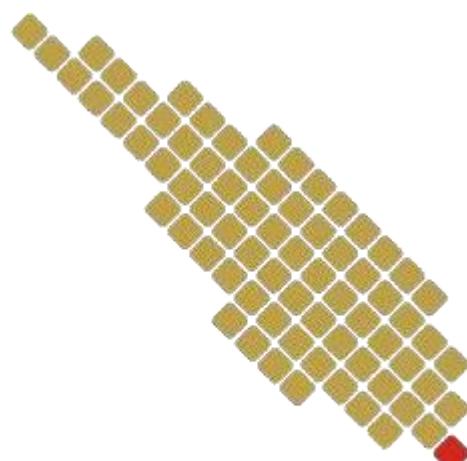


**Modul Praktikum**  
**ALGORITMA PEMROGRAMAN**



**PROGRAM STUDI SAINS DATA FAKULTAS SAINS  
INSTITUT TEKNOLOGI SUMATERA  
2025**

## Value, Assignment, Input/Output, dan Expression

### 1. Tujuan

Setelah mempelajari Bab ini, mahasiswa diharapkan dapat:

Dapat menjelaskan dan menggunakan Value, Assignment, Input/Output, dan Expression pada Python

### 2. Dasar Teori

#### 2.1. *Value*

*Value* merupakan data yang disimpan di dalam variabel atau dapat digunakan dalam operasi. *Value* biasanya yang umum dikenal yaitu 1, 2, ‘a’, ‘z’, dan “Hello Word”. *Value* dibagi menjadi beberapa tipe yang berbeda, misalnya 2 untuk `integer`, “Hello Word” untuk sebuah nilai `string`, dll. Beberapa contoh tipe value:

- a. `Integer` (bilangan bulat): 5, 42, -10
- b. `Float` (bilangan desimal): 3.14, 0.001, -7.5
- c. `String` (teks): "Hello, world!", "Python", "12345"
- d. `Boolean` : True, False

Untuk mencoba memahami values dan type, silakan coba beberapa baris kode pada gambar 1. ([silahkan anda coba menggunakan tipe data selain dari contoh](#))

```
1  print(5)
2  print('Z')
3  print('True')
```

Gambar 1. Potongan kode value

perintah `print` bekerja untuk *value* selain `string`, seperti `integer` (bilangan bulat), `float` (bilangan pecahan), `character` (huruf), atau `boolean` (benar/salah). Python menyediakan fungsi *built-in* untuk melakukan pengecekan tipe data pada *value* dengan menggunakan fungsi `type()`

```
x=5  
print(x, "tipenya adalah ", type(x))  
x=0.2  
print(x, "tipenya adalah ", type(x))  
x="Hai"  
print(x, "tipenya adalah ", type(x))
```

Gambar 2. Potongan kode penggunaan type

Pada gambar 2, dilakukan pengecekan type data, outputnya adalah value dari x adalah 5 dengan type datanya nya `Integer`, 0.2 tipenya adalah `float`, “Hai” tipenya adalah `str` atau `string`. Output dapat dilihat pada gambar 3. Output type pada value. (silahkan anda coba menggunakan tipe data selain dari contoh)

```
5 tipenya adalah <class 'int'>  
0.2 tipenya adalah <class 'float'>  
Hai tipenya adalah <class 'str'>
```

Gambar 3. Output type pada value.

Catatan : Ketika menggunakan bilangan bulat besar, beberapa model penulisan menggunakan tanda koma (,) diantara kelompok tiga digit. Misalnya pada penulisan 1.000.000. Dalam python, akan dianggap sebagai bilangan bulat. `print(1,000,000)` menghasilkan 1,0,0. Hal ini terjadi karena Python menganggap bahwa 1,000,000,000 merupakan kiriman parameter sebanyak 3 parameter pada fungsi print, yaitu 1, 0, dan 0.

## 2.2. Assigment

**Assigment** adalah proses memberikan atau mengubah nilai variabel setelah variabel tersebut di inisialisasi (**Inisialisasi** adalah proses mendeklarasikan sebuah variabel dan memberikan nilai awal dari variabel tersebut). Inisialisasi adalah langkah pertama dalam membuat variabel dapat digunakan. Inisialisasi memastikan bahwa variabel memiliki nilai yang valid sebelum digunakan dalam program.). **Assignment** dalam Python adalah proses menempatkan nilai ke dalam sebuah variabel. Dalam Python, assignment dilakukan dengan menggunakan tanda `=` (**sama dengan**). **Variabel di sisi kiri = akan diberi nilai dari ekspresi di sisi kanan**. Assignment dapat dilakukan berkali-kali selama eksekusi program dapat melakukan perubahan pada nilai variabel.

```
39    x = 10  
40    y = 5 + 7
```

Gambar 4. Assignment.

Pada gambar 4, **x=10** merupakan proses menyimpan nilai 10 ke dalam variabel x, **y = 5 + 7** menyimpan hasil penjumlahan  $5 + 7$  ke dalam variabel y.

### 2.3. *Input/Output*

Sebuah program biasanya memiliki alur kerja Input - Proses – Output. Input adalah data/masukan yang dibutuhkan supaya program bisa berjalan. Proses adalah langkah-langkah yang dilakukan oleh program untuk memecahkan masalah. Sedangkan Output adalah hasil yang didapatkan setelah menjalankan langkah-langkah tersebut. Python juga dapat menangani input dari pengguna. Input dalam hal ini dapat berupa input text yang dimasukkan oleh pengguna. Python menyediakan built-in function yang disebut `input` untuk mendapatkan input dari keyboard. Ketika fungsi ini dipanggil, program akan berhenti dan menunggu pengguna untuk mengetik sesuatu. Ketika pengguna menekan tombol Enter, program akan dilanjutkan dan input akan mengambilkan apa yang diketik oleh pengguna sebagai string. Pada gambar 5, saat anda menuliskan fungsi `input()`, dan melakukan perintah `print`, output adalah input yang diberikan oleh pengguna.

```
33    inp = input()  
34    print(inp)
```

Gambar 5. Potongan kode input

Sebelum mendapatkan input dari pengguna, kita juga dapat mencetak prompt text untuk memberitahu pengguna apa yang harus diinput. String tersebut dapat diteruskan ke input untuk ditampilkan kepada pengguna sebelum berhenti untuk input. Dapat dilihat pada gambar 6. Pada gambar 6 output saat program di running adalah siapa namamu dan menerima inputan pada pengguna. Tanda `\n` pada akhir ‘siapa nama mu ?’ mewakili bars baru atau ganti baris sehingga input pengguna muncul dibawah prompt.

```
36    name = input('Siapa nama mu ?\n')  
37    print(name)
```

Gambar 6. Potongan kode input menggunakan prompt

## 2.4. Expressions

*Expression* merupakan representasi dari nilai dan dapat terdiri dari gabungan antara values, variable dan operator. *Value* dengan sendirinya dapat dianggap sebagai expression dan juga variabel. Berikut adalah beberapa jenis ekspresi dalam python:

1. Ekspresi Aritmatika. Contoh ekspresi aritmatika dapat dilihat pada gambar 7.  
*(silahkan anda coba menggunakan operator aritmatika lain)*

```
13  x = 10
14  y = 3
15
16  print(x + y)
17  print(x - y)
18  print(x % y)
```

Gambar 7. Potongan kode Ekspresi Aritmatika

2. Ekspresi Perbandingan. Contoh ekspresi perbandingan dapat dilihat pada gambar 8.  
*(silahkan anda coba menggunakan operator perbandingan dan menggunakan variable berbeda)*

```
20  a = 10
21  b = 5
22  c = 10
23  print(a == b)
24  print(a != b)
25  print(a > b)
```

Gambar 8. Potongan kode Ekspresi perbandingan

3. Ekspresi Logika. Contoh ekspresi perbandingan dapat dilihat pada gambar 9.  
*(silahkan anda coba menggunakan operator logika yang lain)*

```
27  a = True
28  b = False
29  print(a and b)
30  print(a or b)
31  print(not a)
```

Gambar 8. Potongan kode Ekspresi perbandingan

## 2.5. Kegiatan Praktikum

Kegiatan praktikum yang akan dilakukan adalah sebagai berikut:

1. Membuat dan memberikan nilai variabel.
  2. Melakukan Assigment
  3. Memberikan dan mencetak nilai dalam variabel.
  4. Menggunakan ekspresi
- 

### 1. Membuat dan memberikan nilai variabel.

Misalkan sebuah data pribadi berisi nama, alamat, umur, tempat lahir, tanggal lahir, indeks prestasi kumulatif akan memberikan 6 (enam) buah variabel dengan tipe datanya.

```
1 Nama = input("Nama Anda : ")
2 Alamat = input("Alamat Tinggal Anda : ")
3 Umur = input("Umur Anda : ")
4 TL = input("Tempat Lahir Anda : ")
5 Tgl = input("Tanggal Lahir Anda : ")
6 IPK = input("IPK Anda : ")
7
8 print("DATA AKADEMIK")
9 print("Nama Anda : ", Nama)
10 print("Alamat Tinggal Anda : ", Alamat)
11 print("Umur Anda : ", Umur)
12 print("Tempat Lahir Anda : ", TL)
13 print("Tanggal Lahir Anda : ", Tgl)
14 print("IPK Anda : ", IPK)
```

Output dari potongan *source code* diatas dapat dilihat pada gambar 8

```
DATA AKADEMIK
Nama Anda : Sains Data
Alamat Tinggal Anda : Lampung
Umur Anda : 28
Tempat Lahir Anda : Bandung
Tanggal Lahir Anda : 07
IPK Anda : 5
```

Gambar 8. Ouput data akademik.

Bagaimana dengan tipe data yang lain? Kita akan mencoba untuk membuat program dengan menggunakan tipe data bilangan baik integer maupun float.

```
1 x1 = eval(input("X1 = "))
2 x2 = eval(input("X2 = "))
3 x3 = eval(input("X3 = "))
4 x4 = eval(input("X4 = "))
5 jumlah = x1+x2+x3+x4
6 kali = x1*x2*x3*x4
7 print('-----')
8 print('Hasil Penjumlahan semua bilangan = ', jumlah)
9 print('Hasil Perkalian semua bilangan = ', kali)
10 jumlah = jumlah + 0.5
11 print('Jika ditambah 0.5 hasilnya = ', jumlah)
12 kali = kali * 0.5
13 print('Jika dikali 0.5 hasilnya = ', kali)
```

Output dari potongan *source code* diatas dapat dilihat pada gambar 9

```
X1 = 1
X2 = 2
X3 = 3
X4 = 4
-----
Hasil Penjumlahan semua bilangan = 10
Hasil Perkalian semua bilangan = 24
Jika ditambah 0.5 hasilnya = 10.5
Jika dikali 0.5 hasilnya = 12.0
```

Gambar 9. Ouput data dengan tipe data berbeda.

## 2. Melakukan Assigment

Memberikan nilai dalam variable

1. Lakukan inisiasi variabel atau konstanta dari permasalahan berikut!

Menjumlahkan total harga pada saat konsumen membeli beberapa barang.

**Langkah 1 :**

**Inisiasi Persoalan Variabel/konstanta input :**

```
kode_barang,          nama_barang,          harga_satuan_barang,
jumlah_per_barang_beli, total_harga_per_transaksi.
```

**Proses :**

- harga\_beli\_per\_barang = harga\_satuan\_barang \* jumlah\_per\_barang\_beli

- `total_harga_per_transaksi = harga_beli_per_barang + total_harga_per_transaksi`

### **Output :**

`total_harga_per_transaksi`

### **Langkah 2 :**

Menetapkan Tipe Data:

- `kd_brg, nama_brg` bertipe data string
- `jumlah_brg` bertipe data integer
- `harga_satuan, harga_beli, total_hrg_brg` bertipe data float.

### **Langkah 3 :**

Kode Program

```

1 total_hrg_brg= 0.0
2 kd_brg=input("Kode barang = ")
3 nama_brg=input("Nama barang = ")
4 harga_satuan=eval(input("Harga satuan barang =Rp. "))
5 jumlah_brg=eval(input("Jumlah barang yang dibeli = "))
6 harga_beli = harga_satuan * jumlah_brg
7 total_hrg_brg= harga_beli + total_hrg_brg
8 print("Total harga yang dibayar Rp",total_hrg_brg)

```

Output dari program diatas dapat dilihat pada gambar 10.

```

Kode barang = 001
Nama barang = LAPTOP
Harga satuan barang =Rp. 5000000
Jumlah barang yang dibeli = 1
Total harga yang dibayar Rp 5000000.0

```

Gambar 10. Output data dari kode langkah 3