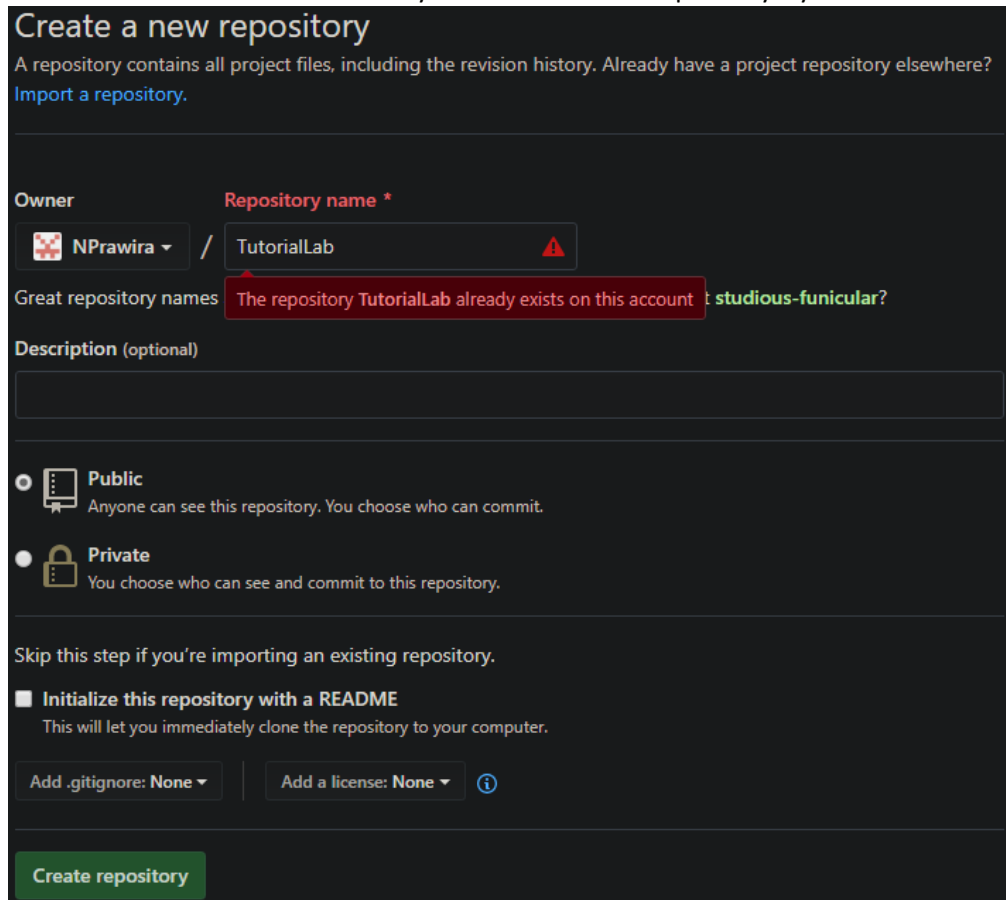


Sebelum memulai tutorial ini, pastikan kalian sudah meng-install Android Studio, GitBash, dan memiliki akun GitHub.

Tutorial Git

1. Login ke akun githubmu. Buat akun github terlebih dahulu jika kamu belum punya.
2. Buat repository baru dengan klik pada tombol New.
3. Akan terbuka halaman berikut, lalu beri nama pada repository (sebagai contoh saya menamakan TutorialLab) kemudian klik tombol Create repository.

*Abaikan error dibawah karena saya sudah membuat repository-nya.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner: N Prawira / Repository name:

Great repository names: The repository TutorialLab already exists on this account [studiois-funicular?](#)

Description (optional):

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: None Add a license: None

Create repository

4. Setelah membuat repository, buat folder yang akan menjadi local repository yang kalian gunakan. Nama folder boleh apa saja (disini saya beri nama folder Lab).
5. Buka folder "Lab" tersebut, lalu klik kanan dan klik pada Git Bash Here.
6. Di dalam folder "Lab", buat folder baru dengan nama Tutorial 1. Lalu kembali ke folder Lab.
7. Buka window Git Bash tadi dan ketikkan command `git init` lalu tekan Enter. Command `git init` akan melakukan inisiasi pada directory yang ditunjuk oleh Command Prompt, jadi pastikan kalian melakukan inisiasi folder pada directory yang benar.

```
Naufal Prawira@DESKTOP-R2NHPD4 MINGW64 ~/Documents/Lab
$ git init
fal Prawira/Documents/Lab/.git/
```

*Pada output command `git init` saya, inisiasi dilakukan pada directory `C:\Users\Naufal Prawira\Documents\Lab\.git\` yang merupakan directory dimana saya meletakkan folder "Lab".

8. Ketikkan command `git status` untuk memastikan inisiasi Git repository sudah berhasil dilakukan. Tekan Enter.
9. Masukkan username dan email GitHub yang akan disambungkan dengan pekerjaanmu di repository Git. Caranya:

1. Ketikkan command `git config --global user.name "<nama-akun-GitHub>"` lalu Enter.

```
Naufal Prawira@DESKTOP-R2NHPD4 MINGW64 ~/Documents/Lab (master)
$ git config --global user.name "NPrawira"
```

2. Ketikkan command `git config --global user.email "<email-akun-GitHub>"` lalu Enter.

```
Naufal Prawira@DESKTOP-R2NHPD4 MINGW64 ~/Documents/Lab (master)
$ git config --global user.email "
```

10. Akan muncul pop-up window Login GitHub, lakukan login seperti biasa dan window login akan menutup otomatis jika berhasil login.
11. Setelah login ke GitHub dari pop-up window tadi, buka folder "Tutorial 1" dan coba buat file Halo.txt.
12. Ketikkan command `git status` dan tekan Enter, lalu akan ditampilkan daftar untracked files dari folder tersebut.

```
Naufal Prawira@DESKTOP-R2NHPD4 MINGW64 ~/Documents/Lab (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Tutorial 1/

nothing added to commit but untracked files present (use "git add" to track)
```

13. Ketikkan command `git add .` untuk setiap untracked files.

```
Naufal Prawira@DESKTOP-R2NHPD4 MINGW64 ~/Documents/Lab (master)
$ git add .
```

14. Ketikkan command `git status` untuk melihat status dan perubahan yang akan dilakukan pada files.

```
Naufal Prawira@DESKTOP-R2NHPD4 MINGW64 ~/Documents/Lab (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Tutorial 1/Halo.txt
```

15. Kemudian, ketikkan command `git commit -m "<isi-pesanmu>"`. Field *isi pesanmu* bisa kamu beri nama apa saja. Command ini berfungsi sebagai nama dan penanda indikator perubahan yang terjadi di dalam files yang sudah dibuat sebelumnya.

```
Naufal Prawira@DESKTOP-R2NHPD4 MINGW64 ~/Documents/Lab (master)
$ git commit -m "<first commit>"
[master (root-commit) f753deb] <first commit>
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Tutorial 1/Halo.txt
```

16. Ketikkan command `git remote add origin "<isi-URL-repository-GitHub-kamu>"`.

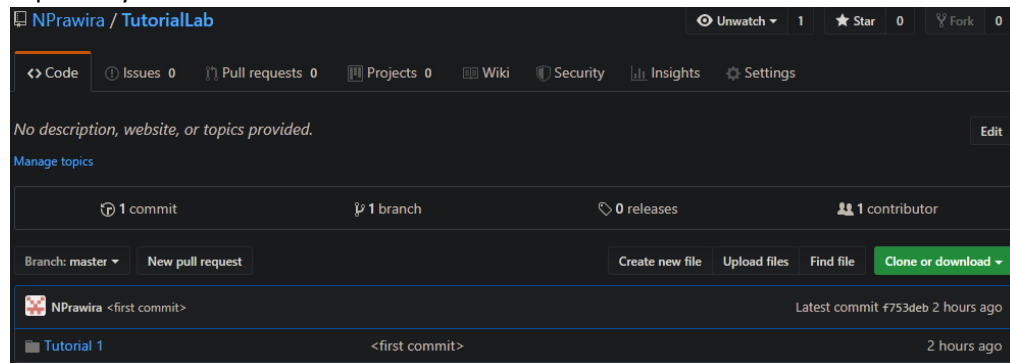
*Contoh: URL repository saya adalah `https://github.com/Nprawira/TutorialLab.git`

```
Naufal Prawira@DESKTOP-R2NHPD4 MINGW64 ~/Documents/Lab (master)
$ git remote add origin https://github.com/Nprawira/TutorialLab.git
```

17. Ketikkan command `git push -u origin master`. Command ini akan melakukan push commit pada local repository kamu ke akun GitHub yang sudah kamu hubungkan tadi.

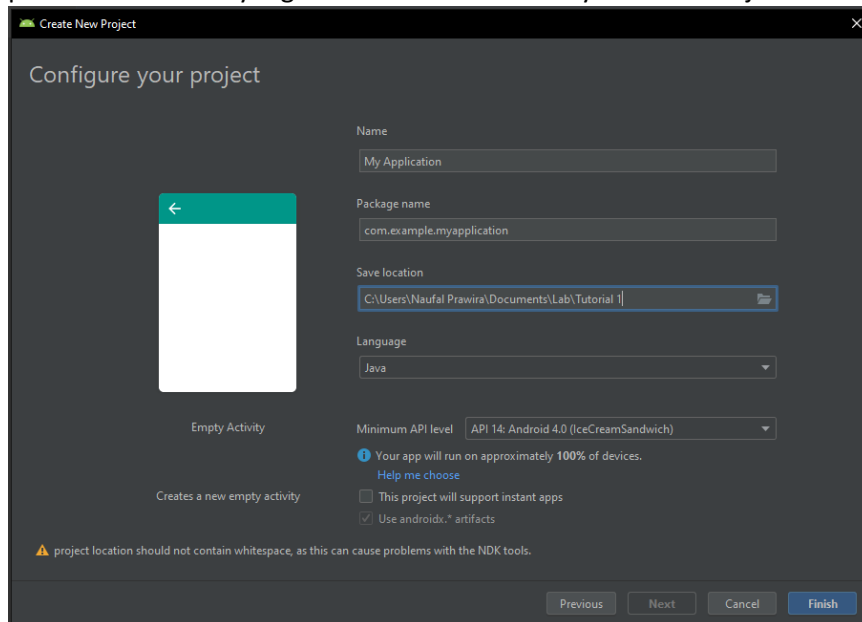
```
Naufal Prawira@DESKTOP-R2NHPD4 MINGW64 ~/Documents/Lab (master)
$ git push -u origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Writing objects: 100% (4/4), 262 bytes | 52.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/Nprawira/TutorialLab.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

18. Buka repository GitHub kamu di web dan lihat apakah files yang kamu commit sudah masuk di repository GitHub. Jika repository yang ada di GitHub sesuai dengan files yang ada di local repository kamu maka kamu sudah berhasil.

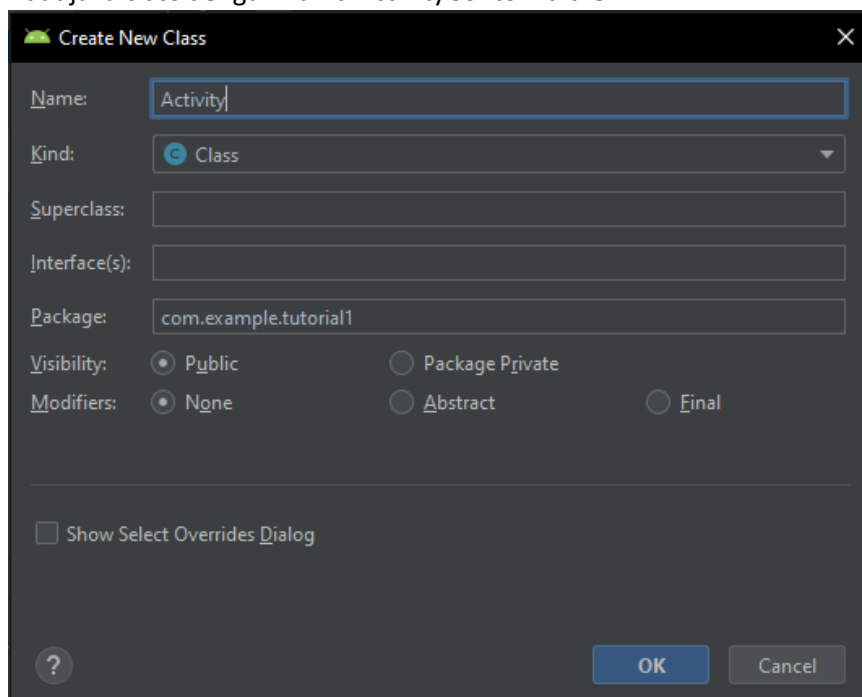


Membuat Aplikasi Pertama (Menampilkan Text Sederhana)

1. Buka aplikasi Android Studio, pilih Start a new Android Studio project.
2. Akan terbuka window Choose your project, untuk tutorial kali ini gunakan Empty Activity.
3. Tuliskan nama projek kamu (nama bebas), bahasa pemrograman gunakan Java, minimum API bebas (contoh saya gunakan API level 14) dan untuk Save Location, arahkan directory pada file Tutorial 1 yang sudah dibuat sebelumnya. Klik Finish jika sudah diatur.



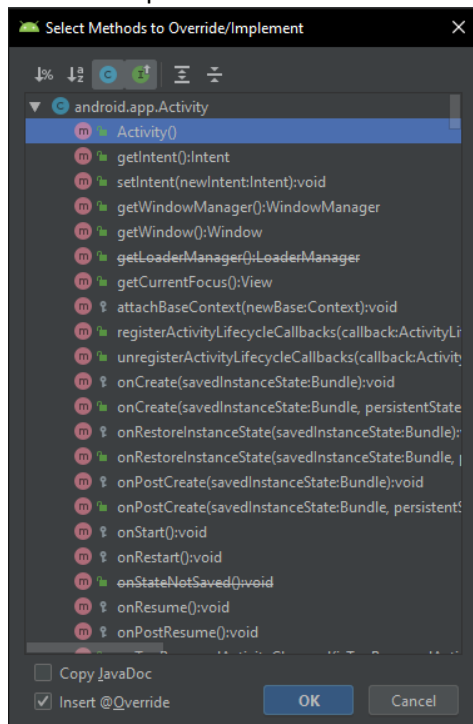
4. Tunggu beberapa saat sampai IDE selesai melakukan build dan muncul window MainActivity.java
5. Expand module App, lalu Expand folder java, klik kanan pada sub folder paling atas lalu pilih New > Java class
6. Buat java class dengan nama ActivityContoh lalu OK



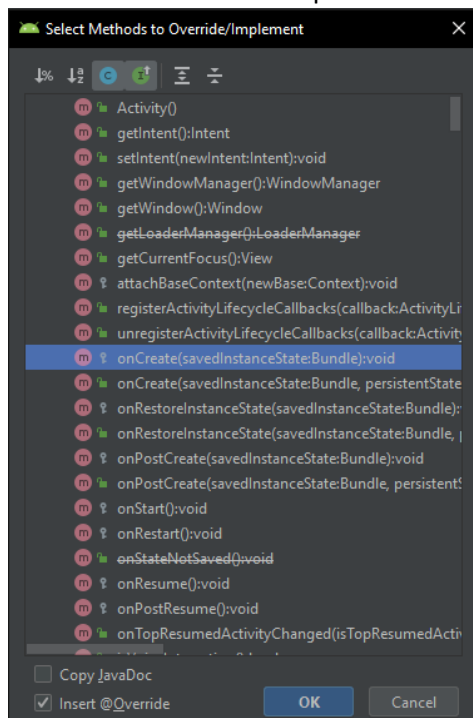
7. Lakukan inheritance (turunan) class Activity pada class yang baru saja dibuat dengan menambah keyword extends seperti berikut.

```
package com.example.tutorial11;  
  
import android.app.Activity;  
  
public class ActivityContoh extends Activity{  
}
```

8. Tekan tombol Ctrl + O pada keyboard dan akan muncul opsi untuk melakukan method Override seperti berikut.



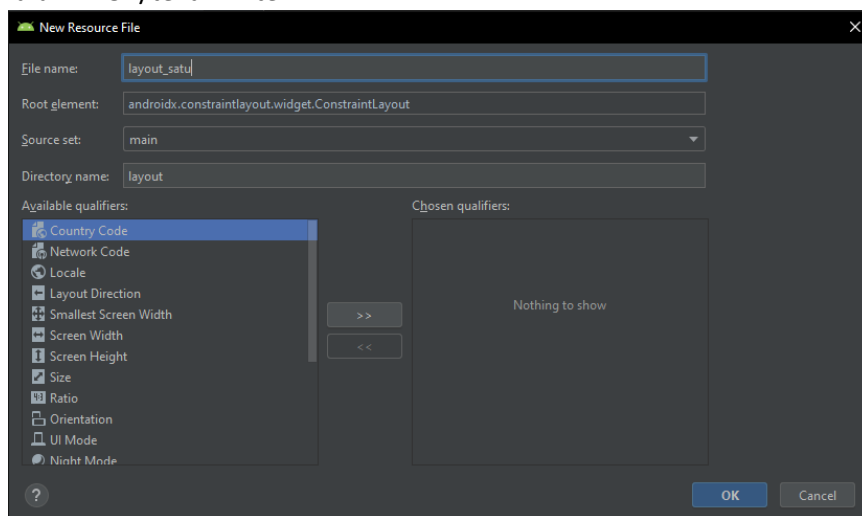
9. Ketikkan huruf "onc" lalu pilih method onCreate seperti berikut.



10. Klik OK atau tekan Enter dan halaman coding akan berubah seperti ini.

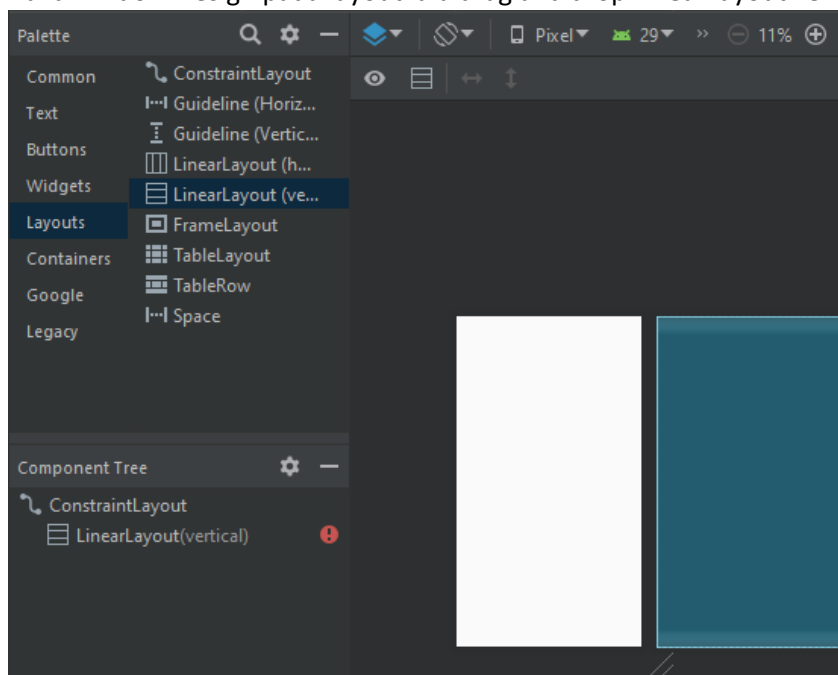
```
package com.example.tutorial1;  
  
import android.app.Activity;  
import android.os.Bundle;  
  
public class ActivityContoh extends Activity{  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
    }  
}
```

11. Setelah ini akan membuat layout dari tampilan aplikasi. Pertama expand folder res, lalu klik kanan di sub folder layout, pilih New > Layout resource file. Buat File name seperti berikut lalu klik OK/tekan Enter.

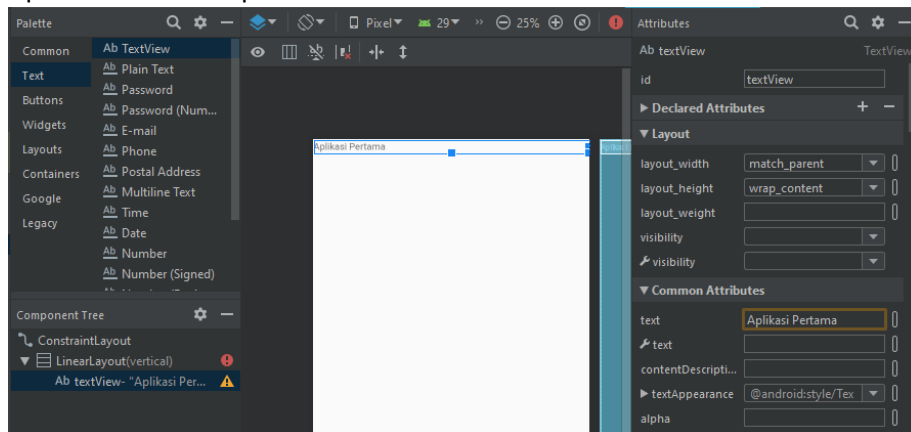


*Pada root element, kosongkan terlebih dahulu lalu tekan Ctrl + Space agar mendapat properti seperti gambar di atas

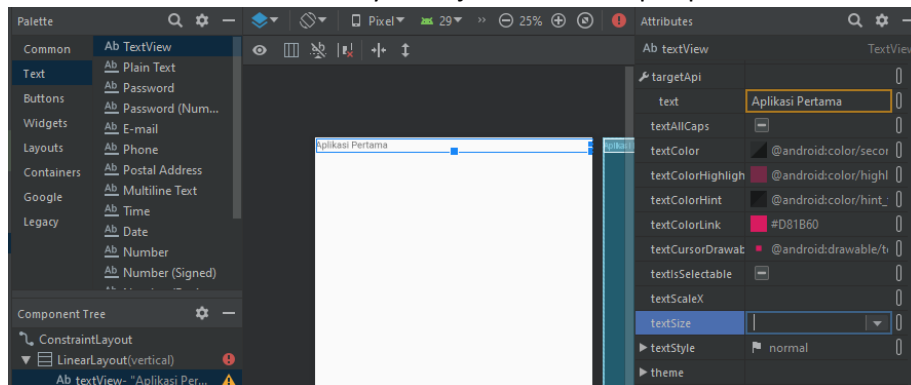
12. Buka window Design pada layout lalu drag and drop LinearLayout ke kotak putih.



13. Lalu coba drag and drop satu TextView dan ubah properties pada bagian text menjadi Aplikasi Pertama seperti berikut.



14. Kamu dapat memperbesar/memperkecil ukuran TextView dengan ubah settingan textSize. Coba buat TextView sebelumnya menjadi ukuran 24sp seperti berikut.



15. Kembali ke window Class ActivityContoh lalu ketikkan code yang di-highlight berikut setelah baris super.onCreate(savedInstanceState).

```
package com.example.tutorial1;

import android.app.Activity;
import android.os.Bundle;

public class ActivityContoh extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout_satu);
    }
}
```

16. Lakukan declare class ActivityContoh yang sudah dibuat sebelumnya ke file AndroidManifest.xml dan ubah code pada baris <activity android:name=".MainActivity"> menjadi seperti berikut.

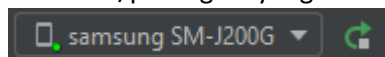
```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Aplikasi Pertama"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".ActivityContoh">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
```

*Mengubah label adalah opsional, ini akan mengubah nama aplikasi yang dibuat.

*Gunakan tombol Ctrl + Space saat mengganti file class Activity agar menghemat waktu.

17. Terakhir, jalankan program dengan menekan tombol Run (tombol Play warna hijau) atau tekan tombol Ctrl + Shift + F10. Akan muncul window untuk memilih emulator/perangkat yang akan digunakan untuk menampilkan hasilnya.



18. Hasilnya akan muncul seperti berikut.

