

1. Tujuan

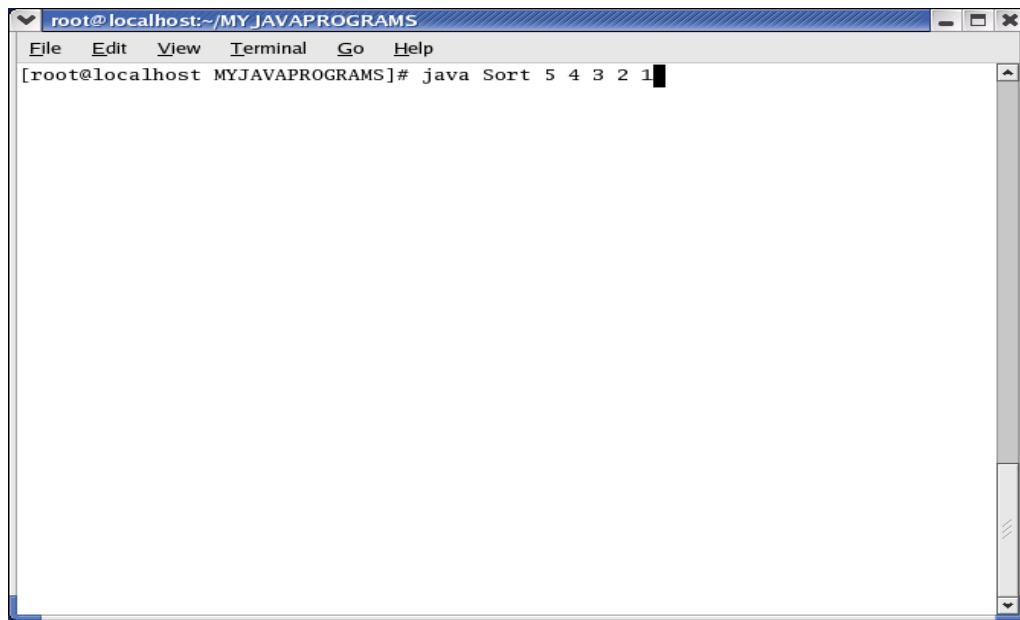
- Mengetahui dan menjelaskan mengenai argumen command-line
- Mendapatkan input dari user dengan menggunakan argumen command-line
- Mempelajari bagaimana memberikan argumen kedalam program pada NetBeans

2. Latar Belakang

Pada bagian ini, kita akan mempelajari bagaimana untuk memproses sebuah input dari command-line dengan menggunakan argumen yang diberikan kepada program Java.

Sebuah aplikasi Java dapat menerima berbagai argumen dari command-line. Argumen dari command line memberikan kesempatan bagi user untuk mempengaruhi operasi dari sebuah aplikasi dalam sekali invokasi. User memasukkan argumen command line pada saat meng-invoke aplikasi dan memberikan spesifikasi kepada aplikasi tersebut setelah nama kelas untuk dijalankan.

Sebagai contoh, apabila Anda memiliki sebuah aplikasi Java dengan nama Sort, yang akan mengurutkan lima nomor, Anda dapat menjalankannya seperti berikut ini:



```
root@localhost:~/MY_JAVAPROGRAMS
File Edit View Terminal Go Help
[root@localhost MYJAVAPROGRAMS]# java Sort 5 4 3 2 1
```

Gambar 1:Menjalankan argument dari Command-Line



Perhatikan bahwa sebuah argumen akan dipisahkan oleh spasi. Di bahasa Java, apabila Anda akan memanggil sebuah aplikasi, sebuah runtime system akan memberikan argumen command-line ke main method lewat array dari String, dimana tiap String dalam array terdiri dari sebuah argumen command-line. Perhatikan deklarasi dari main method,

```
public static void main( String[] args )
```

Argumen yang diberikan kepada program Anda akan disimpan kedalam sebuah array String dengan *identifier* args.

Pada contoh sebelumnya, argumen dari command-line yang akan diberikan kepada aplikasi sort command adalah lima buah String yaitu "5","4","3","2", dan "1". Anda dapat mengetahui berapa banyak argumen dari command-line dengan cara melihat panjang dari attribute array.

Sebagai contoh,

```
int numberOfArgs = args.length;
```

Jika program Anda membutuhkan support sebuah argumen command-line yang numeric. Anda harus mengkonversi String argumen tersebut untuk merepresantasikan sebuah nomor, misalnya "34" menjadi sebuah nomor. Kode dibawah ini adalah sebuah potongan untuk mengkonversi sebuah argumen command-line menjadi integer.

```
int firstArg = 0;
if (args.length > 0){
    firstArg = Integer.parseInt(args[0]);
}
```

parseInt akan mendapatkan NumberFormatException (ERROR) jika format args[0] tidak valid (bukan sebuah nomor).

3. Percobaan

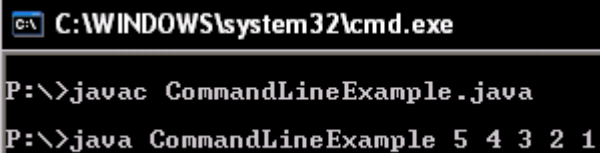
Percobaan 1 Contoh program pada Command Line:

```
public class CommandLineExample
{
    public static void main( String[] args ){

        System.out.println("Number of arguments=" + args.length);

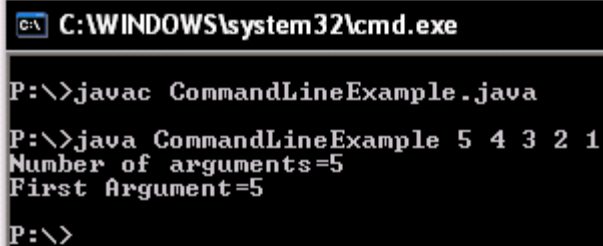
        System.out.println("First Argument="+ args[0]);

    }
}
```



```
C:\> C:\WINDOWS\system32\cmd.exe
P:\> javac CommandLineExample.java
P:\> java CommandLineExample 5 4 3 2 1
```

Hasil Percobaan 1 :



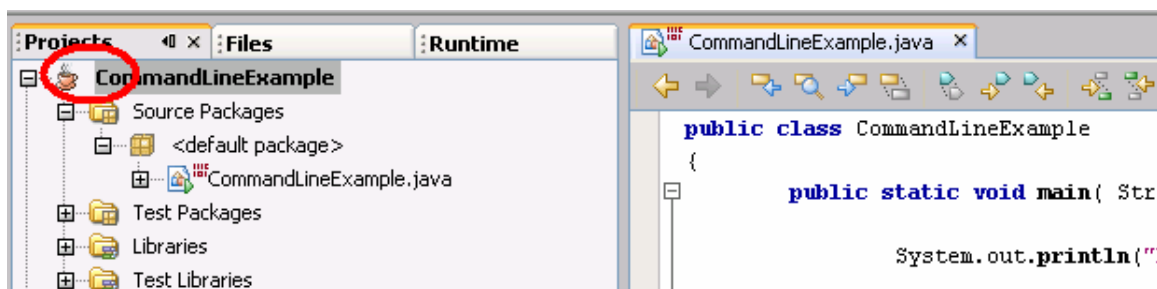
```
C:\> C:\WINDOWS\system32\cmd.exe
P:\> javac CommandLineExample.java
P:\> java CommandLineExample 5 4 3 2 1
Number of arguments=5
First Argument=5
P:\>
```

Percobaan 2 Contoh Program pada Netbeans:

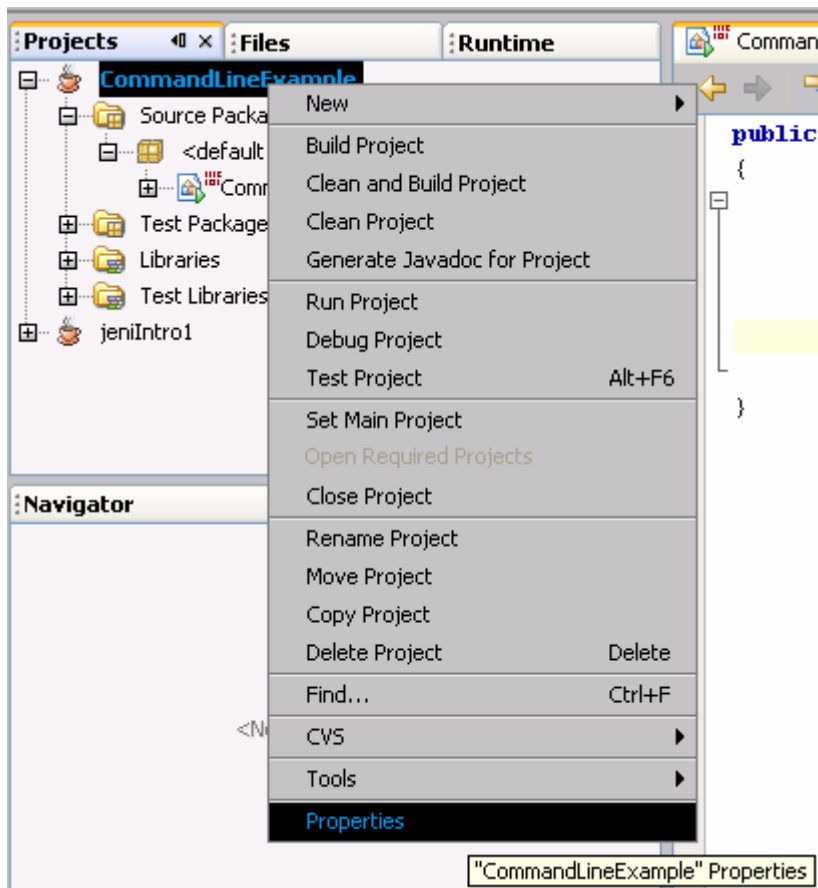
```
public class CommandLineExample
{
    public static void main( String[] args ){
        System.out.println("Number of arguments=" + args.length);
        System.out.println("First Argument="+ args[0]);
    }
}
```

Langkah-langkah

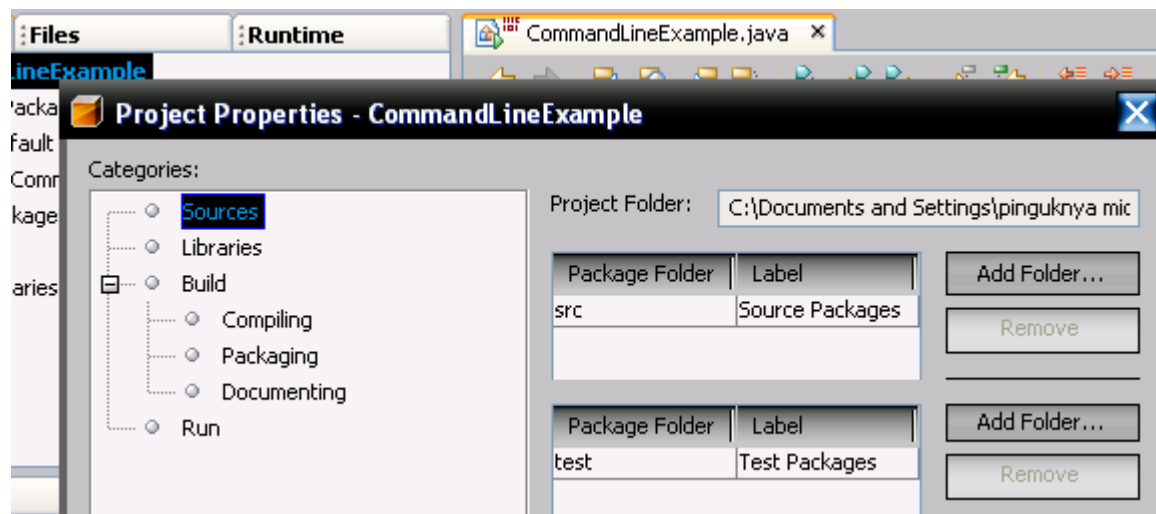
Klik pada daerah yang dilingkari



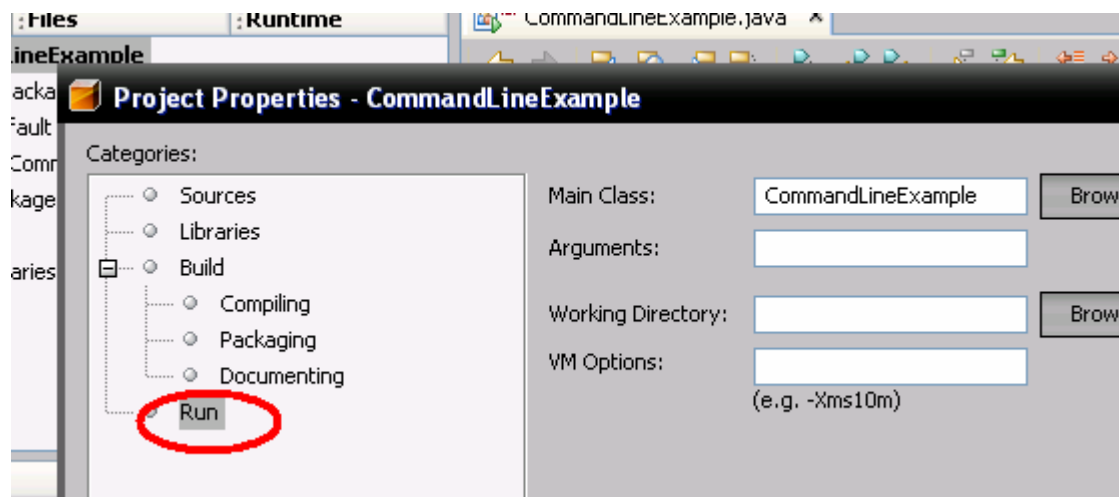
Click kanan pada CommandLineExample, kemudian pilih Properties :



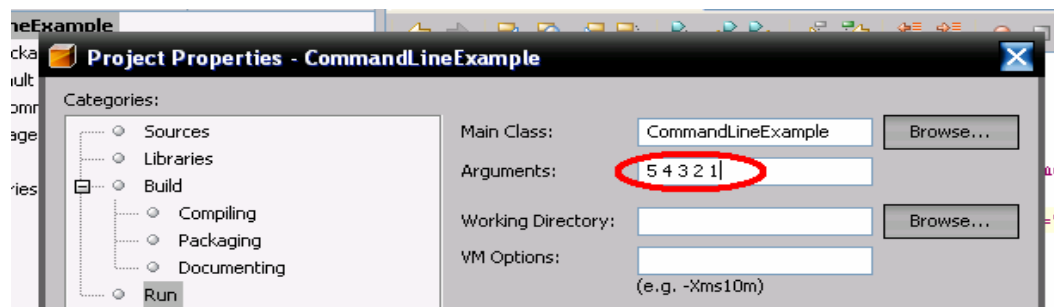
Dialog Project Properties akan muncul



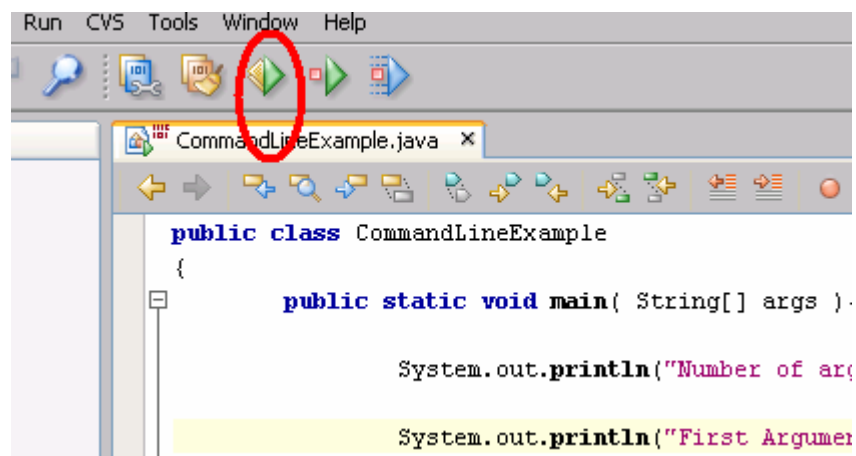
click Run



Dalam textbox argumen, ketik argumen 5 4 3 2 1. Kemudian tekan click OK.



Run program



Hasil Percobaan 2 Output :

```
Output - CommandLineExample (run)
init:
deps-jar:
Created dir: C:\Documents and Settings\pingu
Compiling 1 source file to C:\Documents and
compile:
run:
Number of arguments=5
First Argument=5
BUILD SUCCESSFUL (total time: 1 second)
```