



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231023
Nama Lengkap	Yehezkiel Darren Putra Wardoyo
Minggu ke / Materi	02 / Variable, Expression dan Statements

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2024

BAGIAN 1: MATERI MINGGU INI (40%)

Values dan type

Value adalah komponen utama dalam suatu program, dapat berbentuk suatu angka, huruf, bahkan simbol-simbol. Jenis-jenis *value* dapat dikelompokkan menjadi 3 jenis pada umumnya, yaitu: *integer* (1,2,3, dll) ,*strings* ("Hello World", "Adi", "#"),*float* (1.50,9.46, dll). Jenis-jenis *value* disebut juga sebagai *type*. Perlu diingat : untuk *value* bertipe *strings* harus di beri tanda petik (") diawal dan diakhir.

Dalam bahasa pemrograman Python, kita tidak perlu untuk menuliskan *type* dari *value* dari suatu variabel karena interpreter Python akan otomatis mengidentifikasi jenis *value* yang diberikan. Saya akan menampilkan beberapa contoh :

```
>>> a= 3
>>> type(a)
<class 'int'>
>>> b=3.5
>>> type(b)
<class 'float'>
>>> b="Hello World"
>>> type(b)
<class 'str'>
```

Dari contoh disamping dapat kita ketahui bahwa interpreter Python akan langsung mengidentifikasi jenis *value* yang dimasukkan ke dalam variabel tersebut.

Variable

Seperti contoh yang sudah saya beri diatas, "a" dan "b" yang saya gunakan dinamakan sebagai *variable*. Secara singkat, *variable* adalah lokasi yang digunakan untuk menyimpan nilai-nilai. Kita dapat memberi nama suatu *variable* secara bebas namun ada beberapa ketentuan yang harus diperhatikan dengan cermat. Di materi sebelumnya sudah membahas bagaimana cara kerja interpreter Python dalam menjalankan suatu program. Program yang dijalankan adalah berdasarkan baris-perbaris (awal-akhir).

```
>>> x = 3
>>> x
3
>>> x = 5
>>> x
5
```

Dari program disamping dapat disimpulkan bahwa cara kerja *variable* adalah sebagai tempat penyimpanan sementara. Di awal, kita memberi *value* pada x yaitu 3. Kemudian, kita memberi *value* 5 pada *variable* yang sama yaitu x. Maka python akan membaca *value* yang baru saja di input walaupun sebelumnya sudah di beri suatu *value* pada *variable* yang sama.

Nama Variable dan Keywords

Dalam memberi nama pada suatu *variable*, kita harus memperhatikan beberapa ketentuan:

1. Nama variabel boleh diawali dengan huruf atau garis bawah.
2. Karakter selanjutnya dapat berupa huruf, garis bawah, atau angka.
3. Karakter pada nama variabel bersifat *case-sensitive*. Artinya, huruf besar dan kecil dibaca oleh python sebagai sesuatu yang berbeda.
4. Nama variabel tidak boleh menggunakan kata kunci yang sudah ada dalam python.

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

Keyword python yang tidak boleh digunakan sebagai nama *variable*

```
>>> nama_saya = 'Darren'
>>> Nama_saya

Traceback (most recent call last):
  File "<stdin>", line 1, in <module>

NameError: name 'Nama_saya' is not defined. Did you mean:
'nama_saya'?
```

Variable bersifat *case-sensitive*

Statements

Statements adalah bagian dari code yang dapat dieksekusi oleh interpreter Python. Contoh dari statements adalah print. Kita dapat menggunakan perintah print untuk menampilkan data di terminal.

Operator dan Operand

Operator dan operand adalah dua hal yang berbeda. Operator adalah simbol yang digunakan untuk melakukan operasi matematika maupun operasi logika. Operand adalah nilai/*value* yang padanya dilakukan operasi. Misalnya $A > B$. Tanda ">" berperan sebagai

operator, sedangkan A dan B sebagai operand. Operator pada python memiliki simbol, antara lain :

Operator	Nama	Fungsi
+	Penjumlahan	Menjumlahkan 2 buah operand
-	Pengurangan	Mengurangkan 2 buah operand
/	Pembagian	Membagi 2 buah operand
*	Perkalian	Mengalikan dua buah operand
**	pemangkatan	Memangkatkan bilangan

Dan masih banyak lagi...

Expressions

Expression merupakan representasi dari nilai dan dapat terdiri dari gabungan antara *values*, *variable*, dan *operator*. *Values* dengan sendirinya dapat dianggap sebagai *expression* dan juga variabel. Secara umum, semuanya dapat disebut dengan *expression*.

Ketika kita menggunakan mode interactive, interpreter akan melakukan evaluasi dan langsung menampilkan hasilnya.

```
>>> 5 * 4
```

```
20
```

Urutan Operasi

Dalam operasi aritmatika terdapat suatu rule/aturan yang berlaku, yaitu PEMDAS. PEMDAS merupakan singkatan dari Paratheses, Exponentiation, Multiplication and Division, Operator. Urutan pengeksekusian suatu operasi matematika dimulai dari yang berada di dalam kurung → lalu, operasi pemangkatan/akar pangkat → lalu, perkalian/ pembagian → terakhir, penjumlahan/ pengurangan. Jika terdapat operasi yang memiliki tingkatan yang sejajar maka urutan pengeksekusian dimulai dari kiri ke kanan.

Operator Modulus dan String

Operator modulus adalah sisa hasil bagi dari bilangan pertama dengan bilangan kedua. Perlu diingat, operator ini hanya berlaku untuk tipe data integer. Simbol modulus dalam Python adalah tanda persen(%).

String adalah tipe data untuk kumpulan karakter (char). Operator + ketika bekerja dengan string tidak berarti penjumlahan secara matematika, melainkan penggabungan antar string contoh :

```
>>> first = 26
>>> second = 25
>>> print(first+second)
51
>>> first = '10'
>>> second = '50'
```

```
>>> print(first + second)
1050
```

Untuk operator perkalian (*) ketika bertemu dengan string maka yang terjadi adalah data string tersebut akan diiterasi sesuai dengan banyaknya yang diinginkan. Cara kerjanya adalah dengan data string akan dilakukan iterasi (pengulangan) sebanyak data integer :

```
>>> first = 'Ayah'
>>> second = 2
>>> print(first*second)
AyahAyahAyah
```

Menangani Input dari Pengguna

Alur sebuah program adalah Input-Proses-Output. Pengertian input sendiri adalah kegiatan memasukkan data yang dibutuhkan agar program bisa berjalan. Sedangkan proses adalah langkah-langkah yang ditempuh oleh program dalam memecahkan masalah. Output adalah hasil/keluaran yang didapat setelah program menyelesaikan proses.

Dalam Python sendiri ada syntax yang bisa digunakan sebagai input dan juga output. Untuk menginput data, kita bisa menggunakan *input()*. Sedangkan untuk menampilkan hasil proses, kita menggunakan *print()*. Perlu diingat, *input()* secara default hanya akan mau menerima hanya tipe data string selain itu program akan error. Kita bisa menambahkan tipe data yang ingin kita terima sebagai inputan dengan menambahkan tipe data tersebut di depan input [*float(input())*]. Contoh :

```
>>> x= input("Masukkan apa saja :")
Masukkan apa saja : 9
>>> x
' 9'
>>> x = float(input("Input here : "))
Input here : 9
>>> x
9.0
```

Dari program di samping menunjukan ketika kita hanya menuliskan input saja, maka program akan menerjemahkannya sebagai sebuah string (dapat dibuktikan dengan simbol tanda petik). Namun, program akan mengenali data tersebut sebagai float ketika kita menambahkan float sebelum input.

Komentar

Dalam Python, kita bisa menambahkan komen dalam sebuah program dengan mengawalinya dengan simbol tanda pagar (#). Fungsi komentar digunakan oleh programmer untuk mempermudah maksud dari kode yang dibuat.

Kegiatan Praktikum

Membuat variabel

Saya membuat suatu program kalkulator sederhana (perkalian, pembagian, pengurangan, penambahan).

```
>>> # Program Kalkulator Sederhana
>>> #membuat variable
>>> angka1 = float(input("Input First Number Here : "))
Input First Number Here : 5.25
>>> angka2= float(input("Input Second Number Here : "))
Input Second Number Here : 4
>>> Kali = angka1*angka2
>>> bagi =angka1/angka2
>>> tambah = angka1+angka2
>>> kurang = angka1-angka2
>>> print(Kali, bagi, tambah, kurang)
Output : 21.0 1.3125 9.25 1.25
```

Memberikan nilai pada variabel

Saya akan mencoba membuat sebuah program menghitung total harga yang dibeli dengan memasukkan harga barang dan juga jumlah barang yang dibeli.

Langkah 1 : Inisiasi Persoalan

Variabel :

kode, nama, harga_barang, jumlah_barang, total_harga.

Proses :

total_harga = harga_barang * jumlah_barang

Langkah 2 : menetapkan tipe data

kode dan nama bertipe data *string*.

harga_barang, total_harga bertipe data *float*

jumlah_barang bertipe data *integer*.

Langkah 3 : kode program

```

#program kasir
print("PROGRAM KASIR\n\n")
#variable
kode = input("Masukkan Kode Barang : ")
nama = input("Masukkan Nama Barang : ")
harga_barang = float(input("Harga %s : " %nama))
jumlah_barang = int(input("Jumlah Barang Yang Dibeli : "))
#Proses
total_harga = harga_barang*jumlah_barang

#output
print("Kode Barang : \n%s\n\nNama Barang : \n%s\n\nHarga Satuan : %f\nJumlah Barang : %d\n\nHarga Akhir : %f" %(kode,nama,harga_barang,jumlah_barang,total_harga))

```

Output Program :

```

PROGRAM KASIR

Masukkan Kode Barang : 71231023
Masukkan Nama Barang : Sabun Mama Mia
Harga Sabun Mama Mia : 19880
Jumlah Barang Yang Dibeli : 2
Kode Barang :
71231023

Nama Barang :
Sabun Mama Mia

Harga Satuan : 19880.000000
Jumlah Barang : 2

Harga Akhir : 39760.000000

```

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

Kode Program :

```
print("Program Body Mass Index")
your_mass= float(input("Input Your Mass in Kg :"))
your_height= float(input("Input Your Height in Meter :"))

def BMI(x,y):
    return x/(y**2)

final = float(BMI(your_mass,your_height))
print("Your Mass : %.2f Kg\nYour Height : %.2f Meter\nBMI is : %.2f"%(your_mass,your_height,final))
```

Output :

```
Program Body Mass Index
Your Mass : 55.00 Kg
Your Height : 1.75 Meter
BMI is : 17.96
```

Penjelasan :

1. Langkah pertama, buat variabel untuk menampung massa dan tinggi badan. `your_mass` untuk menampung berat badan dan `your_height` untuk menampung tinggi badan. Lalu, buatlah variabel untuk menghitung *Body Mass Index*, yaitu BMI.
2. Selanjutnya, tentukan tipe data untuk variabel. `your_mass`, `your_height`, dan BMI bertipe data *float*.
3. Setelah itu, definisikan variable BMI sebagai `your_mass/(your_height*your_height)`.
4. Terakhir, tutup program dengan menampilkan hasil program.

SOAL 2

Kode Program :

```
x = int(input("Input x : "))

formula = 2*(x**3)+ 2*x +15/x

print("x :",x)
print("The Result is",formula)
```

Output :


```
x : 10
The Result is 2021.5
```

Penjelasan:

1. Pertama, buat variabel. Umpamakan x sebagai suatu variabel yang ingin kita gunakan sebagai tempat menyimpan nilai.
2. Selanjutnya, buatlah variabel formula dan definisikannya sebagai $2*(x**3)+2*x+(15/x)$
3. Terakhir, akhiri program dengan menampilkan hasil akhir program.

SOAL 3

Kode Program :

```
print("Program Menghitung Pendapatan Budi Selama Liburan Musim Panas")
print("="*80)
#input
salary=int(input("Masukkan gaji per jam Budi : "))
work_hours=float(input("Masukkan jam kerja Budi dalam seminggu : "))

#Function Pendapatan Setelah kena Pajak
def taxSalary(x):
    return x*(1-0.14)

#Function Pendapatan Sebelum kena Pajak
def noTax(x,y,z):
    return x*(y*z)

#Waktu yang diperlukan
num_Weeks=5

#Pendapatan BELUM KENA PAJAK
gross= noTax(salary,work_hours,num_Weeks)

#Pendapatan KENA PAJAK
netIncome= taxSalary(gross)

#Pengeluaran Pendapatan
Accesories= netIncome*0.1 #Pengeluaran untuk baju dan aksesoris
stationery= netIncome*0.01 #Pengeluaran untuk alat tulis
```

```

#Sisa pendapatan pertama
ResidualSalary= netIncome-(Accesories+stationery)

#Sedekah
Charity = ResidualSalary*0.25
poorPeople = 0
YteamPeople =0
#While Loops
i =ResidualSalary*0.25
while i>= 1000:
    poorPeople+=700
    YteamPeople+=300
    i-=1000
Charity-=i

#Print
print("Pendapatan Budi selama 5 minggu dengan :\nGaji Rp.%d dan Total Jam Kerja dalam Seminggu %d Jam " %(salary,work_hours))
print("1. BELUM KENA PAJAK : Rp.%.2f" %gross)
print("2. KENA PAJAK : Rp.%.2f" %netIncome)
print("Pengeluaran Budi :")
print("1. Baju dan Aksesoris : Rp.%.2f" %Accesories)
print("2. Alat Tulis : Rp.%.2f" %stationery)
print("3. Sedekah : Rp.%.2f" %Charity)
print("\t3a. Sedekah untuk Yatim : Rp.%.2f" %poorPeople)
print("\t3b. Sedekah untuk Dhuafa : Rp.%.2f" %YteamPeople)

```

Output :

```

Program Menghitung Pendapatan Budi Selama Liburan Musim Panas
=====
Pendapatan Budi selama 5 minggu dengan :
Gaji Rp.150000 dan Total Jam Kerja dalam Seminggu 15 Jam
1. BELUM KENA PAJAK : Rp.11250000.00
2. KENA PAJAK : Rp.9675000.00
Pengeluaran Budi :
1. Baju dan Aksesoris : Rp.967500.00
2. Alat Tulis : Rp.96750.00
3. Sedekah : RP.2152687.50
    3a. Sedekah untuk Yatim : Rp.1506400.00
    3b. Sedekah untuk Dhuafa : Rp.645600.00

```

Penjelasan:

1. Pertama, kita memasukkan gaji per jam yang Budi terima. Misal, gaji per jam Budi adalah 150.000. Lalu, kita memasukkan total jam kerja Budi dalam seminggu, misalkan dalam seminggu Budi bekerja selama 15 jam. Catatan : Budi bekerja selama 5 minggu.
2. Kedua, program akan mulai menghitung gaji kotor Budi yaitu gaji*(waktu*5). Lalu, menghitung gaji bersih (dipotong PPh) yaitu gaji kotor *(1-0.14).
3. Ketiga, Budi menggunakan gaji bersihnya untuk beberapa keperluan. Pertama, Budi ingin membeli baju dan aksesoris dengan harga 10% dari gaji bersihnya. Kedua, Budi membeli juga alat tulis seharga 1% dari gaji bersihnya. Lalu, untuk menghitung sisa uang budi bisa

dengan menghitung selisih gaji bersih dengan harga baju dan aksesoris ditambah harga alat tulis.

4. Keempat, Budi ingin menggunakan 25% dari sisa gaji bersih untuk sedekah. Dari uang yang di sedekahkannya, setiap 1000, 30% nya untuk yatim dan sisanya (70%) untuk dhuafa. Struktur kontrol untuk case ini yaitu kita akan mengecek apakah uang sedekah lebih dari 1000. Jika ya, maka uang sedekah akan berkurang 1000 dan uang untuk yatim bertambah 300 dan uang untuk dhuafa bertambah 700. Jika tidak, maka berhenti.
5. Terakhir, untuk mengakhiri program maka tampilkan hasil gaji kotor Budi, gaji bersih Budi, harga baju dan aksesoris, harga alat tulis, uang sedekah, uang sedekah untuk yatim, uang sedekah untuk dhuafa.

Repository Link :

https://github.com/YehezkielDarren/PrAIPro_Minggu2.git