux	Mac
Basic Completion	Menampilkan saran dasar untuk variabel, jenis, metode, ekspresi, dan sebagainya. Jika memanggil Basic Completion dua kali berturut-turut, Anda akan melihat lebih banyak hasil, termasuk anggota pribadi dan anggota statis yang tidak diimpor.	Control+Space	Control+Space
Smart Completion	Menampilkan opsi yang relevan berdasarkan konteks. Smart Completion mengetahui jenis dan alur data yang diharapkan. Jika Anda memanggil Smart Completion dua kali berturut-turut, Anda akan melihat lebih banyak hasil, termasuk chain.	Control+Shift+Space	Control+Shift+Space
Statement Completion	Melengkapi pernyataan saat ini secara otomatis, seperti menambahkan tanda kurung, tanda kurung siku, tanda kurung kurawal, pemformatan, dll. yang tidak lengkap.	Control+Shift+Enter	Shift+Command+Enter

## Navigasi

Berikut ini beberapa tips untuk membantu Anda menjelajah di dalam Android Studio.

- Beralih antar file yang baru saja diakses menggunakan tindakan *Recent Files*. Tekan **Control+E** (**Command+E** pada Mac) untuk memunculkan tindakan Recent Files. Secara default, file yang terakhir diakses akan dipilih. Anda juga dapat mengakses jendela fitur mana saja melalui kolom kiri dalam tindakan ini.
- Lihat struktur file saat ini menggunakan tindakan *File Structure*. Munculkan tindakan File Structure dengan menekan **Control+F12** (**Command+F12** pada Mac). Dengan tindakan ini, Anda dapat membuka bagian mana pun dari file saat ini dengan cepat.
- Telusuri dan buka class tertentu dalam project menggunakan tindakan *Navigate to Class*. Munculkan tindakan ini dengan menekan **Control+N** (**Command+O** pada Mac). Navigate to Class mendukung ekspresi canggih, termasuk camel humps, jalur, baris navigasi ke, pencocokan nama tengah, dan banyak lagi. Jika Anda memanggilnya dua kali berturut-turut, hasil dari class project akan ditampilkan.
- Buka file atau folder menggunakan tindakan *Navigate to File*. Munculkan tindakan Navigate to File dengan menekan **Control+Shift+N** (**Command+Shift+O** pada Mac). Untuk menelusuri folder dan bukan file, tambahkan / (garis miring) di akhir ekspresi Anda.
- Buka metode atau kolom menurut nama menggunakan tindakan *Navigate to Symbol*. Munculkan tindakan Navigate to Symbol dengan menekan **Control+Shift+Alt+N** (**Command+Option+O** pada Mac).
- Temukan semua bagian kode yang merujuk ke class, metode, kolom, parameter, atau pernyataan di posisi kursor saat ini dengan menekan **Alt+F7** (**Option+F7** pada Mac).

## Gaya dan Pemformatan

Saat Anda mengedit, Android Studio otomatis menerapkan pemformatan dan gaya seperti yang ditentukan dalam setelan gaya kode Anda. Anda dapat menyesuaikan setelan gaya kode menurut Bahasa pemrograman, termasuk menentukan konvensi untuk tab dan indentasi, spasi, penggabungan, tanda kurung kurawal, dan baris kosong. Untuk menyesuaikan setelan gaya kode

ANDA, KLIK **File > Setting > Editor > Code Style (Preferences > Editor > Code Style** pada Laptop.)

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    // initialize object  
    buttonScan = (Button) findViewById(R.id.buttonScan);  
    textViewNomor = (TextView) findViewById(R.id.textViewNomor);  
    textViewNama = (TextView) findViewById(R.id.textViewNama);  
}
```

Gambar 1.3 Kode Sebelum Pemformatan

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    // initialize object  
    buttonScan = (Button) findViewById(R.id.buttonScan);  
    textViewNomor = (TextView) findViewById(R.id.textViewNomor);  
}
```

Gambar 1.4 Kode Setelah Pemformatan

### Dasar-dasar control versi

Android Studio mendukung berbagai sistem kontrol versi (VCS), termasuk Git, GitHub, CVS, Mercurial, Subversion, dan Google Cloud Source Repositories.

Setelah mengimpor aplikasi Anda ke Android Studio, gunakan opsi menu VCS pada Android Studio untuk mengaktifkan dukungan VCS bagi sistem kontrol versi yang diinginkan, membuat repositori, mengimpor file baru ke kontrol versi, dan menjalankan operasi kontrol versi lainnya:

1. Dari menu **VCS** Android Studio, klik **Enable Version Control Integration**.
2. Dari menu drop-down, pilih sistem kontrol versi yang ingin dikaitkan dengan root project, lalu klik **OK**.

Menu VCS sekarang menampilkan sejumlah opsi kontrol versi berdasarkan sistem yang Anda pilih .

### Sistem build Gradle

Android Studio menggunakan Gradle sebagai dasar dari sistem build, dengan lebih banyak kemampuan khusus Android yang disediakan oleh [plugin Android untuk Gradle](#). Sistem build ini

berjalan sebagai fitur terintegrasi dari menu Android Studio, dan terpisah dari command line. Anda dapat menggunakan fitur-fitur sistem build untuk:

- Menyesuaikan, mengonfigurasi, dan memperluas proses pembuatan build.
- Membuat banyak APK untuk aplikasi Anda, dengan berbagai fitur yang menggunakan project dan modul yang sama.
- Menggunakan kembali kode dan resource di seluruh set sumber.

### **Varian Build**

Sistem build dapat membantu Anda membuat beberapa versi berbeda untuk aplikasi yang sama dari satu project. Hal ini berguna saat Anda menyediakan aplikasi dalam versi gratis dan berbayar, atau jika Anda ingin mendistribusikan beberapa APK untuk berbagai konfigurasi perangkat di Google Play.

### **Dukungan multi-APK**

Dukungan multi-APK memungkinkan Anda membuat beberapa APK sekaligus secara efisien berdasarkan kepadatan layar atau ABI. Misalnya, Anda dapat membuat APK aplikasi terpisah untuk kepadatan layar hdpi dan mdpi, dengan tetap menganggapnya sebagai varian tunggal serta mengizinkannya berbagi setelan APK pengujian, javac, dx, dan ProGuard.

### **Penyusutan resource**

Penyusutan resource di Android Studio otomatis menghapus resource yang tidak digunakan dari aplikasi terpaket dan dependensi library Anda. Misalnya, jika aplikasi Anda menggunakan [layanan Google Play](#) untuk mengakses fungsi Google Drive, dan saat ini Anda tidak menggunakan [Login dengan Google](#), penyusutan resource dapat menghapus beragam aset yang dapat digambar untuk tombol `SignInButton`.

### **Mengelola Dependensi**

Dependensi untuk project Anda ditentukan oleh nama dalam file `build.gradle`. Gradle menangani penemuan dependensi dan menyediakannya di build Anda. Anda dapat mendeklarasikan dependensi modul, dependensi biner jarak jauh, dan dependensi biner lokal di file `build.gradle` Anda.



## Fitur profil dan debug

Gunakan proses debug inline untuk menyempurnakan panduan kode Anda dalam tampilan debugger dengan verifikasi inline untuk referensi, ekspresi, dan nilai variabel. Informasi debug inline meliputi:

- Nilai variabel inline
- Objek perujuk yang merujuk ke objek terpilih
- Nilai yang dihasilkan metode
- Ekspresi operator dan lambda
- Nilai tooltip

## Anotasi di Android Studio

Android Studio mendukung anotasi variabel, parameter, dan nilai kembalian untuk membantu Anda merekam bug, seperti pengecualian pointer null dan konflik jenis resource. Android SDK Manager mengemas library Support-Annotations di Android Support Repository untuk digunakan dengan Android Studio. Android Studio memvalidasi anotasi yang sudah dikonfigurasi selama pemeriksaan kode.

## Pesan Log

Saat membuat dan menjalankan aplikasi dengan Android Studio, Anda bisa melihat output **adb** dan pesan log perangkat di jendela **Logcat**.

## MANFAAT MEMPELAJARI ANDROID STUDIO

- Dengan mempelajari Android Studio dapat membantu Anda untuk mempercepat pembuatan aplikasi yang Anda inginkan.
- Android Studio merupakan sebuah tools yang mudah dipahami dan digunakan.
- Dalam satu tools ini Anda bisa mendapatkan berbagai manfaat mulai dari pembuatan aplikasi hingga testing aplikasi.
- Bahkan, dengan belajar Android Studio maka Anda bisa menghemat waktu kerja untuk dapat lebih produktif.
- Dapat memperdalam ilmu codingan dengan baik. Karena dalam android studi diberikan beberapa referensi ketika Anda mengetik sintaks. Dengan begitu tentunya Anda akan mencari tahu apa saja kegunaan dari sintaks yang terdapat.

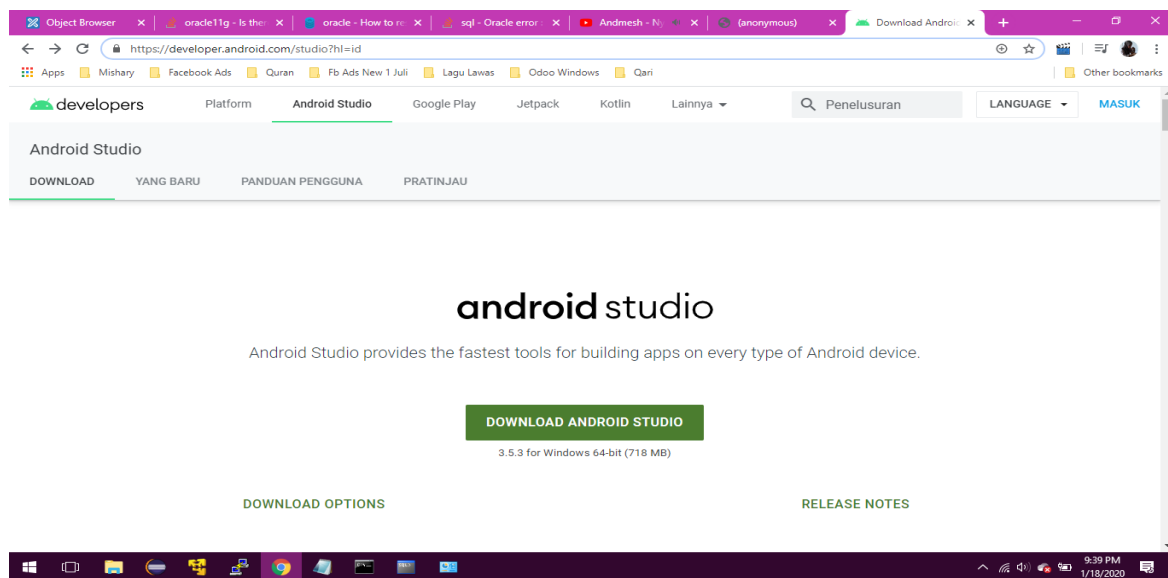
- Sarana pembelajaran coding dan pembuatan aplikasi yang baik dan praktis hanya dengan Android Studio.

## **Langkah-langkah dalam membuat aplikasi Debugger Pada Aplikasi Monitoring Barang Masuk Dan Keluar Pada Inventory Barang Dengan Scanner**

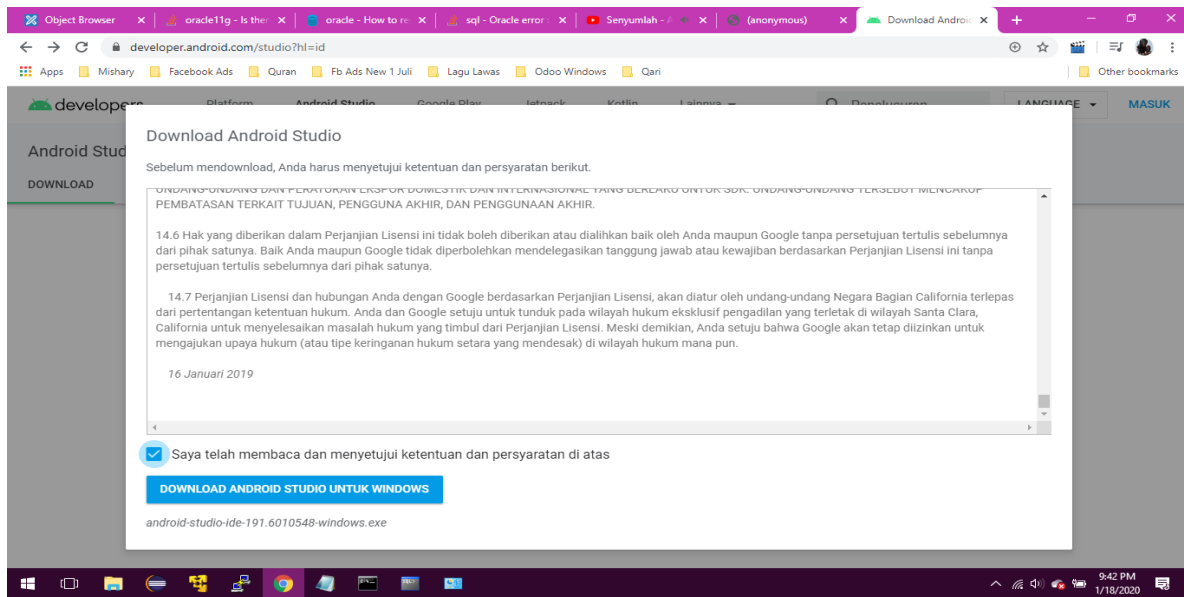
### **1.Download terlebih dahulu aplikasi android dengan langkah-langkah sebagai berikut :**

a. Kunjungi website resmi dari android studio yaitu

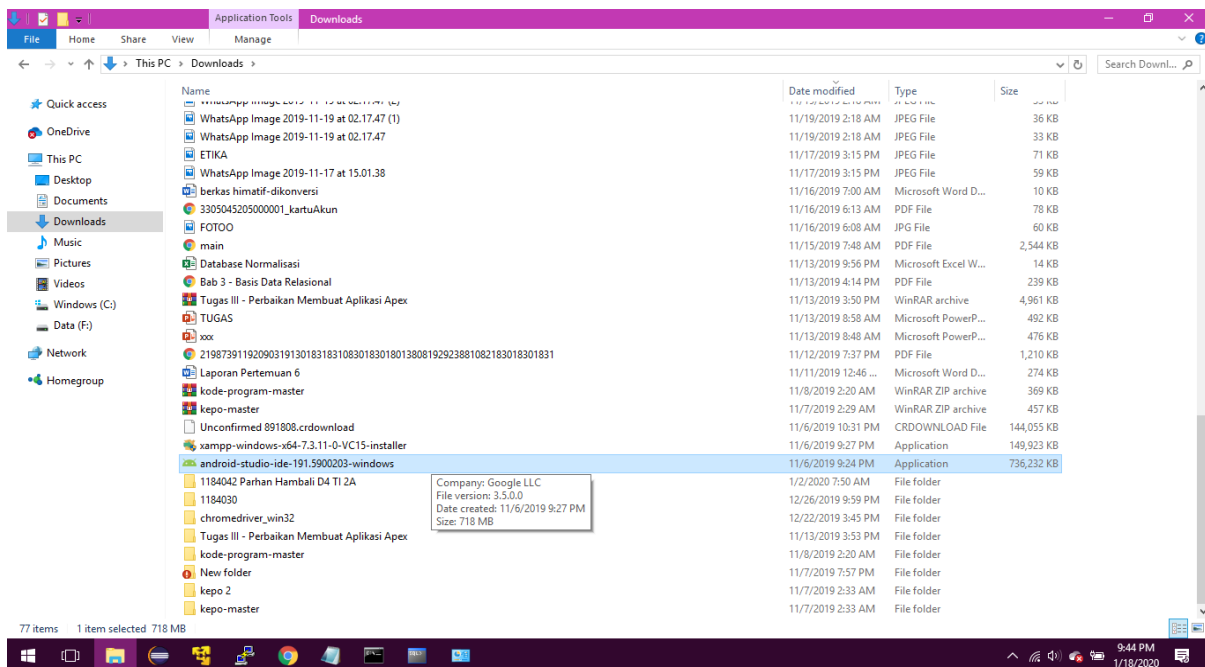
<https://developer.android.com/studio?hl=id> .



b. Untuk mendownload aplikasi android studio ini harus melakukan syarat perizinan terlebih dahulu dengan cara berikan tanda centang pada kotak yang tersedia jika anda menyetujuinya .



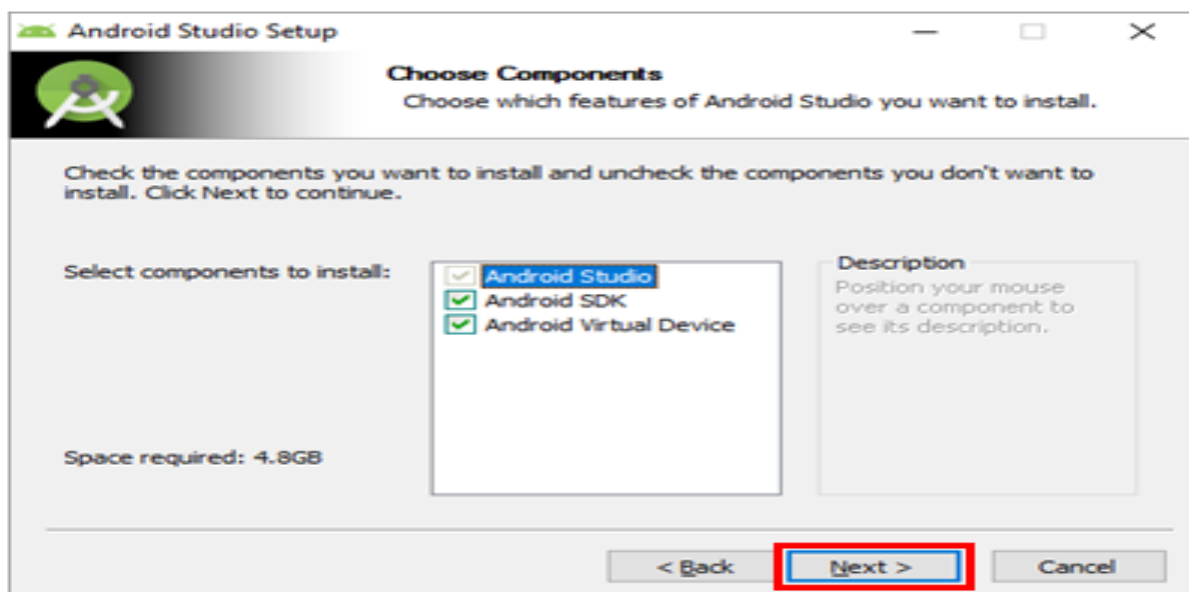
c. Lalu untuk mengecek aplikasi sudah terdownload atau belum anda bisa masuk ke dalam PC lalu lihat dalam Downloads. Dan di dalam download sudah terdapat android studio yang sudah ter download .

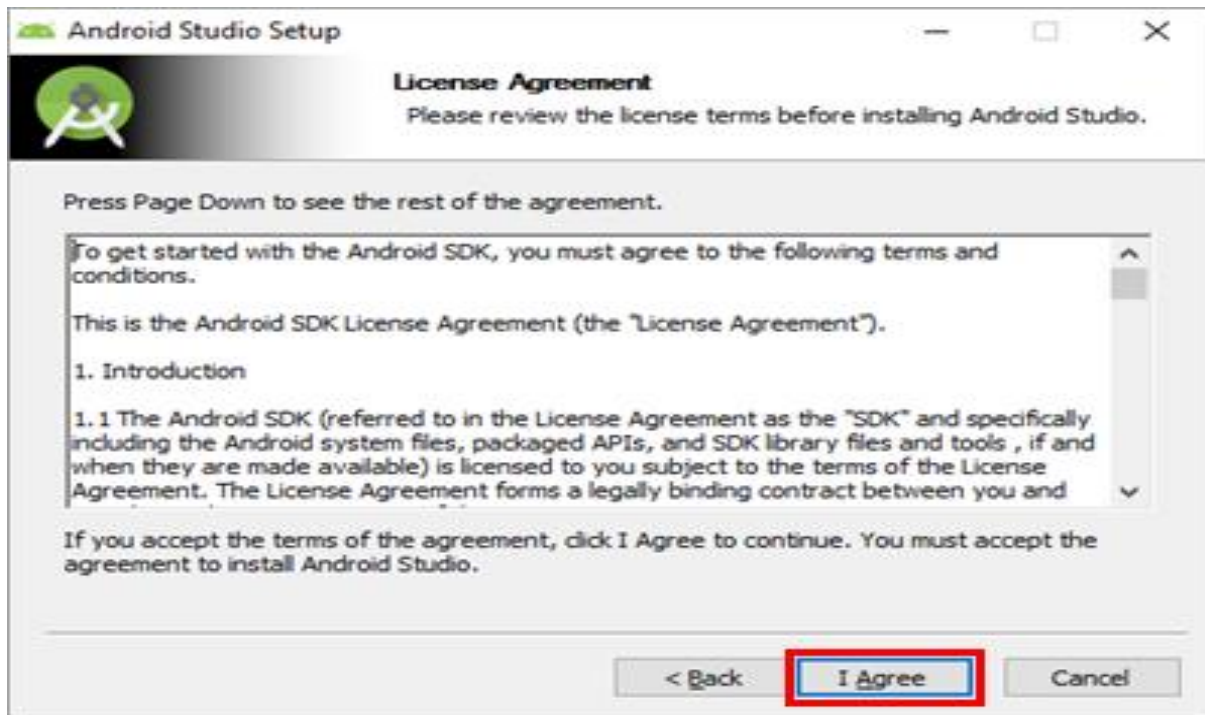


d. Kemudian klik android studio yang sudah di downlad tadi lalu akan mncul tampilan seperti ini, lalu klik Next saja .

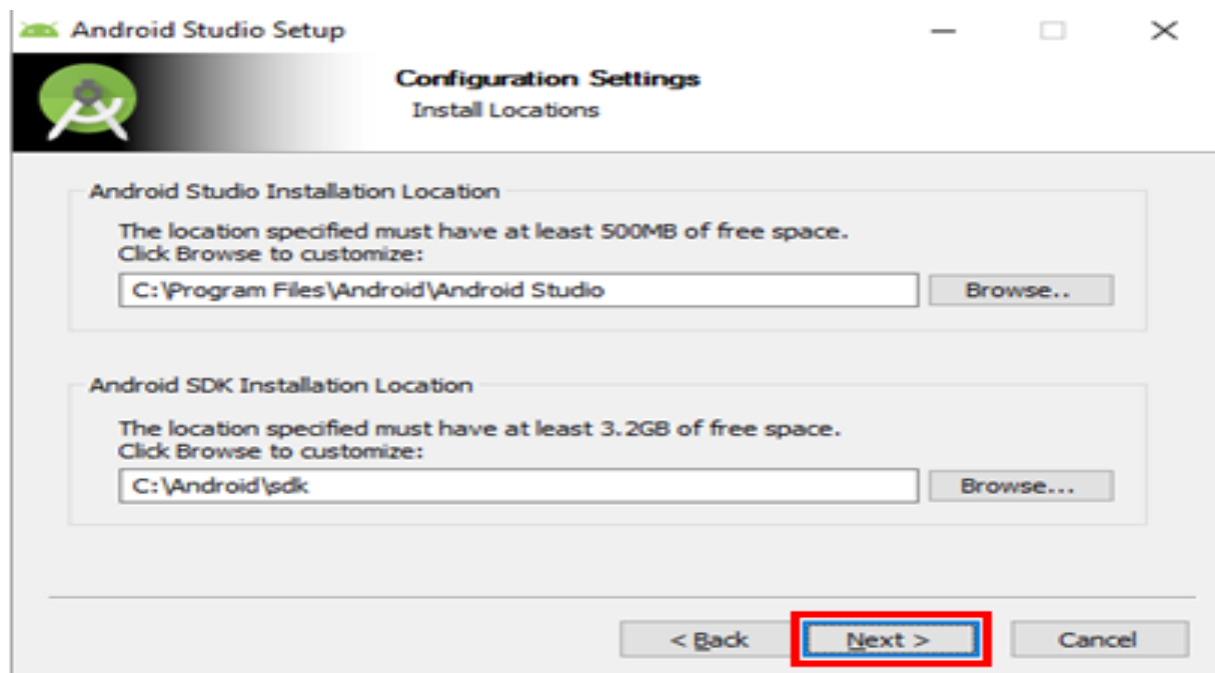


e. Kemudian pilih salah satu komponen untuk menginstall android studio dengan cara memberikan tanda centang pada kotak yang telah disediakan. Setelah memberikan tanda centang pada kotak yang diberikan klik Next .

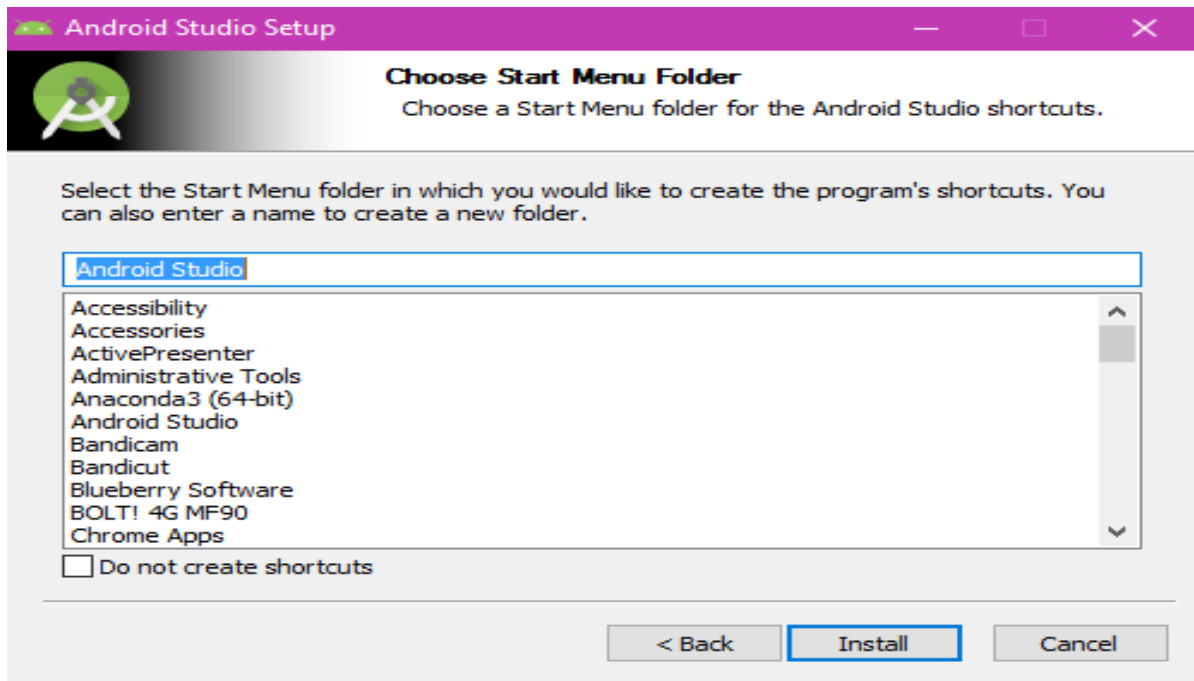




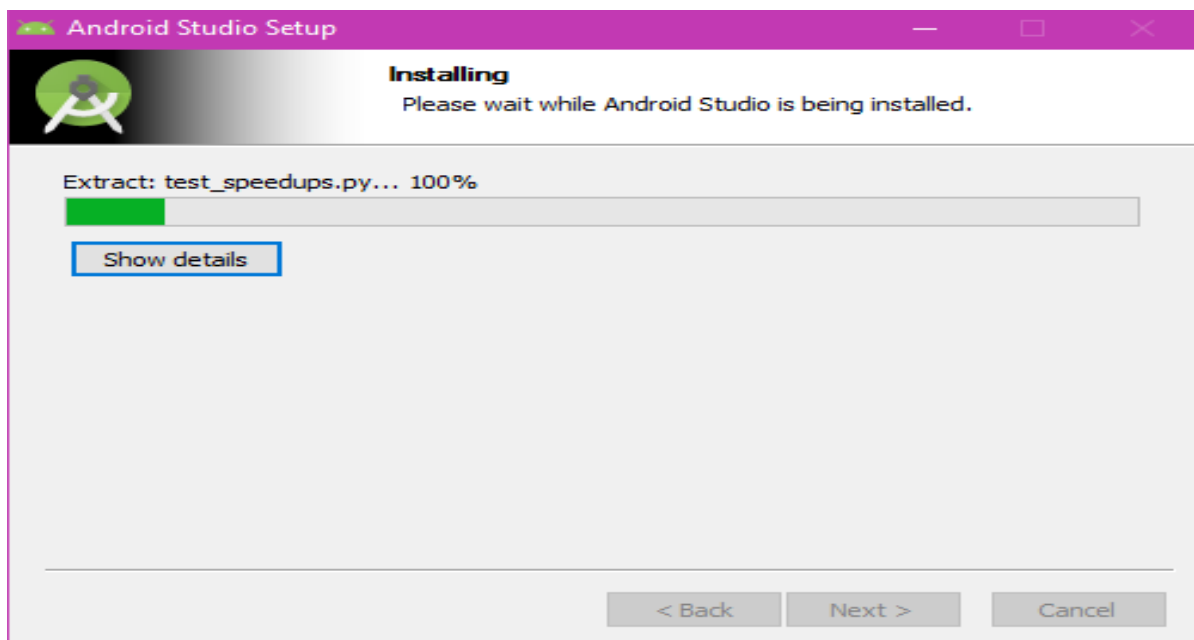
f. Lalu pilih lokasi untuk install Android Studio. Disini lokasi Android Studio saya berada pada Windows C. Untuk install lokasi Android Studio bebas dimana saja dengan syarat penyimpanan tidak kurang dari 500MB lalu Next .



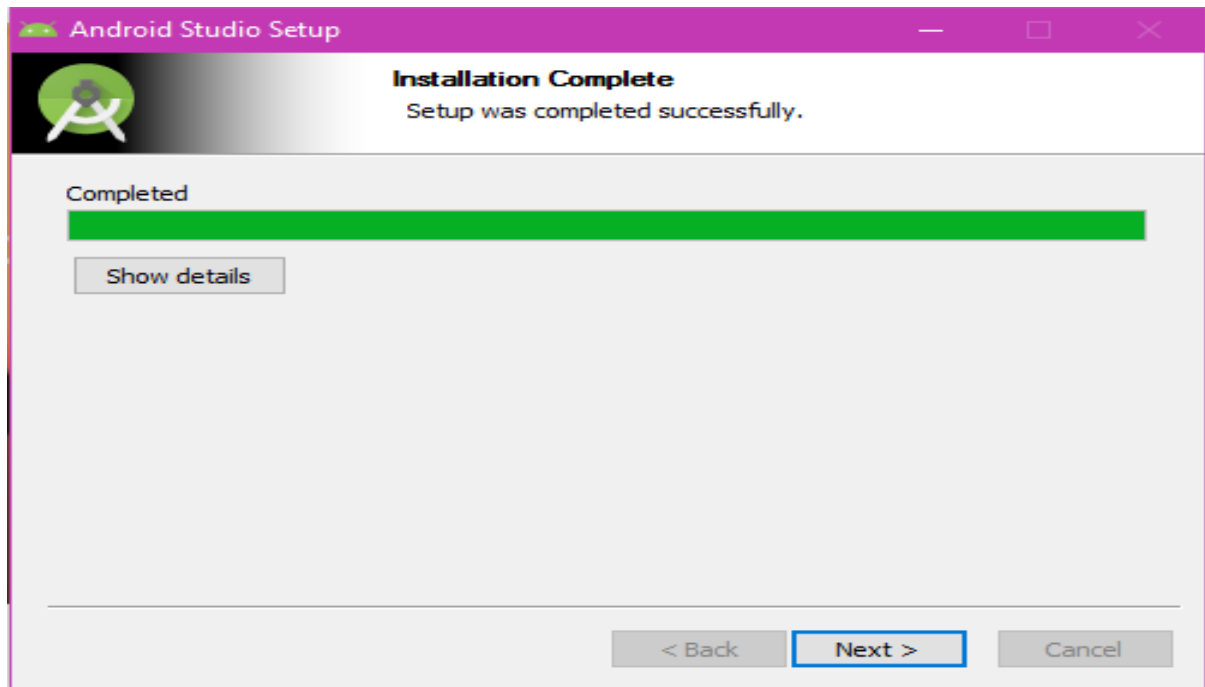
g. Pilih untuk Start Menu Folder dengan memilih Android Studio . Setelah memilih untuk Start Menu Folder lalu klik Install .



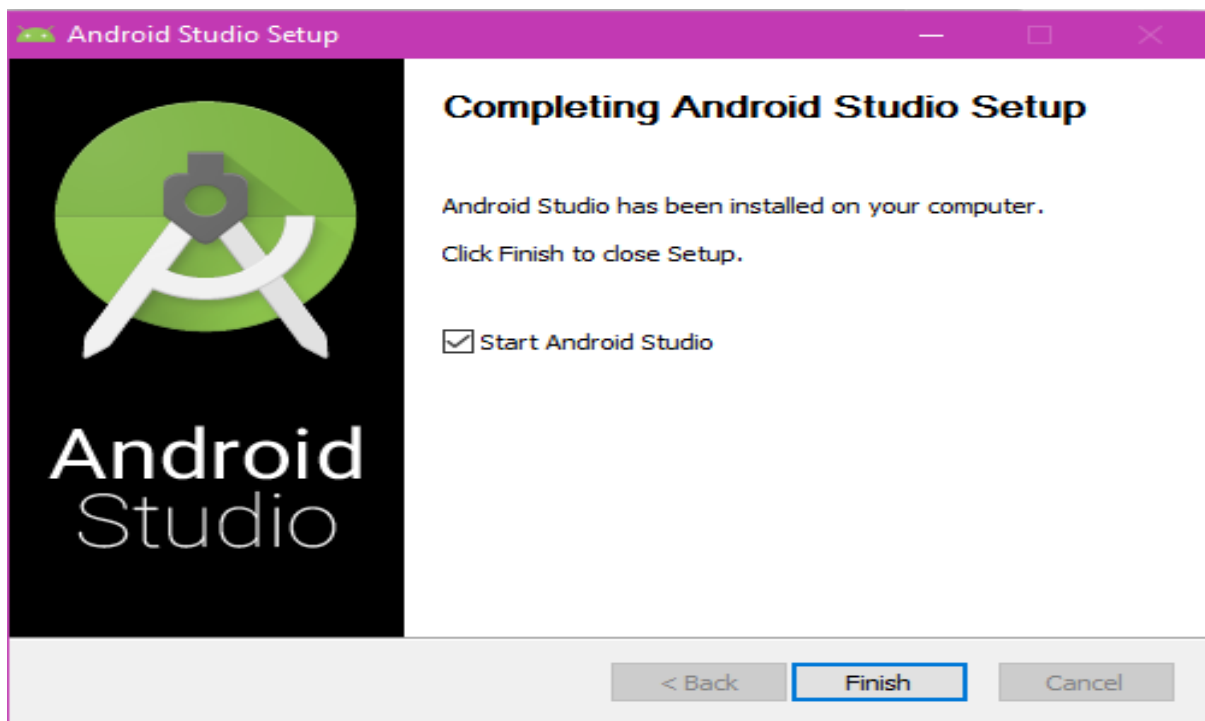
h. Tampilan loading untuk Install Android Studio, tunggu sampai install completed .



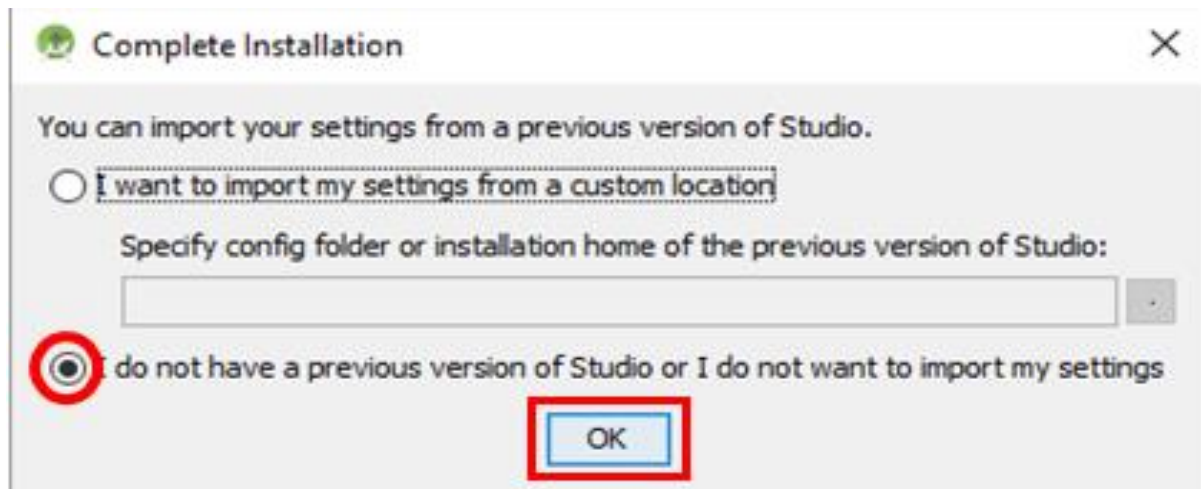
- i. Setelah tampilan pada loading sudah menjadi warna hijau semua atau sudah completed maka lanjutkan klik Next .



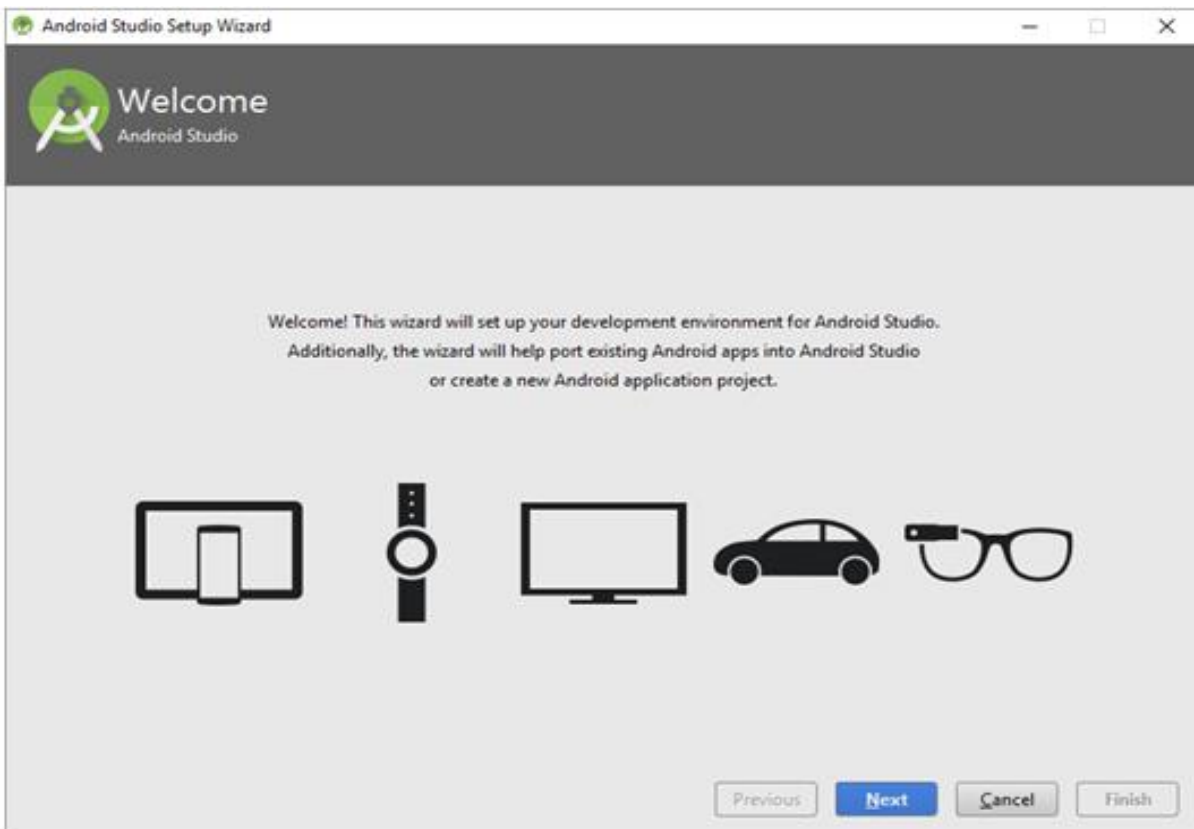
- j. Setelah completed selanjutnya klik Finish .



k. Kemudian akan muncul jendela baru dengan 2 opsi pilihan. Checklist opsi kedua jika kalian belum pernah menginstall IDE Andoid Srtudio dan klik Ok.

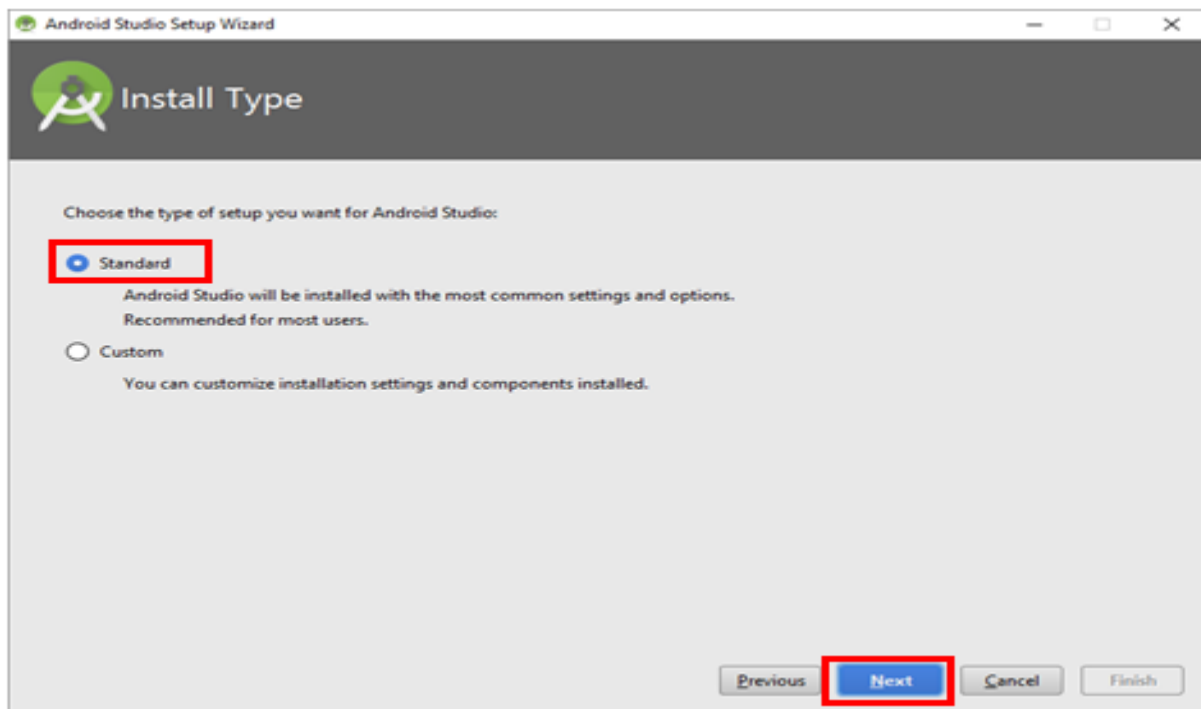


l, Selanjutnya akan muncul jendela welcome dan Next pada Android Studio

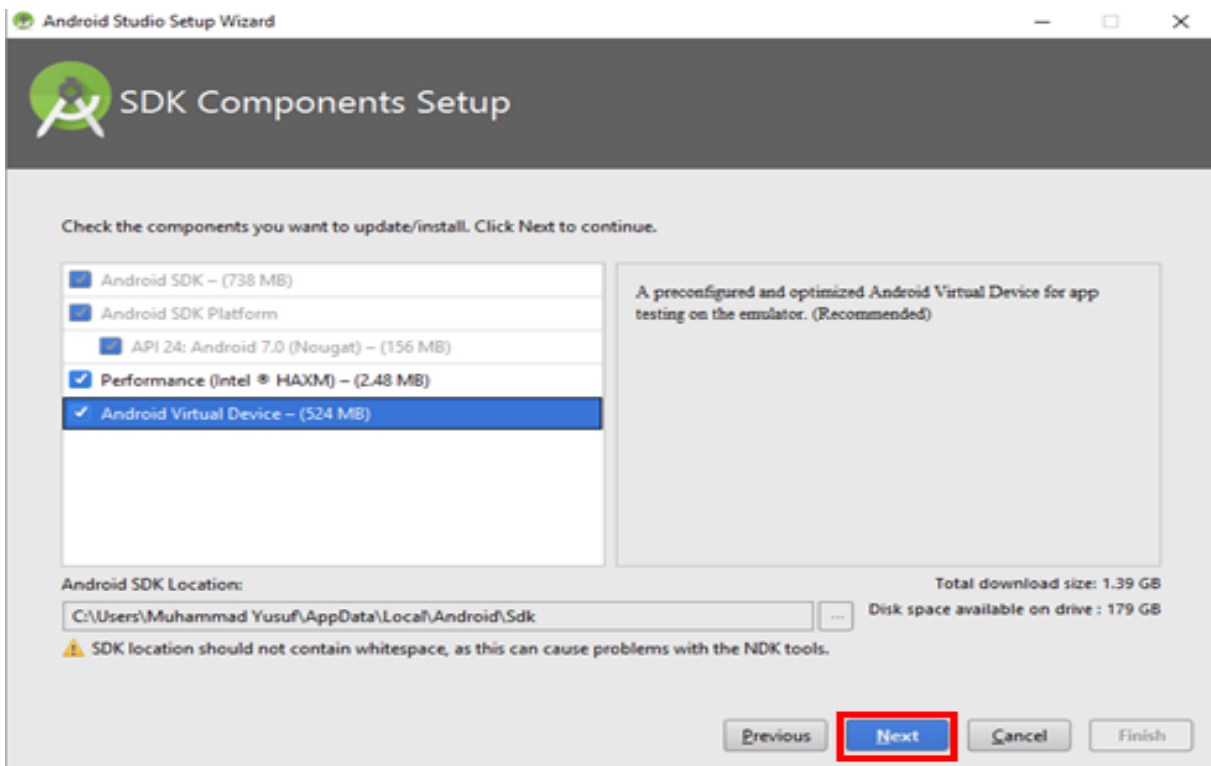




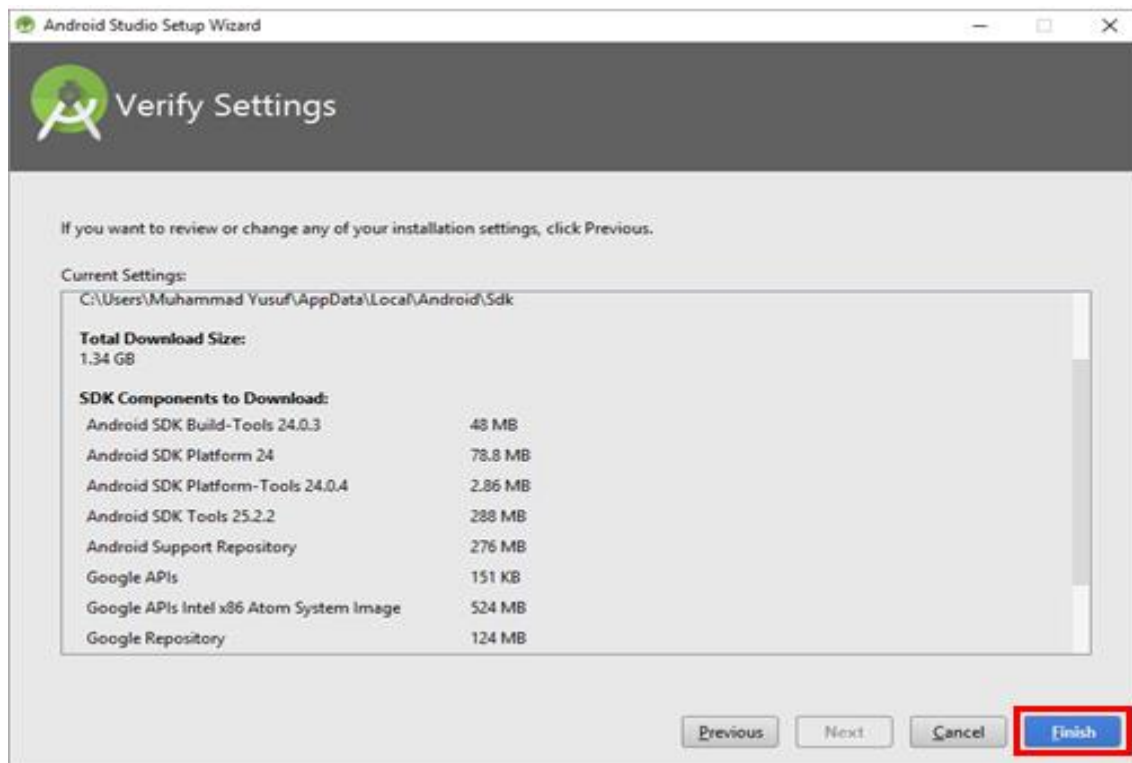
m. Selanjutnya pilih Standard dan klik Next



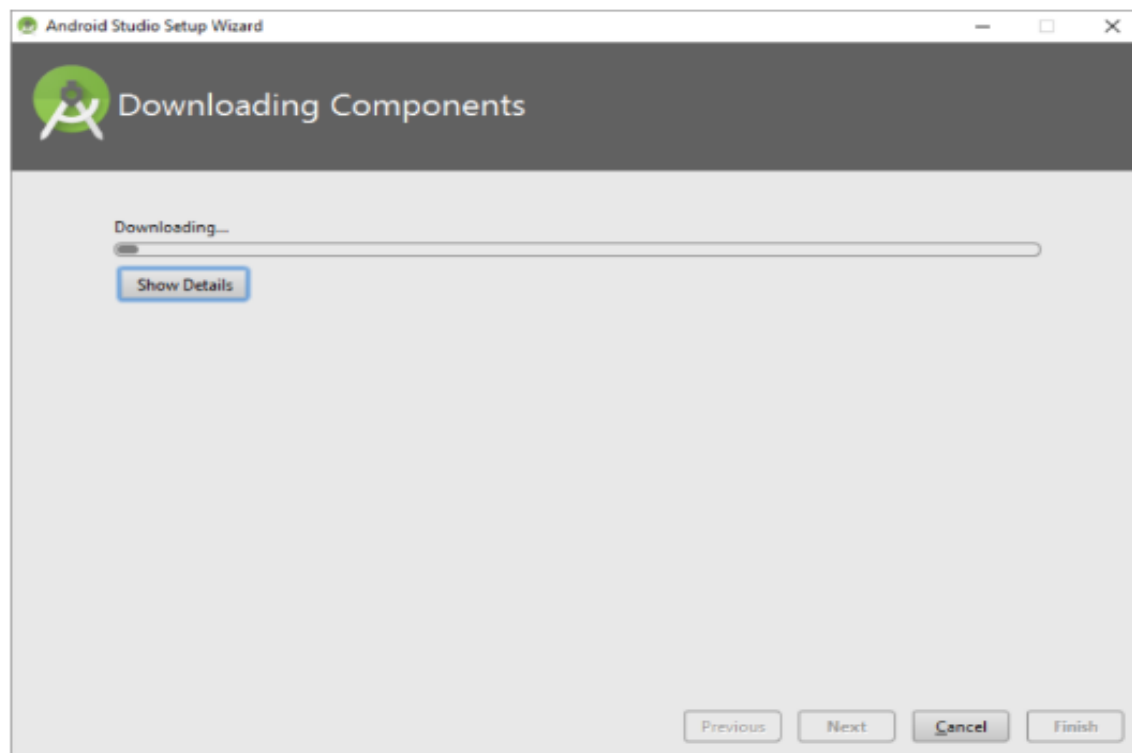
n. Setelah itu akan muncul jendela SDK Component Setup. Pilih yang ingin di install/update



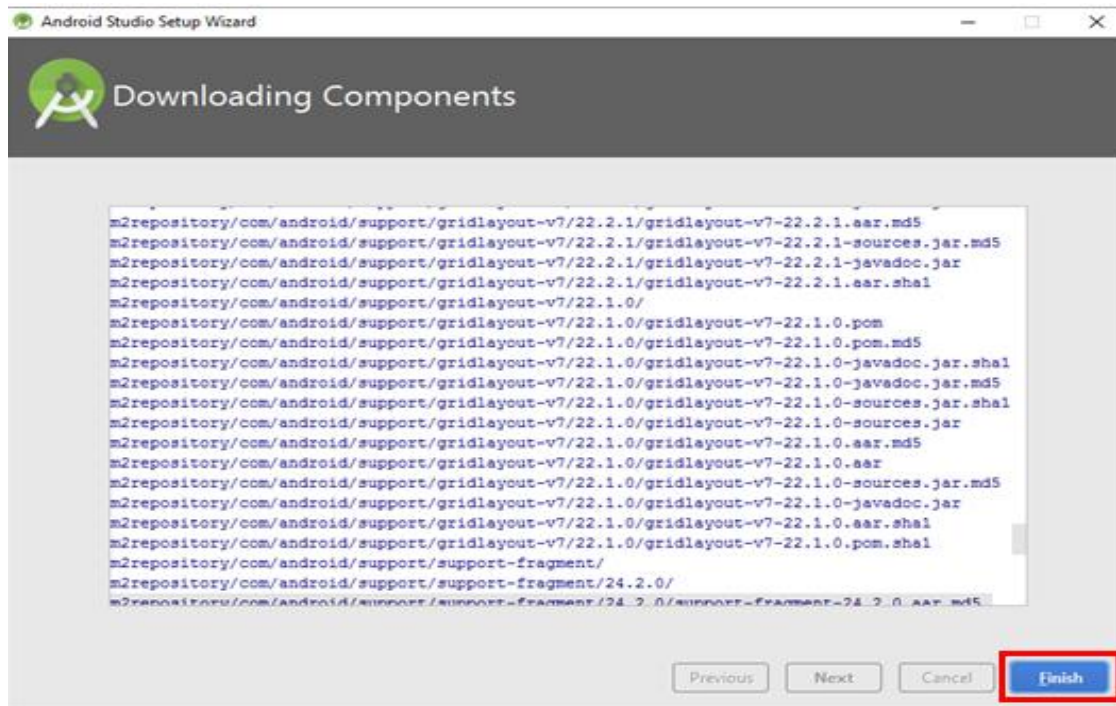
o. Setelah itu klik Finish



p. Maka akan tampil jendela Downloading Componen, tunggu hingga proses downloading selesai



q. Setelah proses download selesai maka proses instalasi Android Studio telah selesai, Klik Finish

















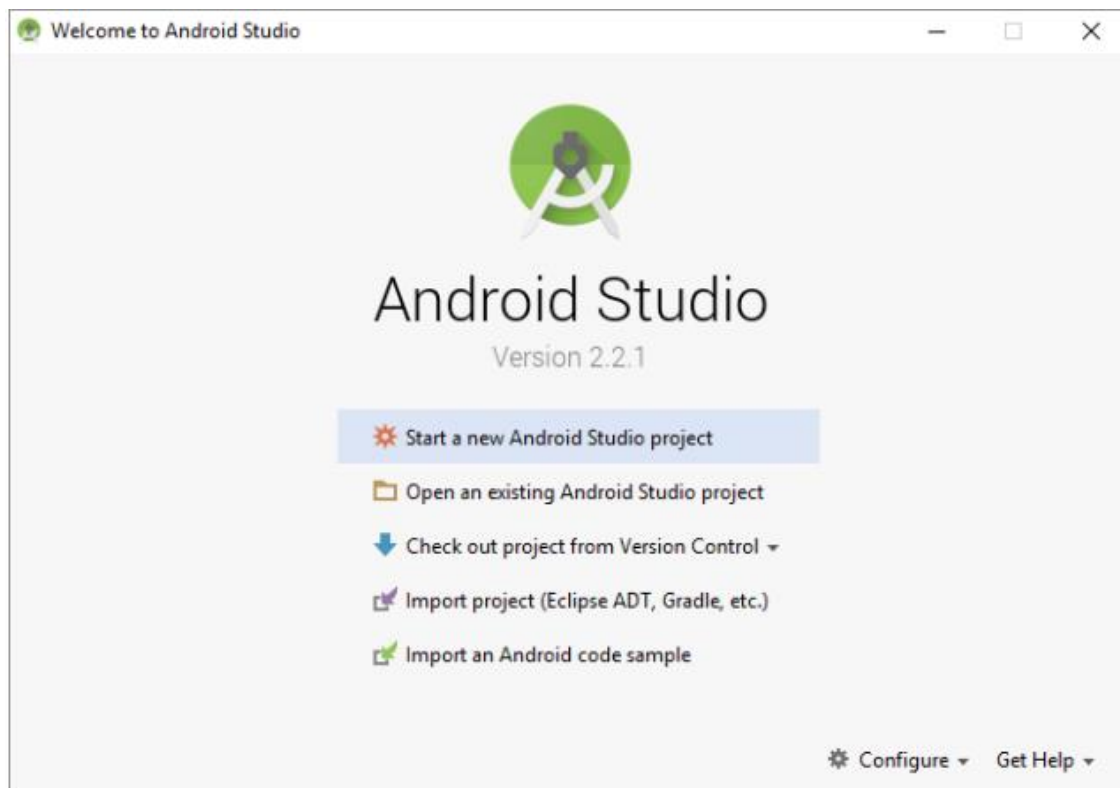




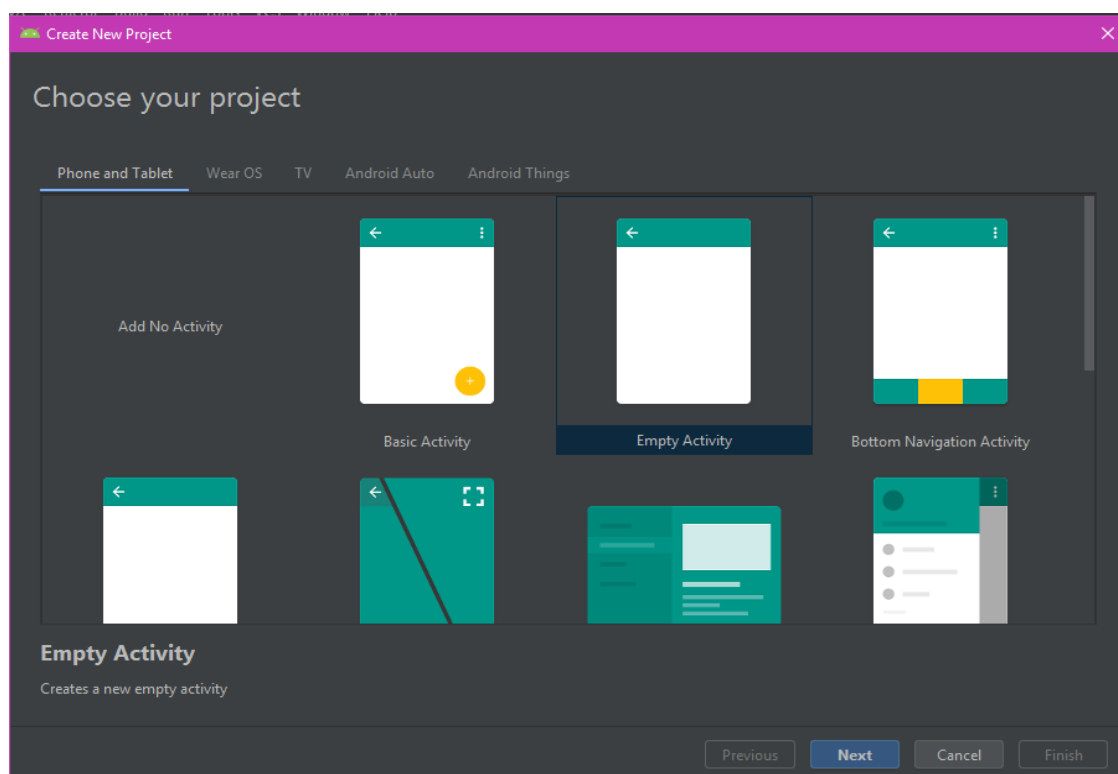




r. Setelah itu akan muncul jendela baru yaitu Welcome to Android Studio



s. Kemudian Pilih salah satu layout activity untuk project Anda



# ZXing

## Apa itu ZXing?

Dikutip dari situs [github](#)-nya bahwasanya ZXing adalah sebagai berikut.

*ZXing (“Zebra crossing”) is an open-source, multi-format 1D/2D barcode image processing library implemented in Java, with ports to other languages.*

Jadi, ZXing merupakan suatu pustaka kode terbuka untuk menangani fungsi-fungsi barcode scanner. Untuk penggunaannya di Android terdapat banyak pustaka-pustaka lain yang bisa Anda gunakan dan salah satunya yang bisa Anda gunakan adalah sebagai berikut.

[dm77/barcodescanner](#)

[barcodescanner](#) – Barcode Scanner Libraries for Android  
[github.com](#)

Developer tersebut telah menulis ulang fungsi-fungsi utama ZXing agar lebih mudah digunakan di Android.

## Memulai Contoh Proyek

Buat proyek di Android Studio dengan nama **Scannerku** dan pastikan Anda meng-checklist Kotlin sebagai bahasa pemrogramannya.

# Konfigurasi build.gradle

Berdasarkan dari situs github-nya <https://github.com/dm77/barcodescanner> kita harus menambahkan dependency berikut pada file **build.gradle**.  
`implementation 'me.dm7.barcodescanner:zxing:1.9.8'`

Apakah Anda mendapatkan pesan error berikut setelah melakukan sync gradle.



Konflik dependency build.gradle

Pada gambar diatas, pesan error tersebut terjadi dikarenakan adanya konflik antara dependency yang Anda masukkan. Dimana, konflik dependency diatas terjadi karena ada konflik antara dependency `com.android.support:appcompat-v7:27.1.1` yang telah memakai versi terbaru sedangkan, dependency `me.dm7.barcodescanner:zxing:1.9.8` ada memakai dependency bawaannya sendiri namun, dengan versi yang berbeda yang dari punya Anda sehingga terjadilah konflik tersebut. Jika Anda sorot kursur Anda ke pesan error-nya maka, Android Studio akan memberitahu dependency apa saja yang



konflik. Contoh, diatas yang konflik itu adalah `com.android.support:support-media-compat` yang masih memakai versi 25.3.1. Untuk solusinya ada 2 yaitu, sebagai berikut.

1. Anda paksa dependency yang konflik tersebut ikut dengan versi Anda caranya ialah dengan menambahkan dependency konflik tersebut ke file **build.gradle**
2. Yang satu lagi ialah Anda membuat pengecualian di dependency parent-nya yang menyebabkan konflik tadi. Pada contoh diatas yang menyebabkan konflik adalah dependency `me.dm7.barcodescanner:zxing:1.9.8` dan untuk membuat pengecualiannya tinggal Anda tambahkan keyword `exclude` seperti berikut dimana, maksud dari syntax tersebut adalah mengecualikan dependency dari `com.android.support` milik si `me.dm7.barcodescanner:zxing:1.9.8` tidak ikut pada projek Anda.

```
implementation('me.dm7.barcodescanner:zxing:1.9.8') {  
    exclude group: 'com.android.support'  
}
```

## Permissions

Untuk menggunakan fitur kamera Anda harus menambahkan permission kamera di **AndroidManifest.xml** seperti berikut.

```
<uses-permission android:name="android.permission.CAMERA" />
```

## Layout Utama

Silakan buka file **activity\_main.xml** dan isi dengan source code berikut.

```
<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.ConstraintLayout  
xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"
```

```

        android:layout_height="match_parent"
        android:padding="16dp"
        tools:context=".MainActivity">

        <android.support.constraint.Guideline
            android:id="@+id/guideline"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            app:layout_constraintGuide_percent="0.8" />

        <TextView
            android:id="@+id/text_view_qr_code_value"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="QR Code Value"
            android:textSize="18sp"
            android:gravity="center_horizontal"
            android:layout_marginTop="16dp"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/guideline" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/button_reset"
            android:text="Reset"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"

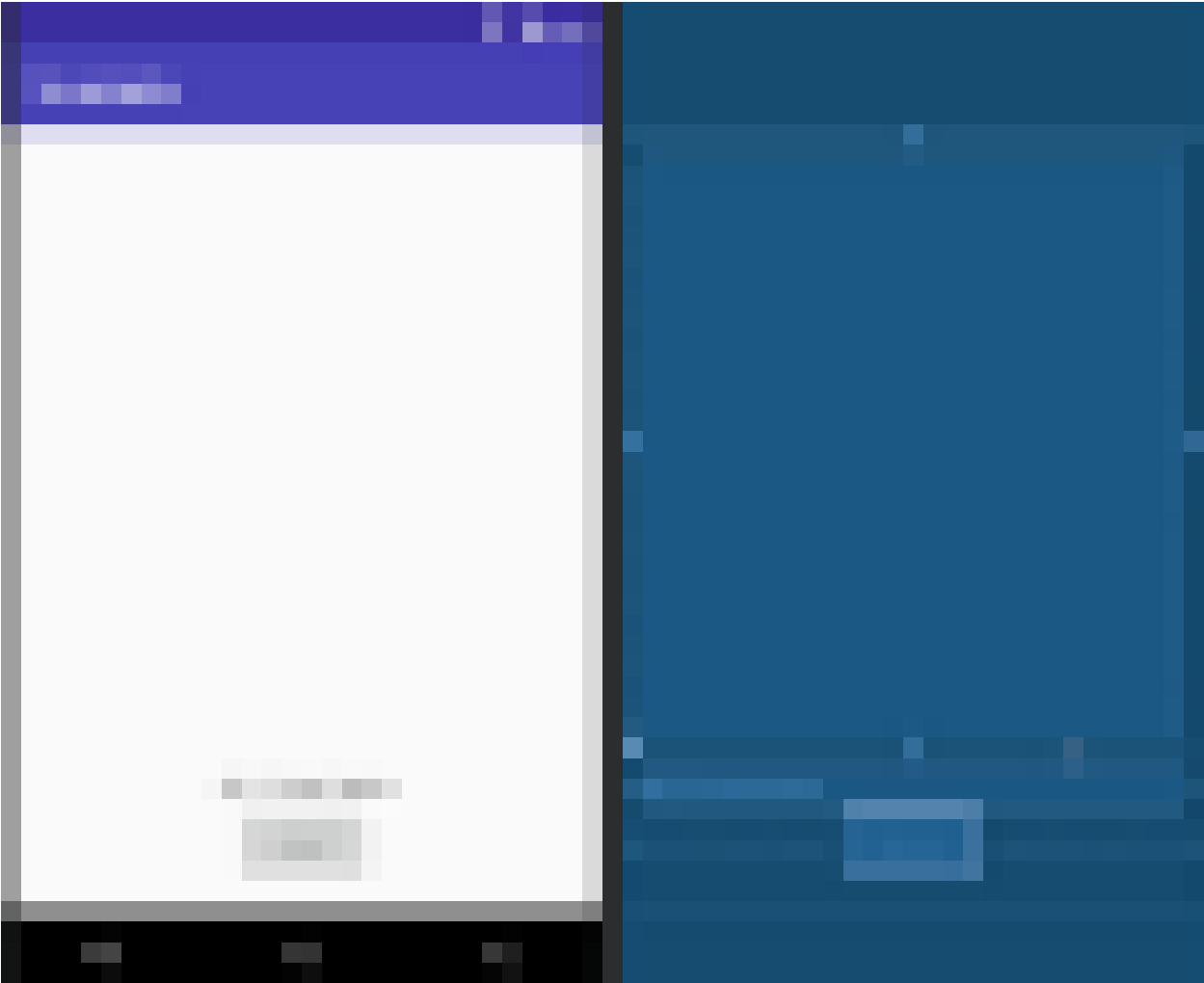
app:layout_constraintTop_toBottomOf="@+id/text_view_qr_code_value"
/>

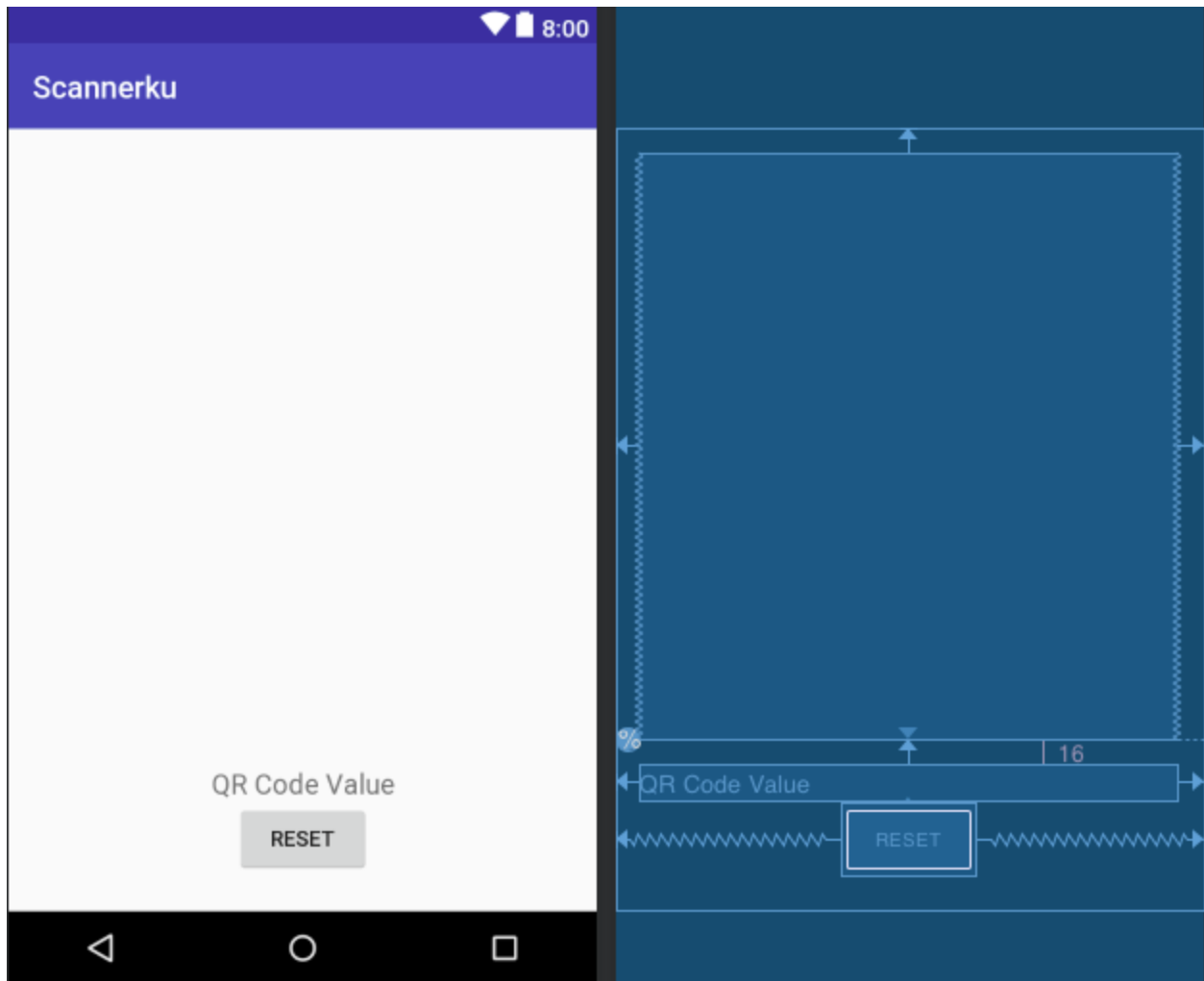
        <FrameLayout
            android:id="@+id/frame_layout_camera"
            android:layout_width="match_parent"
            android:layout_height="0dp"
            app:layout_constraintBottom_toTopOf="@+id/guideline"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

    </android.support.constraint.ConstraintLayout>

```

Dan output dari source code layout diatas adalah seperti berikut.





Design Layout Utama

## Penambahan Fungsi Utama

Selanjutnya, Anda buka file **MainActivity.kt** dan isi dengan source code berikut.

```
package com.ysn.scannerku

import android.Manifest
import android.content.pm.PackageManager
import android.os.Build
import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.view.View
import com.google.zxing.Result
import kotlinx.android.synthetic.main.activity_main.*
```

```

import me.dm7.barcodescanner.zxing.ZXingScannerView

class MainActivity : AppCompatActivity(),
ZXingScannerView.ResultHandler, View.OnClickListener {

    private lateinit var mScannerView: ZXingScannerView
    private var isCaptured = false

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        initScannerView()
        initDefaultView()
        button_reset.setOnClickListener(this)
    }

    private fun initScannerView() {
        mScannerView = ZXingScannerView(this)
        mScannerView.setAutoFocus(true)
        mScannerView.setResultHandler(this)
        frame_layout_camera.addView(mScannerView)
    }

    override fun onStart() {
        mScannerView.startCamera()
        doRequestPermission()
        super.onStart()
    }

    private fun doRequestPermission() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            if (checkSelfPermission(Manifest.permission.CAMERA) !=
PackageManager.PERMISSION_GRANTED) {

requestPermissions(arrayOf(Manifest.permission.CAMERA), 100)
            }
        }
    }

    override fun onRequestPermissionsResult(requestCode: Int,
permissions: Array<out String>, grantResults: IntArray) {
        when (requestCode) {
            100 -> {
                initScannerView()
            }
            else -> {
                /* nothing to do in here */
            }
        }
    }

```

```

        }
    }
}

override fun onPause() {
    mScannerView.stopCamera()
    super.onPause()
}

private fun initDefaultView() {
    text_view_qr_code_value.text = "QR Code Value"
    button_reset.visibility = View.GONE
}

override fun handleResult(rawResult: Result?) {
    text_view_qr_code_value.text = rawResult?.text
    button_reset.visibility = View.VISIBLE
}

override fun onClick(view: View?) {
    when (view?.id) {
        R.id.button_reset -> {
            mScannerView.resumeCameraPreview(this)
            initDefaultView()
        }
        else -> {
            /* nothing to do in here */
        }
    }
}
}
}

```

## Penjelasan:

Untuk menambahkan fungsi scanner pada ZXing Anda harus menggunakan `ZXingScannerView.ResultHandler` dimana, *callback* ini berfungsi untuk menerima deteksi otomatis dari hasil scanner si ZXing. Selanjutnya, initialize object `ZXingScannerView` dan masukkan ke `FrameLayout` yang sudah Anda buat di xml-nya tadi. Kemudian, dikarenakan fitur ini menggunakan kamera jelas Anda memerlukan permission camera dimana, untuk Android versi M keatas itu memerlukan runtime permission sehingga pada kode

diatas Anda ada menambahkan runtime permission. Kemudian, ketika method `handleResult` terpanggil maka, Anda ada melakukan `setText` pada `text_view_qr_code_value` dan menampilkan `button_reset` . Selanjutnya, pada `button_reset` Anda juga ada menambahkan `onClickListener` dimana, didalamnya ada fungsi untuk mereset kamera dan semua view-nya kembali ke awal. Dan terakhir, jangan lupa untuk menambahkan fungsi `startCamera` di method `onStart` dan `stopCamera` di method `onPause` .

## Output

Berikut adalah output dari projek yang Anda buat.





Output Projek

Untuk projeknya bisa Anda lihat di github.

[CoderJava/Scannerku](#)

[Scannerku – Example how to make Scanner Barcode use ZXing in Android](#)  
[github.com](#)

## Custom View

“Lalu bagaimana jika Anda ingin mengubah layout scanner-nya?”  
Bisa dong. Berikut saya berikan Anda contoh untuk membuat layout scanner-nya bisa bergerak keatas bawah pada garis indikator scanner-nya. Silakan buat satu class baru dengan



nama **CustomViewFinderView.kt** dan isi dengan source code berikut.

```
package com.ysn.scannerku

import android.content.Context
import android.graphics.Canvas
import android.graphics.Color
import android.graphics.Paint
import me.dm7.barcodescanner.core.ViewFinderView

class CustomViewFinderView constructor(context: Context) :
    ViewFinderView(context) {

    private val paint = Paint()
    private val SCANNER_ALPHA = intArrayOf(0, 64, 128, 192, 255,
192, 128, 64)
    private var scannerAlpha: Int = 0
    private var cntr = 0
    private var goingup = false
    private val POINT_SIZE = 10
    private val ANIMATION_DELAY = 80L

    init {
        paint.color = Color.WHITE
        paint.isAntiAlias = true
        setSquareViewFinder(true)
        setBorderColor(Color.parseColor("#35B781"))
        setLaserColor(Color.parseColor("#35B781"))
        setLaserEnabled(true)
    }

    override fun drawLaser(canvas: Canvas?) {
        /*super.drawLaser(canvas)*/
        paint.alpha = SCANNER_ALPHA[scannerAlpha]
        scannerAlpha = (scannerAlpha + 1) % SCANNER_ALPHA.size
        var middle = framingRect.height() / 2 + framingRect.top
        middle += cntr
        if (cntr < framingRect.top - mBorderLineLength - 10 &&
!goingup) {
            canvas?.drawRect((framingRect.left + 2).toFloat(),
(middle - 1).toFloat(), (framingRect.right - 1).toFloat(), (middle
+ 2).toFloat(), mLaserPaint)
            cntr += 4
        }

        if (cntr >= framingRect.top - mBorderLineLength - 10 &&
```

```

!goingup) goingup = true

        if (cntr > -framingRect.top + mBorderLineLength + 10 &&
goingup) {
            canvas?.drawRect((framingRect.left + 2).toFloat(),
(middle - 1).toFloat(), (framingRect.right - 1).toFloat(), (middle
+ 2).toFloat(), mLaserPaint)
            cntr -= 4
        }

        if (cntr <= -framingRect.top + mBorderLineLength + 10 &&
goingup) goingup = false

        postInvalidateDelayed(ANIMATION_DELAY,
            framingRect.left - POINT_SIZE,
            framingRect.top - POINT_SIZE,
            framingRect.right + POINT_SIZE,
            framingRect.bottom + POINT_SIZE)
    }
}

```

class tersebut Anda `extends` ke `ViewFinderView` milik ZXing sehingga Anda bisa mengubah layout-nya. Pada class yang barusan Anda buat ada meng-override method `drawLaser` dimana method ini berfungsi untuk menggambar laser pada layout-nya dan Anda ada mengubah pergerakan lasernya pada method tersebut. Selanjutnya, pada class **MainActivity.kt** silakan Anda ubah kode berikut

```
mScannerView = ZXingScannerView(this)
```

menjadi seperti berikut.

```

mScannerView = object : ZXingScannerView(this) {
    override fun createViewFinderView(context: Context?):
IViewFinder {
        return CustomViewFinderView(context!!)
    }
}

```

dimana, Anda meng-inisialisasikan object `mScannerView` dan menambahkan class `CustomViewFinderView` yang tadi Anda buat. Dan berikut adalah outputnya.





































































































































Output

# Uri

[Kotlin](#) | **Java**

```
public abstract class Uri  
extends Object implements Parcelable, Comparable<Uri>  
java.lang.Object  
↳ android.net.Uri
```

---

Immutable URI reference. A URI reference includes a URI and a fragment, the component of the URI following a '#'. Builds and parses URI references which conform to [RFC 2396](#).

In the interest of performance, this class performs little to no validation. Behavior is undefined for invalid input. This class is very forgiving--in the face of invalid input, it will return garbage rather than throw an exception unless otherwise specified.

## Summary

---

## Nested classes

class [Uri.Builder](#)

Helper class for building or manipulating URI references.

## Inherited constants

From interface [android.os.Parcelable](#)

## Fields

public static [CREATOR](#)

final [Creator](#)<[Uri](#)>

Reads Uris from Parcels.

public static [EMPTY](#)

final [Uri](#)

The empty URI, equivalent to "".

## Public methods

abstract [Uri.Builder](#) [buildUpon](#)()

Constructs a new builder, copying the attributes from this Uri.

int [compareTo](#)([Uri](#) other)

Compares the string representation of this Uri with that of another.

static [String](#) [decode](#)([String](#) s)

Decodes '%'-escaped octets in the given string using the UTF-8 scheme.

static [String](#) [encode](#)([String](#) s, [String](#) allow)

Encodes characters in the given string as '%'-escaped octets using the UTF-8 scheme.

static [String](#) [encode](#)([String](#) s)

Encodes characters in the given string as '%'-escaped octets using the UTF-8 scheme.

boolean	<a href="#"><code>equals</code></a> ( <a href="#"><code>Object</code></a> o)	Compares this Uri to another object for equality.
static <a href="#"><code>Uri</code></a>	<a href="#"><code>fromFile</code></a> ( <a href="#"><code>File</code></a> file)	Creates a Uri from a file.
static <a href="#"><code>Uri</code></a>	<a href="#"><code>fromParts</code></a> ( <a href="#"><code>String</code></a> scheme, <a href="#"><code>String</code></a> ssp, <a href="#"><code>String</code></a> fragment)	Creates an opaque Uri from the given components.
abstract <a href="#"><code>String</code></a>	<a href="#"><code>getAuthority</code></a> ()	Gets the decoded authority part of this URI.
boolean	<a href="#"><code>getBooleanQueryParameter</code></a> ( <a href="#"><code>String</code></a> key, boolean defaultValue)	Searches the query string for the first value with the given key and interprets it as a boolean value.
abstract <a href="#"><code>String</code></a>	<a href="#"><code>getEncodedAuthority</code></a> ()	Gets the encoded authority part of this URI.
abstract <a href="#"><code>String</code></a>	<a href="#"><code>getEncodedFragment</code></a> ()	Gets the encoded fragment part of this URI, everything after the '#'. If there is no fragment, returns an empty string.
abstract <a href="#"><code>String</code></a>	<a href="#"><code>getEncodedPath</code></a> ()	Gets the encoded path.
abstract <a href="#"><code>String</code></a>	<a href="#"><code>getEncodedQuery</code></a> ()	Gets the encoded query component from this URI.
abstract <a href="#"><code>String</code></a>	<a href="#"><code>getEncodedSchemeSpecificPart</code></a> ()	Gets the scheme-specific part of this URI, i.e. everything between the scheme separator ':' and the fragment separator '#'. If there is no scheme-specific part, returns an empty string.
abstract <a href="#"><code>String</code></a>	<a href="#"><code>getEncodedUserInfo</code></a> ()	Gets the encoded user information from the authority.
abstract <a href="#"><code>String</code></a>	<a href="#"><code>getFragment</code></a> ()	Gets the decoded fragment part of this URI.



	Gets the decoded fragment part of this URI, everything after the '#'.
abstract <a href="#">String</a>	<a href="#">getHost</a> ()
	Gets the encoded host from the authority for this URI.
abstract <a href="#">String</a>	<a href="#">getLastPathSegment</a> ()
	Gets the decoded last segment in the path.
abstract <a href="#">String</a>	<a href="#">getPath</a> ()
	Gets the decoded path.
abstract <a href="#">List</a> < <a href="#">String</a> >	<a href="#">getPathSegments</a> ()
	Gets the decoded path segments.
abstract int	<a href="#">getPort</a> ()
	Gets the port from the authority for this URI.
abstract <a href="#">String</a>	<a href="#">getQuery</a> ()
	Gets the decoded query component from this URI.
<a href="#">String</a>	<a href="#">getQueryParameter</a> ( <a href="#">String</a> key)
	Searches the query string for the first value with the given key.
<a href="#">Set</a> < <a href="#">String</a> >	<a href="#">getQueryParameterNames</a> ()
	Returns a set of the unique names of all query parameters.
<a href="#">List</a> < <a href="#">String</a> >	<a href="#">getQueryParameters</a> ( <a href="#">String</a> key)
	Searches the query string for parameter values with the given key.
abstract <a href="#">String</a>	<a href="#">getScheme</a> ()
	Gets the scheme of this URI.
abstract <a href="#">String</a>	<a href="#">getSchemeSpecificPart</a> ()
	Gets the scheme-specific part of this URI, i.e. everything between the scheme separator and the fragment separator '#'.

abstract <a href="#">String</a>	<a href="#">getUserInfo()</a>	Gets the decoded user information from the authority.
int	<a href="#">hashCode()</a>	Hashes the encoded string representation of this Uri consistently with <a href="#">equals(java.lang.Object)</a> .
boolean	<a href="#">isAbsolute()</a>	Returns true if this URI is absolute, i.e. if it contains an explicit scheme.
abstract boolean	<a href="#">isHierarchical()</a>	Returns true if this URI is hierarchical like "http://google.com".
boolean	<a href="#">isOpaque()</a>	Returns true if this URI is opaque like "mailto:nobody@google.com".
abstract boolean	<a href="#">isRelative()</a>	Returns true if this URI is relative, i.e. if it doesn't contain an explicit scheme.
<a href="#">Uri</a>	<a href="#">normalizeScheme()</a>	Return an equivalent URI with a lowercase scheme component.
static <a href="#">Uri</a>	<a href="#">parse(String uriString)</a>	Creates a Uri which parses the given encoded URI string.
abstract <a href="#">String</a>	<a href="#">toString()</a>	Returns the encoded string representation of this URI.
static <a href="#">Uri</a>	<a href="#">withAppendedPath(Uri baseUri, String pathSegment)</a>	Creates a new Uri by appending an already-encoded path segment to a base Uri.
static void	<a href="#">writeToParcel(Parcel out, Uri uri)</a>	Writes a Uri to a Parcel.

## Inherited methods

From class [java.lang.Object](#)

From interface [android.os.Parcelable](#)

From interface [java.lang.Comparable](#)

## Fields

---

### CREATOR

Added in [API level 1](#)

```
public static final Creator<Uri> CREATOR
```

Reads Uris from Parcels.

### EMPTY

Added in [API level 1](#)

```
public static final Uri EMPTY
```

The empty URI, equivalent to "".

## Public methods

---

### buildUpon

Added in [API level 1](#)

```
public abstract Uri.Builder buildUpon ()
```

Constructs a new builder, copying the attributes from this Uri.

Returns

[Uri.Builder](#)

## compareTo

Added in [API level 1](#)

```
public int compareTo (Uri other)
```

Compares the string representation of this Uri with that of another.

Parameters

other    Uri

Returns

int

## decode

Added in [API level 1](#)

```
public static String decode (String s)
```

Decodes '%'-escaped octets in the given string using the UTF-8 scheme. Replaces invalid octets with the unicode replacement character ("\\uFFFD").

## Parameters

`s` `String`: encoded string to decode

## Returns

`String` the given string with escaped octets decoded, or null if `s` is null

## encode

Added in [API level 1](#)

```
public static String encode (String s,  
                             String allow)
```

Encodes characters in the given string as '%'-escaped octets using the UTF-8 scheme. Leaves letters ("A-Z", "a-z"), numbers ("0-9"), and unreserved characters ("\_!~'()\*") intact. Encodes all other characters with the exception of those specified in the `allow` argument.

## Parameters

`s` `String`: string to encode

`allow` `String`: set of additional characters to allow in the encoded form, null if no characters should be skipped

## Returns

`String` an encoded version of `s` suitable for use as a URI component, or null if `s` is null

## encode

Added in [API level 1](#)

```
public static String encode (String s)
```

Encodes characters in the given string as '%'-escaped octets using the UTF-8 scheme. Leaves letters ("A-Z", "a-z"), numbers ("0-9"), and unreserved characters ("\_!~'()\*") intact. Encodes all other characters.

#### Parameters

s            **String:** string to encode

#### Returns

[String](#)    an encoded version of s suitable for use as a URI component, or null if s is null

## equals

Added in [API level 1](#)

```
public boolean equals (Object o)
```

Compares this Uri to another object for equality. Returns true if the encoded string representations of this Uri and the given Uri are equal. Case counts. Paths are not normalized. If one Uri specifies a default port explicitly and the other leaves it implicit, they will not be considered equal.

#### Parameters

o            **Object:** the reference object with which to compare.

#### Returns

boolean    true if this object is the same as the obj argument; false otherwise.

## fromFile

Added in [API level 1](#)

```
public static Uri fromFile (File file)
```

Creates a Uri from a file. The URI has the form "file:///". Encodes path characters with the exception of '/'.

Example: "file:///tmp/android.txt"

### Parameters

file    [File](#)

### Returns

[Uri](#)    a Uri for the given file

### Throws

[NullPointerException](#)    if file is null

## fromParts

Added in [API level 1](#)

```
public static Uri fromParts (String scheme,  
                             String ssp,  
                             String fragment)
```

Creates an opaque Uri from the given components. Encodes the ssp which means this method cannot be used to create hierarchical URIs.

### Parameters

scheme    [String](#): of the URI

`ssp`      `String`: scheme-specific-part, everything between the scheme separator (':') and the fragment separator ('#'), which will get encoded

`fragment`      `String`: fragment, everything after the '#', null if undefined, will get encoded

## Returns

[Uri](#)      Uri composed of the given scheme, ssp, and fragment

## Throws

[NullPointerException](#)      if scheme or ssp is null

## See also:

- [if you don't want the ssp and fragment to be encoded](#)

## getAuthority

Added in [API level 1](#)

```
public abstract String getAuthority ()
```

Gets the decoded authority part of this URI. For server addresses, the authority is structured as follows: `[ userinfo '@' ] host [ ':' port ]`

Examples: "google.com", "bob@google.com:80"

## Returns

[String](#)      the authority for this URI or null if not present

## getBooleanQueryParameter

Added in [API level 11](#)



```
public boolean getBooleanQueryParameter (String key,  
                                         boolean defaultValue)
```

Searches the query string for the first value with the given key and interprets it as a boolean value. "false" and "0" are interpreted as `false`, everything else is interpreted as `true`.

#### Parameters

key	String: which will be decoded
defaultValue	boolean: the default value to return if there is no query parameter for key

#### Returns

boolean	the boolean interpretation of the query parameter key
---------	---

## getEncodedAuthority

Added in [API level 1](#)

```
public abstract String getEncodedAuthority ()
```

Gets the encoded authority part of this URI. For server addresses, the authority is structured as follows: `[ userinfo '@' ] host [ ':' port ]`

Examples: "google.com", "bob@google.com:80"

#### Returns

<a href="#">String</a>	the authority for this URI or null if not present
------------------------	---

## getEncodedFragment

Added in [API level 1](#)

```
public abstract String getEncodedFragment ()
```

Gets the encoded fragment part of this URI, everything after the '#'.

#### Returns

[String](#) the encoded fragment or null if there isn't one

## getEncodedPath

Added in [API level 1](#)

```
public abstract String getEncodedPath ()
```

Gets the encoded path.

#### Returns

[String](#) the encoded path, or null if this is not a hierarchical URI (like "mailto:nobody@google.com") or the URI is

## getEncodedQuery

Added in [API level 1](#)

```
public abstract String getEncodedQuery ()
```

Gets the encoded query component from this URI. The query comes after the query separator ('?') and before the fragment separator ('#'). This method would return "q=android" for "http://www.google.com/search?q=android".

#### Returns

[String](#) the encoded query or null if there isn't one

## getEncodedSchemeSpecificPart

Added in [API level 1](#)

```
public abstract String getEncodedSchemeSpecificPart ()
```

Gets the scheme-specific part of this URI, i.e. everything between the scheme separator ':' and the fragment separator '#'. If this is a relative URI, this method returns the entire URI. Leaves escaped octets intact.

Example: "//www.google.com/search?q=android"

### Returns

[String](#) the encoded scheme-specific-part

## getEncodedUserInfo

Added in [API level 1](#)

```
public abstract String getEncodedUserInfo ()
```

Gets the encoded user information from the authority. For example, if the authority is "nobody@google.com", this method will return "nobody".

### Returns

[String](#) the user info for this URI or null if not present

## getFragment

Added in [API level 1](#)

```
public abstract String getFragment ()
```

Gets the decoded fragment part of this URI, everything after the '#'.

## Returns

[String](#) the decoded fragment or null if there isn't one

## getHost

Added in [API level 1](#)

```
public abstract String getHost ()
```

Gets the encoded host from the authority for this URI. For example, if the authority is "bob@google.com", this method will return "google.com".

## Returns

[String](#) the host for this URI or null if not present

## getLastPathSegment

Added in [API level 1](#)

```
public abstract String getLastPathSegment ()
```

Gets the decoded last segment in the path.

## Returns

[String](#) the decoded last segment or null if the path is empty

## getPath

Added in [API level 1](#)

```
public abstract String getPath ()
```

Gets the decoded path.

#### Returns

[String](#) the decoded path, or null if this is not a hierarchical URI (like "mailto:nobody@google.com") or the URI is

## getPathSegments

Added in [API level 1](#)

```
public abstract List<String> getPathSegments ()
```

Gets the decoded path segments.

#### Returns

[List](#)<[String](#)> decoded path segments, each without a leading or trailing '/'

## getPort

Added in [API level 1](#)

```
public abstract int getPort ()
```

Gets the port from the authority for this URI. For example, if the authority is "google.com:80", this method will return 80.

#### Returns

int the port for this URI or -1 if invalid or not present

## getQuery

Added in [API level 1](#)

```
public abstract String getQuery ()
```

Gets the decoded query component from this URI. The query comes after the query separator ('?') and before the fragment separator ('#'). This method would return "q=android" for "http://www.google.com/search?q=android".

### Returns

[String](#) the decoded query or null if there isn't one

## getQueryParameter

Added in [API level 1](#)

```
public String getQueryParameter (String key)
```

Searches the query string for the first value with the given key.

**Warning:** Prior to Jelly Bean, this decoded the '+' character as '+' rather than ' '.

### Parameters

key      [String](#): which will be encoded

### Returns

[String](#) the decoded value or null if no parameter is found

### Throws

[UnsupportedOperationException](#) if this isn't a hierarchical URI

[NullPointerException](#) if key is null

## getQueryParameterNames

Added in [API level 11](#)

```
public Set<String> getQueryParameterNames ()
```

Returns a set of the unique names of all query parameters. Iterating over the set will return the names in order of their first occurrence.

### Returns

[Set](#)<[String](#)> a set of decoded names

### Throws

[UnsupportedOperationException](#) if this isn't a hierarchical URI

## getQueryParameters

Added in [API level 1](#)

```
public List<String> getQueryParameters (String key)
```

Searches the query string for parameter values with the given key.

### Parameters

key                [String](#): which will be encoded

### Returns

[List](#)<[String](#)> a list of decoded values

### Throws

[UnsupportedOperationException](#) if this isn't a hierarchical URI

[NullPointerException](#) if key is null

## getScheme

Added in [API level 1](#)

```
public abstract String getScheme ()
```

Gets the scheme of this URI. Example: "http"

### Returns

[String](#) the scheme or null if this is a relative URI

## getSchemeSpecificPart

Added in [API level 1](#)

```
public abstract String getSchemeSpecificPart ()
```

Gets the scheme-specific part of this URI, i.e. everything between the scheme separator ':' and the fragment separator '#'. If this is a relative URI, this method returns the entire URI. Decodes escaped octets.

Example: "///www.google.com/search?q=android"

### Returns

[String](#) the decoded scheme-specific-part

## getUserInfo



Added in [API level 1](#)

```
public abstract String getUserInfo ()
```

Gets the decoded user information from the authority. For example, if the authority is "nobody@google.com", this method will return "nobody".

### Returns

[String](#) the user info for this URI or null if not present

## hashCode

Added in [API level 1](#)

```
public int hashCode ()
```

Hashes the encoded string representation of this Uri consistently with [equals\(java.lang.Object\)](#).

### Returns

`int` a hash code value for this object.

## isAbsolute

Added in [API level 1](#)

```
public boolean isAbsolute ()
```

Returns true if this URI is absolute, i.e. if it contains an explicit scheme.

### Returns

`boolean` true if this URI is absolute, false if it's relative

## isHierarchical

Added in [API level 1](#)

```
public abstract boolean isHierarchical ()
```

Returns true if this URI is hierarchical like "http://google.com". Absolute URIs are hierarchical if the scheme-specific part starts with a '/'. Relative URIs are always hierarchical.

### Returns

boolean

## isOpaque

Added in [API level 1](#)

```
public boolean isOpaque ()
```

Returns true if this URI is opaque like "mailto:nobody@google.com". The scheme-specific part of an opaque URI cannot start with a '/'.

### Returns

boolean

## isRelative

Added in [API level 1](#)

```
public abstract boolean isRelative ()
```

Returns true if this URI is relative, i.e. if it doesn't contain an explicit scheme.

## Returns

`boolean` true if this URI is relative, false if it's absolute

## normalizeScheme

Added in API level 16

```
public Uri normalizeScheme ()
```

Return an equivalent URI with a lowercase scheme component. This aligns the Uri with Android best practices for intent filtering.

For example, "HTTP://www.android.com" becomes "http://www.android.com"

All URIs received from outside Android (such as user input, or external sources like Bluetooth, NFC, or the Internet) should be normalized before they are used to create an Intent.

This method does *not* validate bad URI's, or 'fix' poorly formatted URI's - so do not use it for input validation. A Uri will always be returned, even if the Uri is badly formatted to begin with and a scheme component cannot be found.

## Returns

[Uri](#) normalized Uri (never null)

### See also:

- [Intent.setData\(Uri\)](#)
- [Intent.setDataAndNormalize\(Uri\)](#)

## parse

Added in API level 1

```
public static Uri parse (String uriString)
```

Creates a Uri which parses the given encoded URI string.

## Parameters

```
uriString  String: an RFC 2396-compliant, encoded URI
```

## Returns

Uri Uri for this given uri string

## Throws

NullPointerException if uriString is null

## toString

Added in API level 1

```
public abstract String toString ()
```

Returns the encoded string representation of this URI. Example: "http://google.com/"

## Returns

String a string representation of the object.

## withAppendedPath

Added in API level 1

```
public static Uri withAppendedPath (Uri baseUri,  
                                   String pathSegment)
```

Creates a new Uri by appending an already-encoded path segment to a base Uri.

### Parameters

`baseUri`      `Uri`: Uri to append path segment to  
`pathSegment`   `String`: encoded path segment to append

### Returns

[Uri](#)              a new Uri based on `baseUri` with the given segment appended to the path

### Throws

[NullPointerException](#)   if `baseUri` is null

## writeToParcel

Added in [API level 1](#)

```
public static void writeToParcel (Parcel out,  
                                Uri uri)
```

Writes a Uri to a Parcel.

### Parameters

`out`   `Parcel`: parcel to write to  
`uri`   `Uri`: to write, can be null

## REFERENSI

<https://developer.android.com/studio/intro?hl=id>

<https://www.indoworx.com/versi-android/>

<https://idcloudhost.com/mengenal-apa-itu-android-studio-fungsi-manfaat-dan-cara-installasinya/>

<https://www.codepolitan.com/mengapa-harus-belajar-android-dengan-android-studio-59bfc3146686f>

<https://galleryteknologi.wordpress.com/2017/05/19/pengertian-android-menurut-para-ahli/>