**PRAKTIKUM SOFTWARE QUALITY ASSURANCE**

**LAPORAN PRAKTIKUM MODUL 4**

**Otomatisasi Proses Build dengan Apache Ant**

**(Pengujian dengan Tools)**



**DANIS APRIYANTO**

**NIM : 205611034**

**PROGRAM STUDI SISTEM INFORMASI PROGRAM SARJANA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA YOGYAKARTA**

**2022**

**PERTEMUAN-4**

**Otomatisasi Proses Build dengan Apache Ant**

**(Pengujian dengan Tools)**

**4.1 Tujuan**

1. Mahasiswa memahami pengertian dari proses build.
2. Mahasiswa memahami berbagai macam software yang berfungsi sebagai build tool
3. Mahasiswa memahami kegunaan Apache Ant
4. Mahasiswa memahami dan bisa melakukan instalasi software Apache Ant
5. Mahasiswa mampu membedakan proses membangun software tanpa build tool serta dengan menggunakan build tool
6. Mahasiswa memahami file-file yang terkait dan diperlukan oleh Apache Ant, yaitu file properties dan build.xml
7. Mahasiswa mampu menggunakan Apache ant dengan cara membuat build.xml untuk keperluan otomatisasi proses build serta menggunakan berbagai target yang telah didefinisikan pada build.xml tersebut.

**4.2 Teori Singkat**

Apache Ant adalah software berbasis Java yang digunakan untuk keperluan build tool. Sebagai build tool, tugas utama dari Ant adalah untuk menyediakan sumber daya dan melaksanakan proses yang memungkinkan membangun suatu software dari bentuk source code menjadi aplikasi yang siap didistribusikan atau bahkan telah diinstall pada remote server. Meskipun definisi tersebut kelihatannya sederhana, kita akan melihat bahwa Ant mempunyai banyak sekali tasks yang bisa digunakan untuk berbagai keperluan sesuai dengan tujuan utama dari build tool. Ada banyak build tool yang tersedia, beberapa diantaranya adalah build tool yang bersifat spesifik untuk sistem operasi tertentu. Beberapa contoh dari build tool, diantaranya adalah:

* GNU Make (ingat siklus ./configure, make, make install)
* Jam
* rake (untuk Ruby), dan lain-lain

Ant merupakan software untuk keperluan build tool yang bersifat cross platform, bisa digunakan pada berbagai sistem operasi yang mendukung Java. Meskipun demikian, pembahasan dalam tulisan ini akan menggunakan Linux. Jika anda menggunakan Windows, silahkan sesuaikan dengan sistem operasi anda (hanya jika diperlukan). Selama tidak menggunakan fitur spesifik dari sistem operasi yang bersangkutan (diantaranya dengan tag <exec>), maka pembahasan ini bersifat cross platform.

**Mengapa Harus Menggunakan Build Tool?**

Build tool tidak diperlukan jika software yang akan kita bangun sangat sederhana (seperti misalnya hanya menampilkan satu atau beberapa baris tulisan) dan kita tidak keberatan untuk menuliskan dan mencantumkan berbagai dependencies dari software tersebut (misalnya berbagai file jar yang kita perlukan). Jika aplikasi yang kita bangun relatif kompleks, melibatkan banyak dependencies, dengan struktur direktori tertentu, serta memerlukan banyak eksekusi atau menjalankan tools (misalnya jar, rmic, javac, bzip2, dan lain-lain), maka kita memerlukan build tool untuk meng-”otomatiskan” proses membangun (compile, test, run, deploy) software tersebut. Anda bisa membayangkan sendiri jika harus mengetikkan berpuluh-puluh baris untuk siklus compile, test, run, deploy jika tidak menggunakan build tool.

**4.3 Praktik**

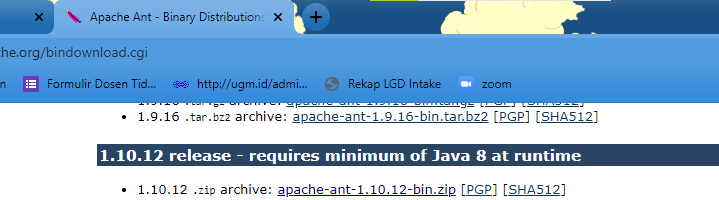
**Instalasi ANT di Windows XP**

**Kebutuhan Paket**

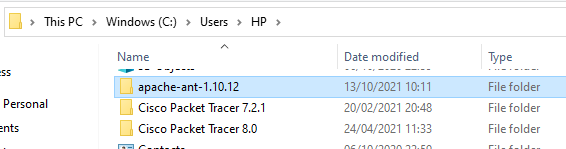
* Sebelumnya Anda sudah harus memiliki instalasi JDK. Untuk referensinya dapat Anda lihat di artikel Install Java di Windows XP Professional

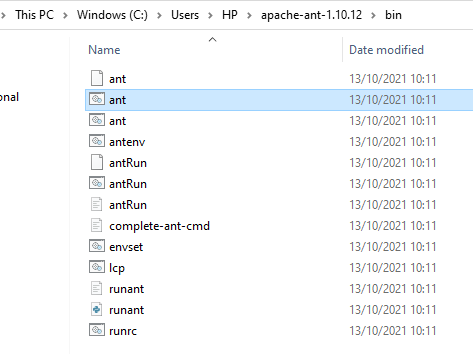
**Instalasi**

* Ambil paket binary dalam bentuk zip file dari situs Apache Ant: <http://ant.apache.org/bindownload.cgi>



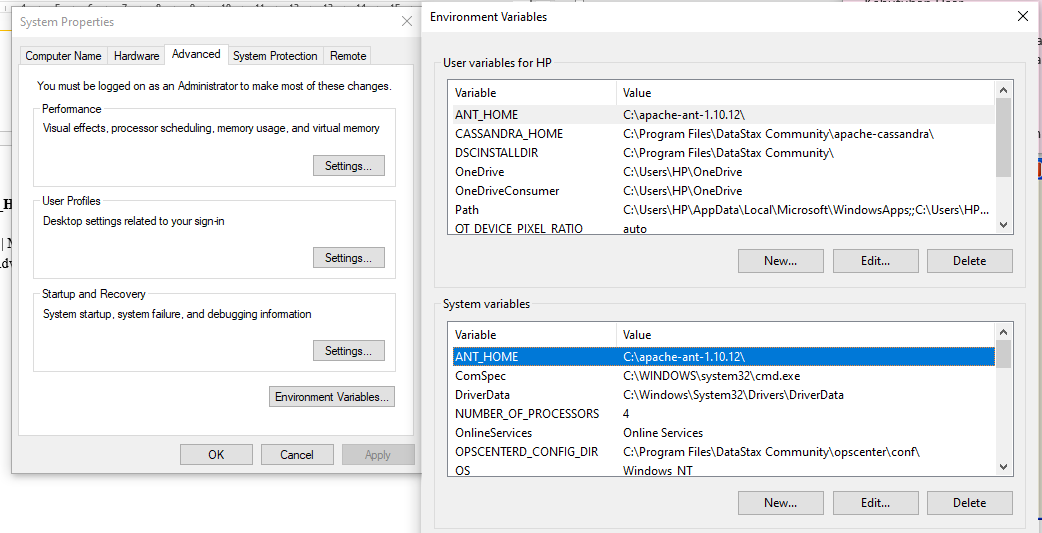
* Ekstrak paket zip yang didownload ke folder C:\
* Struktur folder hasil ekstrak akan tampak seperti gambar di bawah.



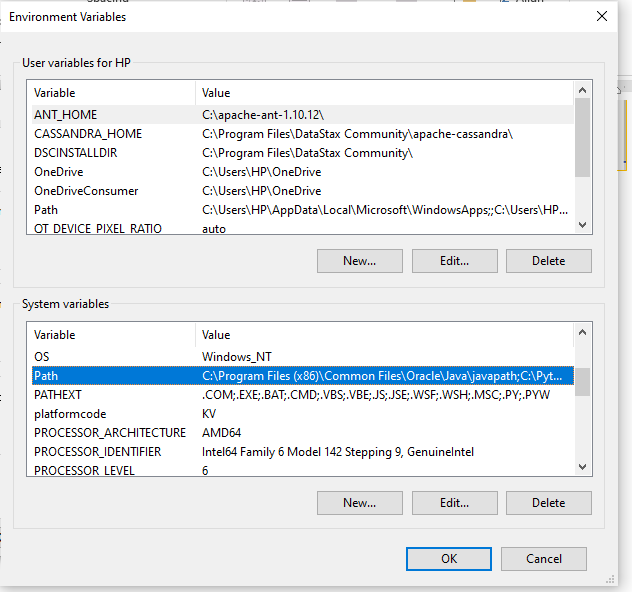


**Konfigurasi ANT\_HOME dan PATH**

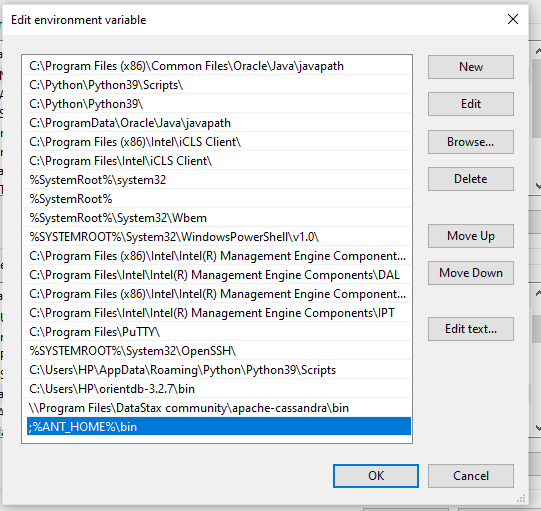
* Masuk ke Start | My Computer | klik kanan lalu pilih Properties
* Masuk ke tab Advanced dan klik tombol Environment Variables dari dialog yang muncul
* Pada dialog berikutnya, di bagian System Variables klik tombol New
* Masukkan folder apache ant ke dalam variable ANT\_HOME :
* Variable Name : ANT\_HOME
* Variable Value : C:\apache-ant-1.10.12\
* Klik tombol OK



* Cari variable PATH di bagian agak bawah dari System Variables dan klik Edit

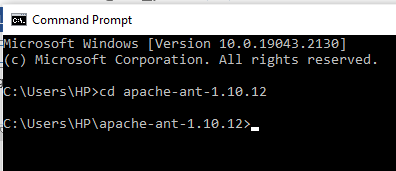


* Tambahkan ;%ANT\_HOME%\bin; di bagian akhir dari Variable Value
* Klik tombol **OK**
* Selesai

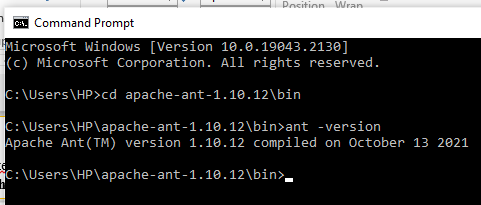


**Testing**

* Masuk ke Start | Run | ketik cmd untuk menjalankan console / command prompt Windows XP (ketik: **cd apache-ant-1.10.12**)

****

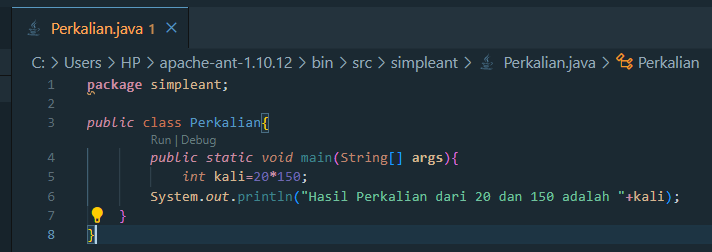
* Ketik ant -version pada command prompt tersebut. Hasil akan tampak seperti pada gambar. Masuk ke folder > **cd apache-ant-1.10.12\bin >** **ant –version**

****

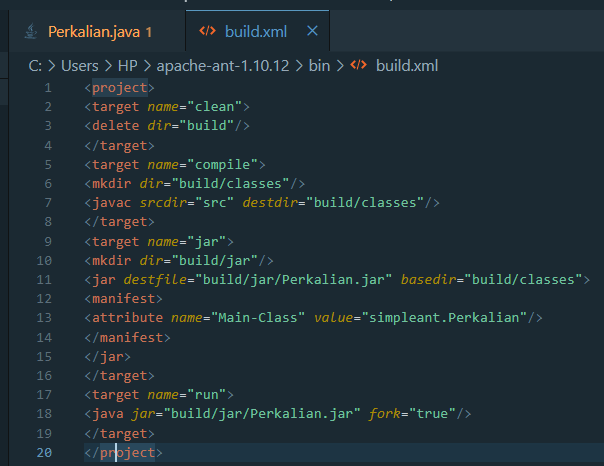
**Latihan**

Gantilah HelloWorld.java dengan program untuk menampilkan hasil perkalian 20\*150 dan nama kelas Perkalian.

1. Mengganti nama program HelloWorld.java menjadi Perkalian.java

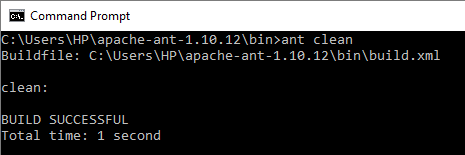


1. Buka file build.xml, lalu ubah kodenya menjadi seperti gambar berikut:



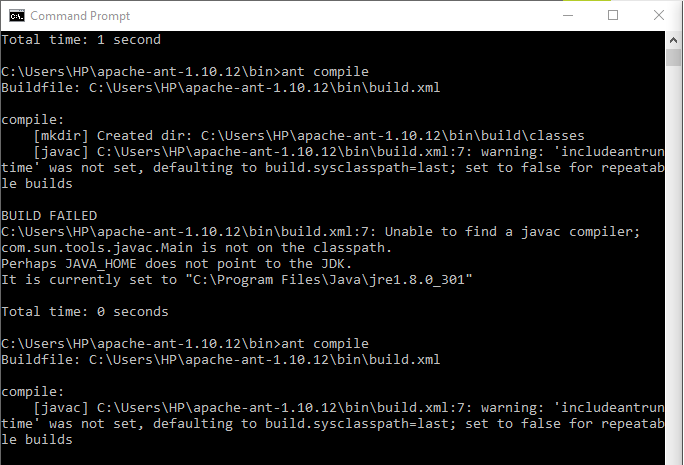
Pembahasan: Agar file build.xml ketika di compile tidak terjadi eror maka kita perlu mengubah kelas HelloWorld menjadi nama Perkalian.

1. Melakukan compile program build, jar dan run,
   1. Instruksi **ant clean**

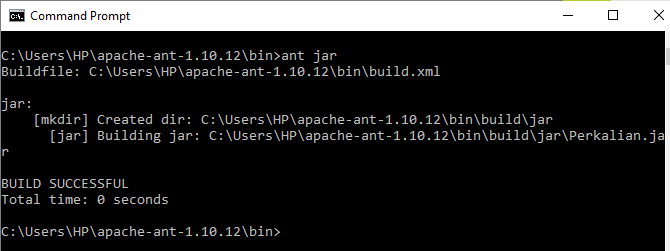
****

Pembahasan : **ant clean** berfungsi mengembalikan project tersebut ke struktur file semua

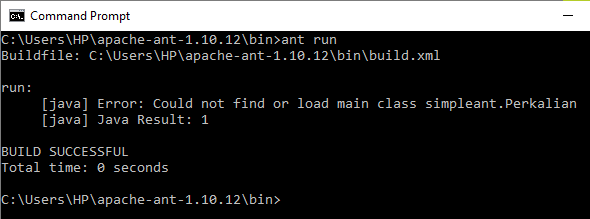
* 1. Perintah **ant compile :** untuk mengompile seluruh source code/Mohon maaf Ibu dosen ant compile saya gagal terus, sudah kroscek berulang kali tidak tahu salahnya dimana.

****

* 1. Perintah **ant jar** : untuk mengabungkan file class ke dalam file jar



* 1. Perintah **ant run** : untuk hasil program, karena ant compile belum berhasil maka hasilnya belum tampil, mohon maaf ibu dosen



**Tugas**

Buatlah proyek baru yang terdiri atas 2 file source code Java. File pertama adalah definisi class, file kedua adalah file utama yang di dalamnya akan membuat instance dari class yang telah didefinisikan pada file pertama. Buatlah file build.xml untuk keperluan seperti pada praktik di atas dan tunjukkan bahwa program yang anda buat tersebut berhasil dikompilasi dan dijalankan.