

BAGIAN IV TIPE DATA, VARIABEL, DAN I/O

Capaian Pembelajaran

1. Mahasiswa mampu membedakan tipe data yang terdapat pada bahasa pemrograman Python
2. Mahasiswa mampu memahami cara menggunakan fungsi input dan output standar dalam bahasa Python
3. Mahasiswa mampu membuat program dengan menggunakan Variabel dengan tipe data yang tepat

A Tipe Data

Tipe data merupakan bagian program yang paling penting karena tipe data mempengaruhi setiap instruksi yang akan dilaksanakan oleh computer. Misalnya saja 5 dibagi 2 bisa saja menghasilkan hasil yang berbeda tergantung tipe datanya. Jika 5 dan 2 bertipe integer maka akan menghasilkan nilai 2, namun jika keduanya bertipe float maka akan menghasilkan nilai 2.5000000. Pemilihan tipe data yang tepat akan membuat proses operasi data menjadi lebih efisien dan efektif.

Dalam bahasa Python terdapat beberapa tipe data dasar, seperti terlihat pada Tabel IV.1

Tabel IV.1: Tipe data dasar pada bahasa Python

Tipe Data	Contoh	Penjelasan
Boolean	True atau False	Menyatakan benar (True) yang bernilai 1, atau salah (False) yang bernilai 0
String	"Sisfo Untan"	Menyatakan karakter/kalimat bisa berupa huruf angka, dll (diapit tanda " atau ')
Integer	25 atau 1209	Menyatakan bilangan bulat
Float	3.14 atau 0.99	Menyatakan bilangan yang mempunyai koma
Hexadecimal	9a atau 1d3	Menyatakan bilangan dalam format heksa (bilangan berbasis 16)
Complex	1 + 5j	Menyatakan pasangan angka real dan imajiner
List	['xyz', 786, 2.23]	Data untaian yang menyimpan berbagai tipe data dan isinya bisa diubah-ubah
Tuple	('xyz', 768, 2.23)	Data untaian yang menyimpan berbagai tipe data tapi isinya tidak bisa diubah
Dictionary	{'nama': 'adi', 'id': 2}	Data untaian yang menyimpan berbagai tipe data

		berupa pasangan penunjuk dan nilai
--	--	------------------------------------

B Variabel

Variabel adalah suatu pengenal (identifier) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program. Berbeda dengan konstanta yang nilainya selalu tetap, nilai dari suatu variabel bisa diubah-ubah sesuai kebutuhan.

Nama dari suatu variabel dapat ditentukan sendiri oleh pemrogram dengan tidak menggunakan keyword yang memang telah digunakan oleh python. Keyword yang tidak boleh digunakan tersebut antara lain:

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

Berikut ini adalah contoh script python yang menggunakan berbagai tipe data yang diambil dari situs resmi python

```
>>> name = "Fred"
>>> f"He said his name is {name!r}."
"He said his name is 'Fred'."
>>> f"He said his name is {repr(name)}." # repr() is equivalent to !r
"He said his name is 'Fred'."
>>> width = 10
>>> precision = 4
>>> value = decimal.Decimal("12.34567")
>>> f"result: {value:{width}.{precision}}" # nested fields
'result:      12.35'
>>> today = datetime(year=2017, month=1, day=27)
>>> f"{today:%b %d, %Y}" # using date format specifier
'January 27, 2017'
>>> number = 1024
>>> f"{number:#0x}" # using integer format specifier
'0x400'
```

C Deklarasi

Deklarasi diperlukan bila kita akan menggunakan pengenal (identifier) dalam program. Identifier dapat berupa variable, konstanta dan fungsi.

C.1. Deklarasi Preprocessor

Pada bahasa Python, program tidak harus diawali dengan deklarasi preprocessor. Preprocessor yang umum dideklarasikan pada kondisi-kondisi tertentu saja. Misalnya ketika akan menggunakan library. Untuk deklarasi preprocessor bagi library insyaa Allah akan dijelaskan pada pembahasan tentang library.

C.2. Deklarasi Variabel

Pada bahasa python tidak perlu untuk menentukan tipe data terlebih dahulu. Bahkan variabel dapat langsung digunakan tanpa harus dibentuk terlebih dahulu. Variabel yang ingin digunakan akan ditulis dan langsung diberikan nilai. Dengan pemberian nilai itulah nantinya yang menentukan tipe data dari variabel tersebut.

Bentuk umum pendeklarasian suatu variable adalah :

```
nama_variabel=nilai
```

Contoh :

```
>>> x = 123.456
>>> print(x)
123.456
>>> x = "This is a string"
>>> print(x + "!")
This is a string!
```

D Komentar

Komentar program hanya diperlukan untuk memudahkan pembacaan dan pemahaman suatu program (untuk keperluan dokumentasi program). Dengan kata lain, komentar program hanya merupakan keterangan atau penjelasan program. Untuk memberikan komentar atau

penjelasan dalam bahasa Python digunakan tanda #. Komentar program tidak akan ikut diproses dalam program (akan diabaikan)

Contoh Kode Program

```
x = 365
y = 7
# this is a comment

print(x % y) # find the remainder
# print (x // y)
# another comment
```

Terdapat pula jenis lain yang mirip dengan komentar yaitu adalah Docstring. Docstring menyajikan tujuan yang sama dengan komentar, karena dirancang untuk menjelaskan kode. Namun lebih spesifik dan memiliki sintaks yang berbeda. Docstring dibuat dengan meletakkan string multiline yang berisi penjelasan tentang fungsi di bawah baris pertama fungsi ini.

Perhatikan contoh berikut:

```
def shout(word) :
    """
    Print a word with an
    exclamation mark following it.
    """
    print(word + "!")

shout("spam")
```

E Input dan Output (I/O)

Dalam pemrograman dua fungsi yang penting untuk digunakan adalah fungsi input dan output. Kedua fungsi tersebut umumnya memberdayakan perangkat input dan perangkat output pada komputer.

E.1. Input

Dalam bahasa Python proses memasukkan suatu data bisa menggunakan sebuah fungsi yaitu `input()`. Perhatikan contoh berikut

```
>>> input("Enter something please: ")
```

```
Enter something please: This is what the user enters!
```

```
'This is what the user enters!'
```

E.2. Ouput

Bentuk umum dari output adalah :

```
print();
```

Contoh :

```
printf("Sistem Informasi Universitas Tanjungpura");
```

Perhatikan contoh program berikut:

```
>>> print(1 + 1)
2
>>> print("Hello\nWorld!")
Hello
World!
```

BAGIAN V OPERATOR DAN EKSPRESI

Capaian Pembelajaran

1. Mahasiswa mampu menjelaskan macam-macam Operator dalam bahasa Python
2. Mahasiswa mampu menjelaskan macam-macam ekspresi dalam bahasa Python
3. Mahasiswa mampu membuat program sederhana dengan menggunakan operator dan ekspresi dalam bahasa Python

A Operator

Operator merupakan simbol-simbol yang digunakan untuk tujuan tertentu. Operator dalam bahasa Python dikelompokkan berdasarkan keperluan atau kegunaannya.

A.1. Operator Penugasan

Operator Penugasan (Assignment operator) dalam bahasa Python berupa tanda sama dengan (“=”). Contoh : nilai = 80; A = x * y; Artinya : variable “nilai” diisi dengan 80 dan variable “A” diisi dengan hasil perkalian antara x dan y.

Ada berbagai operator majemuk dengan Python seperti += 5 yang menambahkan variabel dan kemudian memberikan yang sama. Ini setara dengan a = a + 5.

Tabel V.1: Assignment operators pada Python

Operator	Contoh	Setara Dengan
=	x = 5	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5
//=	x //= 5	x = x // 5
**=	x **= 5	x = x ** 5

<code>&=</code>	<code>x &= 5</code>	<code>x = x & 5</code>
<code> =</code>	<code>x = 5</code>	<code>x = x 5</code>
<code>^=</code>	<code>x ^= 5</code>	<code>x = x ^ 5</code>
<code>>>=</code>	<code>x >>= 5</code>	<code>x = x >> 5</code>
<code><<=</code>	<code>x <<= 5</code>	<code>x = x << 5</code>

A.2. Operator Aritmatika

Bahasa C menyediakan lima operator aritmatika, yaitu :

* : untuk perkalian

/ : untuk pembagian

% : untuk sisa pembagian (modulo) ¹

+ : untuk pertambahan

- : untuk pengurangan

Perhatikan contoh program berikut:

```
>>> 2 * (3 + 4)
14
>>> 10 / 2
5.0
>>> 2 + 2
4
>>> 5 + 4 - 3
6
```

A.3. Operator Hubungan

Operator Hubungan digunakan untuk membandingkan hubungan antara dua buah operand (sebuah nilai atau variabel). Operator hubungan dalam bahasa C dapat dilihat pada Tabel V.2.

¹ operator % digunakan untuk mencari sisa pembagian antara dua bilangan. Misalnya : $9 \% 2 = 1$ atau $9 \% 3 = 0$.

Tabel V.2: Tabel Macam-Macam Operator Hubungan Dalam Bahasa C

Operator	Arti		Contoh
<	Kurang dari	$x < y$	Apakah x kurang dari y
<=	Kurang dari atau sama dengan	$x \leq y$	Apakah x kurang dari atau sama dengan y
>	Lebih dari	$x > y$	Apakah x lebih dari y
>=	Lebih dari atau sama dengan	$x \geq y$	Apakah x lebih dari atau sama dengan y
==	Sama dengan	$x == y$	Apakah x sama dengan y
!=	Tidak sama dengan	$x != y$	Apakah x tidak sama dengan y

Perhatikan contoh program berikut

```
>>> 5<2
False
>>> 4>2
True
>>> 4>=5
False
>>> 4==4
True
```

A.4. Operator Logika

Jika operator hubungan membandingkan hubungan antara dua buah operand, maka operator logika digunakan untuk membandingkan logika hasil dari operator-operator hubungan.

Operator logika ada tiga macam, yaitu :

and : Logika AND (DAN)

or : Logika OR (ATAU)

not : Logika NOT (INGKARAN)

Perhatikan contoh program berikut

```
x = True
y = False

# Output: x and y is False
```



```
print('x and y is',x and y)

# Output: x or y is True
print('x or y is',x or y)

# Output: not x is False
print('not x is',not x)
```

A.5. Operator Bitwise

Operator bitwise digunakan untuk memanipulasi bit-bit dari nilai data yang ada di memori.

Operator bitwise dalam bahasa Python :

Tabel V.3: Operator Bitwise Python

Operator	Kegunaan	Contoh
&	Bitwise AND	$x \& y = 0$ (0000 0000)
	Bitwise OR	$x y = 14$ (0000 1110)
~	Bitwise NOT	$\sim x = -11$ (1111 0101)
^	Bitwise XOR	$x \wedge y = 14$ (0000 1110)
>>	Bitwise right shift	$x \gg 2 = 2$ (0000 0010)
<<	Bitwise left shift	$x \ll 2 = 40$ (0010 1000)

BAGIAN VI STRUKTUR KONTROL DASAR

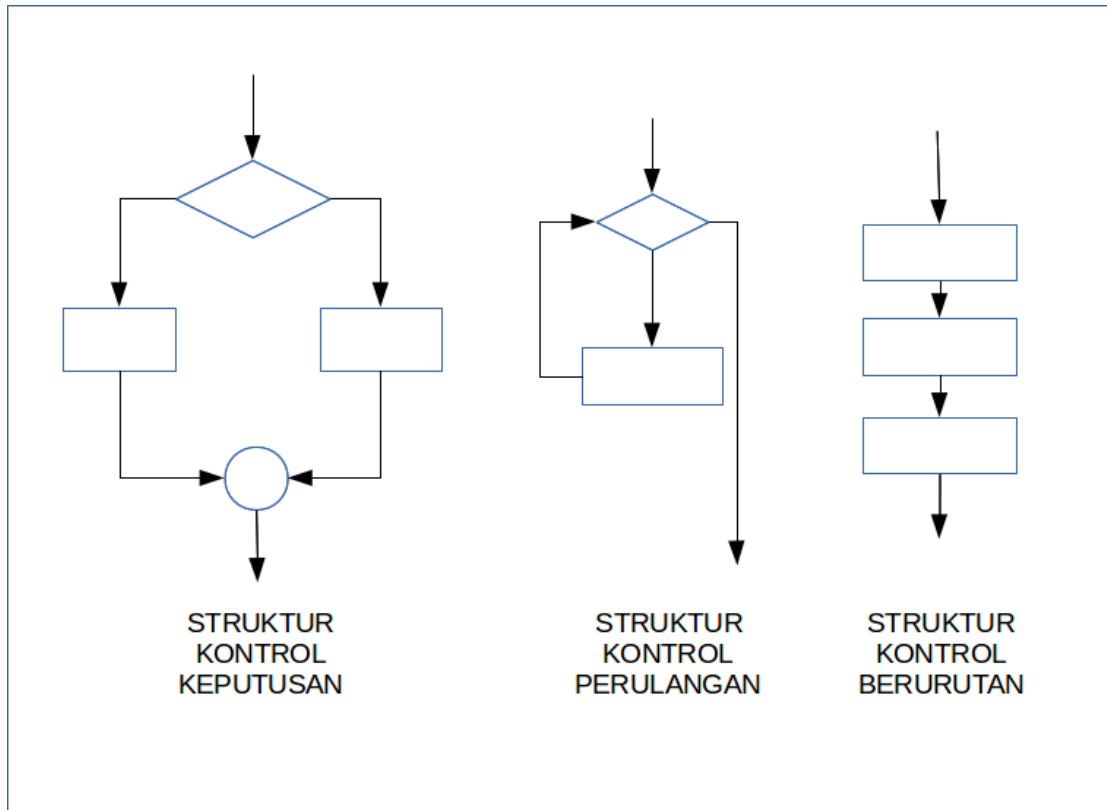
Capaian Pembelajaran

- 1. Mahasiswa mampu menjelaskan konsep Struktur kontrol dasar*
- 2. Mahasiswa mampu membuat program sederhana dengan menggunakan struktur kontrol dasar pada bahasa Python*

Struktur kontrol di dalam pemrograman adalah kumpulan perintah dengan bentuk tertentu yang digunakan untuk melakukan pegontrolan terhadap jalannya program. Pada dasarnya struktur kontrol terdapat beberapa macam diantaranya adalah:

1. Struktur kontrol berurutan (*Sequence Control Structures*)
2. Struktur kontrol keputusan/percabangan/kondisi (*Selection Control Structures*)
3. Struktur kontrol pengulangan (*Iteration Control Structures*)

Bila struktur kontrol digambarkan dalam bentuk flowchart maka dapat dilihat pada Gambar VI.1



Gambar VI.1: Macam-macam struktur kontrol

Struktur kontrol berurutan merupakan struktur kontrol yang paling sederhana dari sebuah program. Pada struktur kontrol berurutan program berjalan berdasarkan perintah-perintah berurutan yang diberikan oleh programmer dari awal kode sumber program hingga akhir kode sumber program. Pada umumnya contoh-contoh program yang telah dibahas pada bagian-bagian sebelumnya, dan memiliki pola sebagai berikut:

```
statement 1
statement 2
statement 3
```

Perhatikan contoh program berikut

```
a = 1;
b = 2;
c = 3;
```

```
d=a*c;  
printf(d);
```

Pada contoh program tersebut, program akan menjalankan perintah mulai dari penugasan/pemberian nilai pada variabel a kemudian memberi nilai pada variabel b, dilanjutkan pemberian nilai pada variabel c, dan diakhiri dengan pemberian nilai d yang merupakan hasil dari perkalian a dan c untuk kemudian ditampilkan di layar monitor.

Dalam suatu program dikenal pula istilah blok kode program yang merupakan kumpulan perintah yang diait oleh tanda kurung kurawal {}. Kumpulan perintah yang diapit oleh tanda tersebut akan dijalankan sesuai dengan fungsi dari blok tersebut. Python menggunakan indentasi (spasi putih di awal baris) untuk membatasi blok kode. Bahasa lain, seperti C, gunakan kurung kurawal untuk membatasi blok kode, namun dengan indentasi Python adalah wajib, program tidak akan bekerja tanpa itu.

BAGIAN VII STRUKTUR KONTROL PERCABANGAN

Capaian Pembelajaran

- 1. Mahasiswa mampu menjelaskan konsep Struktur kontrol percabangan*
- 2. Mahasiswa mampu membuat program sederhana dengan menggunakan struktur kontrol percabangan pada bahasa C*

Banyak istilah dari struktur kontrol percabangan. Pada literatur-literatur lain didapatkan istilah percabangan dengan nama seleksi, ada pula yang menyebutnya sebagai struktur kontrol keputusan. Namun dari semua istilah mengandung makna yang sama, sebab pada beberapa hal berubahnya nama tidaklah dapat mengubah makna dari sesuatu. Begitu pula dengan percabangan dari beberapa istilah yang telah disebutkan sebelumnya merujuk pada makna sebuah struktur program yang digunakan untuk mengarahkan perjalanan suatu proses.

Percabangan dapat diibaratkan sebagai katup atau kran yang mengatur jalannya air. Bila katup terbuka maka air akan mengalir dan sebaliknya bila katup tertutup air tidak akan mengalir atau akan mengalir melalui tempat lain. Fungsi penyeleksian kondisi penting artinya dalam penyusunan bahasa C, terutama untuk program yang kompleks.

A Struktur Kondisi If...

Struktur if dibentuk dari pernyataan if dan sering digunakan untuk menyeleksi suatu kondisi tunggal. Bila proses yang diseleksi terpenuhi atau bernilai benar, maka pernyataan yang ada di dalam blok if akan diproses dan dikerjakan. Bentuk umum struktur kondisi if adalah :

```
if expression:  
    statements
```

Perhatikan contoh program berikut:

```
if 10 > 5:  
    print("10 besar dari 5")  
  
print("Program Selesai")
```

Bentuk if sederhana ini dapat pula bersarang, misalnya seperti pada contoh berikut

```
num = 12
if num > 5:
    print("Bigger than 5")
    if num <=47:
        print("Between 5 and 47")
```

B Struktur Kondisi if... else...

Dalam struktur kondisi if....else minimal terdapat dua pernyataan. Jika kondisi yang diperiksa bernilai benar atau terpenuhi maka pernyataan pertama yang dilaksanakan dan jika kondisi yang diperiksa bernilai salah maka pernyataan yang kedua yang dilaksanakan. Bentuk umumnya adalah sebagai berikut :

```
if kondisi:
    statement jika benar
else:
    statement jika salah
```

Perhatikan contoh program berikut:

```
x = 4
if x == 5:
    print("Yes")
else:
    print("No")
```

If ... else ini dapat juga dibuat bersarang. Perhatikan contoh berikut

```
num = 7
if num == 5:
    print("Angkanya adalah5")
else:
    if num == 11:
        print("Angkanya adalah11")
    else:
        if num == 7:
            print("Angkanya adalah7")
        else:
            print("Number isn't 5, 11 or 7")
```

C Struktur Kondisi elif

Pernyataan elif (singkatan dari else ... if ...) adalah jika pernyataan ada 3 kondisi atau lebih. Serangkaian elif dapat memiliki blok akhir yang lain, yang disebut jika tidak ada ekspresi if atau elif yang Benar. Bentuk umum struktur kondisi elif adalah sebagai berikut

```
if kondisi:  
    pernyataan  
elif kondisi 2:  
    pernyataan  
else:  
    pernyataan
```

Perhatikan contoh kode program berikut:

```
num = 7  
if num == 5:  
    print("Angkanya adalah5")  
elif num == 11:  
    print("Angkanya adalah11")  
elif num == 7:  
    print("Angkanya adalah7")  
else:  
    print("Angkanya bukan 5, 11 atau 7")
```