LAPORAN PRAKTIKUM

**SOFTWARE QUALITY ASSURANCE**

**PERTEMUAN KE 4**



**Disusun Oleh :**

|  |  |
| --- | --- |
| **NAMA** | **: MARFIANA AYU IRAWATI** |
| **NIM** | **: 235611053** |
| **JURUSAN** | **: SISTEM INFORMASI** |
| **JENJANG** | **: S1** |

**Universitas Teknologi Digital Indonesia 2024**

**PERTEMUAN 4**

**OTOMATISASI PROSES BUILD DENGAN APACHE ANT**

**(PENGUJIAN DENGAN *TOOLS*)**

1. **PENDAHULUAN DAN DASAR TEORI**

Dalam pengembangan perangkat lunak, proses build merupakan tahap penting yang memastikan kode sumber dapat dikompilasi, diuji, dan dihasilkan menjadi aplikasi yang dapat dijalankan. Dengan meningkatnya kompleksitas proyek perangkat lunak, manual build process sering kali menjadi tidak efisien dan rentan terhadap kesalahan. Oleh karena itu, otomatisasi proses build menjadi sangat penting. Salah satu alat yang sering digunakan untuk tujuan ini adalah Apache Ant.

Apache Ant adalah alat otomatisasi yang ditulis dalam bahasa pemrograman Java. Alat ini memungkinkan pengembang untuk mendefinisikan proses build dalam bentuk file XML, sehingga memudahkan untuk mengatur, menjalankan, dan memelihara proses build. Selain itu, Ant mendukung berbagai tugas seperti kompilasi, pengujian, dan pengemasan, sehingga memungkinkan integrasi dengan alat pengujian lainnya.

Otomatisasi proses build dengan Apache Ant memberikan banyak keuntungan dalam pengembangan perangkat lunak, mulai dari pengurangan kesalahan manual hingga peningkatan efisiensi. Dengan integrasi alat pengujian, pengembang dapat memastikan bahwa aplikasi yang dibangun memenuhi standar kualitas yang diinginkan. Penggunaan Apache Ant dalam proses build tidak hanya menyederhanakan pengelolaan proyek tetapi juga membantu dalam mengoptimalkan siklus pengembangan perangkat lunak secara keseluruhan.

1. **PEMBAHASAN**

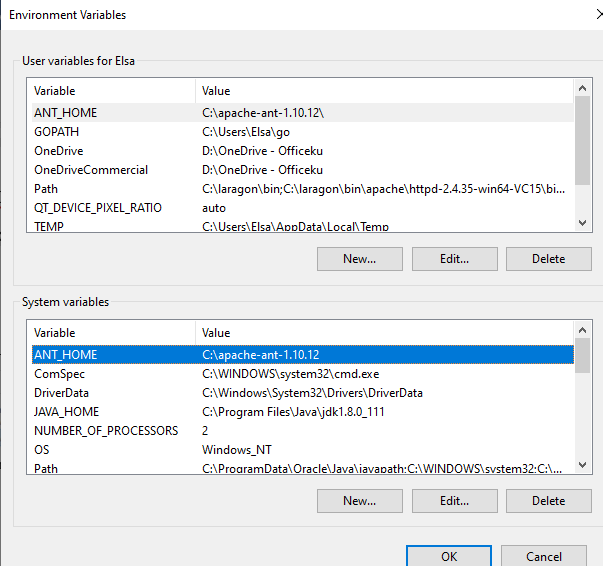
# Praktik :

# Instalasi ANT, yaitu :

# Download file master apache ant, lalu ekstrak file di drive C, seperti gambar berikut.

# Graphical user interface, application, Word, Teams Description automatically generated

* 1. **Konfigurasi ANT\_HOME dan PATH di Environment**
* Buka Advanced system settings, lalu tambahkan system variable ANT\_HOME

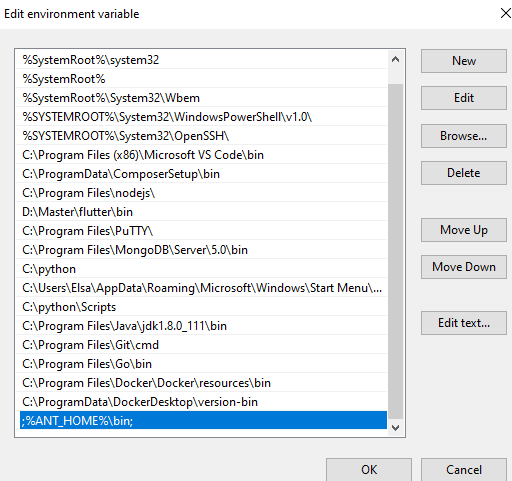


* Double klik Path

Graphical user interface, text, application

Description automatically generated

* Tambahkan “ ;%ANT\_HOME%\bin; “ di path



* Klik ok
* Proses setting konfigurasi selesai
  1. **Testing**
* Buka cmd, arahkan path ke folder **C:\apache-ant-1.10.12\bin**

Text

Description automatically generated

* Cek versi ANT yang telah terinstal, dengan perintah : **ant -version**



* Tandanya sudah sukses dan berhasil terinstall ANT

# Soal Latihan :

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

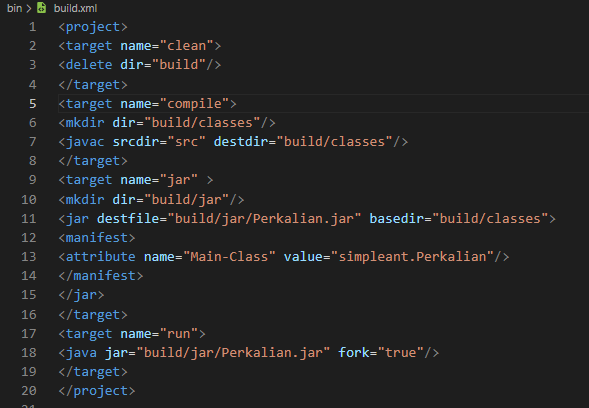
# Mengganti Nama program HelloWorld.Java menjadi Perkalian.java

# Ubah script menjadi seperti ini di file Perkalian.java

**Text

Description automatically generated**

1. Buka file build.xml, lalu ubah kodenya menjadi seperti gambar di bawah ini

****

**Pembahasan :**

Agar file build.xml ketika di compile tidak terjadi eror maka kita perlu mengubah nama kelas **HelloWord** menjadi nama kelas dengan kelas **Perkalian.**

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

Text

Description automatically generated

**Pembahasan :**

ant clean berguna mengembalikan project tersebut ke strukur file semula

1. Perintah **ant compile**

Text

Description automatically generated

**Pembahasan :**

perintah "compile" digunakan untuk mengompile seluruh source code

1. Perintah **ant jar**

Text

Description automatically generated

**Pembasan :**

ant jar di gunakan untuk menyatukan file class ke dalam file jar

1. Perintah **ant run (hasil program)**

Text

Description automatically generated

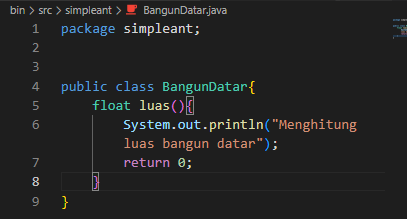
**Pembahasan :**

perintah "run" digunakan untuk menjalankan file .jar yang telah dibuat sebelumnya.

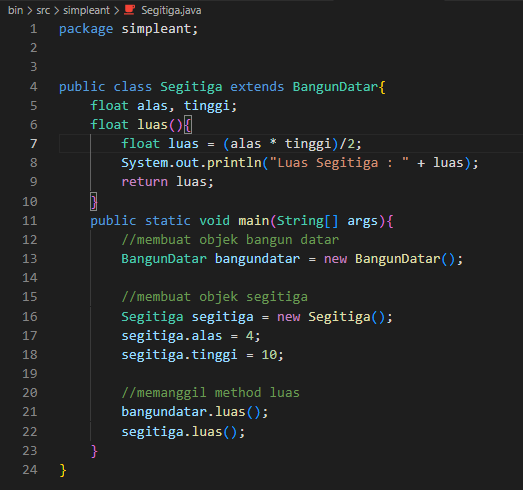
# Soal Tugas :

# 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.

1. Membuat kelas induk BangunDatar.java, untuk menghitung luas bangun datar



1. Membuat program turunan dari BangunDatar.java, yaitu program menghitung luas segitiga

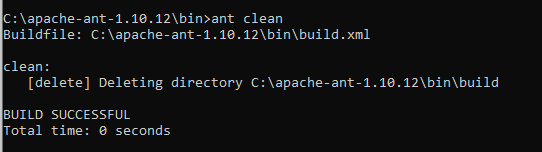


1. Ubah file di build.html, menjadi segitiga.java, seperti berikut ini

Text

Description automatically generated

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



**Pembahasan :**

ant clean berguna mengembalikan project tersebut ke strukur file semula

1. Perintah **ant compile**

Text

Description automatically generated

**Pembahasan :**

perintah "compile" digunakan untuk mengompile seluruh source code

1. Perintah **ant jar**

Text

Description automatically generated

**Pembasan :**

ant jar di gunakan untuk menyatukan file class ke dalam file jar

1. Perintah **ant run (hasil program)**

Text

Description automatically generated

**Pembahasan :**

perintah "run" digunakan untuk menjalankan file .jar yang telah dibuat sebelumnya.

Path

1. 1-2-3-4-5-6(T)-7-16

2. 1-2-3-4-5-8(F)-9(T)-10-16

3. 1-2-3-4-5-8(F)-11(F)-12(T)-13-16

4. 1-2-3-4-5-8(F)-11(F)-14(F)-15-16

Hasil uji coba

1. Suhu celsius ke Reamur (R)

A white background with a black border

Description automatically generated

1. Suhu celsius ke Fahrenhit (F)

A picture containing application

Description automatically generated

1. Suhu Celsius ke Kelvin (K)

A picture containing graphical user interface

Description automatically generated

1. Suhu Celsius ke Kelvin (K)

Graphical user interface

Description automatically generated with low confidence

1. Suhu celsius ke Fahrenhit (F)

Graphical user interface

Description automatically generated with medium confidence

1. **KESIMPULAN**

Otomatisasi proses build dengan Apache Ant merupakan langkah penting dalam pengembangan perangkat lunak modern. Dengan memanfaatkan Ant, pengembang dapat mendefinisikan dan mengelola proses build secara efisien melalui file XML, yang mencakup kompilasi, pengujian, dan pengemasan aplikasi. Keuntungan utama dari otomatisasi ini meliputi:

* 1. Efisiensi dan Konsistensi : Proses build yang otomatis mengurangi waktu yang dibutuhkan dan meningkatkan konsistensi, mengurangi kemungkinan kesalahan manusia.
  2. Integrasi Pengujian : Ant memungkinkan integrasi langsung alat pengujian seperti JUnit, sehingga pengujian dapat dilakukan secara otomatis selama fase build. Ini membantu dalam deteksi dini masalah dan menjaga kualitas kode.
  3. Portabilitas dan Fleksibilitas : Ant dapat dijalankan di berbagai platform dan dapat disesuaikan untuk memenuhi kebutuhan spesifik proyek, menjadikannya alat yang fleksibel bagi tim pengembangan.
  4. Peningkatan Kolaborasi : Dengan otomatisasi, seluruh tim dapat berkolaborasi lebih baik, karena proses build yang terstandarisasi meminimalkan kebingungan dan kesalahan.

Secara keseluruhan, penggunaan Apache Ant dalam otomatisasi proses build tidak hanya meningkatkan efisiensi pengembangan tetapi juga mendukung praktik pengujian yang lebih baik, sehingga menghasilkan perangkat lunak yang lebih berkualitas dan dapat diandalkan.