



Laporan Praktikum Algoritma & Pemrograman

Semester Genap 2025/2026

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.

NIM	71251213
Nama Lengkap	Maryo Aurel Nubatonis
Minggu ke / Materi	02 / Variable Expression dan Statements

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2026**

BAGIAN 1: MATERI MINGGU INI (40%)

Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

MATERI 1

Values dan type

Value adalah komponen utama yang programnya, seoerti huruf atau angka. Value juga biasanya yang kita kenal 1,2,'a','z', dan "Hello Word". Value juga dapat dibagi menjadi beberapa bagian tipe yang beda, misalnya 2 untuk nilai integer dan "Hello Word" untuk sebuah nilai *string*. Karena penulisannya ditutup menggunakan tanda petik ('').

Untuk memehami values dan type, kita akan mencoba beberapa baris code berikut ini menggunakan python interactive mode seperti yang ditunjukan pada code dibawah ini.

```
>>> print(4)
4
>>> print(10.876)
10.876
>>> print('Z')
Z
>>> print('True')
True
>>> print('False')
False
```

Perintah print juga bekerja untuk value selain *string*, seperti integer (bilangan bulat), *float* (bilangan pecahan), character atau huruf, dan *bool* (benar atau salah). Untuk mencoba kita dapat menggunakan Perintah python untuk jalankan interpreter.

Setiap value yang memiliki type untuk mengatahui tipe data tersebut. Python menyediakan fungsus built-in melakukan pengecekan tipe data pada value, menggunakan fungsi *type()*

```
>>> x=5
>>> print (x, "tipenya adalah ", type (x))
5 tipenya adalah <class 'int'>
>>> x=2.0
>>> print (x, "tipenya adalah ", type(x))
2.0 tipenya adalah <class 'float'>
>>> x=1+2j
>>> print (x, "tipenya adalah ", type(x))
(1+2j) tipenya adalah <class 'complex'>
```

Ketika menggunakan bilangan bulat besar, beberapa model penulisan menggunakan tanda koma (,) diantara kelompok tiga digit. Contohnya pada penulisan 1.000.000. Dalam python, akan dianggap sebagai bilangan bulat seperti dibawah ini.

```
>>> print (1,000,000)
1 0 0
```

Hal ini terjadi karena python menganggap bahwa 1,000,000 merupakan kiriman parameter yang sebanyak 3 parameter pada fungsi print, yaitu 1,0, dan 0.

MATERI 2

Variabel

Variabel adalah salah satu fitur powerfull dalam bahasa pemrograman dan kemampuannya dapat melakukan manipulasi *variable*. *Variabel* juga merupakan Lokasi memori yang dicadangkan untuk menyimpan nilai-nilai. Ini juga berarti bahwa ketika Anda membuat sebuah *variable* Anda dapat memesan beberapa ruang dimemori. *Variabel* menyimpan data yang dilakukan selama proses program dieksekusi, yang nantinya isi dari

variable tersebut yang dapat diubah kapanpun oleh operasi – operasi tertentu pada program yang menggunakan variable seperti dibawah ini.

```
>>> pesan = 'selamat malam guys, mari kita belajar python bersama'  
>>> n = 17  
>>> pi = 3.1415926535897931
```

Variabel merupakan suatu elemen dalam pemrograman yang digunakan untuk menyimpan berbagai macam tipe data. Dalam bahasa pemrograman Python, variabel memiliki sifat dinamis, yaitu variabel tidak perlu dideklarasikan dengan tipe data tertentu terlebih dahulu. Selain itu, nilai yang tersimpan dalam variabel Python dapat diubah saat program dijalankan.

Potongan kode di atas merupakan contoh penggunaan variabel dalam Python. Contoh pertama adalah variabel pesan yang menyimpan data bertipe *string*. Contoh kedua adalah variabel n yang menyimpan nilai bertipe *integer*, yaitu angka 17. Contoh ketiga adalah variabel pi yang menyimpan nilai konstanta π (phi) dengan tipe data *float*.

Untuk menampilkan nilai yang tersimpan di dalam suatu variabel, Python menyediakan perintah `print()`, yang berfungsi untuk menampilkan output ke layar.

```
>>> print (n)  
17  
>>> print(ipk)  
3.29
```

MATERI 3

Nama Variabel dan Keywords

Pemberian nama pada variabel mengacu pada panduan berikut ini.

1. Nama variable boleh diawali menggunakan huruf atau garis bawah (_), contoh:
`nama, _nama,`

`namaKu, nama_variable.`

2. Karakter selanjutnya dapat berupa huruf, garis bawah (_) atau angka, contoh:
`_nama, n2,`

nilai1.

3. Karakter pada nama variable bersifat sensitif (case-sensitif). Artinya huruf besar dan kecil

dibedakan. Misalnya, variabel_Ku dan variabel_ku, keduanya adalah variabel yang berbeda.

4. Nama variabel tidak boleh menggunakan kata kunci yang sudah ada dalam python seperti if,

while, for, dsb

Python sendiri memiliki 35 keyword yang tidak boleh digunakan untuk memberi nama variable.

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

Berikut ini adalah contoh penggunaan variable dalam bahasa pemrograman Python.

```
nama = "Aryo Nubatonis"

print(nama)

umur = 18
print(umur)
type(umur)
umur = "sembilan belas"
print(umur)
type(umur)
namaDepan = "Aryo"
namaBelakang = "Nubatonis"
nama = namaDepan + " " + namaBelakang
umur = 20
hobi = "Futsal"
print("Biodata\n", nama, "\n", umur, "\n", hobi)

inivariabel = "Halo"
ini_juga_variabel = "Hai"
_inivariabeljuga = "Hi"
inivariabel222 = "Bye"
panjang = 10
lebar = 5
luas = panjang * lebar
print(luas)
```

MATERI 3

Statements

Statements merupakan instruksi dalam bahasa pemrograman Python yang dapat dieksekusi oleh interpreter. Contoh statement antara lain print, expression statements, dan assignment statements.

Pada penggunaan Python dalam mode interaktif, setiap statement yang dituliskan akan langsung dieksekusi dan hasilnya ditampilkan secara otomatis. Hal ini berbeda dengan penggunaan Python dalam mode script, di mana program dituliskan dalam sebuah berkas dan berisi kumpulan statement yang saling berhubungan serta dieksekusi secara berurutan (sekuensial).

```

week2.py > ...
1  print(1)
2  x=2
3  print(x)
y=3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\LEGION\OneDrive\Documents\aryo kuliah semester 1> python -u "c:\Users\LEGION\OneDrive\Documents\aryo kuliah semester 1\week2.py"
1
2
3
PS C:\Users\LEGION\OneDrive\Documents\aryo kuliah semester 1>

```

Contoh statement dan ouputnya.

MATERI 3

Operator dan Operand

Operator merupakan simbol tertentu yang digunakan untuk melakukan operasi aritmatika maupun logika. Nilai yang dikenai suatu operasi disebut sebagai *operand*. Sebagai contoh pada operasi $2 + 3$, tanda $+$ berperan sebagai operator penjumlahan, sedangkan angka 2 dan 3 merupakan operand.

Pada bagian ini dibahas secara khusus operator aritmatika dalam bahasa pemrograman Python. Operator aritmatika adalah operator yang digunakan untuk melakukan operasi matematika, seperti penjumlahan, pengurangan, perkalian, pembagian, dan operasi matematika lainnya. Jenis-jenis operator aritmatika tersebut ditunjukkan pada tabel berikut.

Operator pada python

Operator	Nama dan Fungsi	Contoh
$+$	Penjumlahan, menjumlahkan 2 buah operand	$X + y$
$-$	Pengurangan, mengurangkan 2 buah operand	$X - y$
$*$	Perkalian, mengalikan 2 buah operand	$X * y$
$/$	Pembagian, membagi 2 buah operand	X / y
$**$	Pemangkatan, memangkatkan bilangan	$X ** y$

Beberapa contoh penggunaan operator aritmatika pada Python dapat dilihat pada potongan kode berikut ini.

```
>>> 32 + 30
62
>>> hour = 5
>>> print (hour - 1)
4
>>> minute = 60
>>> print (hour*6+minute)
90
>>> print (minute/60)
1.0
>>> 5**2
25
>>> (5+9)*(16-7)
126
```

MATERI 3

Expressions

Expression merupakan representasi dari suatu nilai yang dapat tersusun atas kombinasi *value*, *variabel*, dan *operator*. Sebuah *value* maupun variabel secara mandiri juga dapat dianggap sebagai sebuah *expression*. Dengan demikian, secara umum setiap bentuk yang menghasilkan nilai dalam Python dapat disebut sebagai *expression*.

17

X

X + 17

Pada penggunaan expression dalam mode interaktif, *interpreter* Python akan melakukan evaluasi terhadap expression tersebut dan secara langsung menampilkan hasil evaluasinya.

```
>>> 1 + 1
2
>>> 3 + 2
5
```

MATERI 4

Urutan Operasi

Urutan operasi berlaku ketika terdapat lebih dari satu operator dalam sebuah expression. Urutan tersebut ditentukan oleh aturan prioritas operator. Untuk operasi matematika, Python mengikuti konvensi matematika standar. Aturan prioritas ini umumnya dikenal dengan singkatan **PEMDAS**, yaitu *Parentheses* (tanda kurung), *Exponentiation* (pangkat), *Multiplication* dan *Division* (perkalian dan pembagian), serta *Addition* dan *Subtraction* (penjumlahan dan pengurangan).

- Parentheses (tanda kurung) merupakan prioritas operasi tertinggi dan digunakan untuk menentukan urutan evaluasi expression secara eksplisit. Sebagai contoh, expression $2 * (3 - 1)$ menghasilkan nilai 4, sedangkan $(1 + 1) ** (5 - 2)$ menghasilkan nilai 8. Penggunaan tanda kurung juga membantu meningkatkan keterbacaan expression, misalnya $(\text{minute} * 100) / 60$, yang dituliskan secara ringkas namun tetap jelas.
- Exponentiation (pemangkatan) merupakan prioritas operasi setelah tanda kurung. Sebagai contoh, expression $2**1 + 1$ menghasilkan nilai 3, bukan 4, dan $3 * 1**3$ menghasilkan nilai 3, bukan 27, karena operasi pemangkatan dievaluasi terlebih dahulu sebelum operasi aritmatika lainnya.
- Multiplication and Division (perkalian dan pembagian) memiliki prioritas yang sama dan lebih tinggi dibandingkan penjumlahan dan pengurangan. Sementara itu, addition dan subtraction (penjumlahan dan pengurangan) juga memiliki prioritas yang sama. Sebagai contoh, expression $2 * 3 - 1$ menghasilkan nilai 5, bukan 4, dan $6 + 4 / 2$ menghasilkan nilai 8, bukan 5, karena operasi perkalian dan pembagian dievaluasi terlebih dahulu.
- Operators - operator memiliki prioritas yang sama, dibaca dari kiri ke kanan. Contoh $5-3-1$ hasilnya 1 bukan 3 karena operasi pengurangan 5-3 terlebih dahulu baru kemudian hasilnya dikurangi dengan 1.

Jika terjadi keraguan, silakan letakkan tanda kurung di dalam ekspresi untuk memastikan bahwa komposisinya sesuai dengan yang diinginkan

MATERI 5

Operator Modulus dan String

Modulus

Operator Modulus adalah hasil bagi dari bilangan pertama dengan bilangan kedua. Operator ini hanya berlaku pada tipe data integer. Dalam Python, operator modulus diambangkan dengan tanda persen (%).

```
>>> quotient = 7 // 3
>>> print (quotient)
2
>>> oprmomulus = 7 % 3
>>> print (oprmomulus)
1
```

7 dibagi dengan 3 menghasilkan 2 dengan sisa hasil bagi 1.

Operator modulus (%) digunakan untuk:

- Mengecek apakah suatu bilangan habis dibagi bilangan lain (jika $x \% y == 0$).
- Mengambil digit terakhir atau beberapa digit dari suatu bilangan, misalnya $x \% 10$ untuk digit terakhir dan $x \% 100$ untuk dua digit terakhir.

String

Operator + ketika bekerja dengan string, dan tidak berarti penjumlahan secara matematika, melainkan penggabungan antar string.

Contoh :

```
>>> first = 10
>>> second = 15
>>> print (first + second)
25
>>> first = '100'
>>> second = '150'
>>> print (first + second)
100150
```

Operator * juga bekerja dengan string dengan melakukan perkalian antara content string dan integer.

```
>>> first = 'Test '
>>> second = 5
>>> print (first * second)
Test Test Test Test Test
```

MATERI 6

Menangani Input dari Pengguna

MATERI 7

Komentar

Tanda pagar (#) digunakan untuk menulis komentar di Python. Komentar tidak dieksekusi oleh interpreter dan hanya berfungsi sebagai penjelasan kode agar mudah dipahami programmer.

Inputnya

```
#komentar pertama
print ("hai, my name is Aryo") #komentar kedua
```

Outputnya

```
hai, my name is Aryo
```

Python juga tidak memiliki fitur komentar multibaris. Jadi kita harus mengomentari satu persatu baris seperti dibawah ini.

```
# ini komentar
# ini juga adalah komentar
# ini juga masih komentar
```

Kegiatan praktikum

Kegiatan praktikum dibagi menjadi 4 yaitu :

1. Membuat variabel.

2. Memberikan nilai dalam variabel.
3. Mencetak nilai dalam variabel.
4. Separator, tipe data dan fungsi type

MATERI 8

Membuat Variabel

Misalnya kita membuat sebuah data pribadi yang berisi nama, alamat, umur, tempat lahir, tanggal lahir, dan indeks prestasi kumulatif akan memberikan 6 (enam) buah variable dengan tipe datanya.

```
Nama = input("Masukkan nama Anda: ")
Alamat = input("Masukkan alamat Anda: ")
Umur = input("Masukkan umur Anda: ")
TL = input("Masukkan tempat lahir Anda: ")
Tgl = input("Masukkan tanggal lahir Anda: ")
IPK = input("Masukkan IPK Anda: ")
print("====")
print("DATA AKADEMIK")
print("Nama: ", Nama)
print("Alamat: ", Alamat)
print("Umur: ", Umur)
print("Tempat Lahir: ", TL)
print("Tanggal Lahir: ", Tgl)
print("IPK: ", IPK)
```

Ouputnya

```
Masukkan umur Anda: 18
Masukkan tempat lahir Anda: KUPANG
Masukkan tanggal lahir Anda: 23
Masukkan IPK Anda: 3.0
=====
DATA AKADEMIK
Nama: Aryo Nubatonis
Alamat: KLITREN
Umur: 18
Tempat Lahir: KUPANG
Tanggal Lahir: 23
IPK: 3.0
```

Bagaimana dengan tipe data lainnya, kita akan mencoba dengan program yang menggunakan tipe data bilangan baik integer maupun float.

```
x1 = eval(input("x1: "))
x2 = eval(input("x2: "))
x3 = eval(input("x3: "))
x4 = eval(input("x4: "))

jumlah = x1 + x2 + x3 + x4
kali = x1 * x2 * x3 * x4

print ('hasil semua bilangan =', jumlah)
print ('hasil kali semua bilangan =', kali)
jumlah = jumlah + 0.5

print ('jika ditambah 0.5 maka hasilnya =', jumlah)
kali = kali * 0.5

print ('jika dikali 0.5 maka hasilnya =', kali)
```

Ouputnya

```
x1: 7
x2: 4
x3: 8
x4: 2
hasil semua bilangan = 21
hasil kali semua bilangan = 448
jika ditambah 0.5 maka hasilnya = 21.5
jika dikali 0.5 maka hasilnya = 224.0
```

MATERI 9

Memberikan nilai dalam variable

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 = 0
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
jum_brg bertipe data integer
harga_satuan, harga_beli, total_hrg_brg bertipe data float.

Langkah 3 : Kode Program

```

total_hrg_brg= 0.0

kd_brg=input("Kode barang = ")
nama_brg=input("Nama barang = ")

harga_satuan=eval(input("Harga satuan barang =Rp. "))

jumlah_barang=eval(input("Jumlah barang yang dibeli = "))
harga_beli = harga_satuan * jumlah_barang

total_hrg_brg= harga_beli + total_hrg_brg
print("Total harga yang dibayar Rp",total_hrg_brg)

```

Outputnya

```

Kode barang = 23
Nama barang = Mouse
Harga satuan barang =Rp. 20.000
Jumlah barang yang dibeli = 7
Total harga yang dibayar Rp 140.0

```

Mencetak nilai dalam variable

```

x = 50
print(x)
type(x)

y = 5 * int(x)
print(y)

print ("Tipe data X dikonversi ke int agar dapat dihitung Y=5*X")

z = y + float(x)
print("z", z)

print ("Tipe data X dikonversi ke float untuk dijumlahkan dengan tipe data Y, "
"hasilnya yang ditampung bertipe float juga")

P = True
L = False

P != L
print ("Tanda != artinya tidak sama dengan")

P == L
print ("Tanda == artinya sama dengan")

```

MATERI 10

Separator, tipe data dan fungsi type

Konversi type data pada pemrograman python gunakan fungsi berikut :

1. str() = Untuk konversi type data ke String
2. int() = Untuk konversi type data ke Integer
3. float() = Untuk konversi type data ke Float

Ada dua macam variasi print :

1. Jika ada simbol, gunakan kutip dua atau gunakan backslash (\) sebelum menuliskan simbol.
2. Dipisahkan dengan tanda koma.
3. Diganti dengan :
 - %d : mewakili integer
 - %f : mewakili float
 - Untuk membuat n angka di belakang koma, gunakan %.nf
 - Misal untuk dua angka di belakang koma, berarti gunakan %.2f
 - %s : mewakili string

```
Kode = 'ANS23'
NamaBarang = 'Karbulator'
HargaSatuan = 500000
Stock = 10
print('Kode barang %s \n Nama Barang= %s \n Harga Satuan=Rp %d \n Stok Barang=%d' %
(Kode,NamaBarang,HargaSatuan,Stock))

HargaBarang = float(HargaSatuan)
print('Harga Satuan Barang= Rp %.2f' % (HargaBarang))
```

Outputnya

```
Kode barang ANS23
Nama Barang= Karbulator
Harga Satuan=Rp 500000
Stok Barang=10
Harga Satuan Barang= Rp 500000.00
```

BAGIAN 2: LATIHAN MANDIRI (60%)

Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

Link GitUP : <https://github.com/maryonubatonis1-web/71251213---Maryo-Aurel-Nubatonis>

SOAL 1

```
tinggi = float(input("Masukkan tinggi badan (meter): "))
bmi = float(input("Masukkan BMI yang diharapkan: "))

berat = bmi * (tinggi ** 2)

print("Berat badan yang diperlukan:", berat, "kg")
```

```
Masukkan tinggi badan (meter): 3
Masukkan BMI yang diharapkan: 5
Berat badan yang diperlukan: 45.0 kg
```

Penjelasan:

Program tersebut meminta pengguna memasukkan tinggi badan dalam satuan meter dan nilai BMI yang diharapkan, kemudian menghitung berat badan yang diperlukan menggunakan rumus $\text{berat} = \text{BMI} \times \text{tinggi}^2$. Hasil perhitungan tersebut selanjutnya ditampilkan ke layar dalam satuan kilogram sebagai output dari program.

SOAL 2

```
x = int(input("Masukkan nilai x: "))

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

print("Hasil f(x) =", f)
```

```
Masukkan nilai x: 2
Hasil f(x) = 27.5
```

Penjelasan:

Program tersebut meminta pengguna memasukkan nilai x sebagai bilangan bulat, kemudian menghitung nilai fungsi $f(x) = 2x^3 + 2x + 15/x$ berdasarkan nilai x yang diberikan. Setelah perhitungan dilakukan, hasil dari $f(x)$ ditampilkan ke layar sebagai output program.

SOAL 3

```
gaji_per_jam = float(input("Masukkan gaji per jam: "))
jam_per_minggu = int(input("Masukkan jumlah jam kerja per minggu: "))

MINGGU = 5
PAJAK = 0.14
PAKAIAN = 0.10
ALAT_TULIS = 0.01
SEDEKAH = 0.25
ANAK_YATIM = 0.30
DHUAFA = 0.70

pendapatan_kotor = gaji_per_jam * jam_per_minggu * MINGGU

pendapatan_bersih = pendapatan_kotor * (1 - PAJAK)

uang_pakaian = pendapatan_bersih * PAKAIAN
uang_alat_tulis = pendapatan_bersih * ALAT_TULIS

sisa_uang = pendapatan_bersih - (uang_pakaian + uang_alat_tulis)

uang_sedekah = sisa_uang * SEDEKAH
uang_anak_yatim = uang_sedekah * ANAK_YATIM
uang_dhuafa = uang_sedekah * DHUAFA

print("Pendapatan sebelum pajak:", pendapatan_kotor)
print("Pendapatan setelah pajak:", pendapatan_bersih)
print("Uang untuk pakaian & aksesoris:", uang_pakaian)
print("Uang untuk alat tulis:", uang_alat_tulis)
print("Uang yang disedekahkan:", uang_sedekah)
print("Uang untuk anak yatim:", uang_anak_yatim)
print("Uang untuk kaum dhuafa:", uang_dhuafa)
```

```
Masukkan gaji per jam: 2000000
Masukkan jumlah jam kerja per minggu: 33600000
Pendapatan sebelum pajak: 336000000000000.0
Pendapatan setelah pajak: 288960000000000.0
Uang untuk pakaian & aksesoris: 288960000000000.0
Uang untuk alat tulis: 288960000000000.0
Uang yang disedekahkan: 642936000000000.0
Uang untuk anak yatim: 192880800000000.0
Uang untuk kaum dhuafa: 450055200000000.0
```

Penjelasan:

Program ini menerima input gaji per jam dan jumlah jam kerja per minggu, kemudian menghitung pendapatan Budi selama 5 minggu sebagai pendapatan kotor. Selanjutnya program menghitung pendapatan bersih setelah dipotong pajak sebesar 14%. Dari pendapatan bersih tersebut, dihitung pengeluaran untuk pakaian dan aksesoris sebesar 10% serta alat tulis sebesar 1%, lalu menentukan sisa uang yang dimiliki. Dari sisa uang tersebut,

25% disedekahkan, kemudian uang sedekah dibagi menjadi 30% untuk anak yatim dan 70% untuk kaum dhuafa, dan seluruh hasil perhitungan ditampilkan sebagai output.