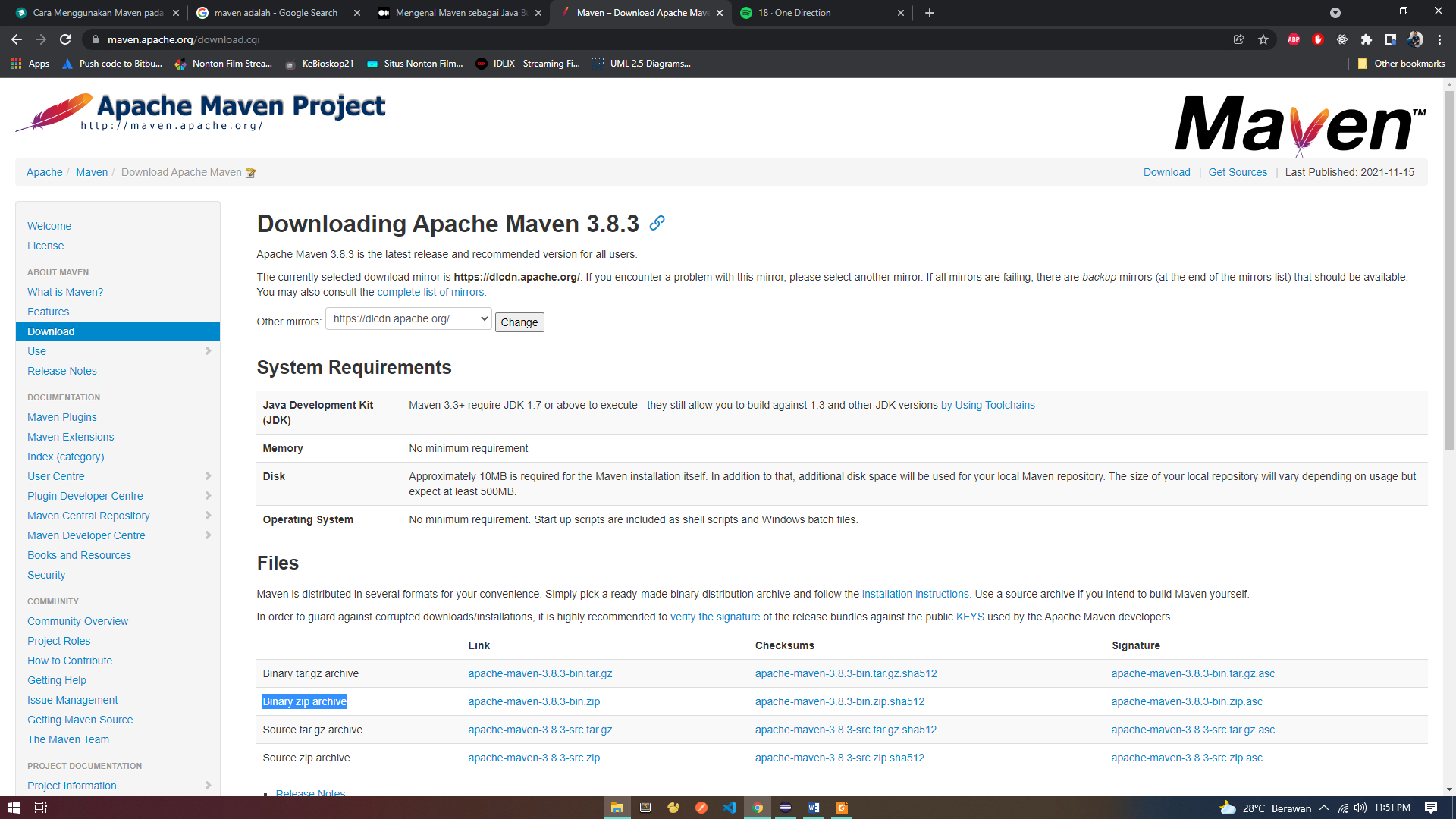
|  |  |
| --- | --- |
|  | **Mata Kuliah : Pemograman Berbasis Objek dan Lab**  **Materi : Apache Maven**  **Kelas : TI-20-PA / TI-20-KA**  **Pengajar : Febri Damatraseta S.T, M.Kom** |

***Apache Maven*** merupakan salah satu *build tools* yang paling umum digunakan dalam pembuatan projek pemograman JAVA. Jika sebelumnya kalian membuat sebuah project java tanpa adanya struktur kerangka kerja atau membuat langsung dengan default bawaan IDE kalian masing-masing. Berbeda dengan *Apache Maven*, *java build tools* ini menggunakan konsep *Project Object Model* (POM). POM sendiri berisi tentang informasi dan konfigurasi yang digunakan oleh *Maven* untuk membuat sebuah projek. Pada dasarnya POM itu sendiri merupakan sebuah file XML yang terdapat pada project *Maven* dan didalam File tersebut berisi tentang konfigurasi dari project yang akan dibuat. Keunggulan dari penggunaan *Maven* ini diantaranya membuat struktur project sendiri sehingga project tersebut dapat dibuka oleh berbagai IDE yang telah mendukung *Apache Maven*. IDE yang telah mendukung *Apache Maven* sebagian besar seperti *Eclipse, Netbeans,* dan *Jetblin/Intellij.* Selain pemograman JAVA, Apache Maven dapat digunakan oleh bahasa pemograman *C#, Kotlin, Ruby, Scala*, dan lain-lain.

* **Installasi *Apache Maven* pada Windows**

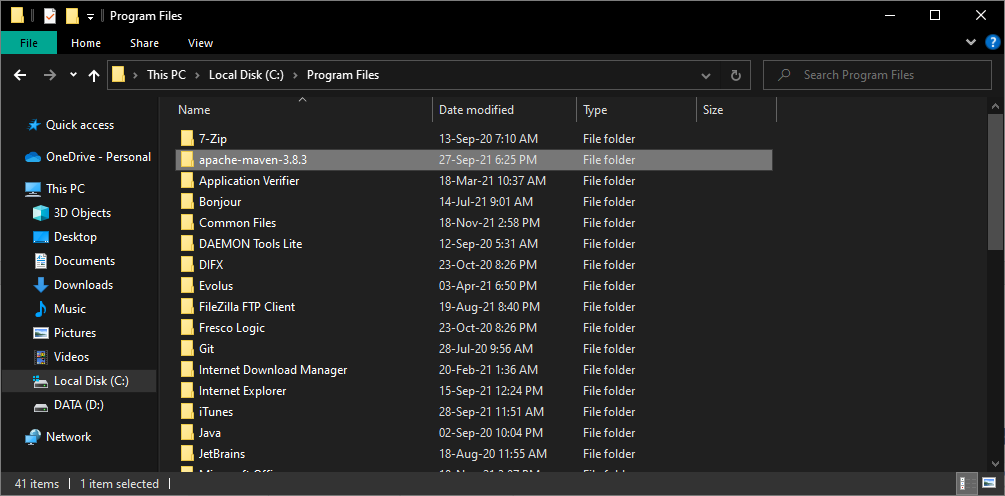
Untuk dapat menggunakan Maven, diharuskan mengunduh *File Binary Apache Maven* di website *Apache Maven*: <https://maven.apache.org/download.cgi>



Gambar 1. Halaman Website *Apache Maven*

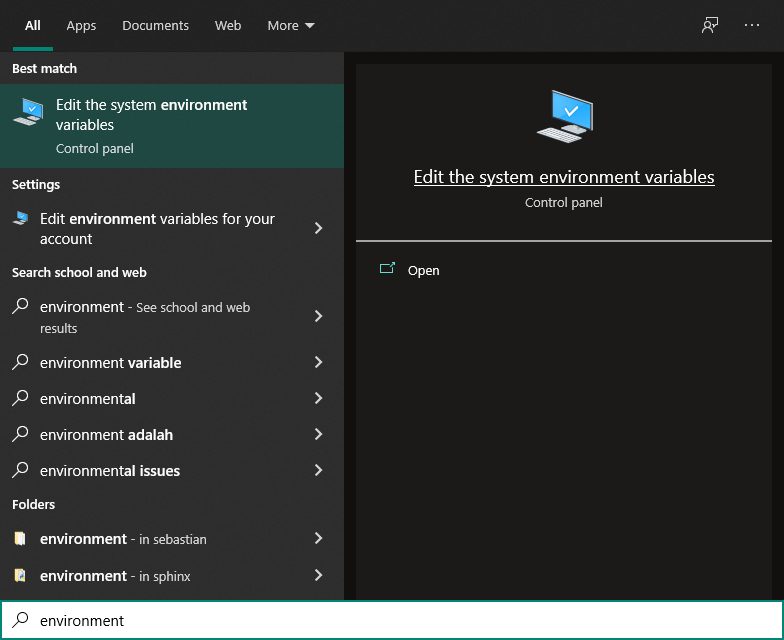
Seperti pada gambar diatas versi *Apache Maven* yang terbaru saat ini ialah versi 3.8.3, setelah kalian berhasil membuka halaman tersebut silakan mengunduh ***Files Binary zip achive***. Setelah berhasil mengunduhnya, silakan lakukan tahap seperti dibawah ini:

1. Ekstrak file *zip*, lalu *cut* folder ***apache-maven-3.8.3-bin*** dan *paste* kedalam directori ***C:\Program Files\***



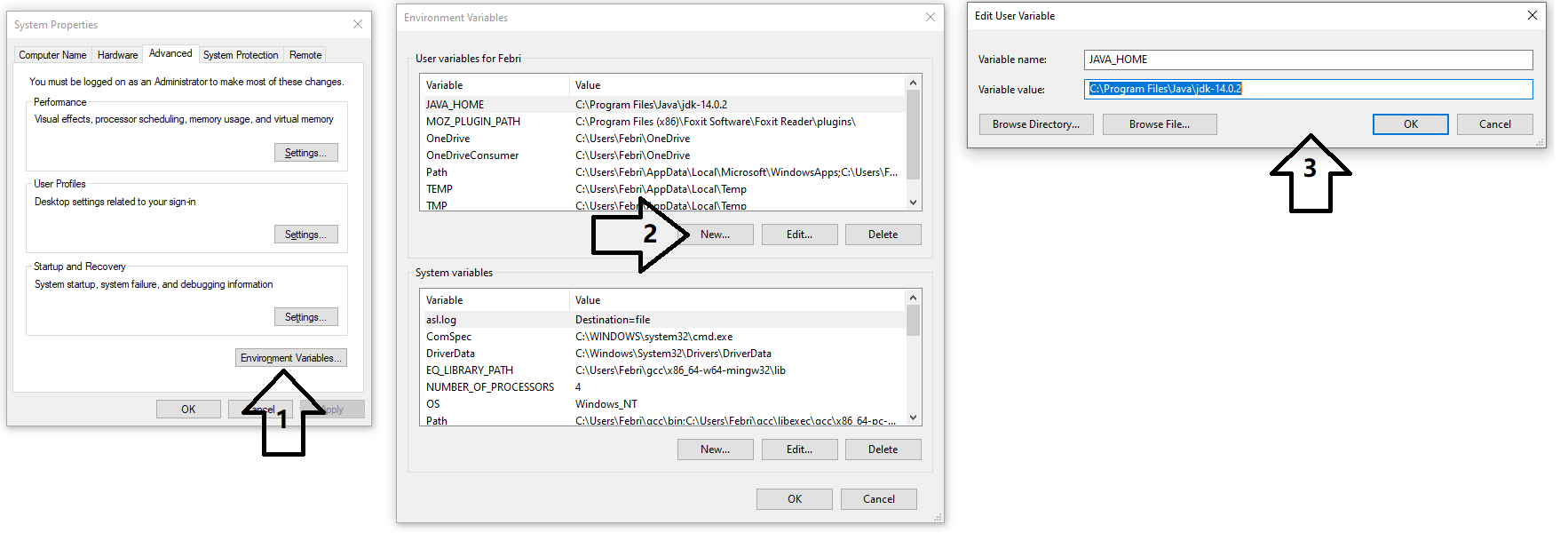
Gambar 2. Directory path Apache Maven

1. Menambahkan *environment variable* milik *maven*, tekan tombol Start lalu tuliskan environment pada search taskbar. Pilihlah applikasi yang bernama *Edit the system environment variables*.

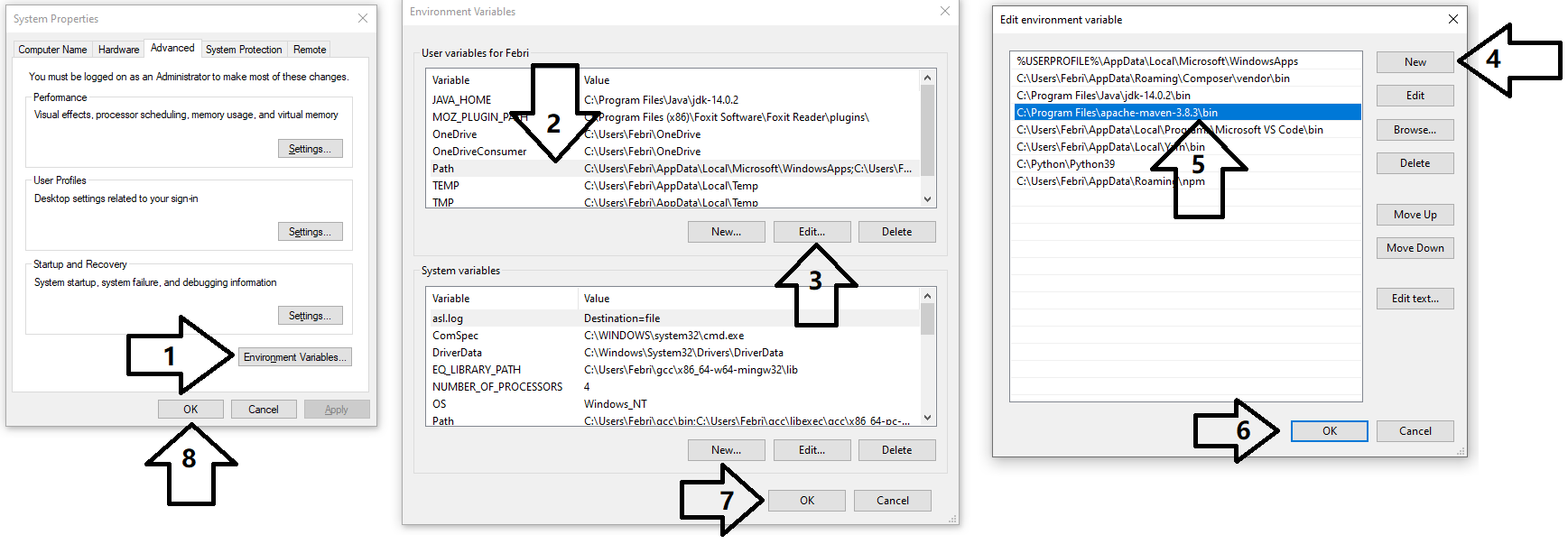


Gambar 3. Icon environment variables

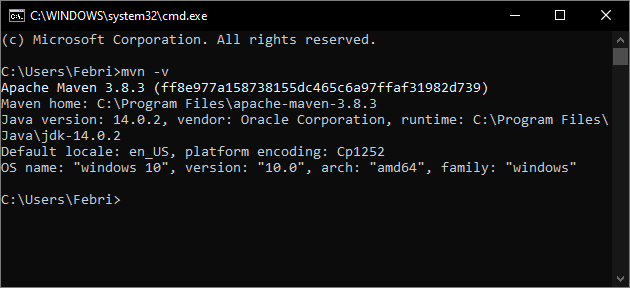
1. Menambahkan variable JAVA\_HOME dengan cara klik tombol Environment Variables muncul pop up windows, pada bagian User variables klik tombol NEW lalu isikan pada windows New User Variable, kolum isian Variable Name adalah JAVA\_HOME dan kolum isian Variable value adalah directory path installer java jdk. Setelah itu klik OK.



Masih pada popup windows Environment Variable, pilih variable path lalu klik tombol Edit. Masukan directory path Apache Maven kedalam variable path dengan cara klik tombol New. Setelah memasukan directory path Apache Maven klik OK, OK dan OK sehingga menutup windows dialog System Properties.



1. Buka command prompt pada tombol start dan ketikan ***mvn -v*** jika muncul seperti dibawah ini berarti installasi maven telah selesai dilakukan.



Gambar 4. Installasi maven berhasil

* **Cara menggunakan Apache Maven sebagai project JAVA**

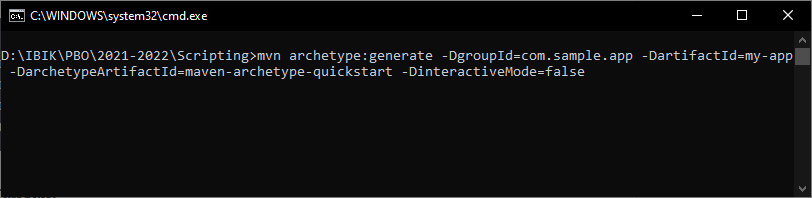
Untuk membuat sebuah project maven hal yang perlu dilakukan ialah membuka command prompt dan masukan perintah ini tanpa ada break line:

|  |
| --- |
| mvn archetype:generate  -DgroupId=com.sample.app  -DartifactId=my-app  -DarchetypeArtifactId=maven-archetype-quickstart  -DinteractiveMode=false |

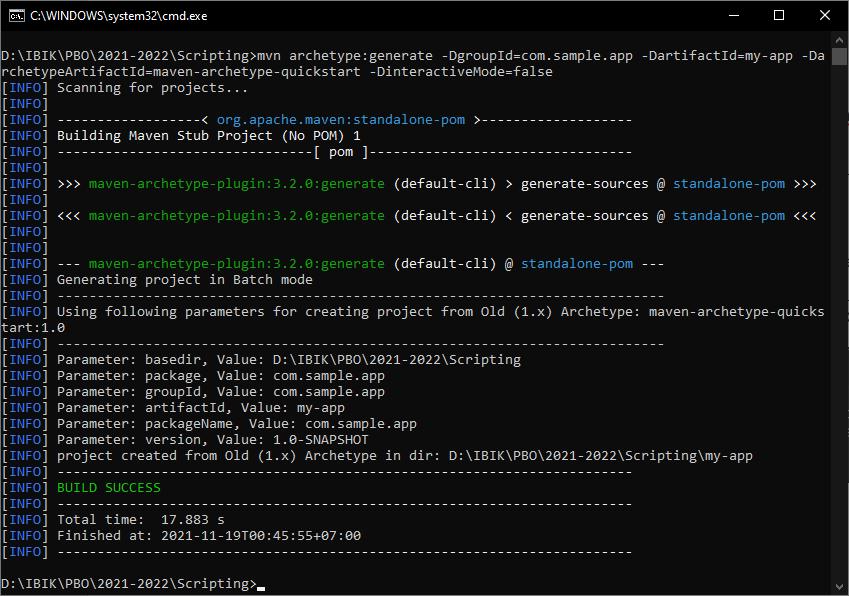
Keterangan:

* Argumen archetype:generate artinya kita akan membuat sebuah archetype;
* Argumen -DgroupId=com.sample.app adalah namespace atau nama package dari aplikasi;
* Argumen -DartifactId=my-app adalah nama direktori dari aplikasi;
* Argumen -DarchetypeArtifactId=maven-archetype-quickstart artifact atau template yang akan digunakan;
* Argumen -DinteractiveMode=false untuk mematikan mode iteraktif.

Berikut ini adalah contoh implementasi syntax untuk membuat project dengan maven:



Setelah menuliskan syntax seperti diatas lalu tekan tombol Enter pada keyboard anda. Maka akan men-generate project maven seperti dibawah ini:



Gambar 5. Build Project Maven

Berikut ini adalah gambaran dari struktur project maven:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | | Direktori atau Nama Fail | Deskripsi | | project home | berisi file pom.xml dan semua direktori | | src/main/java | berisi kode program | | src/main/resources | berisi file pendukung untuk project seperti gambar, dokumen, dll. | | src/test/java | berisi kode program untuk melakukan testing | | src/test/resources | berisi file pendukung untuk melakukan testing | | target | berisi file hasil kompilasi seperti .jar dan .class | | pom.xml | berisi konfigurasi project | |

Sedangkan untuk menjalankan program project Maven ini dengan cara masuk kedalam directory path project di command prompt, dan tuliskan perintah:

|  |
| --- |
| mvn clean package |

Perintah tersebut digunakan untuk mengcompile, sedangkan perintah dibawah ini dijalankan untuk menjalankan projek maven:

|  |
| --- |
| mvn exec:java -Dexec.mainClass=com.example.app.App |

Perintah diatas adalah argument untuk menjalankan main class pada projek maven, “com.example.app.App” menandakan penulisan untuk memanggil file berdasarkan packge yang telah ditentukan.

