

PERTEMUAN IV

INPUT PADA PEMROGRAMAN JAVA

4.1 Mengambil Input Pada java

Dalam sebuah peprogram komputer terdiri dari tiga komponen utama, yaitu: input, proses, dan output. Input adalah suatu nilai dimana dapat kita berikan nilai dan masukan ke program, sedangkan proses suatu langkah langkah yang di proses dalam suatu program untuk mendapatkan hasil yang diinginkan oleh programmer, sedangkan output merupakan hasil yang didapat dari peolahan input dan proses dalam program komputer.

Semua bahasa pemrograman telah menyediakan fungsi-fungsi untuk melakukan input dan output. Java sendiri sudah menyediakan tiga class untuk mengambil input yaitu yang pertama adalah Class Scanner, kedua adalah Class BufferedReader dan Class Console. Ketiga kelas ini digunakan untuk input dalam program berbasis teks (konsol). Pada saat yang bersamaan, GUI menggunakan kelas-kelas lain yang berbentuk seperti JOptionPane dan inputbox yang akan dipelajari pada pertemuan semester berikutnya. Pada saat yang sama, untuk keluaran, Java menyediakan fungsi print (), println () dan format ().

1 Mengambil Input dengan Class Scanner

Scanner merupakan class yang menyediakan fungsi-fungsi untuk mengambil input dari keyboard. Untuk class Scanner perlu mengimpornya/menambahkan ke dalam kode dibawah ini, padabagian atas/ kepala program, jika menggunakan package di netbean maka tambahkan diatasnya, jika memakai notepad/notepad++ maka tidak perlu, akan diberikan contoh pada latihan 1, anda bisa mempraktekan dibawah ini.

```
import java.util.Scanner;  
package pertemuan2;
```

B. Output Pada java

Program berbasis teks dapat diambil dengan input pada keyboard. Fungsi yang sudah disediakan java diantaranya adalah fungsi system.out.print(), fungsi system.out.println() dan fungsiyang ketiga adalah system.out.format(). Adapun fungsi tersebut dibedakan sebagai berikut:

- a. Fungsi print() vs println()

Kedua fungsi `print ()` dan `println ()` dimanfaatkan sebagai fungsi yang dapat menampilkan teks. Perbedaan antara kedua fungsi ini adalah jika fungsi `print ()` dapat menampilkan teks sesuai adanya. Pada saat yang sama, `println ()` dapat menampilkan teks dalam baris baru saat ditambahkan.

b. Menggabungkan String

fungsi `print()` maupun `println()`, pada saat digunakan adakalanya perlu mengambil teks dari variabel untuk menggabungkannya bersama teks lain.

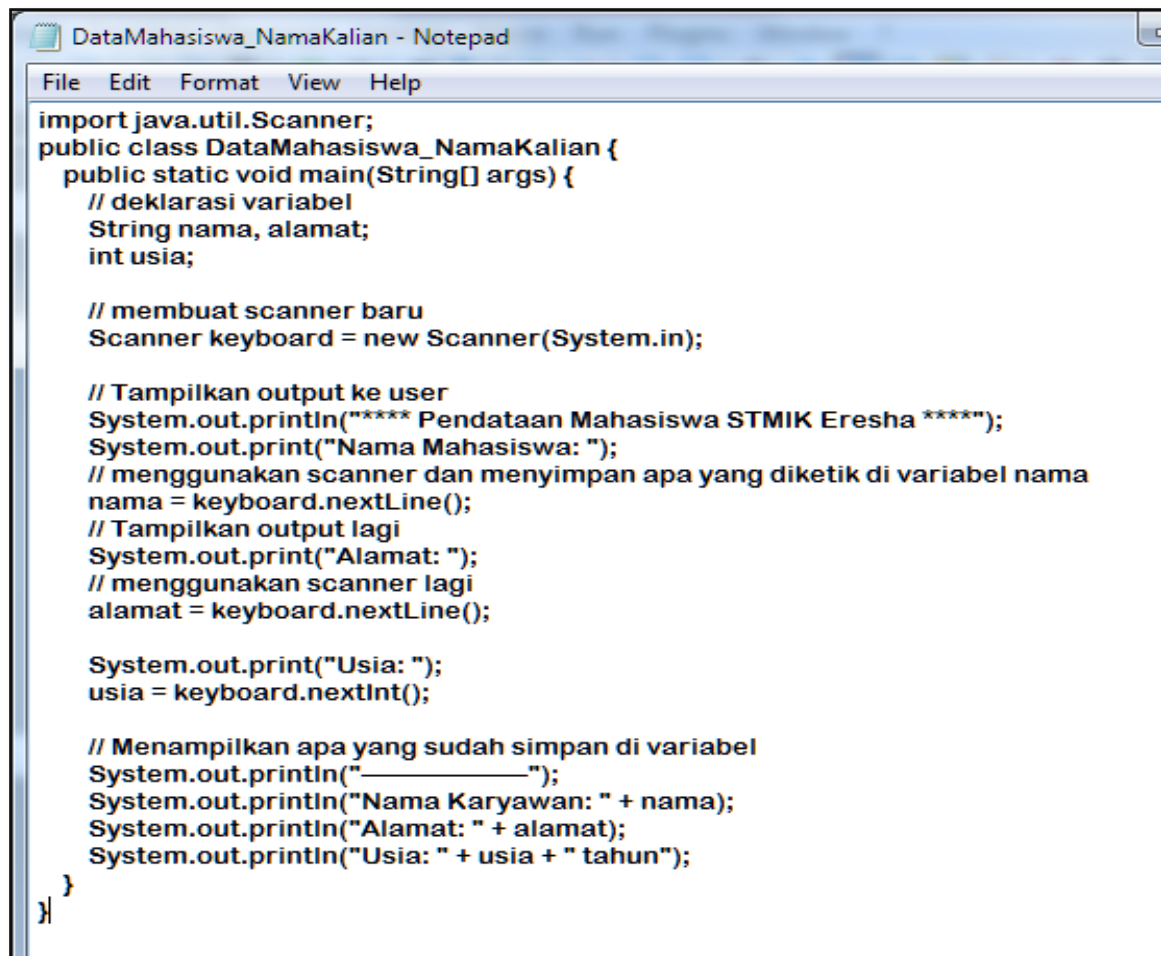
c. Format String

Fungsi `format()` digunakan untuk menggabungkan String yang lebih kompleks . Untuk memperdalam pemahaman ini silahkan mempraktikkan langsung latihan 5 agar lebih memahami format penggunaan String, seperti dibawah ini

F. Prosedur Dan Mekanisme Praktik

Latihan 1

Silahkan ketik pada notepad/notepad++/neatbean, simpan source code dengan `DataMahasiswa_NamaKalian.java` pada drive D/E/F(sesuai perangkat komputer kalian),



```
import java.util.Scanner;
public class DataMahasiswa_NamaKalian {
    public static void main(String[] args) {
        // deklarasi variabel
        String nama, alamat;
        int usia;

        // membuat scanner baru
        Scanner keyboard = new Scanner(System.in);

        // Tampilkan output ke user
        System.out.println("**** Pendataan Mahasiswa STMIK Eresha ****");
        System.out.print("Nama Mahasiswa: ");
        // menggunakan scanner dan menyimpan apa yang diketik di variabel nama
        nama = keyboard.nextLine();
        // Tampilkan output lagi
        System.out.print("Alamat: ");
        // menggunakan scanner lagi
        alamat = keyboard.nextLine();

        System.out.print("Usia: ");
        usia = keyboard.nextInt();

        // Menampilkan apa yang sudah simpan di variabel
        System.out.println("_____");
        System.out.println("Nama Karyawan: " + nama);
        System.out.println("Alamat: " + alamat);
        System.out.println("Usia: " + usia + " tahun");
    }
}
```

Perhatikan bahwa saat menggunakan fungsi untuk mengambil data, ini bergantung pada tipe data yang digunakan. Misalnya jika tipe datanya adalah String, maka fungsi atau metode yang digunakan adalah `nextLine ()`, `nextInt ()` sedang digunakan, dan Double menggunakan `nextDouble ()`, maka hasilnya adalah sebagai berikut.

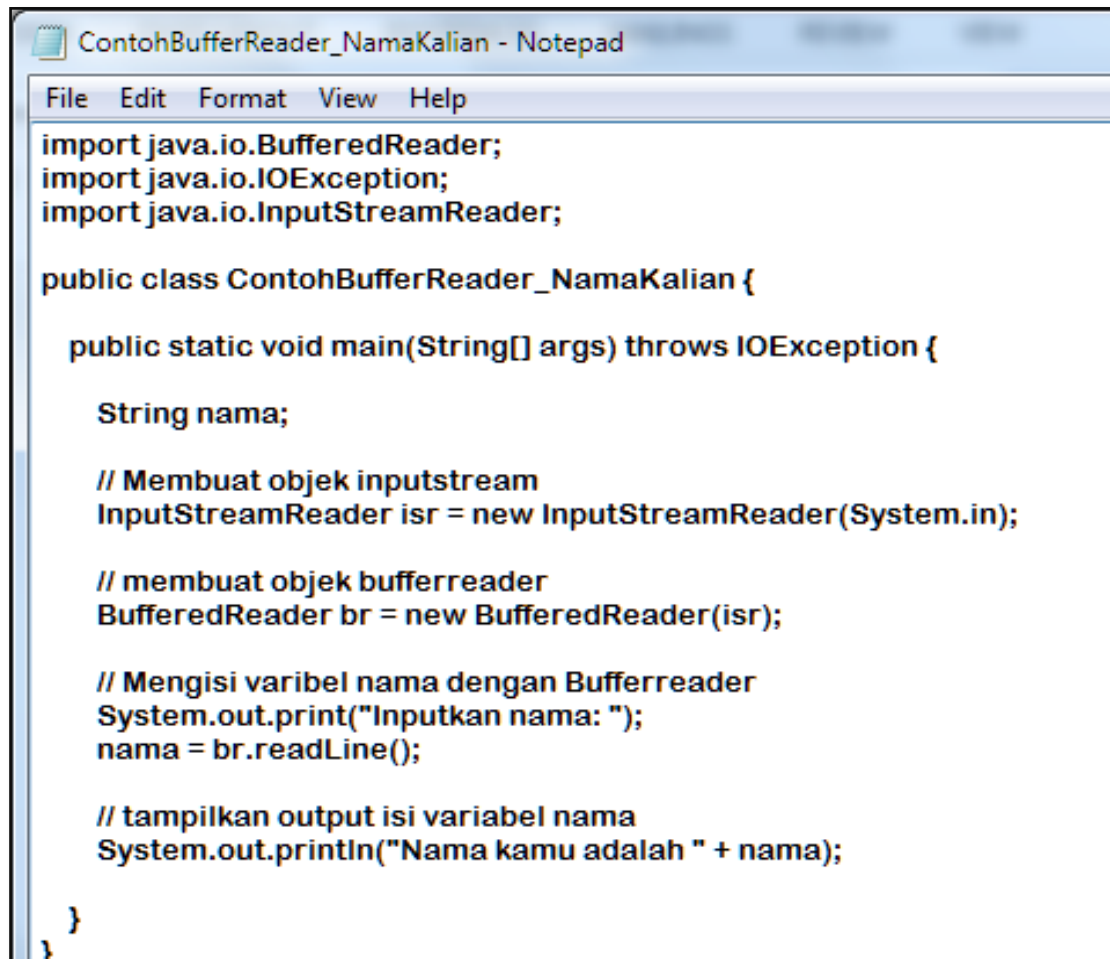
2. Gunakan kelas `BufferedReader` untuk input

Kelas `BufferedReader` tidak hanya digunakan untuk mendapatkan input dari keyboard. Kelas ini juga dapat digunakan untuk membaca masukan dari file dan jaringan. Kelas ini terletak di `java.io`. Harap tambahkan atau impor untuk menggunakan kelas `BufferedReader`.

```
import java.io.BufferedReader;
```

Latihan 2

Silahkan ketik pada notepad/notepad++/neatbean, simpan source code dengan ContohBufferedReader_NamaKalian.java pada drive D/E/F(sesuai perangkat komputer kalian),



```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class ContohBufferedReader_NamaKalian {

    public static void main(String[] args) throws IOException {

        String nama;

        // Membuat objek inputstream
        InputStreamReader isr = new InputStreamReader(System.in);

        // membuat objek bufferreader
        BufferedReader br = new BufferedReader(isr);

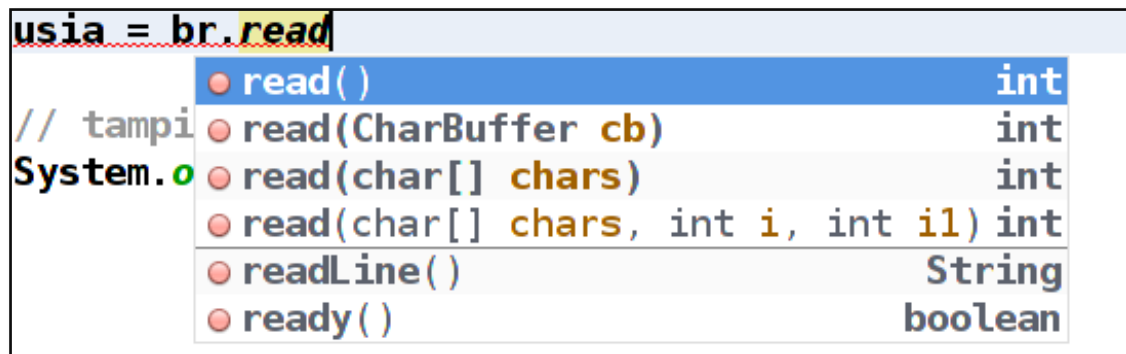
        // Mengisi variabel nama dengan Bufferreader
        System.out.print("Inputkan nama: ");
        nama = br.readLine();

        // tampilkan output isi variabel nama
        System.out.println("Nama kamu adalah " + nama);

    }
}
```

Pada class BufferedReader tidak dapat melakukan atau bekerja sendirian membutuhkan class InputStreamReader dan class IOException.. Jika di compile maka hasil program seperti dibawah ini.

Perbedaan BufferedReader dan Scanner dapat dilihat dari fungsi atau metode yang digunakan. Pemindai menggunakan next (), sedangkan BufferedReader menggunakan readLine (). Untuk tipe data integer, BufferedReader menggunakan fungsi read (). Kita bisa melihat ini di prompt pelengkapan otomatis (jika menggunakan netbean, lihat di bawah).



- Silahkan Tekan Ctrl +Spasi pada saat menulis kode untuk menampilkan hint autocomplete .

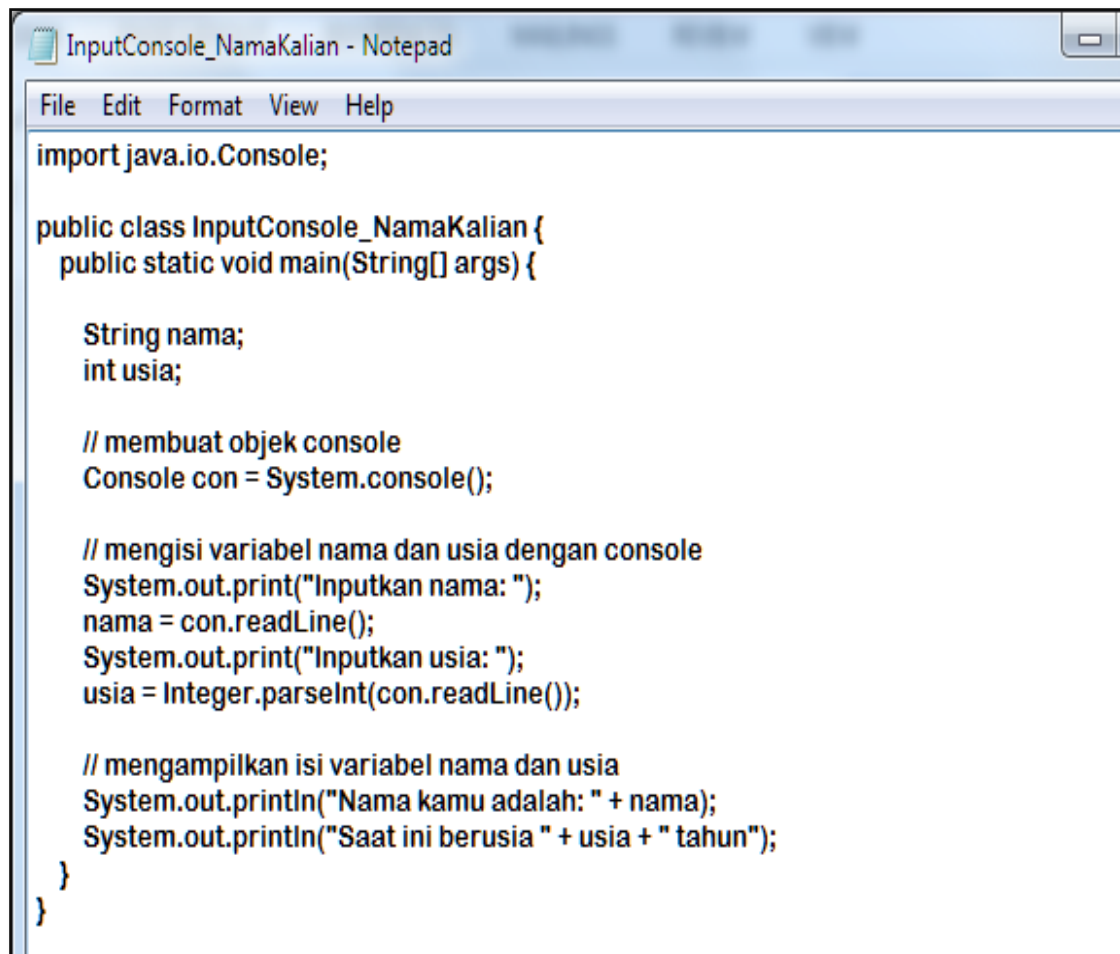
3 Gunakan Kelas Console untuk Input

Kelas Console memiliki kesamaan BufferedReader. Penggunaan fungsi readLine () pada Kelas Console berfungsi juga sebagai mengambil input. Pada kelas ini dapat digunakan hanya pada lingkungan konsol (seperti Terminal dan CMD). Kelas Konsol tidak dapat digunakan secara langsung di Netbeans. Oleh karena itu, kita harus mengkompilasinya secara manual. Dalam menerapkan dalam kelas ini, diperlukan tindakan mengimpornya atau mengetik perintah di bawah ini lebih dahulu.

```
import java.io.Console;
```

Latihan 3

Silahkan ketik source code dibawah ini pada notepad/notepad++/neatbean, simpan source code dengan InputConsole_NamaKalian.java pada drive D/E/F(sesuai perangkat komputer kalian), pada folder Latihan Algo.



```
import java.io.Console;

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

        String nama;
        int usia;

        // membuat objek console
        Console con = System.console();

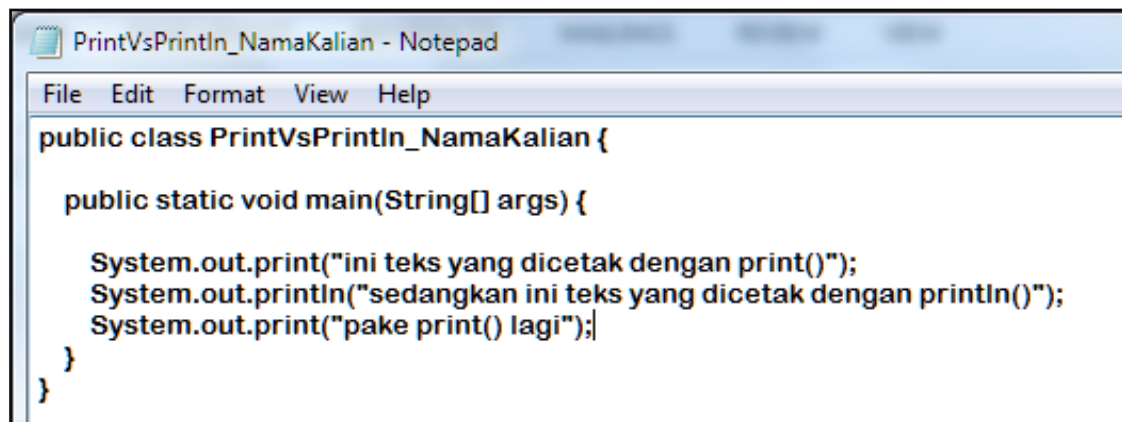
        // mengisi variabel nama dan usia dengan console
        System.out.print("Inputkan nama: ");
        nama = con.readLine();
        System.out.print("Inputkan usia: ");
        usia = Integer.parseInt(con.readLine());

        // menampilkan isi variabel nama dan usia
        System.out.println("Nama kamu adalah: " + nama);
        System.out.println("Saat ini berusia " + usia + " tahun");
    }
}
```

Dalam kode di atas, kami menggunakan fungsi `Integer.parseInt (con.readLine ())` untuk tipe data integer. Dengan kata lain, kami mengubah tipe data String menjadi Integer. Karena konsol tidak memiliki nilai kembalian integer dari fungsi `read ()`. Setelah itu, kompilasi melalui terminal atau CMD. Buka terminal dan masukkan direktori tempat kode disimpan. Ketik perintah: `javac InputConsole.java` untuk dikompilasi. Ketika kode sumber di atas dikompilasi dan dijalankan di Cmd, hasilnya adalah sebagai berikut.

Latihan 4

Silahkan ketik pada notepad/notepad++/neatbean, simpan source code dengan PrintVsPrintln_NamaKalian.java pada drive D/E/F(sesuai perangkat komputer kalian),



```
public class PrintVsPrintln_NamaKalian {  
    public static void main(String[] args) {  
        System.out.print("ini teks yang dicetak dengan print()");  
        System.out.println("sedangkan ini teks yang dicetak dengan println()");  
        System.out.print("pake print() lagi!");  
    }  
}
```

2. Menggabungkan String

fungsi print() maupun println(), pada saat digunakan adakalanya perlu mengambil teks dari variabel untuk menggabungkannya bersama teks lain. Contohnya pada saat akan menampilkan variabel **namaDepan** dan **namaBelakang**, dibawah ini.;

```
String nama_Depan = "Matakuliah";  
String nama_Belakang = "Praktikum Algoritma dengan Java";
```

Untuk menampilkannya dengan fungsi `print()`, maka hanya perlu memasukkannya ke sana, sebagai contoh di bawah ini.

```
System.out.print(nama_Depan);  
System.out.print(nama_Belakang);
```

Kode yang akan dihasilkan: kursus latihan algoritma Java. Faktanya, metode ini kurang efisien saat menggunakan dua fungsi print () karena menggabungkannya dengan operator +. Misalnya contoh berikut:

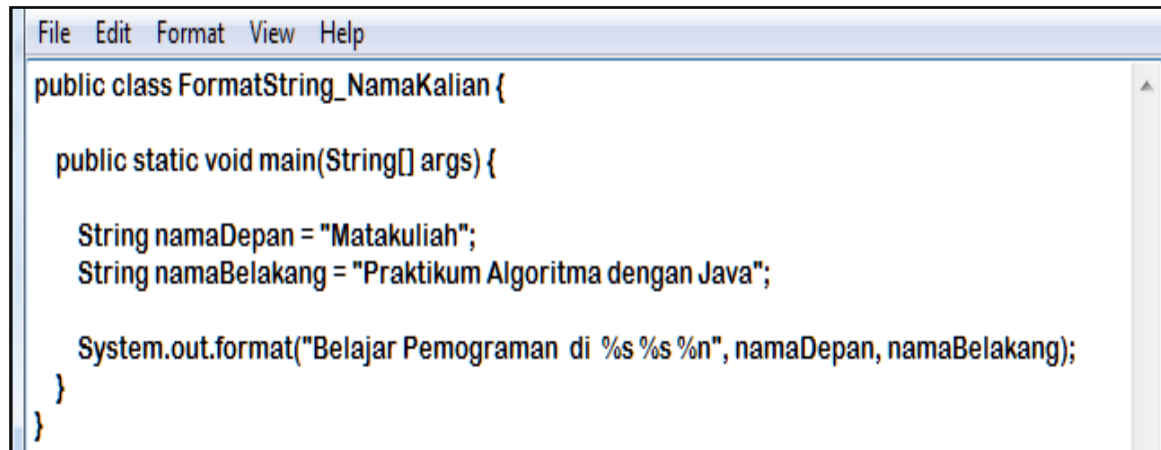
```
System.out.print (nama depan + nama belakang);  
Untuk memiliki ruang, cukup tambahkan spasi:  
System.out.print (nama depan + " " + nama belakang);
```

3. Format String

Fungsi format() digunakan untuk menggabungkan String yang lebih kompleks . Untuk memperdalam pemahaman ini silahkan mempraktikan langsung latihan 5 agar lebih memahami format penggunaan String, seperti dibawah ini>

Latihan 5

Silahkan ketik pada notepad/notepad++/neatbean, simpan source code dengan FormatString_NamaKalian.java pada drive D/E/F(sesuai perangkat komputer kalian), pada Folder Latihan Algo.



```
File Edit Format View Help
public class FormatString_NamaKalian {

    public static void main(String[] args) {

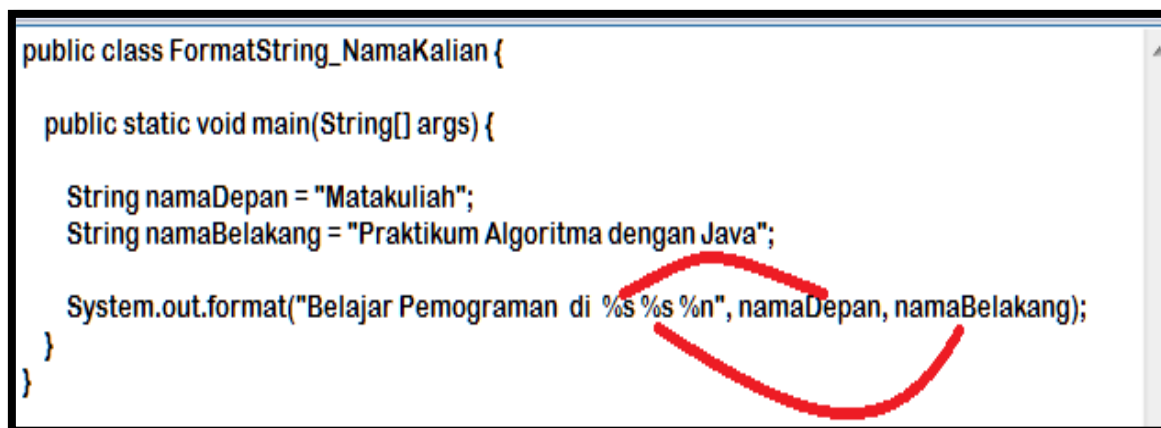
        String namaDepan = "Matakuliah";
        String namaBelakang = "Praktikum Algoritma dengan Java";

        System.out.format("Belajar Pemograman di %s %s %n", namaDepan, namaBelakang);
    }
}
```

Pada contoh diatas penggunaan simbol %s berfungsi mengambil nilai yang diperoleh dari variabel terdekatnya, %s artinya string. Selain %s, ada juga simbol lain:

- %d berfungsi sebagai pengambil nilai untuk desimal atau angka;
- %f berfungsi sebagai pengambil nilai untuk bilangan pecahan;
- %n berfungsi sebagai untuk baris baru, bisa juga pakai \n;

Sebagai contoh penempatan dan pembacaan string dalam penjelasan di deskripsikan dalam gambar dibawah ini.



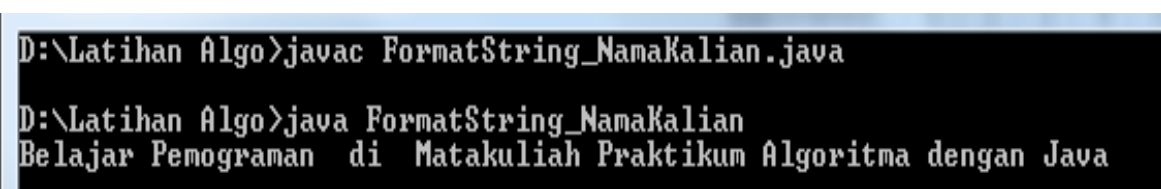
```
public class FormatString_NamaKalian {

    public static void main(String[] args) {

        String namaDepan = "Matakuliah";
        String namaBelakang = "Praktikum Algoritma dengan Java";

        System.out.format("Belajar Pemograman di %s %s %n", namaDepan, namaBelakang);
    }
}
```

Kode di atas akan menghasilkan



```
D:\Latihan Algo>javac FormatString_NamaKalian.java
D:\Latihan Algo>java FormatString_NamaKalian
Belajar Pemograman di Matakuliah Praktikum Algoritma dengan Java
```