**LAPORAN PRAKTIKUM 1**

**PEMROGRAMAN II**



**STEFFI NOVA MARGARETH**

**(4512212043)**

**TEKNIK INFORMATIKA**

**FAK. TEKNIK – UNIVERSITAS PANCASILA**

**BAB I**

**PENDAHULUAN**

Suatu ketika kita ingin membuat sebuah project java, apa yang kita lakukan pertama kali? Biasanya kita akan langsung membuka IDE lalu mulai membuat project dari sana. Dari sini sekilas nampak tidak ada masalah, namun yang terjadi adalah project yang kita buat dengan IDE tertentu belum tentu bisa dibuka dari IDE yang lain (bisa tapi dengan cara yang cukup melelahkan), karena masing­masing IDE memiliki standarnya sendiri.

Sekarang kita akan coba menambahkan library ke dalam project tadi. Dengan cara manual kita bisa mendownload semua file yang kita butuhkan dan menaruhnya didalam classpath project. Tetapi kadang tidak semudah itu, bisa jadi library itu membutuhkan dependency library yang lain dan dependency library itu membutuhkan dependency library yg lain lagi dan seterusnya. Bayangkan kalo harus mencari satu persatu, tentu sangat melelahkan. Inilah yang dinamakan dengan jar hell.

Masalah belum berhenti sampai disini, misalnya kita ingin berbagi project kita dengan orang lain. Biasanya kebanyakan orang malas untuk mengikutkan library dari project yang akan dibagikan tersebut, karena library yang banyak mengakibatkan file membesar dan dengan bandwidth yang pas­ pasan akan memakan waktu lama pada saat upload. Akibatnya adalah orang lain yang ingin memanfaatkan project ini harus mencari library yang dibutuhkan satu persatu, sangat tidak praktis.

Untuk mengatasi permasalahan diatas kita bisa menggunakan tools dari apache yang bernama **maven**. **Maven** memiliki struktur project standar yang membuat maven ramah terhadap IDE apapun. Maven juga menyediakan online repository yang memudahkan kita untuk download secara otomatis library yang kita butuhkan + dependency­nya.

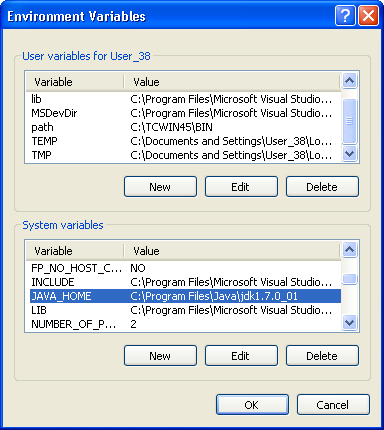
**BAB II**

**LANDASAN TEORI**

1. **Instalasi Maven**

* **JDK and JAVA\_HOME**

Pastikan JDK terinstal, dan "JAVA\_HOME" variabel ditambahkan dalam environment variables Windows, dan arahkan ke folder JDK.



* **Download Apache Maven**

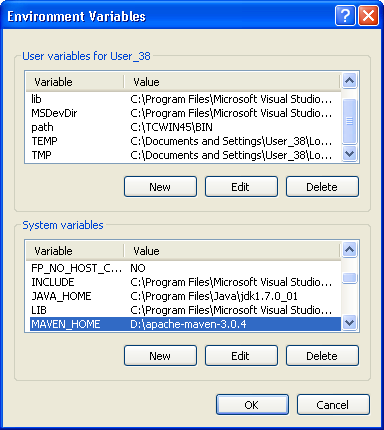
Kunjungi website Maven resmi ini, pilih versi dan klik pada link download, misalnya apache-maven-2.2.1-bin.zip.

* **Extract it**

Ekstrak file zip download. Misalnya pada drive D:

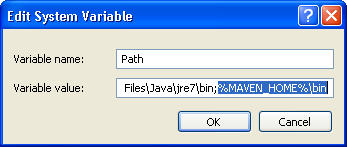
* **Add MAVEN\_HOME**

Tambahkan variabel MAVEN\_HOME baru ke environment Windows, dan arahkan ke folder Maven Anda.

****

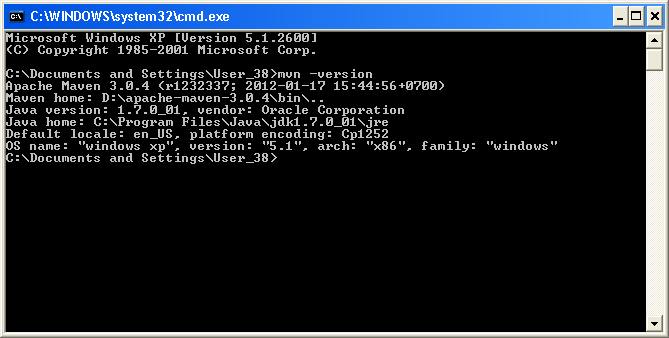
* **Add Path**

Perbarui variabel PATH, tambahkan "Maven bin folder" jalan, sehingga Anda dapat menjalankan perintah Maven di mana-mana.



* **Verification**

Cek apakah maven sudah terinstal, pada cmd ketik "mvn-version".



Beberapa perintah maven yang sering digunakan antara lain:

­ mvn clean : untuk membersihkan compile sebelumnya.

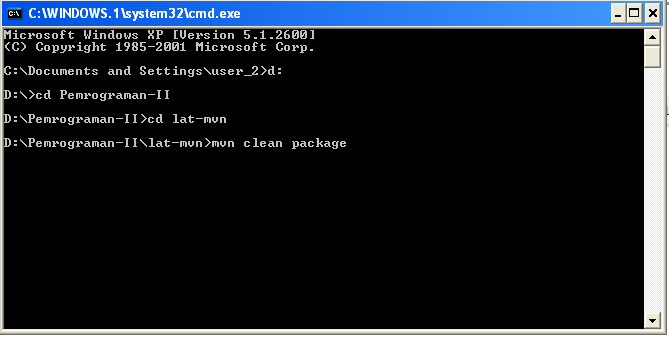
­ mvn install : untuk memasang hasil build project ke dalam local repository.

­ mvn package : untuk build project.

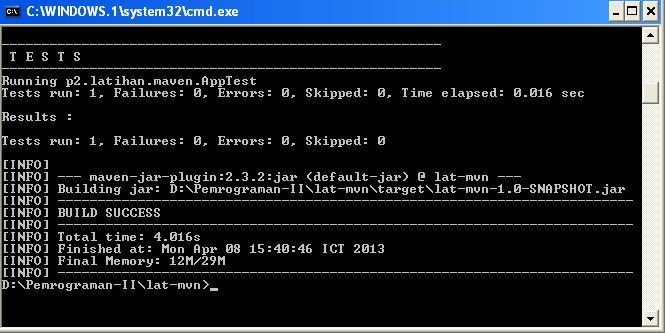
**BAB III**

**HASIL PRAKTIKUM**

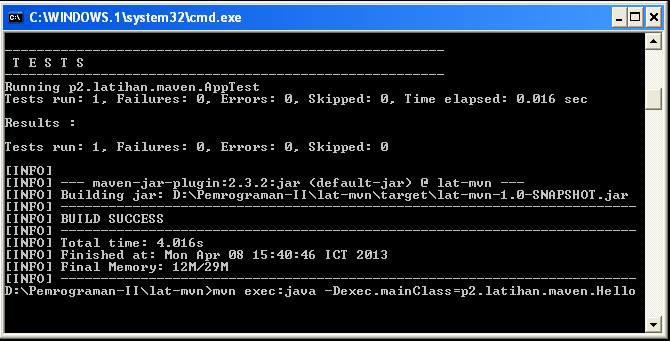
1. **Compile menggunakan maven**

****

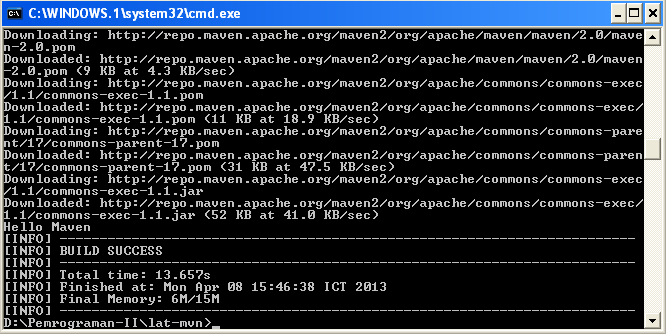
**Hasilnya:**

****

1. **Menjalankan program menggunakan maven**

****

**Hasilnya:**

****