**Pertemuan 3 : Input Output**

**Topik Sesi** :

1. Pengertian input dan penulisan fungsi Input()
2. Pengertian output dan penulisan Print() pada Bahasa Python

**Capaian pembelajaran mata kuliah** :

1. Mahasiswa mampu menjelaskan konsep dan penggunaan input/output dalam pemrograman;
2. Mahasiswa mampu menjelaskan dan mempraktikkan fungsi Input() dan Print() pada Bahasa Python.
3. **Pengertian Input Output**

Input adalah masukan yang kita berikan ke program.

Program akan memprosesnya dan menampilkan hasil outputnya.

Input, proses, dan output adalah inti dari semua program komputer.

Python menyediakan banyak fungsi *built-in* yang bisa digunakan. Salah satunya adalah yang berkenaan dengan fungsi i/o atau input output. Fungsi bawaan untuk melakukan operasi output adalah print(), dan fungsi untuk melakukan operasi input adalah fungsi input().

1. **Operasi Input Menggunakan Fungsi input()**

Agar program kita lebih interaktif, kita bisa meminta input atau masukan dari user. Python memiliki fungsi input() untuk melakukan hal tersebut. Sintaksnya adalah seperti berikut :

**Input([prompt])**

Prompt bersifat opsional berfungsi sebagai string yang kita ingin tampilkan di layar. Agar inputannya tidak langsung hilang, maka harus disimpan ke dalam variabel. Perhatikan contoh berikut :

Nama = input(“Masukkan nama : “)

Bila kita menginput bilangan, misalnya integer lewat fungsi input(), maka hasil inputan tersebut adalah string dan bukan integer. Kita harus mengubahnya terlebih dahulu menjadi tipe integer menggunakan fungsi int(). Contohnya adalah sebagai berikut :

panjang = input(“Masukkan nilai panjang : “)

lebar = input(“Masukkan nilai lebar : “)

luas = int(panjang) \* int(lebar)

print (“Luas : “, luas)

Jika kita aplikasikan melalui IDE, hasilnya adalah sebagai berikut :



1. **Operasi Output Menggunakan Fungsi print()**

Untuk menampilkan hasil masukan, kita bisa menggunakan fungsi print().

Contoh :

nama = (“Safitri Jaya”)

print(nama)

a = 15

print(“Nilai a : “, a)

selain mencetak ke layar, fungsi print() juga bisa digunakan untuk mencetak data ke file. Sintaknya adalah sebagai berikut :

print(\*objects, sep=' ', end='\n', file=sys.stdout, flush=False)

Pada sintaks tersebut,  **object** adalah nilai yang hendak dicetak. Fungsi  **print()** akan mengubah semua objek menjadi string terlebih dahulu sebelum dicetak.

**Sep** adalah pemisah(separator) yang berfungsi sebagai tanda pemisah antar objek yang dicetak. Defaultnya adalah tanda spasi.

**End** adalah karakter yang dicetak di akhir baris. Defaultnya adalah tanda *newline* (baris baru).

**File** adalah nama file kemana objek akan dicetak. Defaultnya adalah ke sys.stdout (layar).

**Flush** adalah opsi apakah keluarannya diflush atau tidak.

Contoh :

Print (1, 3, 5, 7)

Hasilnya : 1 3 5 7

Print(1, 2, 3, 4, sep = ‘ \* ‘)

Hasilnya : 1 \* 2 \* 3 \* 4

Print(1, 2, 3, 4, sep = ‘ # ’, end = ‘ & ‘)

Hasilnya : 1 # 2 # 3 # 4 &

**Menampilkan variabel dan teks**

Tanda koma digunakan untuk menampilkan gabungan antara teks dan variabel.

Contoh :

nama = “Dian”

print (“Hello”, nama)

**Menggunakan fungsi format()**

Fungsi format() akan menggabungkan isi variabel dengan teks.

Contoh 1:

nama = input(“Nama : “)

print (“halo {} apa kabar?”.format(nama))

Tanda {} akan otomatis diganti sesuai dengan nilai yang diinputkan ke variabel nama

**Hasilnya :**

Nama : Safitri (diinput)

Halo **Safitri** apa kabar?

Contoh 2:

nama = input("Nama : ")

alamat = input("Alamat : ")

print ("halo {} Kamu tinggal di {} ya".format(nama, alamat))

**Hasilnya :**

Nama : Safitri (diinput)

Alamat : Graha Raya (diinput)

halo **Safitri** Kamu tinggal di **Graha Raya** ya

**Penggabungan teks dan variabel cara lama menggunakan symbol persen (%)**

Contoh :

nama = input("Nama : ")

print ("halo %s" %nama)

**Hasilnya :**

Nama : Safitri

Halo **Safitri**

Tanda %s untuk tipe data teks, %d untuk angka (desimal), dan %f untuk bilangan pecahan

**Sumber referensi :**

* 1. Zelle, J. M. (2002). Programming : An Introduction to Computer Science. Wartburg College.
	2. Knuth, DE. The Art of Computer Programming: Fundamental Algorithms, Addison-Wesley, 1997.
	3. Abdul Kadir (2018) Dasar Pemrograman Python.
	4. [www.python.org](http://www.python.org)
	5. [www.pythonindo.com](http://www.pythonindo.com)