

LAPORAN PRAKTIKUM
MATA KULIAH DASAR PEMROGRAMAN

Dosen Pengampu : Triana Fatmawati, S.T, M.T
PERTEMUAN 1 DASAR PEMROGRAMAN



Nama : Yonanda Mayla Rusdiaty

NIM : 2341760184

Prodi : D-IV Sistem Informasi Bisnis

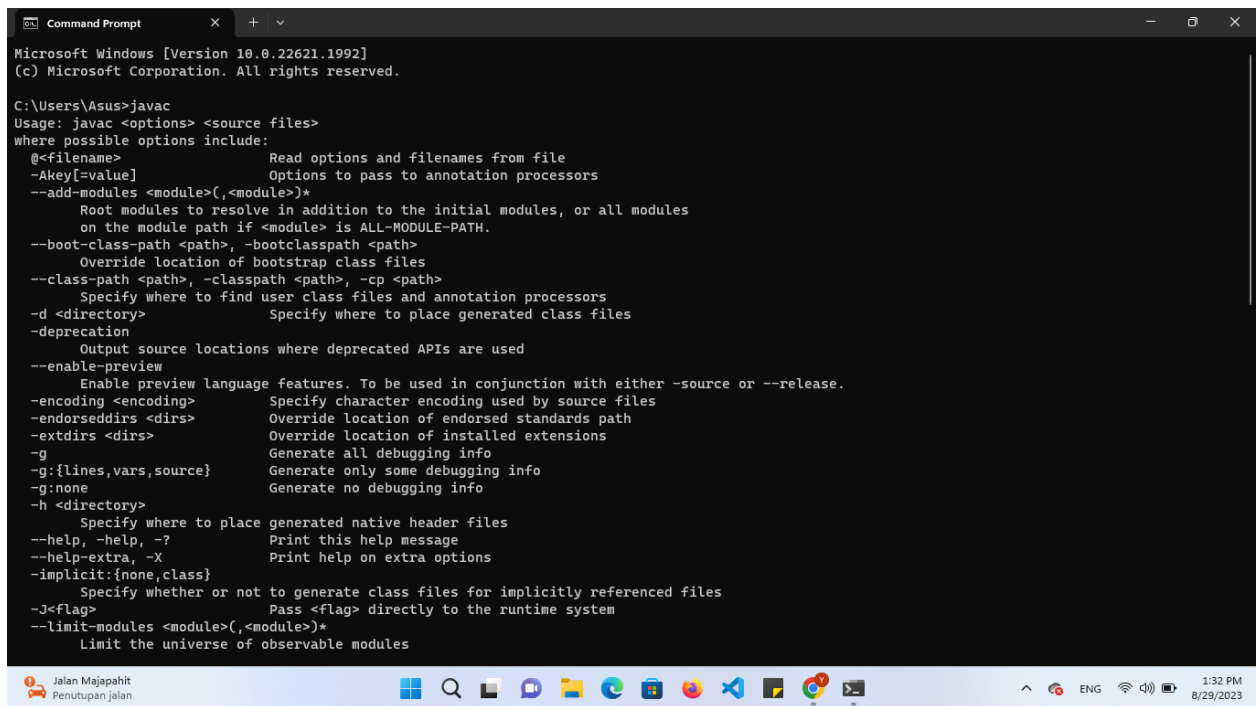
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2023

2.1 Percobaan 1 : Instalasi *Java Development Kit/ JDK*

Berikut adalah hasil dari praktikum percobaan 1 : Instalasi *Java Development Kit/ JDK*.

Setelah berhasil *mendownload* dan *menginstall* JDK, langkah selanjutnya yaitu membuat *setting PATH* agar perintah java dapat dikenali. Jika sudah, maka ketikkan perintah *javac* pada *Command Prompt* (CMD) dan kemudian akan muncul gambar seperti pada Gambar 1, yang berarti bahwa perintah tersebut dapat dikenali dan lingkungan operasi *Windows* telah mendukung program *java*.



```
Microsoft Windows [Version 10.0.22621.1992]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Asus>javac
Usage: javac <options> <source files>
where possible options include:
  @<filename>                Read options and filenames from file
  -Akey[=value]              Options to pass to annotation processors
  --add-modules <module>(<module>)*
                             Root modules to resolve in addition to the initial modules, or all modules
                             on the module path if <module> is ALL-MODULE-PATH.
  --boot-class-path <path>, -bootclasspath <path>
                             Override location of bootstrap class files
  --class-path <path>, -classpath <path>, -cp <path>
                             Specify where to find user class files and annotation processors
  -d <directory>             Specify where to place generated class files
  -deprecation                Output source locations where deprecated APIs are used
  --enable-preview            Enable preview language features. To be used in conjunction with either -source or --release.
  -encoding <encoding>       Specify character encoding used by source files
  -endorseddirs <dirs>       Override location of endorsed standards path
  -extdirs <dirs>            Override location of installed extensions
  -g                          Generate all debugging info
  -g:{lines,vars,source}     Generate only some debugging info
  -g:none                    Generate no debugging info
  -h <directory>             Specify where to place generated native header files
  --help, -help, -?          Print this help message
  --help-extra, -X           Print help on extra options
  --implicit:{none,class}    Specify whether or not to generate class files for implicitly referenced files
  -J<flag>                   Pass <flag> directly to the runtime system
  --limit-modules <module>(<module>)*
                             Limit the universe of observable modules
```

Gambar 1

Pertanyaan

1. Jelaskan apa kegunaan memasukkan lokasi folder bin dari Java ke dalam variabel PATH!
2. Jelaskan kegunaan perintah javac ketika masuk di command prompt!

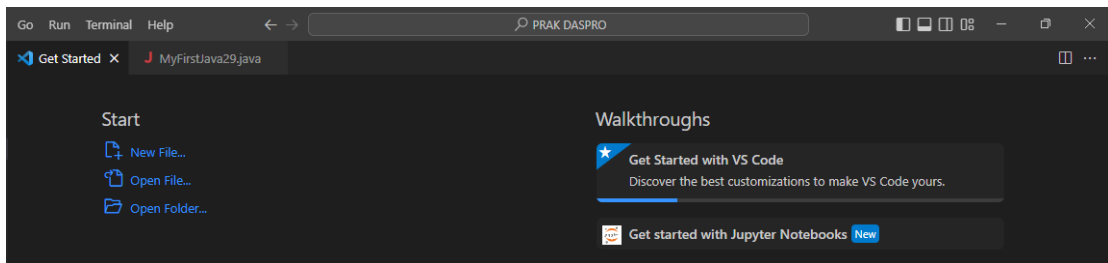
Jawaban :

1. Jadi kegunaan memasukkan lokasi folder bin dari *Java* ke dalam variable *PATH* adalah untuk mengenali perintah. Sehingga dapat memudahkan akses ke berkas-berkas eksekusi dari berbagai direktori di dalam sistem. Hal ini berguna ketika kita ingin menjalankan program *Java* dari mana saja tanpa harus selalu berpindah ke direktori tempat berkas eksekusinya disimpan. Selain itu kegunaanya juga untuk membantu proses kompilasi dan pemanggilan ketika kita memiliki kelas Java yang saling memanggil satu sama lain dan berada dalam folder yang berbeda.
2. Kegunaan perintah ***javac*** ketika masuk di *Command Prompt* (CMD) yaitu sebagai penanda bahwa operasi *Windows* telah mendukung program *java*, jika hasil dari perintah ***javac*** tersebut sudah berhasil dikenali.

2.2 Percobaan 2 : Struktur Dasar *Java*

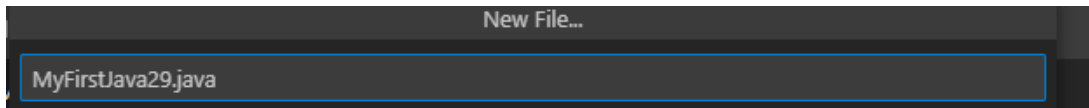
Berikut adalah hasil dari praktikum percobaan 2 : Struktur Dasar *Java*

- 1) Setelah *download* dan *install visual studio code*, langkah selanjutnya adalah membuka *visual studio code* tersebut seperti pada tampilan Gambar 2.



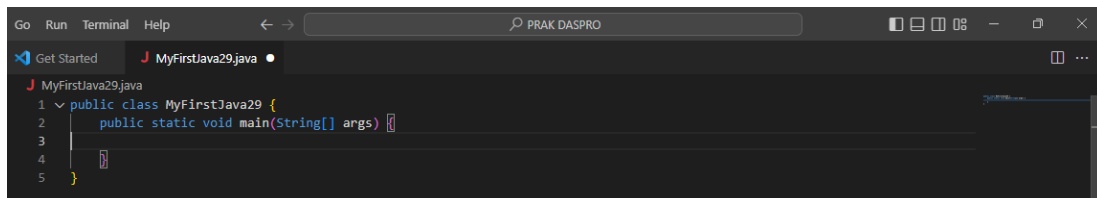
Gambar 2

Kemudian membuat file baru dan disimpan dengan nama ***MyFirstJava29.java*** (sesuai nomor absen).



Gambar 3

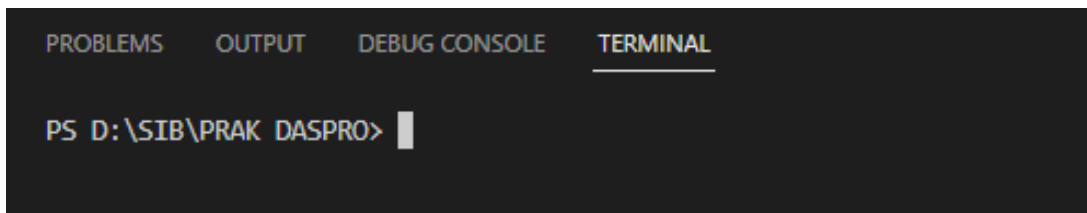
- 2) Menuliskan kode program seperti yang tertera pada Gambar 4. Jika sudah simpan kode program tersebut.



Gambar 4

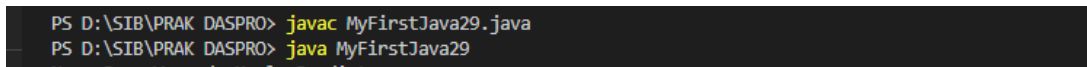
Keterangannya yaitu :

- **Public** adalah sebuah *keyword* di *Java* yang menandakan bahwa objek, *method*, atau atribut dapat diakses dari class lain.
 - **Class** adalah sebuah *keyword* di *Java* yang digunakan untuk membuat sebuah *class*.
 - **Static** adalah sebuah *keyword* untuk membuat sebuah *method* dengan tidak perlu menginstansiasi terlebih dahulu.
 - **Void** adalah sebuah *keyword* untuk membuat sebuah *method* tidak me-*return* nilai apapun atau kosong
 - **Nb : Nama class dan nama file harus sama**
- 3) Buka terminal pada *visual studio code*. Pada terminal ini perlu adanya pengarahannya ke direktori tempat kita menyimpan file. Hasilnya akan seperti pada Gambar 5.



Gambar 5

- 4) Langkah selanjutnya yakni melakukan *compile* dengan mengetikkan **javac** *MyFirstJava29.java* seperti pada Gambar 6.



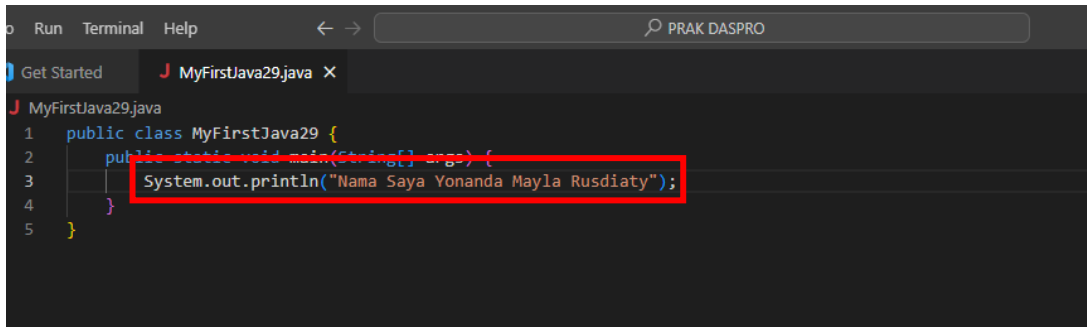
Gambar 6

- 5) Jika proses kompilasi berhasil dan tidak terdapat *error*, maka akan terdapat file baru dengan nama **MyFirstJava29.class** seperti pada Gambar 7.

 MyFirstJava29.class	8/29/2023 2:20 PM	CLASS File
 MyFirstJava29	8/29/2023 2:20 PM	Java Source File

Gambar 7

- 6) Pada percobaan ini belum ada output yang ditampilkan di layar terminal karena kode program yang dituliskan hanya struktur dasar program *java*. Oleh karena itu perlu adanya modifikasi pada file ***MyFirstJava29.java*** dengan menambahkan kode seperti pada Gambar 8. Kemudian klik simpan dan lakukan kompilasi kembali.



```
1 public class MyFirstJava29 {
2     public static void main(String[] args) {
3         System.out.println("Nama Saya Yonanda Mayla Rusdiaty");
4     }
5 }
```

Gambar 8

- 7) Output dari kompilasi Gambar 8 adalah seperti yang ada pada Gambar 9.

```
PS D:\SIB\PRAK DASPRO> javac MyFirstJava29.java
PS D:\SIB\PRAK DASPRO> java MyFirstJava29
Nama Saya Yonanda Mayla Rusdiaty
PS D:\SIB\PRAK DASPRO> |
```

Gambar 9

Pertanyaan!

1. Jelaskan fungsi perintah `javac MyFirstJava00.java` pada percobaan diatas!
2. Jelaskan fungsi perintah `java MyFirstJava00` pada percobaan diatas!

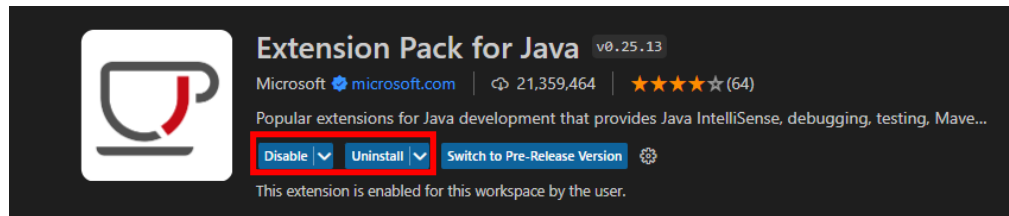
Jawaban :

1. Fungsi dari perintah ***javac MyFirstJava29.java*** pada percobaan tersebut adalah untuk melakukan perintah dalam proses kompilasi. Perintah ini dijalankan ketika kita sudah membuka terminal dan mengarahkan ke direktori tempat penyimpanan file dengan menggunakan perintah **cd**.
2. Fungsi perintah ***java MyFirstJava29*** pada percobaan tersebut adalah untuk menjalankan program yang telah dikompilasi sebelumnya, tepatnya saat proses kompilasi berhasil tidak ada error dan akan muncul file baru dengan nama ***MyFirstJava29.class***.

2.3 Percobaan 3 : Menambahkan *Extension* pada *Visual Studio Code*

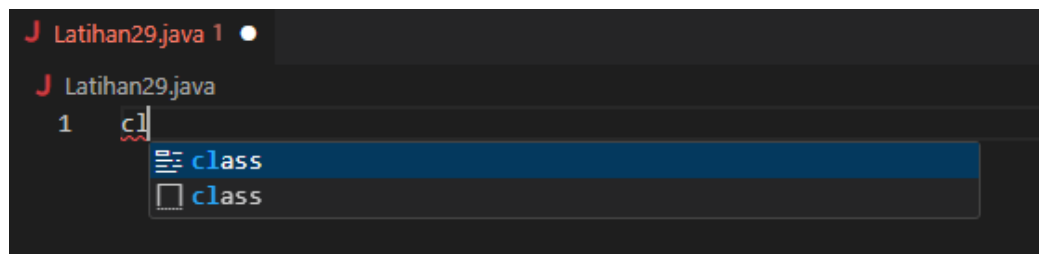
Selanjutnya yaitu menambahkan *Extension Pack Java* pada aplikasi *visual studio code*.

1. Langkah pertama, buka menu *extension* lalu cari *extension pack for java* dan klik *install*.



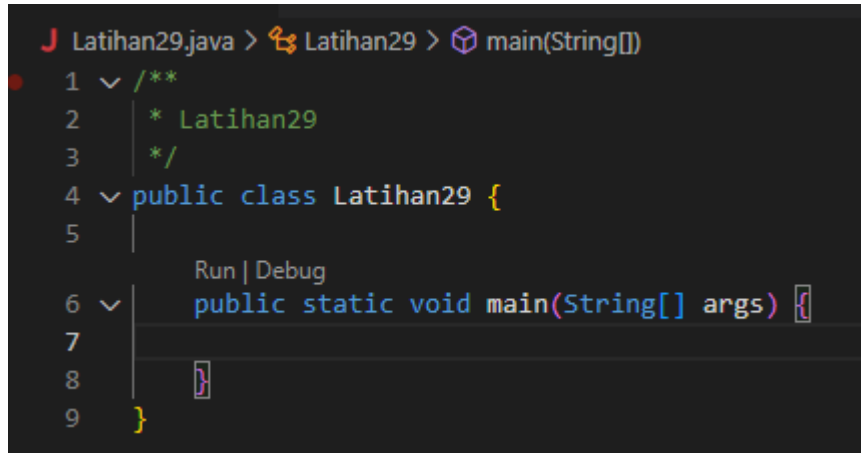
Gambar 10

2. Setelah berhasil terinstall, kita harus mengaktifkan dulu *extension* tersebut dengan mengklik *disable*. Jika sudah, langkah selanjutnya yaitu kita perlu membuat file baru yang diberi nama **Latihan29.java** (sesuai nomor absen).
3. Kegunaan dari adanya *extension* tersebut yaitu kita tidak perlu lagi menuliskan setiap program satu persatu dan hanya cukup menuliskan beberapa huruf yang relevan dengan struktur dasar java, maka akan muncul *autotext* atau *rekoemndasi* yang sesuai dan klik “tab”.



Gambar 11

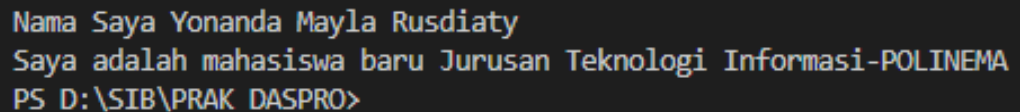
4. Selanjutnya, tuliskan kode program sesuai dengan Gambar 12 dan klik tombol *run* yang berada di pojok kanan atas untuk menjalankan program.



```
J Latihan29.java > Latihan29 > main(String[])
1  /**
2   * Latihan29
3   */
4  public class Latihan29 {
5
6      Run | Debug
       public static void main(String[] args) {
7
8
9  }
```

Gambar 12

5. Output yang dihasilkan dari kode program pada Gambar 12 adalah seperti pada Gambar 13 di bawah ini.



```
Nama Saya Yonanda Mayla Rusdiaty
Saya adalah mahasiswa baru Jurusan Teknologi Informasi-POLINEMA
PS D:\SIB\PRAK DASPRO>
```

Gambar 13

Jawaban dari Pertanyaan pada Percobaan 3 :

1. Mengganti perintah *System.out.println* menjadi *System.out.print* pada baris ke 7 dan 8

```
public static void main(String[] args) {  
    System.out.print(s:"Nama Saya Yonanda Mayla Rusdiaty");  
    System.out.print(s:"Saya adalah mahasiswa baru Jurusan Teknologi Informasi-POLINEMA");  
}
```

2. Perbedaan dari kedua kode program tersebut yang diletakkan pada baris ke 7 dan 8 yaitu, jika kita menggunakan *System.out.println* (“...”) maka output yang dihasilkan akan memiliki jarak antar baris. Namun jika kita menggunakan kode program *System.out.print* (“...”), maka *output* yang dihasilkan tidak memiliki jarak antar baris.
3. Berikut adalah hasil dari perintah *System.out.print* (“\nSaya adalah Mahasiswa Jurusan Teknologi Informasi – POLINEMA”); pada baris ke 8.

```
System.out.println(x:"\nSaya adalah Mahasiswa Jurusan Teknologi Informasi-POLINEMA");|
```

```
Nama Saya Yonanda Mayla Rusdiaty  
Saya adalah Mahasiswa Jurusan Teknologi Informasi-POLINEMA
```

4. Fungsi dari \n pada baris ke 8 adalah untuk memisahkan baris ke 8 dengan baris sebelumnya yaitu baris ke 7. Dalam artian fungsinya seperti tombol *enter* pada *keyboard*.
5. Hasil dari menambahkan karakter \\ pada awal baris ke 8 yaitu barisan tersebut tidak akan dieksekusi dan tampilannya hanya ada pada baris ke 7 saja seperti gambar di bawah ini.

```
//System.out.println("\nSaya adalah Mahasiswa Jurusan Teknologi Informasi-POLINEMA");
```

```
Nama Saya Yonanda Mayla Rusdiaty
```

6. Fungsi dari karakter `\\` pada awal barisan kode program adalah menjadikan kode program pada barisan yang diberi karakter `\\` menjadikan barisan tersebut tidak tereksekusi atau tidak ditampilkan. Penambahan karakter `\\` ini bisa disebut dengan komentar.