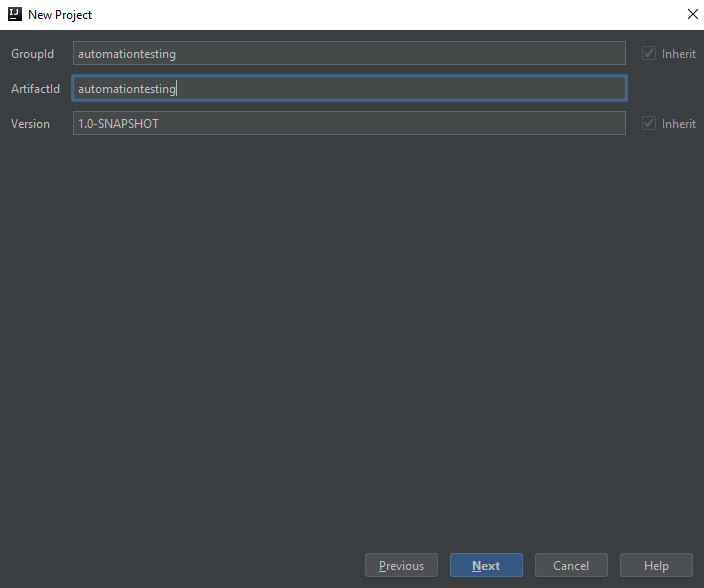
Berikut saya berikan langkah – langkah automation testing menggunakan appium pada aplikasi GoJek

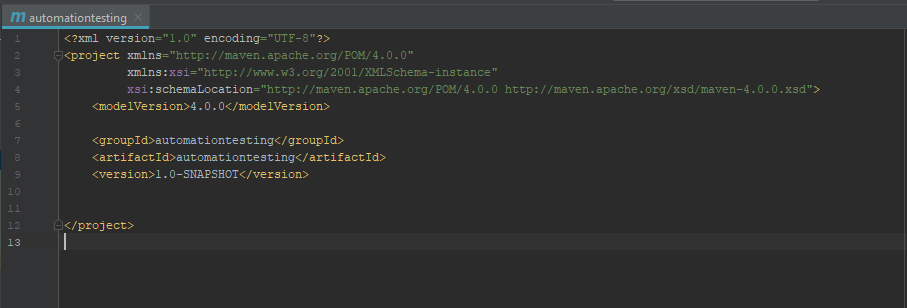
1. Membuat Project Baru



Langkah untuk membuat project baru pada Android Studio – Intellij IDE adalah *File -> New -> Project* , next hingga proses pembuatan project *Finish* . Pada GroupID dan ArtifactID isikan dengan nama *automationtesting* .

1. Membuat Aturan Dependencies

Dependencies disini berfungsi sebagai kumpulan library yang nantinya akan kita panggil, bisa saat melakukan configure dari device testing atau saat proses testing itu sendiri. Dalam menginsertkan library, terdapat beberapa basic Dependency yang harus di insertkan agar dapat menjalankan proses automation test. Berikut adalah contohnya



<groupId>automationtesting</groupId>

<artifactId>automationtesting</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>io.appium</groupId>

<artifactId>java-client</artifactId>

<version>6.1.0</version>

</dependency>

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>3.14.0</version>

</dependency>

<dependency>

<groupId>info.cukes</groupId>

<artifactId>cucumber-core</artifactId>

<version>1.2.5</version>

</dependency>

<dependency>

<groupId>info.cukes</groupId>

<artifactId>cucumber-java</artifactId>

<version>1.2.5</version>

</dependency>

<dependency>

<groupId>info.cukes</groupId>

<artifactId>cucumber-jvm</artifactId>

<version>1.2.5</version>

<type>pom</type>

</dependency>

<dependency>

<groupId>info.cukes</groupId>

<artifactId>gherkin</artifactId>

<version>2.12.2</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-api</artifactId>

<version>3.141.59</version>

</dependency>

<dependency>

<groupId>io.appium</groupId>

<artifactId>java-client</artifactId>

<version>6.1.0</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>io.appium</groupId>

<artifactId>java-client</artifactId>

<version>6.1.0</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>io.appium</groupId>

<artifactId>java-client</artifactId>

<version>6.1.0</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>io.appium</groupId>

<artifactId>java-client</artifactId>

<version>6.1.0</version>

<scope>test</scope>

</dependency>

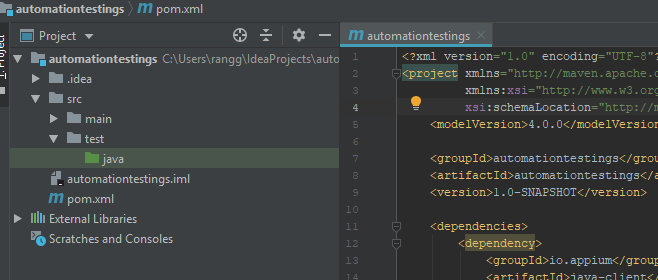
</dependencies>

Jika ingin menambahkan dependency lain kita dapat mengakses Repository dari Maven tersebut di

<https://mvnrepository.com/>

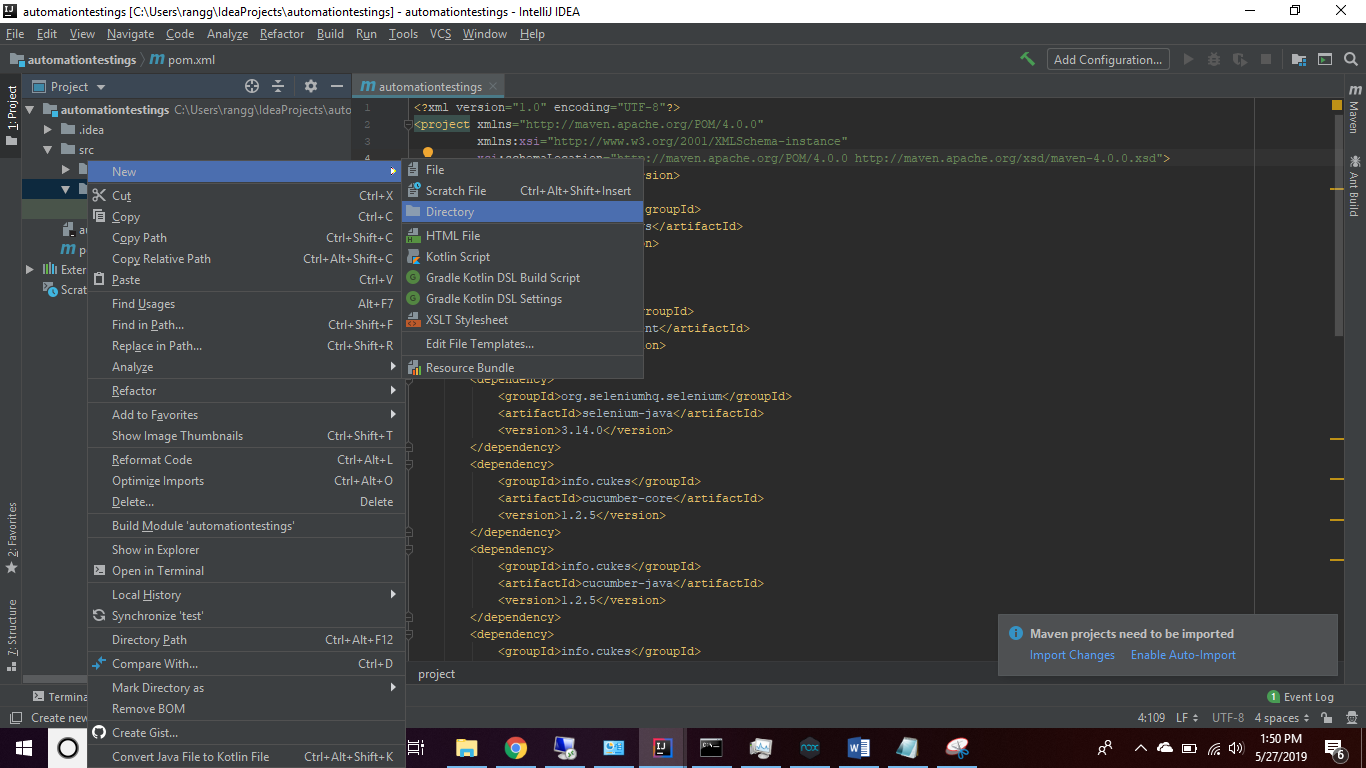
Note : Pengalaman pada saat pertama kali membuat dependency tersebut eror dan code banyak berwarna merah, hal tersebut karena memang aplikasi sedang melakukan download beberapa *plugin* yang di butuhkan aplikasi, tunggu saja hingga proses download selesai atau setelah eror tersebut menghilang.

1. Membuat Scenario Testing

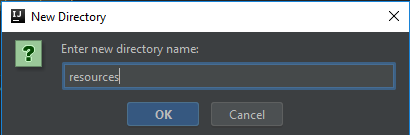


Scenario ini digunakan untuk menjalankan *automationtest* pada aplikasi yang ingin di test.

Langkahnya adalah *expand folder src -> klik kanan folder test -> New -> Directory*

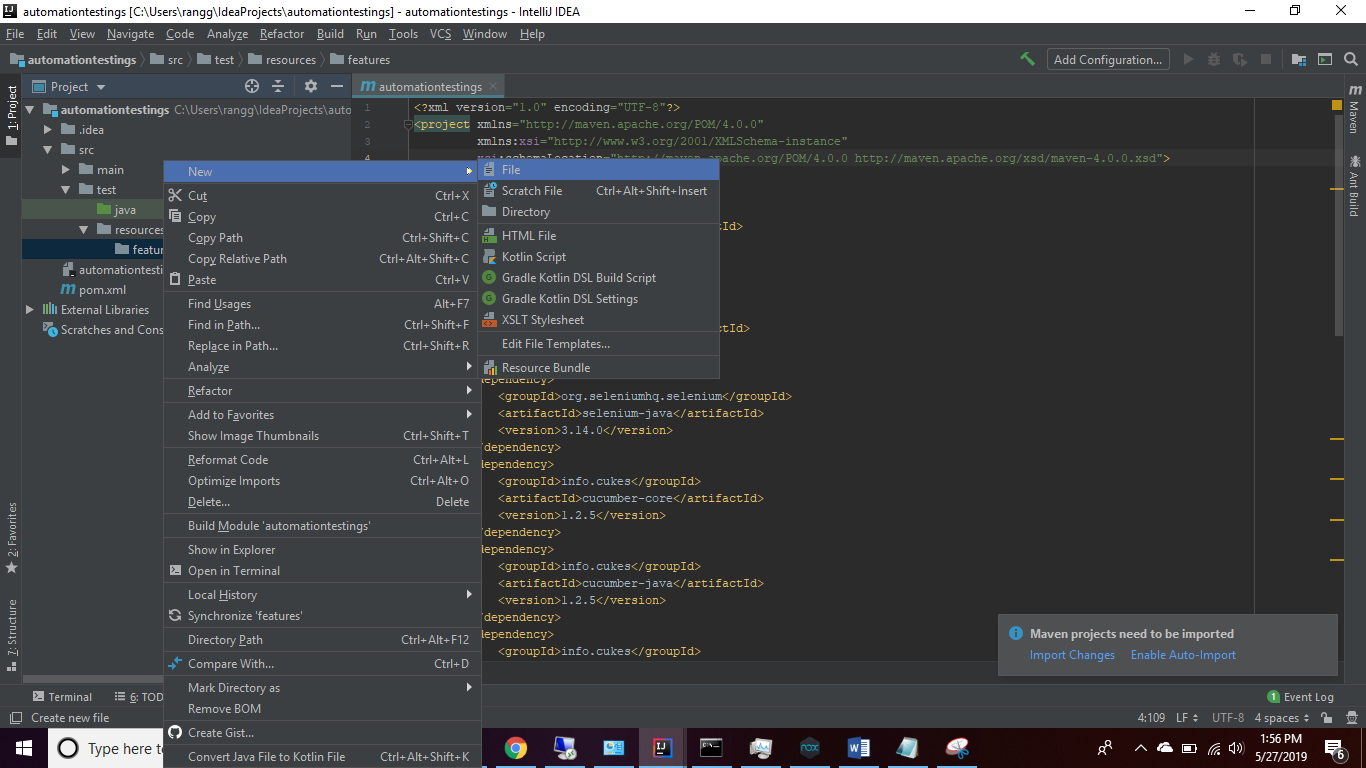


Kita beri nama *resources* untuk directory yang barusan kita bikin



Selanjutnya adalah membuat sebuah directory baru di dalam directory resources tadi, langkahnya *expand folder src -> expand folder test -> klik kanan folder resources -> New -> Directory* dan berikan nama *features* untuk nama directory tersebut.

Selanjutnya membuat file scenarionya, langkahnya *expand folder src -> expand folder test -> klik kanan folder resources -> New -> File* dan berikan nama *scenario.feature* untuk nama file tersebut.



Kemudian pada *scenario.feature* tersebut diisikan

@GoJek

Feature: Gojek

@LaunchingGojek

Scenario: Launching Gojek

Given User launch Gojek

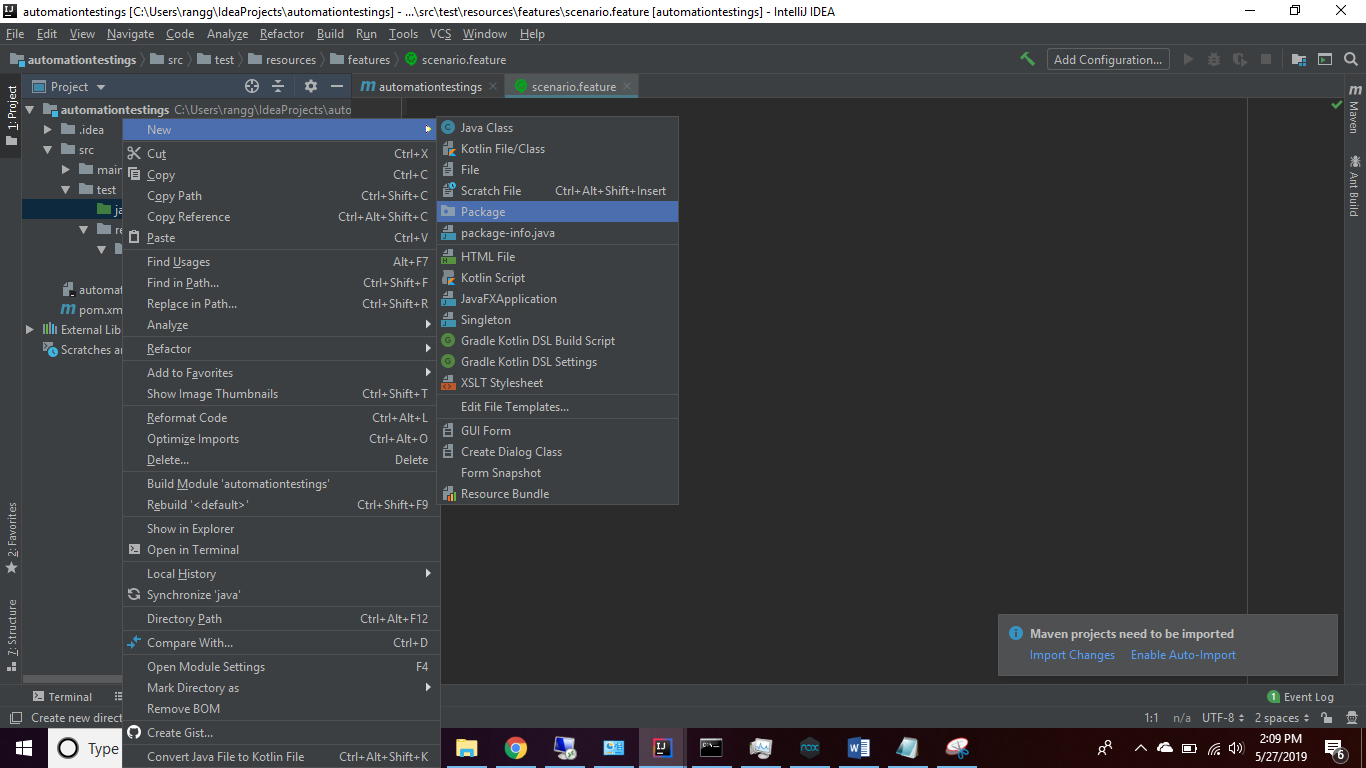
Scenario tersebut untuk menjalankan / membuka aplikasi gojek yang ada di device android.

Note : penamaan scenario diberikan .feature yang menandakan bahwa file tersebut sebagai file feature

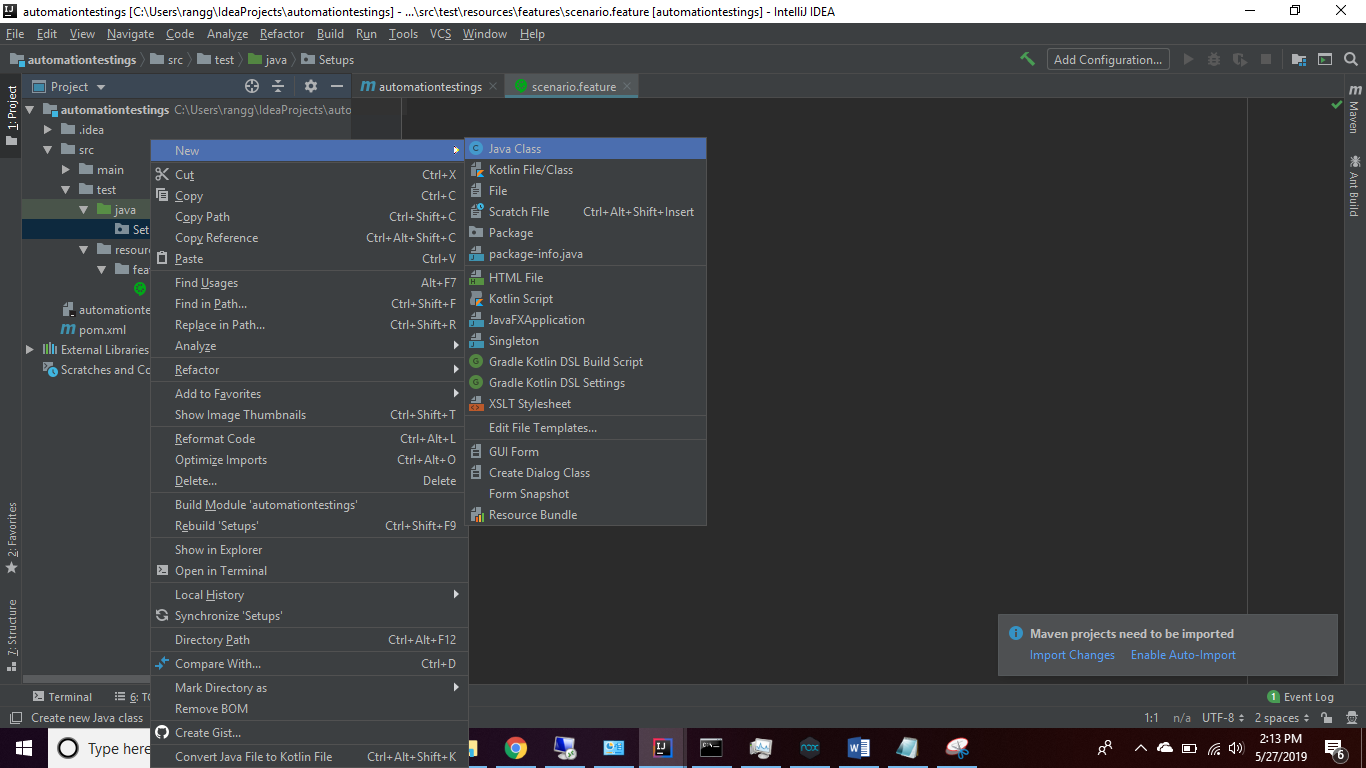
1. Membuat Class Capabilities

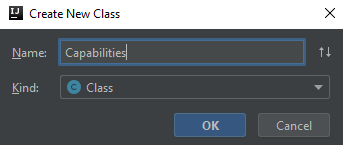
Class Capabilities digunakan untuk melakukan configure device automation, seperti versi PlatformName, OS Android, Device Name, Aplikasi Package, Aplikasi, permission pop up, paralel test, dll.

Langkahnya adalah *expand folder src -> expand folder test -> klik kanan folder java -> New -> Package* dan berikan nama *Setups* untuk nama package tersebut.



Selanjutnya membuat file capabilities, langkahnya *expand folder src -> expand folder test -> klik kanan folder java -> New -> Java Class* dan berikan nama *Capabilities* untuk nama file tersebut.





Pada Class Capabilities yang barusan di buat, isikan code berikut

package Setups;

import java.net.URL;

import java.util.concurrent.TimeUnit;

import io.appium.java\_client.android.AndroidDriver;

import io.appium.java\_client.service.local.AppiumDriverLocalService;

import org.openqa.selenium.remote.DesiredCapabilities;

public class Capabilities extends DesiredCapabilities {

protected AndroidDriver driver;

private AppiumDriverLocalService service;

protected void preparation() throws Exception {

DesiredCapabilities capabilities = new DesiredCapabilities();

capabilities.setCapability("platformName", "Android");

capabilities.setCapability("platformVersion", "4.4.2");

capabilities.setCapability("deviceName", "Android Emulator");

capabilities.setCapability("appPackage", "com.gojek.app");

capabilities.setCapability("appActivity", "com.gojek.app.home.HomeActivity");

service = AppiumDriverLocalService.buildDefaultService();

service.start();

String service\_url = service.getUrl().toString();

System.out.println("Appium Service Address: " + service\_url);

driver = new AndroidDriver(new URL(service\_url), capabilities);

driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);

}

public void stopServer() {

service.stop();

}

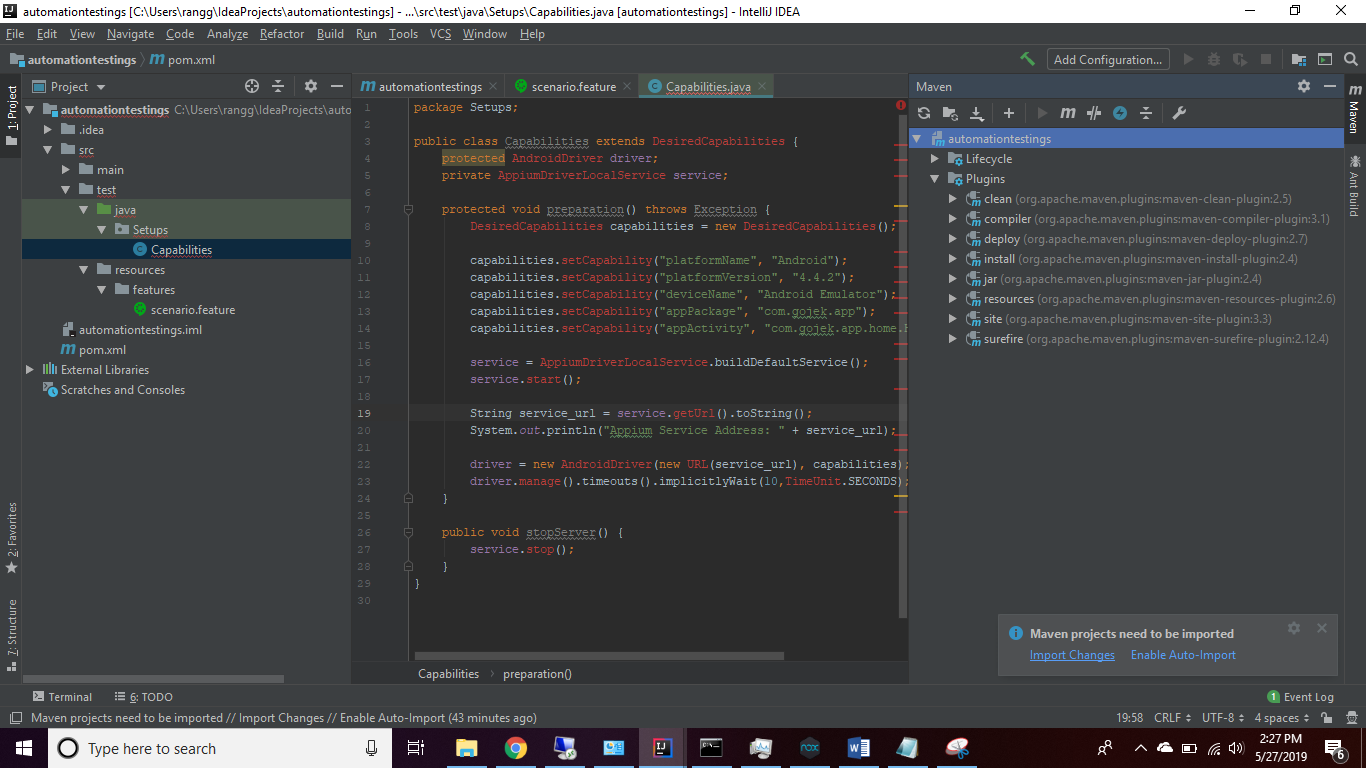
}

Note : Seperti yang dijelaskan tadi, Class Capabilities digunakan untuk melakukan configure device automationnya, dicode tersebut pada code capabilities.setCapability digunakan untuk mengatur device dan aplikasi apa yang akan di automation.

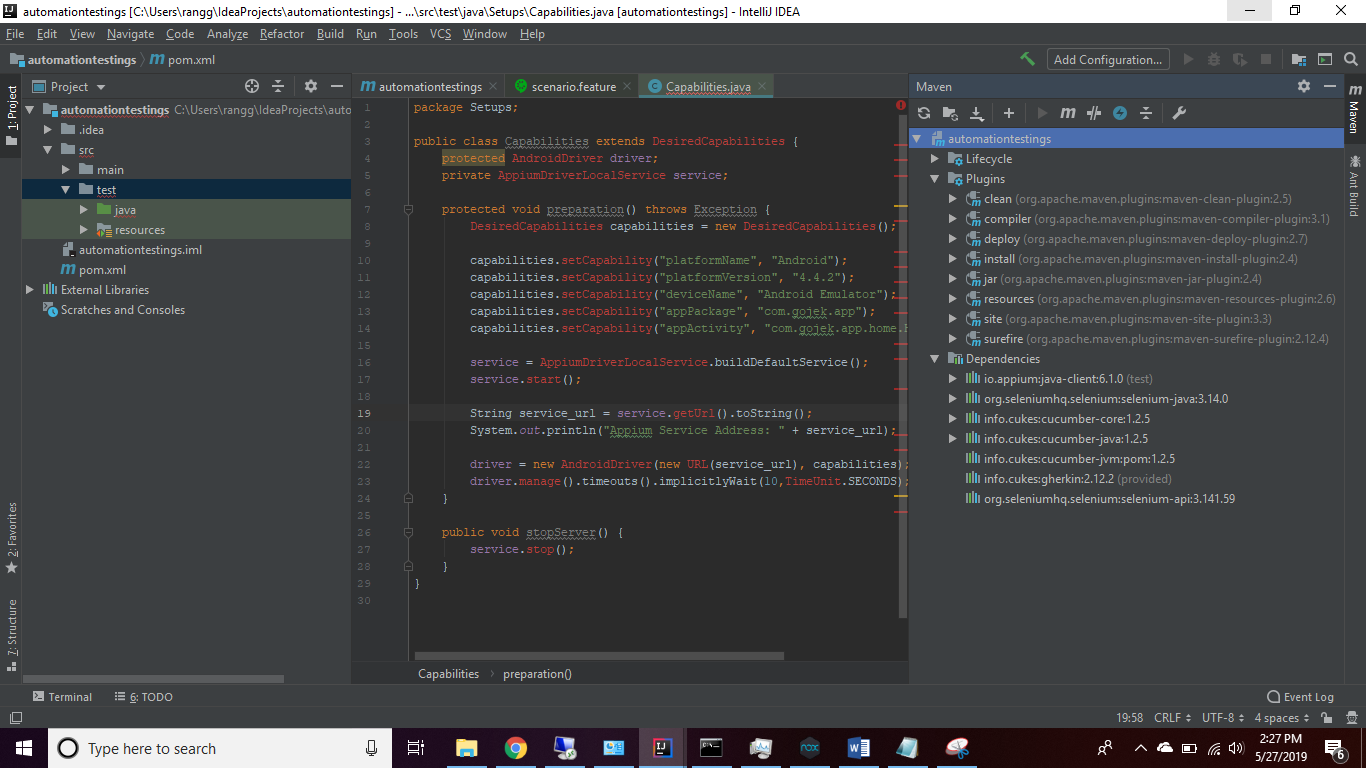
Pada bagian code capabilities.setCapability untuk platformVersion kita dapat mengecek versi android device di about phone dan untuk deviceName kita bisa cek dengan menggunakan adb devices yang dapat kita jalankan di cmd untuk mendapatkan UDID dari devices yang ingin kita testing.

Pada saat code tersebut dimasukkan kedalam class Capabilities, class tersebut akan eror karena resource yang tadi kita buat belum ke set pada Intellij IDEA dan kita belum melakukan import library dari dependencies.

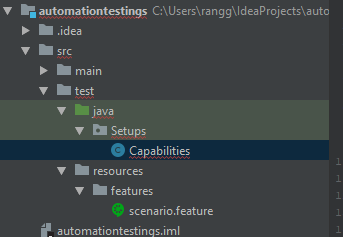
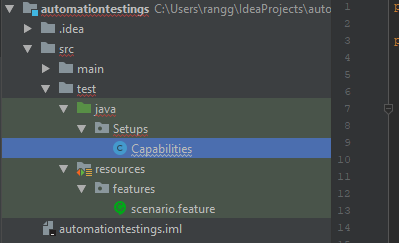
Langkah pertama yang perlu kita lakukan adalah masuk di *tab Maven* di bagian kanan page dan dapat dilihat bahwa library dependencies yang tadi kita buat tidak ada dan jika kita lihat pada pojok kanan bawah ada notif bahwa project membuthkan imported, maka kita pilih Import Change ( agar dependencies berhasil ke import ) atau Enable Auto-Import ( aplikasi akan selalu otomatis melakukan import apa saja yang dibutuhkan class Capabilities )



Setelah kita memilih pilihan dari notif tersebut maka Import Dependencies berhasil dilakukan

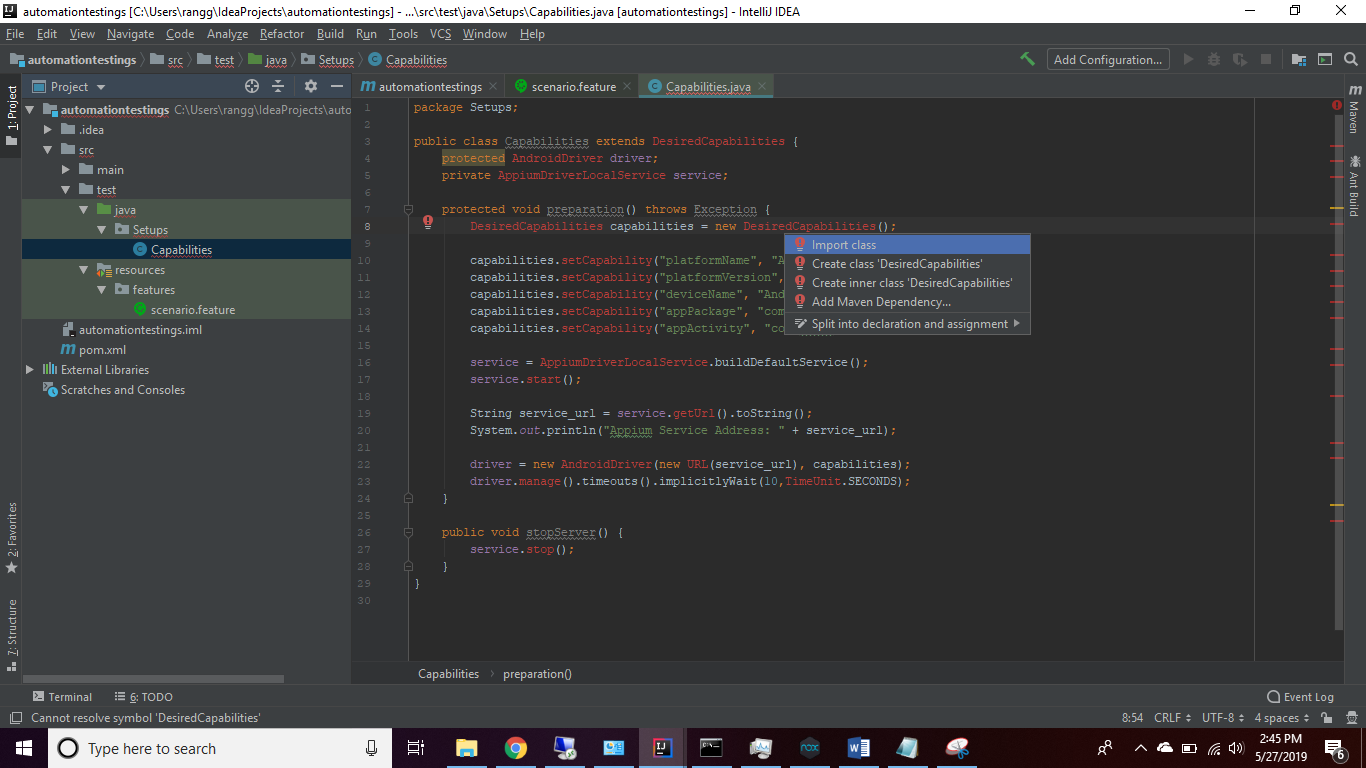


Dan jika kita sadar bahwa bentuk dari directory resources berubah dan itu menandakan bahwa import yang barusan kita buat berhasil dilakukan.

SEBELUM IMPORT SETELAH IMPORT

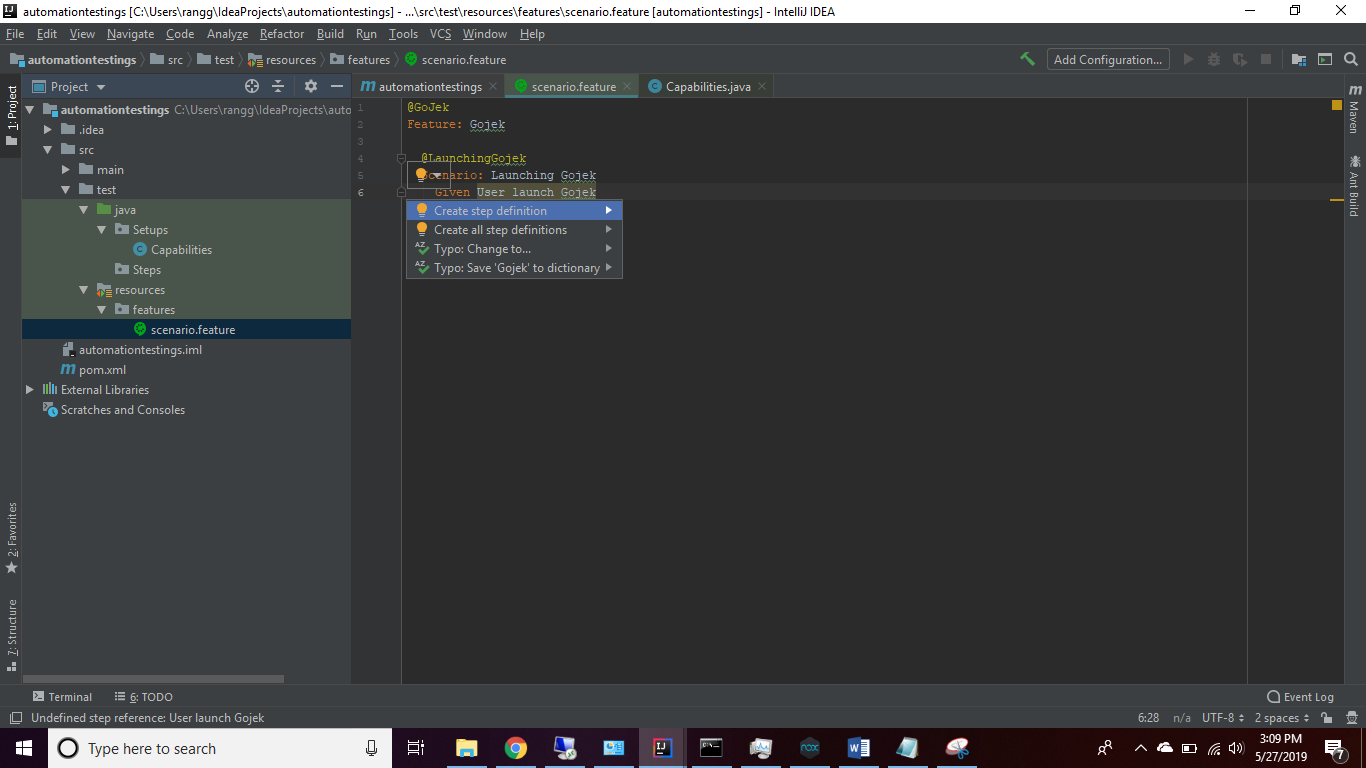
Langkah selanjutnya setelah kita melakukan import dependencies adalah menuliskan code import library dependencies tadi ke class Capabilities, bisa dengan mengetikan code import class atau bisa dengan melakukan kombinasi **alt + enter** pada kalimat yang berwarna merah (warna merah berarti code tersebut masih error sedangkan warna kuning code tersebut warning) dan pilih **Import Class**. Lakukan langkah tersebut hingga semua import yang dibutuhkan berhasil di insertkan pada code tersebut / hingga tidak ada kalimat yang berwarna merah.



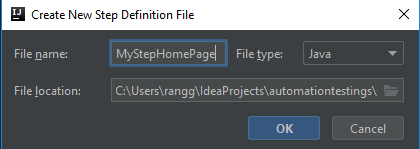
1. Membuat Steps Definition

Langkah ini digunakan untuk mengisikan fungsi atau langkah – langkah yang kita gunakan dalam proses automation, seperti melakukan klik tombol ataupun mengisikan isian textarea.

Langkahnya sama saat membuat class Capabilities, dari pas membuat packagenya, yang membedakan adalah saat membuat file class Stepsnya, yakni kita masuk scenario.feature -> arahkan mouse ke Userlaunch Gojek -> tunggu hingga muncul lampu, jika tidak muncul maka arahkan mouse ke sisi lain di User launch Gojek tersebut -> expand tanda arrow -> pilih ‘Create step definition’



Selanjutnya berikan nama file class step yang kita buat dengan nama MyStepHomePage dengan File type nya Java.



Selanjutnya kita akan menuliskan code java pada class step yang barusan kita buat, dapat kita isikan seperti ini :

package Steps;

import Setups.Capabilities;

import cucumber.api.PendingException;

import cucumber.api.java.Before;

import cucumber.api.java.After;

import cucumber.api.java.en.Given;

import org.openqa.selenium.By;

public class MyStepHomePage extends Capabilities {

@Before({"@GoJek"})

public void setup() throws Exception {

preparation();

}

@Given("^User launch Gojek$")

public void userLaunchGojek() throws Throwable {

//driver.findElement(By.id("com.android.packageinstaller:id/permission\_allow\_button")).click();

driver.findElement(By.id("com.gojek.app:id/button\_login")).click();

driver.findElement(By.id("com.gojek.app:id/btn\_login")).click();

driver.findElement(By.id("com.gojek.app:id/auth\_back\_button")).click();

driver.findElement(By.id("com.gojek.app:id/btn\_signup")).click();

driver.findElement(By.id("com.gojek.app:id/input\_name")).sendKeys("Rangga Deputra");

driver.findElement(By.id("com.gojek.app:id/input\_email")).sendKeys("Rangga.Depoetra@Gmail.com");

driver.findElement(By.id("com.gojek.app:id/input\_phone")).sendKeys("82134503856");

//driver.findElement(By.id("com.android.packageinstaller:id/permission\_allow\_button")).click();

}

@After({"@GoJek"})

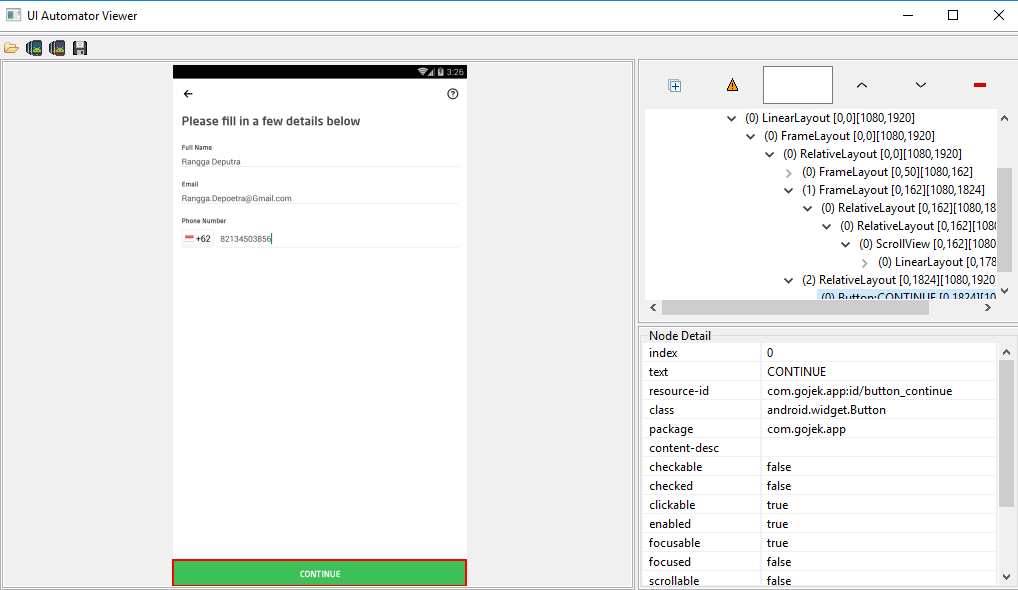
public void stop(){

stopServer();

}

}

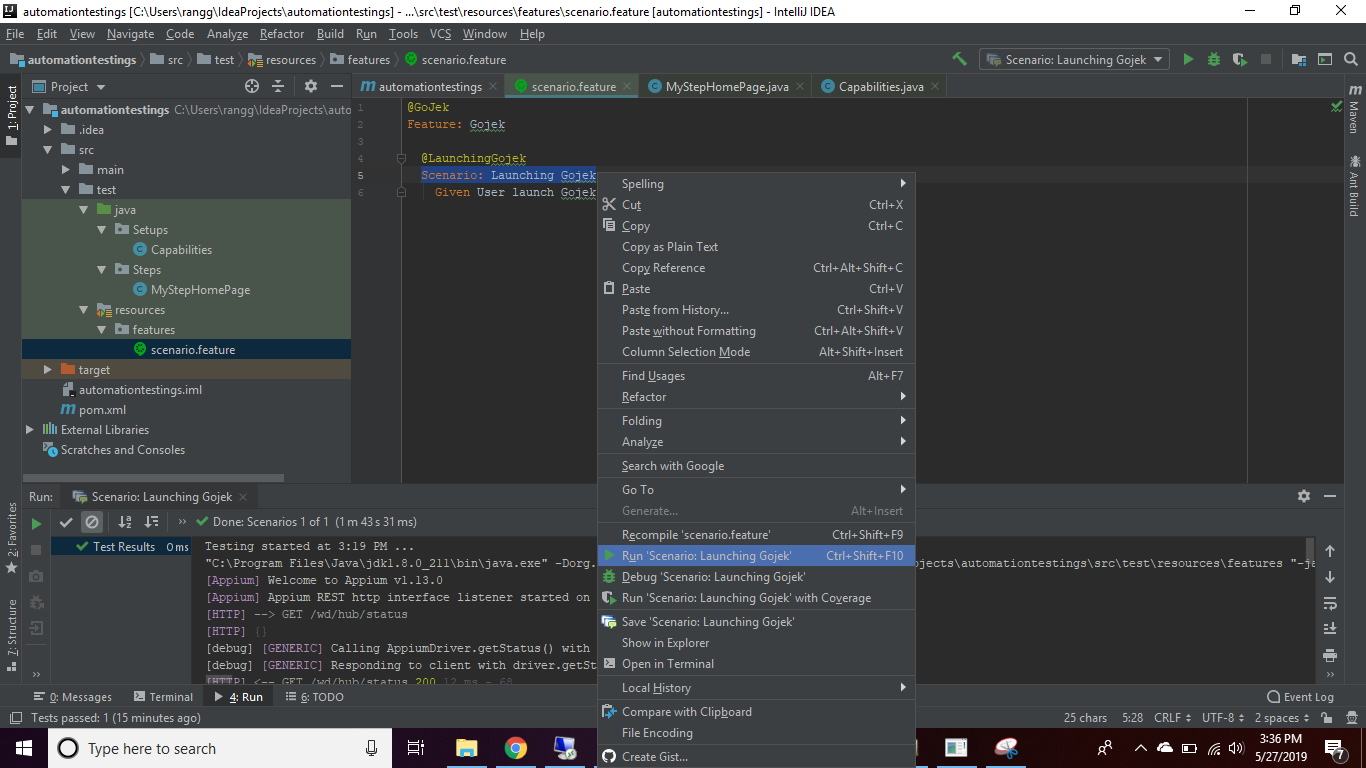
Pada code driver.findElement(By.id kita dapat mengetahui ID dari tombol ataupun inputan pada aplikasi tester dengan menggunakan tool yang telah di sediakan di sdk, yakni **uiautomatorviewer** yang ada di directory **Android/Sdk/tools/bin** .



1. Running Automation

Tahap selanjutnya adalah menjalankan Automation yang telah kita buat dari awal, caranya adalah :

1. Masuk tab scenario.feature
2. Klik mouse pada scenario Scenario: Launching Gojek yang telah dibuat
3. Klik kanan -> Run ‘Scenario: Launching Gojek’



Dan Proses Automation akan otomatis berjalan.