

Nama : I Dewa Ayu Puspa Dewi  
NIM : 255150700111045  
Kelas : TI - D  
Bab : Input, Output, Proses  
Asisten : 1. Alya Hamidah  
2. Anindya Ratna Paramitha.

## LANGKAH 1

### 1. Soal

Buatlah program untuk menghitung pemakaian daya listrik dirumah tangga secara sederhana. Tampilan program sebagai berikut.

Program penghitung pemakaian listrik sederhana

Masukkan Nama : Bpk Asisten

Kelurahan : Java

Masukkan posisi awal Kwh Meter : 8000

Masukkan posisi akhir Khw Meter : 9000

Masukkan biaya beban saat ini : 140

Masukkan PPJ (dalam persen) : 10

=====PLN Java=====

Nama : Bpk Asisten

Kelurahan : Java

Pemakaian bulan ini : 1000 Kwh Meter

Tarif Listrik : Rp 140000,-

PPJ 10% : Rp 14000,-

Total Bayar : Rp 154000,-

=====

## 2. Screenshoot (Output)

```
Program Perhitungan Pemakaian Listrik Sederhana
Masukkan Nama : Bpk Asisten
Kelurahan : Java
Masukkan posisi awal Kwh Meter : 8000
Masukkan posisi akhir Kwh Meter : 9000
Masukkan biaya beban saat ini : 140
Masukkan PPJ (dalam persen) : 10
=====PLN Java=====
Nama : Bpk Asisten
Kelurahan : Java
Pemakaian bulan ini : 1000 Kwh Meter
Tarif Listrik : Rp 140000,-
PPJ 10% : Rp 14000,-
Total Bayar : Rp 154000,-
=====
```

## 3. Syntax

MODUL1.java	
	<pre>import java.util.Scanner;  public class MODUL1 {      public static void main(String[] args) {          Scanner input = new Scanner(System.in);          //INPUT DATA          //Keterangan beberapa variabel input :          //PawKwh : posisi awal Kwh Meter          //PakKwh : posisi akhir Kwh Meter          //konversiPPJ : pengubahan PPJ dari persen ke bentuk rupiah           System.out.println("Program Perhitungan Pemakaian Listrik Sederhana");          System.out.print("Masukkan Nama : ");          String nama = input.nextLine();          System.out.print("Kelurahan : ");          String kelurahan = input.nextLine();          System.out.print("Masukkan posisi awal Kwh Meter : ");</pre>

```

int PawKwh = input.nextInt();

System.err.print("Masukkan posisi akhir Kwh Meter : ");

int PakKwh = input.nextInt();

System.out.print("Masukkan biaya beban saat ini      : ");

int biayaBeban = input.nextInt();

System.out.print("Masukkan PPJ (dalam persen)        : ");

float ppj = input.nextFloat();

//PROCESS

int besarPemakaian = PakKwh-PawKwh;

int tarifListrik = biayaBeban*besarPemakaian;

float konversiPPJ = (tarifListrik*(ppj/100));

//float konversiPPJ = (tarifListrik*(ppj/100));

float totalBayar = tarifListrik+konversiPPJ;

//OUTPUT

System.out.println("=====PLN
Java=====");

System.out.println("Nama                : " +nama);

System.out.println("Kelurahan            : " +kelurahan);

System.out.println("Pemakaian bulan ini : " +besarPemakaian
+" Kwh Meter");

System.out.println("Tarif Listrik        : Rp " + (int)
tarifListrik +",-");

System.out.println("PPJ 10%              : Rp " + (int)
konversiPPJ +",-");

System.out.println("Total Bayar          : Rp " + (int)
totalBayar +",-");

System.out.println("=====
=====");

input.close();

}

```

	}
--	---

#### 4. Penjelasan

MODUL1.java	
	<pre>import java.util.Scanner;  public class MODUL1 {     public static void main(String[] args) {         Scanner input = new Scanner(System.in);</pre>

Pertama-tama, kita perlu memasukkan perintah `import java.util.Scanner;` untuk mengizinkan program menggunakan kelas `Scanner` dari paket `java.util` yang memiliki fungsi untuk membaca input dari pengguna (melalui keyboard). Tanpa perintah ini program tidak dapat membaca dari luar.

Di bagian bawah lagi ada `public class MODUL1 {` yang merupakan instruksi untuk membuat sebuah kelas baru bernama `MODUL1` yang dapat diakses oleh bagian program lainnya. Kelas menjadi wadah untuk semua kode (variable dan metode) yang terkait dengan modul atau fungsi tertentu dalam program.

Selanjutnya, baris kode `public static void main(String[] args) {` adalah titik awal dari setiap program Java yang berdiri sendiri. Ini adalah “gerbang” di mana program mulai dieksekusi oleh mesin virtual Java (JVM). Bagian ini penting karena tanpa baris ini, JVM tidak tau dimana harus memulai. Setiap program Java yang dijalankan dari baris perintah atau IDE akan mencari dan mengeksekusi kode di dalam blok ini. Jika dijabarkan dengan lebih detail, bagian ini dapat dipisahkan menjadi bagian-bagian kecil seperti di bawah ini:

- `Public`, ini adalah pengubah akses yang berarti metode ini dapat dipanggil dari mana saja.
- `Static`, kata kunci ini berarti `main` milik kelas itu sendiri, bukan milik objek yang dibuat dari kelas tersebut.
- `Void`, ini berarti metode ini tidak mengembalikan nilai apa pun setelah selesai dieksekusi
- `Main`, ini adalah nama standar untuk metode utama. JVM secara spesifik mencari metode dengan nama ini untuk memulai eksekusi program. Nama ini tidak bisa diubah.
- `String[]`, menunjukkan bahwa argument yang diterima adalah sebuah array dari string.
- `Args`, ini adalah nama variable untuk array tersebut.

Terakhir, baris kode `Scanner input = new Scanner(System.in);` memberikan instruksi dalam Bahasa Java untuk membuat sebuah objek yang bisa membaca input dari pengguna (*user*)

	<pre>//INPUT DATA  //Keterangan beberapa variabel input : //PawKwh : posisi awal Kwh Meter //PakKwh : posisi akhir Kwh Meter //konversiPPJ : pengubahan PPJ dari persen ke bentuk rupiah</pre>
--	--

	<pre> System.out.println("Program Perhitungan Pemakaian Listrik Sederhana"); System.out.print("Masukkan Nama : "); String nama = input.nextLine(); System.out.print("Kelurahan : "); String kelurahan = input.nextLine(); System.out.print("Masukkan posisi awal Kwh Meter : "); int PawKwh = input.nextInt(); System.out.print("Masukkan posisi akhir Kwh Meter : "); int PakKwh = input.nextInt(); System.out.print("Masukkan biaya beban saat ini : "); int biayaBeban = input.nextInt(); System.out.print("Masukkan PPJ (dalam persen) : "); float ppj = input.nextFloat(); </pre>
--	--

Sekarang, kita masuk ke bagian input data. Penginputan data yang dilakukan adalah dengan menggunakan mekanisme memasukkan data oleh *user* melalui terminal. Hal ini mungkin dilakukan karena adanya `Scanner input = new Scanner(System.in);` yang memberikan ruang bagi *user* untuk mengisi data nya melalui keyboard dan media input lainnya. Indikasi dari penggunaan metode `int PawKwh = input.nextInt();` adalah adanya nama tipe data primitif (`int`, `string`, `float`) sebelum variable. Variabel yang perlu dimasukkan kumpulan kata akan diawali `String`. Sementara itu, variable yang biasa diisi bilangan bulat diberikan awal `int` dan `float` untuk merepresentasikan bahwa datanya adalah bilangan desimal / pecahan dengan presisi tinggi.

Disediakan pula tampilan instruksi yang output yang dikerluarkan melalui monitor. Beberapa baris dengan awalan serupa `System.out.print("Masukkan Nama : ");` untuk menampilkan output kepada user dan memberikan instruksi tertampil untuk mengisikan data user yang hendak di proses. Di sini bentuk yang digunakan adalah `System.out.print` yang membuat data tertampil tanpa garis baru.

	<pre> //PROCESS  int besarPemakaian = PakKwh-PawKwh; int tarifListrik = biayaBeban*besarPemakaian; float konversiPPJ = (tarifListrik*(ppj/100)); //float konversiPPJ = (tarifListrik*(ppj/100)); float totalBayar = tarifListrik+konversiPPJ; </pre>
--	--

Setelah pemasukan (input) data selesai maka data akan diproses. Pada bagian ini, bagaimana data diproses akan dijelaskan. Proses data pada tahap ini diproses dengan operator binary menggunakan operator aritmatika yang diperlihatkan dengan penggunaan tambah, kurang, kali, dan bagi. Dalam proses tersebut terdapat penerapan beberapa rumus, yaitu:

- `int besarPemakaian = PakKwh-PawKwh;` digunakan untuk menghitung pemakaian dengan mengurangi posisi akhir dan awal Kwh Meter. Tipe data adalah integral karena diisi bilangan bulat.
- `int tarifListrik = biayaBeban*besarPemakaian;` digunakan untuk menghitung tarif listrik dengan cara mengalikan biaya beban dengan besar pemakaian diatasnya. Tipe datanya adalah integral karena diisi bilangan bulat
- `float konversiPPJ = (tarifListrik*(ppj/100));` digunakan untuk mengubah PPJ yang semulanya berbentuk persen menjadi angka dalam rupiah. Alasan penggunaan float sebab PPJ bisa berbentuk pecahan desimal.

- `float totalBayar = tarifListrik+konversiPPJ;` digunakan untuk menghitung total bayar dengan cukup menambahkan tarif listrik dan konversi PPJ. Tipe data juga float karena sebelumnya konversi PPJ adalah float.

```
//OUTPUT

System.out.println("=====PLN
Java=====");
System.out.println("Nama                : " +nama);
System.out.println("Kelurahan           : " +kelurahan);
System.out.println("Pemakaian bulan ini : " +besarPemakaian
+" Kwh Meter");
System.out.println("Tarif Listrik       : Rp " + (int)
tarifListrik +",-");
System.out.println("PPJ 10%            : Rp " + (int)
konversiPPJ +",-");
System.out.println("Total Bayar        : Rp " + (int)
totalBayar +",-");

System.out.println("=====PLN
Java=====");
input.close();
}
```

Terakhir, kita masuk ke bagian output. Bagian ini akan memberikan tampilan dari hasil pemrosesan data yang dimasukkan *user* dengan rapi dan terstruktur. Perintah output yang digunakan adalah `System.out.println` yang bertujuan menampilkan teks, lalu secara otomatis baris baru (new line) terbentuk. Tiga baris sebelum `System.out.println` terakhir ada penambahan  `+ (int) totalBayar + ",-"`; sebelum variabelnya. Ini dilakukan karena data total bayar, konversi PPJ, dan tarif listrik memuat data float. Apabila tetap dimasukkan tanpa penambahan  `+ (int)` akan menyebabkan error pada program yang dijalankan.

Lalu, kita beralih ke bagian paling bawah ada `input.close();` yang memberikan instruksi Java untuk menutup objek `Scanner` yang dibuat sebelumnya. Di awal, kita membuka aliran data melalui perintah `Scanner input = new Scanner(System.in);`. Aliran ini perlu ditutup. Jika tidak ditutup, sumber daya akan tetap “terbuka” dan tidak bisa digunakan oleh program lain, bahkan setelah program selesai. Demikianlah serangkaian sistem perhitungan pemakaian listrik sederhana yang kita buat dalam pemrograman dapat diakses oleh user setelah melalui berbagai tahapan.

## 5. Link GitHub

<https://github.com/AIU7AgrataEleigh/MODUL-1-PRAKTIKUM-PEMDAS->