



Challenge

Event

Job

Developer

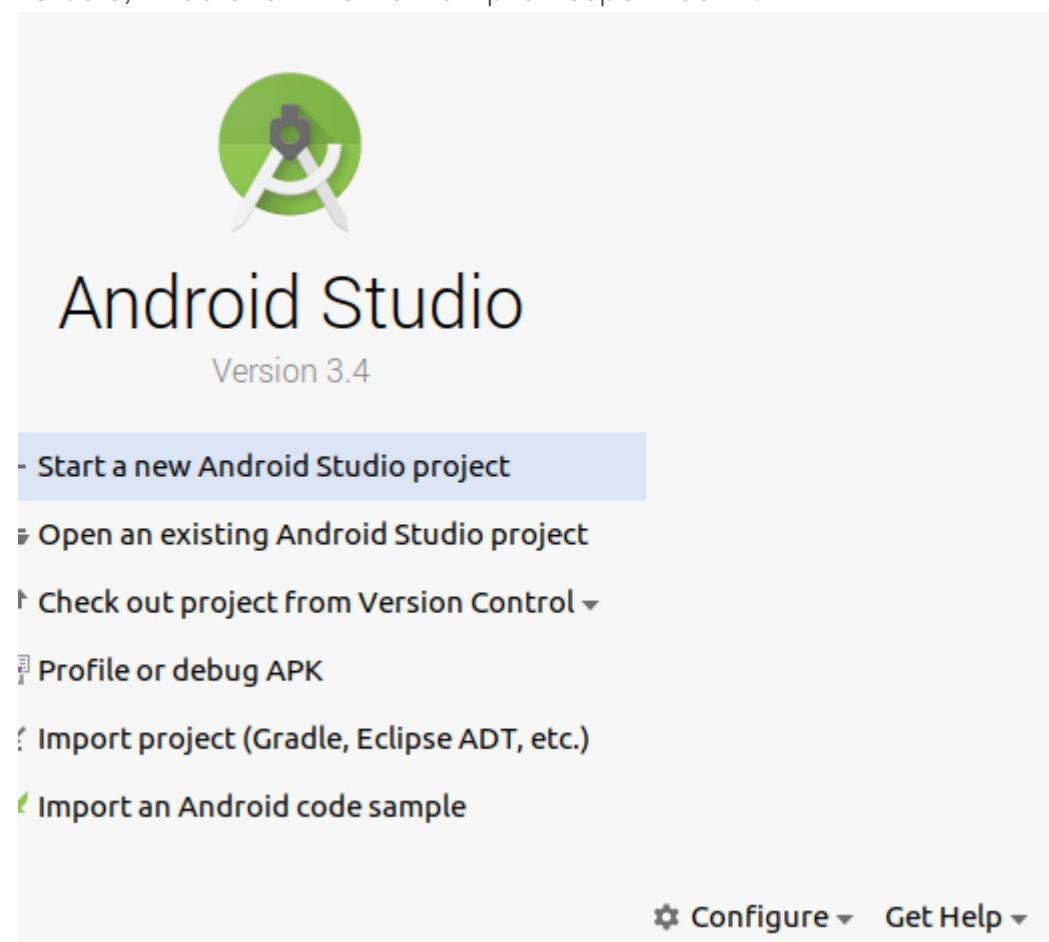


## Android Studio

Android Studio, baiknya Anda tahu bagaimana strukturnya terlebih dahulu. Bila terbiasa dengan Android Studio lainnya, maka akan mudah bagi Anda untuk menavigasi tata letak dan struktur Android Studio. Komponen tambahan pendukung yang membantu pengembangan dan pembuatan aplikasi Android, *right from the git-go!!*

Anda

Android Studio, Anda akan melihat tampilan seperti berikut ini.



Anda akan melihat tampilan seperti berikut ini.

Android Studio, mari kita mulai membuat sebuah aplikasi **Hallo World**, aplikasi pertama Anda.

Android Studio, mari kita mulai membuat sebuah aplikasi **Hallo World**, aplikasi pertama Anda. Ketika Anda memilih **Start a new Android Studio project**, Anda diminta untuk melakukan konfigurasi dalam pembuatan aplikasi. Dalam dialog ini Anda bisa memilih **template** dari Aplikasi yang akan Anda buat. Terdapat beberapa template yang bisa Anda pilih, seperti **Empty Activity**, **Login Activity**, **Navigation Drawer Activity** dan lain-lain. Di dalam dialog ini juga Anda bisa memilih **get devices** dari aplikasi yang akan kita buat.

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



9

- Challenge
- Event
- Job
- Developer

[sini](#), dan untuk melihat perubahan antara sebelum dan sesudah **AndroidX**, atau cara migrasi bisa lihat di [sini](#). Perlu Anda ketahui juga, ketika Anda sudah menggunakan **AndroidX** maka hapus library lama atau sebelum **AndroidX**.

udio

di Project Wizard, *done!* Kali ini kita akan menemui tampilan penuh Android Studio. Untuk kita bahas lebih jauh tentang antarmuka (*interface*) dari Android Studio ini.

1 penuh IDE Android Studio berbasis IntelliJ IDEA. Mungkin tampilan tersebut akan berbeda karena perbedaan konfigurasi dan versi Android Studio.

akan dalam *development* seperti *copy/paste*, *build*, menjalankan aplikasi, hingga

dalam (*depth*) dan posisi proyek yang saat ini sedang dibuka.

ditor

Android Studio di mana kita menuliskan kode. Pada tampilan di atas, sebelah kiri adalah sidebar dan editor. Bagian ini akan dibahas lebih detail di poin selanjutnya.

or ini merupakan button yang dapat di-*expand* ataupun untuk menampilkan Tools secara

Android Studio, ia berfungsi untuk menampilkan status proyek kita dan pesan peringatan (*warning*

risi satu atau beberapa modul dengan file kode sumber dan file sumber daya. Jenis-jenis

n menampilkan berkas proyek Anda dalam tampilan proyek Android, seperti yang ditampilkan

Tampilan disusun berdasarkan modul untuk memberikan akses cepat ke berkas sumber utama proyek Anda.



- Challenge
- Event
- Job
- Developer

- Antisipasi tampilan *project structure* sesuai kebutuhan.
- Siapkan proyek yang baru saja kita buat.
- Siapkan folder berikut:

yang harus ada di dalam sebuah proyek Android. Manifest akan memberikan beragam informasi tentang aplikasi Android. Sistem perlu mengetahui apa yang akan digunakan oleh aplikasi sebelum dijalankan. Struktur manifest adalah sebagai berikut.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.dicoding.codiaploma.myapplication">
```

Sebagai identitas unik dari sebuah aplikasi. Identitas ini juga yang akan digunakan di dalam aplikasi dengan aplikasi lainnya. Jangan pernah mengganti value di dalam package name ini sebagai aplikasi yang lain, jika sudah masuk ke dalam store.

Definisikan komponen dari aplikasi mulai dari activity, services, broadcast receiver, dan content

```
    <activity
        android:name=".MainActivity"
        android:label="@string/app_name"
        android:icon="@mipmap/ic_launcher_round"
        android:theme="@style/Theme.AppCompat.NoActionBar">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

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

    <receiver
        android:name=".MyReceiver"
        android:enabled="true"></receiver>
```

Activity berada di antara tag <application>. Ia juga berfungsi sebagai penamaan kelas yang akan dijalankan dan mendeskripsikan kemampuannya seperti intent-filter, di mana fungsinya untuk menentukan komponen itu adalah yang pertama kali dijalankan.

Sebelum menambahkan apa saja yang harus dimiliki oleh aplikasi untuk akses ke dalam komponen API seperti internet, kita juga perlu menambahkan izin yang diperlukan. Sebagai contoh ini adalah kode untuk menambahkan izin internet.

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

Penempatan kode ini di atas tag <application> dan masih di dalam tag <manifest>



Challenge



bisa juga diganti sesuai dengan bahasa yang dipakai, misal "kotlin."

Event



Job



Developer



, atau sebuah kelas dari proyek Android.

ang mana bukan berupa kode, melainkan *layout* aplikasi, sumber gambar, ikon, hingga *style*.  
at sejumlah folder yang sudah diatur dan dikategorikan sesuai kebutuhan, seperti :

ar maupun ikon.

dipakai untuk file desain aplikasi.

n berbagai dimensi.

er data, seperti **colors.xml** untuk warna, **strings.xml** untuk teks, **dimens.xml** untuk ukuran,  
iat style atau template.

*ild automation system. **Automation system** berguna untuk mengotomatisasi proses  
n proses-proses terkait lainnya termasuk *compile source code* menjadi *binary code*, *packaging*  
omated test.*

on)

teratas yang meliputi keseluruhan dari proyek dari sebuah aplikasi. Di dalamnya berisi  
a di dalam proyek.

da pada setiap modul di dalam proyek sebuah aplikasi. Beberapa konfigurasi yang diedit di  
s, **defaultConfig** dan **productFlavors**, **buildTypes**, dan **dependencies**.



- Challenge
- Event
- Job
- Developer

```
ng.picodiploma.myapplication"

tal)

"androidx.test.runner.AndroidJUnitRunner"
```

ultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

```
'libs', include: ['*.jar'])
ompat:appcompat:1.0.2'
traintlayout:constraintlayout:1.1.3'
it:4.12'
droidx.test:runner:1.2.0'
```

bisa menetapkan compileSDKVersion dan buildToolsVersion.

ties seperti **applicationID** , **minSdkVersion** , **targetSdkVersion** dan test information.

ties dari debuggable, *ProGuard enabling*, *debug signing*, *version name suffix* dan *test*

asi tentang *library* yang digunakan oleh aplikasi.

u di bagian **versionCode** , misal kita sudah *publish* aplikasi ke PlayStore dengan *version code* 1, *te* aplikasinya lagi ke Playstore, kita perlu merubah *version code*-nya menjadi 2 (*incremental*), *ore* akan menolak APK yang kita *upload*.

nasi di dalam build.gradle kita harus melakukan sinkronisasi terlebih dahulu. Tombol **sync now** atas ketika terjadi perubahan.

maka akan muncul informasi pada log apakah proses sinkronisasi berhasil atau tidak.

< manajemen segala *resource* yang ada di proyek Anda seperti gambar, warna, layout dll. **res**, dengan menggunakan Resource Manager semua *resource* akan tampil di sini dan Anda tuhan Anda.

### roid Studio

as yang *powerful* di bawah IntelliJ IDEA ini. Banyak *tools* milik Android yang membantu kita iri kita bahas *tools* yang sering digunakan dan manfaatnya.

- **Shift+Shift**  
*Search Everywhere*, atau dapat dikatakan pencarian semua jenis berkas yang masih dalam 1 proyek.



- Challenge
- Event
- Job
- Developer

- i seluruh berkas proyek.
- au perintah-perintah yang ada di Android Studio.
- alam berkas.

ertentu.

deklarasi yang dipilih.

ya.

ya.

tentu.

ecent file).

**:+F7**  
riabel/objek yang diklik.

/objek yang dipilih di seluruh berkas proyek.

ari variabel/objek yang dipilih.

dipilih.

tampilan minimal.

nting ketika melihat method dari Android atau library lain.

mpilkan saran untuk melengkapi kode Anda.

ampilkan saran kode untuk melengkapi kode Anda dengan lebih pintar (menampilkan apa  
ngan kode Anda).

1 (*generate*) kode. Perintah ini sangat memudahkan ketika membuat constructor dan

xpikan kode.

- **Ctrl+Y**  
*Delete One Line*, Menghapus satu baris kode.



- Challenge
- Event
- Job
- Developer

□ s yang diblok menjadi sebuah fungsi.



□ ima suatu file atau variabel maupun fungsi.



arkas, package atau modul.

*emulator* atau *devices*.

ke *emulator* atau *devices* dalam mode **Debug**, biasanya untuk keperluan *testing*.

*ipo*) dalam pemanggilan class, method hingga variabel sebaiknya kita memanfaatkan Code apat dua (2) jenis Code Completion yang sering digunakan di Android Studio:

tion

on standar untuk membantu kita melengkapi kode.

sebuah variabel, cukup ketikkan code completion di atas. Saran pun akan diberikan.  
ion

U karena kita bisa menyelesaikan kode tanpa harus mengetik lengkap dan tanpa tanda urawal, dan banyak macam pemformatan lainnya.  
lum menggunakan shortcut:

i Statement Completion. Lihat apa yang terjadi!



## Challenge

## Event

## Job

## Developer



empelajarinya di sini: <https://www.jetbrains.com/help/idea/2016.1/code-completion-2.html>



dalam pemrograman. Kita memiliki *signature style* masing-masing, Semua tergantung pilihan memperhatikan bagaimana tata letak kode, apalagi bila suatu saat nanti kita membuat yang rapi itu enak dilihat dan memudahkan, baik kita maupun orang lain untuk membacanya. berikan code style formatting untuk tata letak kode yang kita miliki. Untuk menyesuaikan **rgs > Editor > Code Style (Android Studio > Preferences > Editor > Code Style** pada Mac.)

kode di atas? Ya tidak ada yang salah. Namun Code Style berantakan dan tidak indah untuk

*formatting* dengan menggunakan shortcut **Ctrl+Alt+L**.

sedikit, tidak terlalu berpengaruh. Tapi bila baris kode sudah mulai kompleks, *formatting code*

menemukan kode yang berkualitas dan *best practice*-nya. Melalui Google, Android Studio kita gunakan dan manfaatkan untuk kebutuhan kita belajar atau membuat aplikasi Android. **Import Sample**, kita punya banyak pilihan contoh kode yang bisa dipakai.

ini: <https://developer.android.com/studio/write/sample-code.html>

menggunakan alat bantu dari Android Studio ini. Tak lain agar membantu dan mempercepat





Mamun amri



Challenge

Event

Job

Developer



ERUSAHAAN

[tentang Kami](#)

[log](#)

[erita Terbaru](#)



PROGRAM

[Academy](#)

[Challenge](#)

[Event](#)

[Job](#)

[Rewards](#)

SUPPORT

[Bantuan](#)

[FAQ](#)

[Hubungi Kami](#)

gths reserved.

[Terms](#)

[Privacy](#)