

Challenge



» Modul ini dibuat pada tanggal 13 June 2017. Pembaharuan terakhir adalah: Migrasi ke Kotlin.  
» [« Lihat riwayat](#)

Event



Job



Developer



## Android Studio

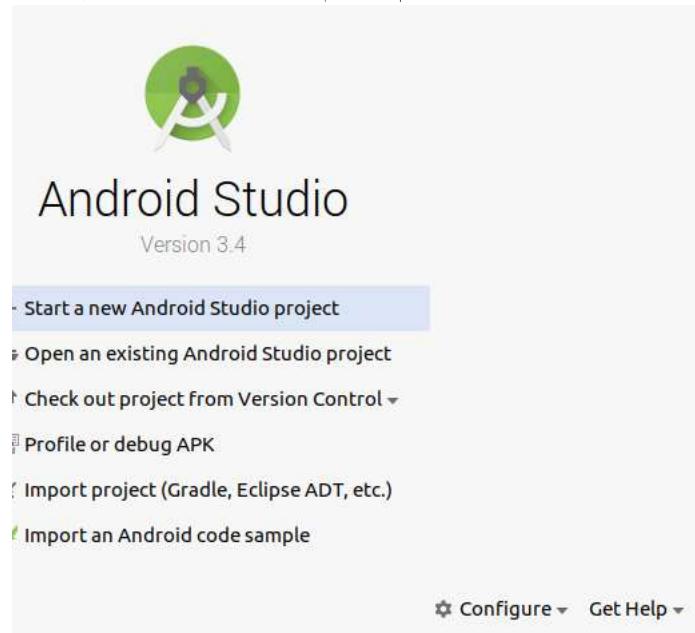
### : Project



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

A

ndroid Studio, Anda akan melihat tampilan seperti berikut ini.



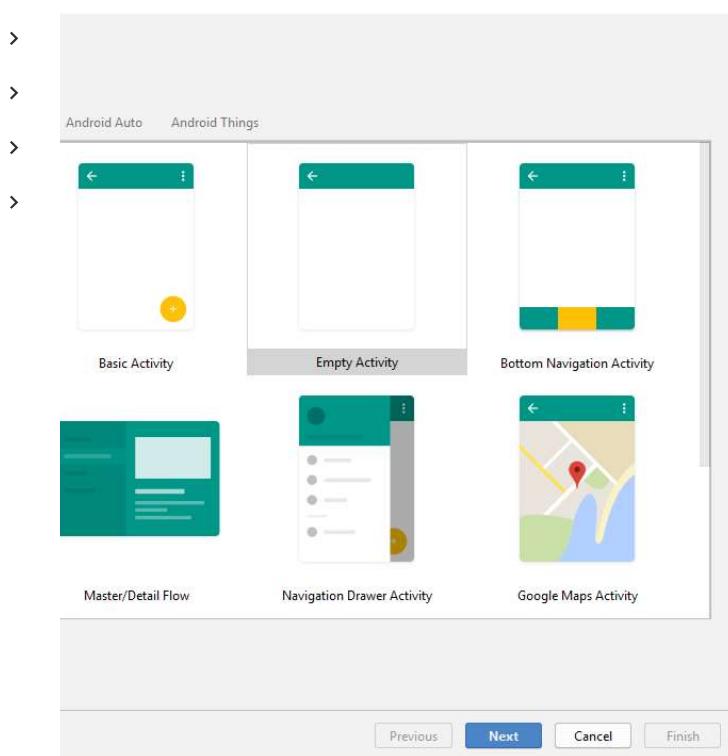
“Start a new Android Studio project”.

udio, mari kita mulai membuat sebuah aplikasi *Hello World*, aplikasi pertama Anda.

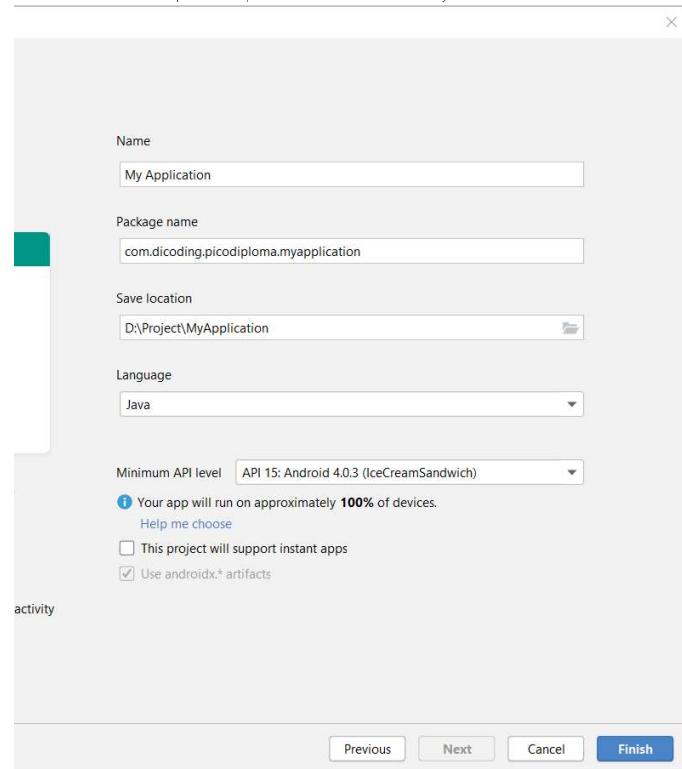
1. Setelah melakukan *Start a new Android Studio project*, Anda diminta untuk melakukan konfigurasi dalam pembuatan proyek baru Anda. Dalam dialog ini Anda bisa memilih *template* dari Aplikasi yang akan Anda buat. Terdapat beberapa



- Challenge >
- Event >
- Job >
- Developer >



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





- Challenge
- Event
- Job
- Developer

melihat perubahan antara sebelum dan sesudah AndroidX atau cara migrasi menggunakan  
perlu Anda ketahui juga, ketika Anda sudah menggunakan AndroidX maka Anda tidak bisa  
menggunakan AndroidX lagi atau sebelum AndroidX.

>

## audio

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.

The screenshot shows the Android Studio interface with the following details:

- Top Bar:** Build, Run, Tools, VCS, Window, Help.
- Toolbars:** Standard toolbar with icons for file operations like Open, Save, Find, etc.
- Left Sidebar:** Shows the project structure with packages: com, dicoding, picodiploma, myapplication. The MainActivity.java file is selected.
- Main Editor:** Displays the Java code for MainActivity.java:
 

```
package com.dicoding.picodiploma.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```
- Bottom Status Bar:** Shows the current file (MainActivity.java), encoding (UTF-8), and context (no context).
- Right Side Panels:** Device File Explorer and Event Log.

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



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

ini > java > com > dicoding > picodiploma > myapplication > MainActivity >

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

ditor



- Challenge
- Event
- Job
- Developer

```
activity_main.xml x MainActivity.java x
1 package com.dicoding.picodiploma.myapplication;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity {
6     @Override
7     protected void onCreate(Bundle savedInstanceState) {
8         super.onCreate(savedInstanceState);
9         setContentView(R.layout.activity_main);
10    }
11
12 }
13
14
```

The screenshot shows the Android Studio interface with the code editor open. The code is written in Java and defines a MainActivity that extends AppCompatActivity. It overrides the onCreate method to set the content view to activity\_main. The code editor has syntax highlighting and a vertical scrollbar. To the left of the code editor is a navigation bar with project items like Challenge, Event, Job, and Developer. On the right side of the screen, there are several toolbars and panels: Gradle, Device File Explorer, and an Event Log at the bottom.

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

1: Project   2: Structure   3: Favorites   4: Build Variants   5: Layout Captures   6: Gradle   7: Device File Explorer

Orang ini merupakan *button* yang dapat di-expand ataupun untuk menampilkan tools secara detail

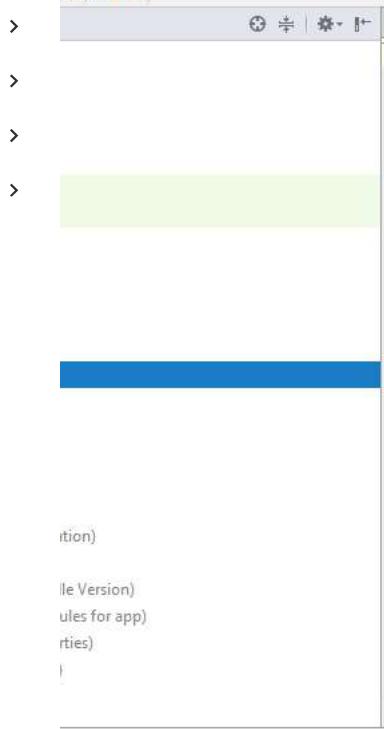
checked state) (28 minutes ago)

Selain Android Studio, ia berfungsi untuk menampilkan status proyek kita dan pesan peringatan

terdiri satu atau beberapa modul dengan berkas kode sumber dan berkas sumber daya. Jenis-



Challenge



Event

Job

Developer

dul untuk memberikan akses cepat ke berkas sumber utama proyek Anda.

Pada proyek baru, Android Studio akan menampilkan struktur yang lebih ringkas dan cepat sesuai standar Android. Bila ingin melihat struktur proyek dalam bentuk selain standar Android, kita dapat menggunakan dropdown yang terdapat di atas *project structure*.

Untuk tampilan *project structure* sesuai kebutuhan.

Pada proyek yang baru saja kita buat.



- > Challenge
- > Event
- > Job
- > Developer

```
application)
i)
: (Gradle Version)
ard Rules for app)
Properties)
ttings)
tion)
```

si folder berikut

yang harus ada di dalam sebuah proyek Android. Manifest akan memberikan beragam android. Sistem perlu mengetahui apa yang akan digunakan oleh aplikasi sebelum dijalankan.

rn Android Manifest adalah sebagai berikut:

```
'http://schemas.android.com/apk/res/android"
;.picodiploma.myapplication">
```

oertindak sebagai identitas unik dari sebuah aplikasi. Identitas ini juga yang akan digunakan di kan satu aplikasi dengan aplikasi lainnya. Jangan pernah mengganti *value* di dalam package li sebagai aplikasi yang lain jika sudah masuk ke dalam *store*.

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





Challenge

```
>     <g><app_name>  
>         <ipmap/ic_launcher_round"  
>         'true"  
>     </AppTheme>  
>     <activity<name=".MainActivity">  
>         <intent-filter>  
>             <action>android.intent.action.MAIN</action>  
>             <category>android.intent.category.LAUNCHER</category>  
>         </intent-filter>  
>     </activity>
```

Event

```
'IntentService"  
:"false" />  
  
'Receiver"  
'true"  
:"true"></receiver>
```

Job

Developer

berada di antara tag `<application>`, ia juga berfungsi sebagai penamaan kelas yang dan mendeskripsikan kemampuannya seperti intent-filter, di mana fungsinya ponen itu adalah yang pertama kali dijalankan.

apa saja yang harus dimiliki oleh aplikasi untuk akses ke dalam komponen API seperti internet, juga untuk berinteraksi dengan aplikasi lainnya. Sebagai contoh ini adalah kode untuk

```
name="android.permission.INTERNET"/>
```

di atas tag `<application>` dan masih di dalam tag `<manifest>`

n  
vity

sering dipakai, berisi berkas *source code* kita yang ditulis dalam bahasa Java/Kotlin, `InstrumentationTest` (Instrumentation Test).

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

atau sebuah kelas dari proyek Android.





- Challenge >
- Event >
- Job >
- Developer >

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

mbar maupun ikon.

dipakai untuk berkas desain aplikasi.

n berbagai dimensi.

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

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

```
application)
)
(Gradle Version)
ard Rules for app)
Properties)
ttings)
tion)
```

on)

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

la pada setiap modul di dalam proyek sebuah aplikasi. Beberapa konfigurasi yang didefinisikan di  
, `defaultConfig` dan `productFlavors`, `buildTypes`, dan `dependencies`.





- Challenge
- Event
- Job
- Developer

```

        androidx.test:runner:AndroidJUnitRunner"
    }

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

    ...
    ^: 'libs', include: ['*.jar'])
    opcompat:appcompat:1.0.2'
    constraintlayout:constraintlayout:1.1.3'
    junit:4.12'
    'androidx.test:runner:1.2.0'
    'androidx.test.espresso:espresso-core:3.2.0'

```

isa menetapkan compileSDKVersion dan buildToolsVersion.

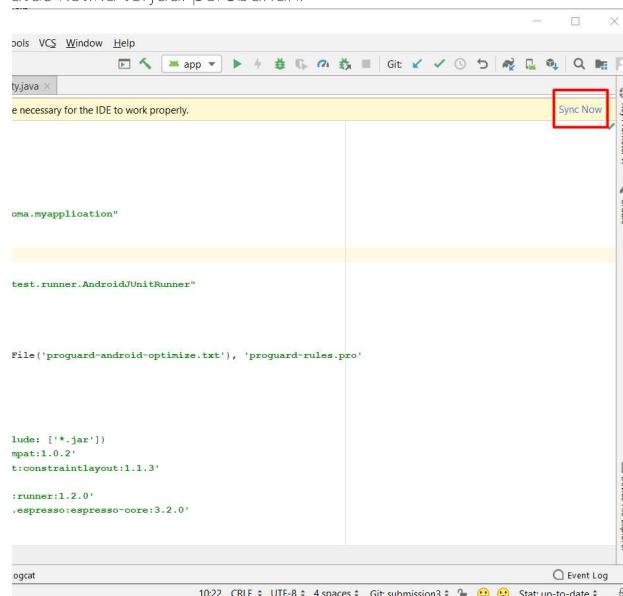
ties seperti applicationID, minSdkVersion, targetSdkVersion dan test information.

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

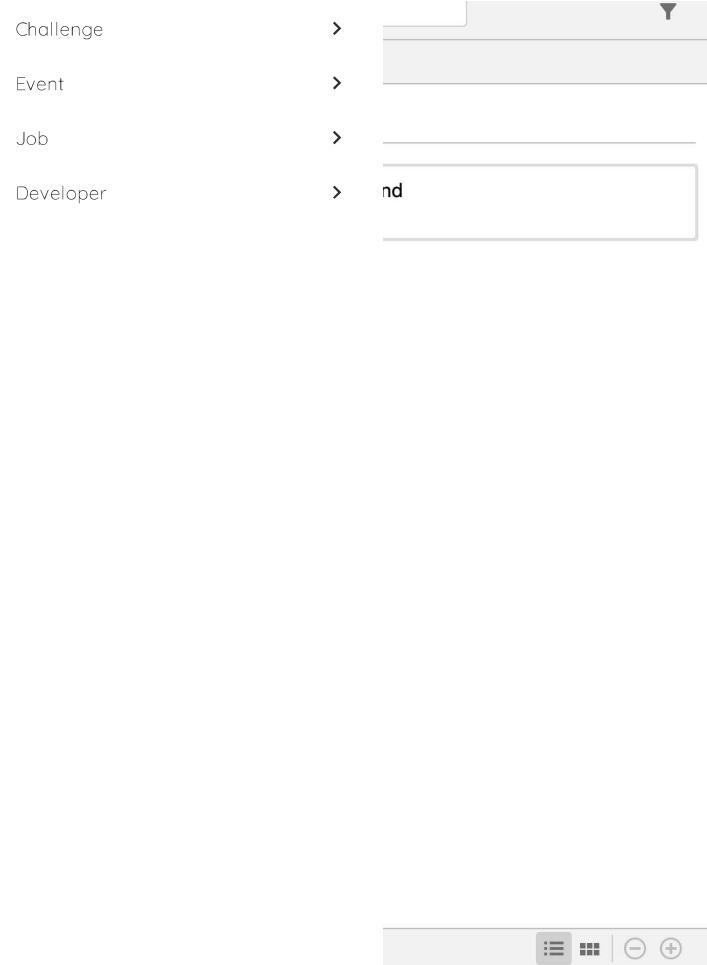
asi tentang library yang digunakan oleh aplikasi.

J di bagian `versionCode`, misal kita sudah publish aplikasi ke PlayStore dengan *version code* 1, 'e aplikasinya lagi ke Playstore, kita perlu merubah *version code*-nya menjadi 2 (*incremental*), :tore akan menolak APK yang di-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.



< memanajemen segala *resource* yang ada di projek 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.

at dikatakan pencarian semua jenis berkas yang masih dalam 1 projek.

lah satu berkas.

i seluruh berkas proyek.

au perintah-perintah yang ada di Android Studio.

replace, mengganti teks di dalam berkas.

Navigasi:



Challenge >

Event > deklarasi yang dipilih.

Job >

Developer > ya.

ya.

tentu.

*recent file).*

+F7  
variabel/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.

mpulkan saran untuk melengkapi kode Anda.

ampulkan saran kode untuk melengkapi kode Anda dengan lebih pintar (menampilkan apa yang kode Anda).

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

xpikan kode.

satu baris kode.

s yang diblok menjadi sebuah variabel.

s yang diblok menjadi sebuah fungsi.

ima suatu file atau variabel maupun fungsi.

- Ctrl+F9

*Make project, build project.*





Challenge

&gt; emulator atau devices.

Event

&gt;

&gt; ke emulator atau devices dalam mode Debug, biasanya untuk keperluan testing.

Job

&gt;

Developer

&gt;

(*ipos*) dalam pemanggilan *class*, *method* hingga variabel sebaiknya kita memanfaatkan Code

ada dua jenis *code completion* yang sering digunakan di Android Studio:

tion

n standar untuk membantu kita melengkapi kode.

tActivity {

stanceState) {  
;  
ain);

```
String
String[]
(AccessibilityEvent .. boolean
boolean
File
File
text, String name) .. boolean
void
int mode, Cursor. SQLiteDatabase
int mode, Cursor. SQLiteDatabase
in completion lookups as well? »
```

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

ion

karena kita bisa menyelesaikan kode tanpa harus mengetik lengkap dan tanpa tanda urawal, dan banyak macam pemformatan lainnya.

um menggunakan shortcut:

```
e savedInstanceState) {
naceState);
activity_main);
```

Statement Completion. Lihat apa yang terjadi!

```
le savedInstanceState) {
anceState);
activity_main);
```

intas akan diselesaikan oleh Android Studio. Tentu hal ini akan mempercepat waktu kita dalam

empelajarinya di [sini](#).

alam pemrograman. Kita memiliki *signature style* masing-masing. Semua tergantung pilihan nemperhatikan bagaimana tata letak kode, apalagi bila suatu saat nanti kita membuat

aplikasi bersama orang lain. Kode yang rapi itu enak dilihat dan memudahkan, baik kita maupun orang lain untuk membacanya. Secara default Android Studio memberikan *code style formatting* untuk tata letak kode yang kita miliki. Untuk menyesuaikan setelan *code style*, klik *File > Settings > Editor > Code Style* (*Android Studio > Preferences > Editor > Code Style* pada Mac.)



Challenge

> `vity_main);`

Event

&gt;

Job

&gt;

Developer

&gt;

› kode di atas? Ya tidak ada yang salah. Namun *code style* berantakan dan tidak indah untuk *Formatting* dengan menggunakan shortcut *Ctrl+Alt+L*.

```
Create(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    TextView tv = (TextView) findViewById(R.id.tv);
    tv.setText("Hello dunia");
```

```
onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    TextView tv = (TextView) findViewById(R.id.tv);
    tv.setText("Hello dunia");
```

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. [sini](#).

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

[NYA](#)[LANJUTKAN KE MATERI BERIKUTNYA →](#)

## ERUSAHAAN

[Bantang Kami](#)[log](#)[erita Terbaru](#)

## PROGRAM

[Academy](#)[Challenge](#)[Event](#)[Job](#)[Rewards](#)

## SUPPORT

[Bantuan](#)[FAQ](#)[Hubungi Kami](#)



- Challenge >
- Event >
- Job >
- Developer >

