



**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	71251233
Nama Lengkap	Mikael Gratianus Satrio Adi Kuncoro
Minggu ke / Materi	02 / Variable, Expression dan Statements

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

Values dan type

Value merupakan komponen utama dari program seperti huruf dan angka, seperti 1, 2, 'a', 'z', dan "Hello World!". *Value* dibagi menjadi beberapa tipe yang berbeda, misalnya 2 untuk sebuah nilai integer dan "Hello World!" untuk sebuah nilai *string* karena penulisan nya ditutup menggunakan tanda petik (*quotation mark*). Berikut merupakan contoh untuk memahami *values dan type*:

```
>>> 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* (huruf), atau *bool* (benar/salah). Setiap *value* pasti memiliki *type* untuk mengetahui tipe data tersebut. Python menyediakan fungsi *built-in* untuk melakukan pemeriksaan tipe data pada value dengan 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. Misalnya 1.000.000. Di python, akan dianggap bilangan bulat.

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

Variable

Salah satu fitur powerfull dalam bahasa pemrograman adalah kemampuannya untuk melakukan manipulasi *variable*. *Variable* merupakan lokasi memori yang dicadangkan untuk menyimpan nilai-nilai. Ketika membuat sebuah variable berarti memesan beberapa ruang di memori. *Variable* menyimpan data yang dilakukan selama program dieksekusi, yang nantinya isi dari variabel tersebut dapat diubah beberapa operasi tertentu pada program yang menggunakan *variable*.

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

Variable dapat menyimpan berbagai macam tipe data. Di python, *variable* mempunyai sifat dinamis, artinya *variable* python tidak perlu dideklarasikan tipe data tertentu dan dapat diubah saat program dijalankan. Potongan code diatas merupakan contoh penggunaan dari *variable*. Contoh pertama adalah *variable* pesan yang berisi *string*, contoh kedua adalah variabel n yang berisi nilai *integer* 17 dan contoh ketiga merupakan nilai dari **pi** (π). Untuk menampilkan nilai dari *variable*, dapat menggunakan perintah print.

```
>>> print (n)  
17
```

Nama Variable dan Keywords

Pemberian nama pada *variable* 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 variable yang berbeda.
4. Nama *variable* 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*. Berikut dapat dilihat pada Gambar 1.1.

```

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

```

Gambar 1.1: 35 *keyword* yang tidak boleh digunakan untuk memberi nama *variable*.

Berikut ini contoh penggunaan *variable* dalam bahasa pemrogramaan python.

```

#proses memasukan data ke dalam variabel
nama = "Agung Sejagat"

#proses mencetak variabel
print(nama)

#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
namaDepan = "Joko"
namaBelakang = "Widodo"
nama = namaDepan + " " + namaBelakang
umur = 22
hobi = "Berenang"
print("Biodata\n", nama, "\n", umur, "\n", hobi)

#contoh variabel lainya
inivariabel = "Halo"
ini_juga_variabel = "Hai"
_inivariabeljuga = "Hi"
inivariabel1222 = "Bye"
panjang = 10
lebar = 5
luas = panjang * lebar
print(luas)

```

Gambar 1.2: penggunaan *variable* dalam bahasa pemrogramaan python.

Statements

Statements adalah perintah dalam Python yang bisa dijalankan oleh interpreter, seperti *print*, *expression*, dan *assignment*. Pada mode interaktif, Python langsung mengeksekusi perintah dan menampilkan hasilnya, sedangkan pada script mode, kode berisi kumpulan statement yang dijalankan secara berurutan.

```

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/statemenet.py"
1
2
192:~ macintosh$ █

```

Gambar 1.3: Contoh statement dan outputnya.

Operator dan *Operand*

Operator adalah simbol untuk melakukan operasi aritmatika atau logika pada suatu nilai yang disebut operand. Dalam Python, operator aritmatika digunakan untuk operasi matematika seperti penjumlahan, pengurangan, perkalian, dan pembagian.

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

Gambar 1.4: Tabel Operator pada Python

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) * (15-7)
112

```

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 *variable*. Secara umum, semuanya dapat disebut dengan *expression*. Ketika menggunakan expression dalam model intercative, interpreter akan melakukan evaluasi dan menampilkan hasilnya.

Urutan Operasi

Urutan operasi berlaku bila ada lebih dari satu operator dalam *expression* dan bergantung pada aturan prioritas. Untuk operasi matematika, python mengikuti konversi matematika. Urutan operasi sering disingkat dengan PEMDAS - *Parantheses, Exponentiation, Multiplication and Division, Operator*.

- *Paranthese* (Tanda kurung) - prioritas tertinggi, memaksa urutan perhitungan dan membuat ekspresi lebih mudah dibaca. Contoh: $2*(3-1) = 4$, $(1+1)^2 * (5-2) = 8$.
- *Exponentiation* (Eksponensial/Pemangkatan) - dihitung setelah tanda kurung. Contoh: $2^2 + 1 = 3$, $3 * 1^2 = 3$.
- *Multiplication and Divison* (Perkalian dan Pembagian) - prioritas sama dan lebih tinggi dari penjumlahan/pengurangan. Contoh: $2 * 3 - 1 = 5$, $6 + 4 / 2 = 8$.
- *Operators* - dihitung dari kiri ke kanan. Contoh: $5 - 3 - 1 = 1$.

Operator Modulus dan String

- 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 (%).

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

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

Operator **modulus (%)** digunakan untuk mengecek sisa hasil pembagian. Jika $x \% y = 0$, maka x habis dibagi y. Selain itu, modulus juga dapat digunakan untuk mengambil digit terakhir suatu bilangan, misalnya $x \% 10$ menghasilkan digit paling kanan, dan $x \% 100$ menghasilkan dua digit terakhir.

- 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
```

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

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

Menangani Input dari Pengguna

Program memiliki alur **Input–Proses–Output**, di mana input adalah data yang diperlukan, proses adalah pengolahan data untuk menyelesaikan masalah, dan output adalah hasilnya; contohnya pada ATM, kartu, PIN, dan nominal uang sebagai input diproses melalui pengecekan saldo dan validasi, lalu menghasilkan uang keluar dan saldo berkurang sebagai output, sedangkan dalam Python input pengguna diambil dengan fungsi input() yang menunggu hingga Enter ditekan dan mengembalikan data sebagai string.

```
>>> name = input('Siapa nama mu ?\n')  
Siapa nama mu ?  
Sancaka  
>>> print(name)  
Sancaka
```

Tanda \n pada akhir prompt mewakili baris baru atau ganti baris sehingga input pengguna muncul dibawah prompt.

Ketika mengharapkan pengguna untuk mengetik bilangan bulat, dapat dilakukan dengan mengonversi nilai kembali ke int menggunakan fungsi int():

```
>>> prompt = 'Berapa suhu ruangan sekarang?\n'  
>>> suhu = input(prompt)  
Berapa suhu ruangan sekarang?  
24  
>>> int(suhu)  
24  
>>> int(suhu) + 5  
29
```

Komentar

Tanda pagar (#) digunakan untuk menandai komentar di python. Komentar tidak akan diproses oleh interpreter Python.

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

dengan output yang dihasilkan adalah **hai dunia**

LINK GITHUB

: <https://github.com/Rio-code-07/71251233-MikaelRio-PrakAlpro.git>

Soal 1

- Program menghitung berat badan Ideal berdasarkan tinggi badan dan BMI.

The screenshot shows a code editor with a Python file named 'Soal1.py'. The code prompts the user for height and BMI, calculates the ideal weight using the formula $BMI \times (height^2)$, and prints the result. To the right, a terminal window shows the execution of the script and its output: it asks for height (1.7), BMI (21), and then prints the calculated weight (60.7 kg).

```
Soal1.py > ...
1 #input tinggi badan dan Body Mass Index
2 Tinggi_badan = float(input("Masukkan Tinggi Badan (m): "))
3 Body_Mass_Index = float(input("Masukkan BMI yang diharapkan: "))
4
5 #perhitungan
6 Berat_badan = Body_Mass_Index * (Tinggi_badan ** 2)
7
8 #output
9 print("Berat badan yang diperlukan adalah", round(Berat_badan, 1), "kg")
10
```

```
PS D:\New folder - Copy\MikaelRio_Coding> python -u "d:\New folder - Copy\MikaelRio_Coding\Soal1.py"
Masukkan Tinggi Badan (m): 1.7
Masukkan BMI yang diharapkan: 21
Berat badan yang diperlukan adalah 60.7 kg
PS D:\New folder - Copy\MikaelRio_Coding> ]
```

Penjelasan:

Source code ini merupakan program python untuk menghitung berat badan ideal berdasarkan tinggi badan dan BMI (Body Mass Index). Program diawali dengan fungsi input dan dua variabel berupa variabel **Tinggi_badan** untuk mamasukan data tinggi badan dalam meter dan variabel **Body_Mass_Index** untuk memasukan BMI (Body Mass Index) yang diharapkan, inputan ini menyimpan data sebagai angka desimal (float) karena biasanya tinggi badan dan BMI ada angka desimal nya. Setelah itu di baris 6, program diketikkan untuk menghitung berat badan yang diperlukan dengan rumus: BMI dikali tinggi badan pangkat dua (tinggi dikuadratkan), karena rumus untuk menentukan BMI sendiri adalah $BMI = \text{berat} / \text{tinggi}^2$ maka tinggal membalik rumus menjadi $\text{berat} = BMI \times \text{tinggi}^2$, hasilnya disimpan dalam variabel **Berat_badan**. Terakhir, program menampilkan hasil perhitungan dengan fungsi **print()**, dimana berat badan dibulatkan menjadi 1 angka di belakang koma menggunakan fungsi **round**, dan ditampilkan bersama satuan "kg". Jadi kalau input tingginya 1.7 m dan BMI-nya 21, program akan hitung $21 \times (1.7)^2 = 60.69$ kg, lalu dibulatkan jadi 60.7 kg seperti yang terlihat di output sebelah kanan.

Soal 2

- Program menghitung hasil dari fungsi $f(x)= 2x^3 + 2x + 15/x$.

The screenshot shows a code editor with a Python file named 'Soal2.py'. The code prompts the user for a whole number, calculates the value of the function $f(x) = 2x^3 + 2x + 15/x$ for that number, and prints the result. To the right, a terminal window shows the execution of the script and its output: it asks for a whole number (5), calculates the function value (263.0), and prints the result.

```
Soal2.py > ...
1 # input
2 x = int(input("Masukan bilangan bulat: "))
3
4 #fungsi / rumus
5 Rumus = 2 * (x**3) + 2 * x + (15 / x)
6
7 #output
8 print("Hasil fungsi f(x)= 2x^3 + 2x + 15/x adalah:", Rumus)
```

```
PS D:\New folder - Copy\MikaelRio_Coding> python -u "d:\New folder - Copy\MikaelRio_Coding\Soal2.py"
Masukan bilangan bulat: 5
Hasil fungsi f(x)= 2x^3 + 2x + 15/x adalah: 263.0
PS D:\New folder - Copy\MikaelRio_Coding> ]
```

Penjelasan:

Source code ini merupakan program python untuk menghitung hasil dari fungsi $f(x)= 2x^3 + 2x + 15/x$. Program diawali dengan fungsi input dengan variabel **x** menggunakan fungsi **int** agar hasilnya berupa bilangan bulat. Kemudian, program menghitung fungsi matematika dengan rumus 2 dikali x pangkat 3, ditambah 2 dikali x, ditambah 15 dibagi x, dan hasilnya disimpan dalam variabel **Rumus**. Terakhir program menampilkan hasil perhitungan dengan fungsi **print()** sebagai output yang menunjukan teks "Hasil fungsi $f(x)= 2x^3 + 2x + 15/x$ adalah:" diikuti dengan nilai hasil perhitungan yang tersimpan di variabel **Rumus**. Jadi kalau

input x-nya = 5 seperti di output sebelah kanan, program akan menghitung $2 \times (5^3) + 2 \times 5 + 15/5 = 2 \times 125 + 10 + 3 = 250 + 10 + 3 = 263.0$, yang ditampilkan sebagai hasil akhir.

Soal 3

- Program perencanaan keuangan pribadi Budi

```
Soal3.py > ...
1  # Input
2  Gaji_per_jam = float(input("Masukkan gaji per jam: "))
3  Jumlah_jam_kerja = float(input("Masukkan jumlah jam kerja dalam 1 minggu: "))
4
5  # Perhitungan gaji
6  Total_jam_kerja = Jumlah_jam_kerja * 5
7  Total_Gaji_Kotor = Gaji_per_jam * Total_jam_kerja
8
9  # Pajak 14%
10 Pajak = Total_Gaji_Kotor * 0.14
11 Total_Gaji_bersih = Total_Gaji_Kotor - Pajak
12
13 # Pengeluaran dari gaji bersih
14 Aksesoris_baju = Total_Gaji_bersih * 0.10
15 Alat_tulis = Total_Gaji_bersih * 0.01
16
17 # Sisa uang setelah pengeluaran
18 Sisa_uang = Total_Gaji_bersih - (Aksesoris_baju + Alat_tulis)
19
20 # Sedekah 25% dari sisa uang
21 Sedekah = Sisa_uang * 0.25
22
23 # Pembagian sedekah
24 Anak_yatim = Sedekah * 0.30
25 Kaum_dhuafa = Sedekah * 0.70
26
27 # Output
28 print("\n==OUTPUT==")
29 print("1. Pendapatan Budi sebelum melakukan pembayaran pajak: Rp", int(Total_Gaji_Kotor))
30 print("2. Pendapatan Budi selama libur musim panas setelah melakukan pembayaran pajak: Rp", int(Total_Gaji_bersih))
31 print("3. Jumlah uang yang akan Budi habiskan untuk membeli pakaian dan aksesoris: Rp", int(Aksesoris_baju))
32 print("4. Jumlah uang yang akan Budi habiskan untuk membeli alat tulis: Rp", int(Alat_tulis))
33 print("5. Jumlah uang yang akan Budi sedekahkan: Rp", int(Sedekah))
34 print("6. Jumlah uang yang akan diterima anak yatim: Rp", int(Anak_yatim))
35 print("7. Jumlah uang yang akan diterima kaum dhuafa: Rp", int(Kaum_dhuafa))
```

Penjelasan:

Source code ini merupakan program python untuk menghitung perencanaan keuangan pribadi Budi. Pertama di baris 2-3, program diketikkan variabel **Gaji_per_jam** dan **Jumlah_jam_kerja** per minggu, keduanya disimpan sebagai angka desimal (float). Kemudian di baris 6-7, program menghitung **Total_jam_kerja** selama 5 minggu dan **Total_gaji_kotor** dengan mengalikan **Gaji_per_jam** dengan **Total_jam_kerja**. Di baris 10-11, program menghitung **Pajak** 14% dari **Total_Gaji_kotor** dan menghitung **Total_Gaji_bersih** dengan mengurangkan **Total_Gaji_kotor** dengan **Pajak**. Selanjutnya di baris 14-15, program menghitung pengeluaran untuk **Aksesoris_baju** (10% dari **Total_Gaji_bersih**) dan **Alat_tulis** (1% dari **Total_Gaji_bersih**). Di baris 18, program menghitung **Sisa_uang** setelah membeli **Aksesoris_baju** dan **Alat_tulis** dengan mengurangi **Total_Gaji_bersih** dengan kedua pengeluaran tersebut. Di baris 21, program menghitung jumlah **Sedekah** yaitu 25% dari **Sisa_uang**. Kemudian di baris 24-25, program membagi **Sedekah** menjadi dua: 30% untuk **Anak_yatim** dan 70% untuk **Kaum_dhuafa**. Terakhir di baris 28-35, program menampilkan semua hasil perhitungan dengan fungsi **print()** mulai dari pendapatan sebelum pajak, setelah pajak, pengeluaran untuk pakaian, alat tulis, sedekah, bagian anak yatim, dan bagian kaum dhuafa, semuanya dikonversi ke bilangan bulat (int) agar tampil tanpa desimal. \n pada komentar output merupakan karakter khusus untuk membuat baris baru atau pindah ke baris berikutnya, fungsi nya seperti menekan tombol Enter pada keyboard.