



---

# 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	71251214
Nama Lengkap	Raffiel Gideon Nyolo Nyolo
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%)

### Values dan type

Value merupakan komponen utama dari program, seperti huruf atau angka. *Value* yang sering kita lihat biasanya adalah 1,2,"a","b", dan "Hello World". Value dibagi menjadi beberapa tipe, misalnya 1 untuk sebuah nilai integer dan "Hello Word" untuk sebuah nilai *string*. Interpreter dapat melakukan identifikasi terhadap *string* karena pada penulisannya harus dibuka dan ditutup menggunakan tanda petik (*quotation mark*).

Contoh untuk *value integer*:

```
>>> print(4)
4
>>> print(10.876)
10.876
```

Gambar 1.1

Contoh untuk *value string*:

```
>>> print('True')
True
>>> print('False')
False
```

Gambar 1.2

Perintah `print` juga bekerja untuk *value* selain *string*, seperti *integer* (bilangan bulat), *float* (bilangan pecahan), *character* (huruf), atau *bool* (benar/salah). Setiap *value* yang pasti memiliki *type* untuk mengetahui tipe data tersebut. Python menyediakan fungsi built-in untuk melakukan pengecekan tipe data pada *value* dengan menggunakan fungsi `type()`.

Contoh:

```
>>> 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">
```

Gambar 1.3

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.

Contohnya: 

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

Gambar 1.4

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

## Variabel

Salah satu fitur dalam bahasa pemrograman adalah melakukan manipulasi variabel. Variabel merupakan lokasi memori yang dicadangkan untuk menyimpan nilai. Ini berarti bahwa ketika membuat sebuah variable, itu sama seperti memesan beberapa ruang di memori. Variabel menyimpan data yang selama program dijalankan, yang nantinya isi dari variabel tersebut dapat diubah oleh operasi-operasi tertentu pada program yang menggunakan variabel.

Contoh: 

```
>>> pesan = 'selamat pagi, mari belajar python'
>>> n = 17
>>> pi = 3.1415926535897931
```

Gambar 1.5

Contoh pertama adalah variabel 'pesan' yang berisi *string*, contoh kedua adalah variabel 'n' yang berisi integer 17 dan contoh ketiga merupakan nilai dari pi ( $\pi$ ).

Untuk menampilkan nilai dari variabel, dapat digunakan perintah print.

Contoh: 

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

Gambar 2.1

Di dalam pemrograman Python, variabel mempunyai sifat yang dinamis, artinya variabel Python tidak perlu dideklarasikan tipe data tertentu dan variabel Python dapat diubah saat program dijalankan.

## 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. 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 variabel antara lain: and, del, from, None, True, as, elif, global, nonlocaly, 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 dan wait.

Contoh penggunaan variabel (Python):

```
#nilai dan tipe data dalam variabel dapat diubah
umur = 20
print(umur)
type(umur)
umur = "dua puluh satu" #nilai setelah diubah
print(umur) #mencetak nilai umur
type(umur) #mengecek tipe data umur
```

Gambar 2.2

## Statements

*Statements* merupakan bagian dari code interpreter Python yang dapat dieksekusi. Misalnya pada *statement print*, dapat berupa *expression statements* dan *assignment*.

```
1 print(1)
2 x=2
3 print(x)
```

---

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

```
192:~ macintosh$ /Library/Frameworks/Python.framework/Versions/3.7/bin/python3 "
python/source-code/bab 02/statemnet.py"
1
2
192:~ macintosh$ █
```

Gambar 2.3: Contoh *statement* dan *outputnya*

## Operator dan Operand

Operator adalah simbol tertentu yang digunakan untuk melakukan operasi aritmatika maupun logika. Operand merupakan nilai yang padanya dilakukan operasi. Misalnya 2+3. Tanda + adalah operator penjumlahan. 2 dan 3 merupakan *operand*.

Operator aritmatika adalah operator yang digunakan untuk melakukan operasi matematika, seperti penjumlahan, pengurangan, perkalian, pembagian, dan sebagainya.

Contoh:

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

Gambar 2.4: Tabel operator pada Python

## 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.

## Urutan Operasi

Urutan operasi berlaku bila ada lebih dari satu operator dalam expression. Urutan operasi bergantung pada aturan prioritas. Untuk operasi matematika, Python mengikuti konversi matematika. Urutan operasi sering disingkat dengan **PEMDAS** - **P**arantheses, **E**xponentiation, **M**ultiplication and **D**ivision, **O**perator. Bahasa lainnya yaitu **BODMAS** - **B**rackets

(Kurung), *Orders* (Pangkat/Akar), *Division* (Pembagian), *Multiplication* (Perkalian), *Addition* (Penjumlahan), dan *Subtraction* (Pengurangan).

## Operator Modulus dan String

### 1. Modulus

Operator Modulus merupakan sisa hasil bagi dari bilangan pertama dengan bilangan kedua. Operator ini hanya berlaku pada tipe data integer. Dalam python, operator modulus dilambangkan dengan tanda persen (%).

Contoh:

```
>>> oprmomulus = 7 % 3
>>> print(oprmomulus)
1
```

Gambar 2.5: Operasi menggunakan Modulus

Penjelasan gambar 2.5: 7 dibagi dengan 3 menghasilkan 2 dengan sisa hasil bagi 1.

### 2. String

Operator + ketika bekerja dengan string 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
```

Gambar 3.1: Operasi + (penjumlahan) pada *string*

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

Contoh:

```
>>> first = 'Test '
>>> second = 3
>>> print(first * second)
Test Test Test
```

Gambar 3.2: Operasi \* (perkalian) pada *string*

## Menangani Input dari Pengguna

Input adalah data yang dibutuhkan agar program dapat berjalan, proses adalah langkah-langkah yang dilakukan untuk menyelesaikan masalah, dan output adalah hasil yang diperoleh setelah proses selesai. Contohnya pada pengambilan uang di ATM: memasukkan kartu, PIN, memilih menu, dan nominal termasuk input; mesin melakukan pengecekan saldo, masa berlaku kartu, dan status blokir sebagai proses; lalu uang yang keluar, bukti transaksi, serta berkurangnya saldo merupakan output.

Dalam Python, input dari pengguna dapat diambil menggunakan fungsi `input()`, yang akan menghentikan program sementara hingga pengguna mengetik dan menekan Enter. Nilai yang dimasukkan akan disimpan sebagai string, sehingga jika ingin mengolahnya sebagai angka perlu dikonversi menggunakan `int()`. Prompt dapat ditambahkan dalam `input()` untuk memberi petunjuk kepada pengguna, dan tanda `\n` digunakan untuk membuat baris baru. Jika pengguna memasukkan data yang bukan angka saat dikonversi ke `int`, maka akan terjadi error `ValueError`.

## Komentar

Tanda pagar (`#`) digunakan untuk menandai komentar di python. Komentar tidak akan diproses oleh interpreter Python. Komentar hanya berguna untuk programmer untuk memudahkan memahami maksud dari kode.

Contoh: 

```
# Komentar pertama
print("hai dunia!") # Komentar kedua
```

Gambar 3.3: penggunaan komentar (`#`)

Outputnya akan mengeluarkan “hai dunia”


## 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 Github: [https://github.com/fieliel-star/71251214\\_raffiel.git](https://github.com/fieliel-star/71251214_raffiel.git)

Catatan: Penulis menggunakan extender yang bernama CodeSnap untuk mengambil *screenshot* pada program.

### SOAL 1

A screenshot of a code editor with a dark background and light-colored text. The code is a Python script for calculating BMI. It consists of seven lines of code, numbered 1 to 7. The code uses the input() function to get user input for weight and height, converts them to floats, calculates BMI using the formula weight / (height / 100)^2, and prints the result rounded to two decimal places.

```
1 beratBadan = float(input("Masukan berat badan: "))
2 tinggi = float(input("Masukan tinggi badan: "))
3
4 tinggi = tinggi / 100
5 BMI = beratBadan / tinggi**2
6
7 print (round (BMI,2))
```

Penjelasan:

Program tersebut bertugas untuk menghitung *Body Mass Index* (BMI) atau Indeks Massa Tubuh (IMT) berdasarkan berat badan dan tinggi badan yang dimasukan. Pada baris pertama Pada baris pertama, program meminta pengguna memasukkan berat badan melalui fungsi input(), lalu nilai tersebut langsung dikonversi menjadi tipe data float agar bisa digunakan dalam perhitungan, dan disimpan dalam variabel beratBadan. Baris kedua melakukan hal yang sama untuk tinggi badan dan menyimpannya dalam variabel tinggi. Pada baris keempat, nilai tinggi yang awalnya diasumsikan dalam satuan sentimeter diubah ke meter dengan cara membaginya dengan 100, karena rumus BMI menggunakan satuan meter. Selanjutnya pada baris kelima, program menghitung BMI dengan rumus berat badan dibagi tinggi badan kuadrat (tinggi\*\*2 berarti tinggi pangkat dua), lalu hasilnya disimpan dalam variabel BMI. Terakhir, pada baris



ketujuh, hasil BMI ditampilkan ke layar menggunakan fungsi `print()`, dengan `round(BMI, 2)` untuk membulatkan hasil hingga dua angka di belakang koma agar lebih rapi dan mudah dibaca.

Output:

Pengguna memasukan 84 sebagai berat badan dan 160.5 sebagai tinggi badan. Perhitungan mengeluarkan hasil 32.61. dalam tabel BMI, jika seseorang memiliki BMI  $\geq 30$ , maka disimpulkan bahwa orang tersebut obesitas.

```
Masukan berat badan: 84
Masukan tinggi badan: 160.5
32.61
```

## SOAL 2

```
1 x = int(input("Masukan nilai x: "))
2
3 if x != 0:
4     fungsi = (2*x**3) + (2*x) + (15/x)
5     print(fungsi)
6 else:
7     print("x tidak boleh 0")
```

Penjelasan:

Program tersebut bertujuan untuk menghitung nilai suatu fungsi matematika berdasarkan nilai  $x$  yang dimasukkan oleh pengguna. Pada baris pertama, program meminta pengguna memasukkan nilai  $x$  melalui fungsi `input()`, kemudian langsung mengonversinya menjadi tipe data `int` agar bisa digunakan dalam operasi perhitungan bilangan bulat. Nilai tersebut disimpan dalam variabel  $x$ . Selanjutnya, pada baris ketiga digunakan percabangan `if` untuk memeriksa apakah nilai  $x$  tidak sama dengan 0 ( $x \neq 0$ ). Pemeriksaan ini penting karena pada rumus terdapat operasi pembagian  $15/x$ , yang akan menyebabkan error (pembagian dengan nol) jika  $x$  bernilai 0.

Jika x tidak sama dengan 0, maka program akan menghitung nilai fungsi  $(2 \cdot x^3) + (2 \cdot x) + (15/x)$ . Bagian  $x^3$  berarti x dipangkatkan tiga, lalu dikalikan 2, kemudian ditambah dengan  $2 \cdot x$ , dan ditambah lagi dengan hasil pembagian 15 oleh x. Hasil perhitungan tersebut disimpan dalam variabel fungsi dan kemudian ditampilkan menggunakan print(). Namun, jika x bernilai 0, maka kondisi if tidak terpenuhi dan program akan menjalankan bagian else, yaitu menampilkan pesan "x tidak boleh 0" sebagai peringatan bahwa nilai tersebut tidak valid untuk perhitungan fungsi.

Output:

Pengguna memasukan 25 sebagai nilai x yang akan di hitung dan mengeluarkan hasil 31300,6. Ketika dimasukan 0 akan mengeluarkan hasil "x tidak boleh 0". Ini di karenakan dalam program terdapat  $15/x$  (15 dibagi x), jika  $x = 0$  maka akan mengeluarkan hasil tak terdefinisi.

Masukan nilai x: 25	Masukan nilai x: 0
31300.6	x tidak boleh 0

### SOAL 3

```
1 gajiPerjam = int(input("masukan gaji: "))
2 jamKerjaDalamSeminggu = int(input("masukan jam kerja: "))
3
4 totalJam = jamKerjaDalamSeminggu * 5
5 gajiKotor = gajiPerjam * totalJam
6 gajiSetelahPajak = gajiKotor - (gajiKotor * 0.14)
7 keperluan = gajiSetelahPajak * 0.10
8 alatTulis = gajiSetelahPajak * 0.01
9 sisaUang = gajiSetelahPajak - keperluan - alatTulis
10 sedekah = sisaUang * 0.25
11 yatim = sedekah * 0.30
12 dhuafa = sedekah * 0.70
13
14 print("Gaji sebelum bayar pajak:", "Rp", round(gajiKotor, 2))
15 print("Gaji setelah bayar pajak:", "Rp", round(gajiSetelahPajak, 2))
16 print("Pakaian dan aksesoris:", "Rp", round(keperluan, 2))
17 print("Alat tulis:", "Rp", round(alatTulis, 2))
18 print("Sedekah:", "Rp", round(sedekah, 2))
19 print("Diterima anak yatim:", "Rp", round(yatim, 2))
20 print("Diterima kaum dhuafa:", "Rp", round(dhuafa, 2))
```

#### Penjelasan:

Program tersebut berfungsi untuk menghitung pembagian gaji berdasarkan jam kerja, pajak, kebutuhan pribadi, serta sedekah. Pada baris pertama dan kedua, pengguna diminta memasukkan gaji per jam dan jumlah jam kerja dalam seminggu. Input tersebut dikonversi menjadi tipe data int agar dapat digunakan dalam operasi aritmetika. Selanjutnya, pada baris keempat dihitung totalJam, yaitu jumlah jam kerja dalam lima minggu (jam kerja per minggu dikalikan 5). Pada baris kelima dihitung gajiKotor, yaitu gaji per jam dikalikan total jam kerja. Kemudian pada baris keenam dihitung gajiSetelahPajak, yaitu gaji kotor dikurangi pajak sebesar 14% dari gaji kotor.

Setelah memperoleh gaji bersih (setelah pajak), program membagi pengeluaran ke beberapa pos. Baris ketujuh menghitung 10% dari gaji setelah pajak untuk keperluan seperti pakaian dan aksesoris, dan baris kedelapan menghitung 1% untuk alat tulis. Baris kesembilan menghitung sisaUang, yaitu gaji setelah pajak dikurangi keperluan dan alat tulis. Dari sisa uang tersebut, pada baris kesepuluh dihitung sedekah sebesar 25%. Selanjutnya, sedekah tersebut dibagi lagi: 30% untuk anak yatim (baris sebelas) dan 70% untuk kaum dhuafa (baris dua belas). Pada bagian akhir (baris 14–20), program menampilkan seluruh hasil perhitungan menggunakan print(). Setiap nilai uang dibulatkan hingga dua angka di belakang koma menggunakan fungsi round(variabel, 2) agar hasilnya lebih rapi dan mudah dibaca. Dengan demikian, program ini menggambarkan proses lengkap mulai dari perhitungan gaji kotor, pengurangan pajak, pembagian kebutuhan, hingga distribusi sedekah secara sistematis.

#### Output:

Pengguna memasukan 10000 sebagai gaji dan 40 sebagai jam kerja dalam seminggu (Rata-rata jam kerja di Indonesia adalah 40 jam per minggu). Mengeluarkan hasil sebagai berikut:

masukan gaji: 10000  
masukan jam kerja: 35  
Gaji sebelum bayar pajak: Rp 1750000  
Gaji setelah bayar pajak: Rp 1505000.0  
Pakaian dan aksesoris: Rp 150500.0  
Alat tulis: Rp 15050.0  
Sedekah: Rp 334862.5  
Diterima anak yatim: Rp 100458.75  
Diterima kaum dhuafa: Rp 234403.75