



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>71230971</b>
<b>Nama Lengkap</b>	<b>James Marvin Santoso</b>
<b>Minggu ke / Materi</b>	<b>02 / Variable, Expression dan Statements</b>

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

### Values dan Type

Value merupakan komponen utama dari sebuah program, seperti huruf atau angka. Beberapa contoh value yang sering digunakan dalam Python adalah 1, 2, 'a', 'z', dan "Hello Word". Value dalam Python dibagi menjadi beberapa tipe yang berbeda, misalnya, 2 adalah nilai integer dan "Hello Word" adalah nilai string. Interpreter Python dapat mengidentifikasi string karena ditulis di antara tanda petik (quotation mark).

Untuk mencoba memahami values dan tipe data dalam Python, ada contoh yang bisa dicoba beberapa baris kode di bawah ini menggunakan Python interactive mode:

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

Perintah print dalam Python tidak hanya berfungsi untuk mencetak nilai string, tetapi juga untuk mencetak nilai selain string, seperti integer (bilangan bulat), float (bilangan pecahan), character (huruf), atau bool (benar/salah)

### Variabel

Variable dalam pemrograman adalah lokasi memori untuk menyimpan nilai. Dalam Python, variable bersifat dinamis, tidak perlu mendeklarasikan tipe data, dan nilai dapat diubah. Variable memungkinkan manipulasi data, seperti perhitungan matematika, menyimpan teks, atau status.

Dalam variabel juga terdapat kata kunci yang bisa digunakan, antara lain :

1. Nama variable dalam Python dapat diawali dengan 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. Nama variable bersifat case-sensitive, artinya huruf besar dan kecil dibedakan. Misalnya, variabel\_Ku dan variabel\_ku adalah variabel yang berbeda.
4. Nama variabel tidak boleh menggunakan kata kunci yang sudah ada dalam Python seperti if, while, for, dan sebagainya.

Namun dari kata kunci yang bisa digunakan juga terdapat kata kunci yang tidak bisa untuk digunakan, antara lain :

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

## Statements

Statements adalah bagian dari kode Python yang dapat dieksekusi. Contoh statements meliputi print, expression statements, dan assignment. Saat menggunakan Python dalam mode interaktif, interpreter akan langsung mengeksekusi perintah dan menampilkan hasilnya. Namun, saat menggunakan mode skrip, biasanya terdapat serangkaian statements yang dieksekusi secara berurutan.

Berikut contoh Statements beserta outputnya, antara lain :

```

contoh.py > ...
1  print(1)
2  x = 5
3  print("Nilai x adalah:", x)

```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```

PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe c:/Tugas/contoh.py
1
Nilai x adalah: 5
PS C:\Tugas>

```

## Operator dan Operand

Operator adalah simbol tertentu yang digunakan untuk melakukan operasi aritmatika maupun logika. Nilai yang dikenai operasi disebut operand. Misalnya, dalam ekspresi  $2 + 3$ , tanda  $+$  adalah operator penjumlahan, sedangkan 2 dan 3 adalah operand.

Operator aritmatika pada Python digunakan untuk melakukan operasi matematika seperti penjumlahan, pengurangan, perkalian, dan pembagian. Berikut adalah jenis operator aritmatika, antara lain :

Tabel 2.1: Operator pada Python

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	$x ** y$

- 1) Penjumlahan (+): Menambahkan dua nilai. Contoh:  $2 + 3 = 5$
- 2) Pengurangan (-): Mengurangkan nilai kedua dari nilai pertama. Contoh:  $5 - 3 = 2$
- 3) Perkalian (\*): Mengalikan dua nilai. Contoh:  $2 * 3 = 6$
- 4) Pembagian (/): Membagi nilai pertama dengan nilai kedua. Contoh:  $6 / 2 = 3$
- 5) Pangkat (\*\*): Mengembalikan hasil pemangkatan. Contoh:  $2 ** 3 = 8$

## Operator Modulus dan String Modulus

Operator Modulus (%) digunakan untuk mengembalikan sisa hasil bagi dari pembagian dua bilangan bulat. Operator ini hanya berlaku pada tipe data integer. Dalam python, operator modulus dilambangkan dengan tanda persen (%). Berikut adalah contoh penggunaan operator modulus dalam Python:

```

contoh.py > ...
1  x = 7
2  y = 2
3  sisa = x % y
4
5  print("Sisa pembagian", x, "dengan", y, "adalah", sisa)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe c:/Tugas/contoh.py
Sisa pembagian 7 dengan 2 adalah 1
PS C:\Tugas> 
```

## String

Operator + pada string digunakan untuk menggabungkan (concatenate) dua string menjadi satu. Operasi ini tidak melakukan penjumlahan matematika seperti pada tipe data numerik. Berikut adalah contoh penggunaan operator + pada string dalam Python:

```
contoh.py > nama_depan
1  nama_depan = "James"
2  nama_belakang = "Marvin"
3  nama_lengkap = nama_depan + " " + nama_belakang
4
5  print("Nama lengkap:", nama_lengkap)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe c:/Tugas/contoh.py
Nama lengkap: James Marvin
PS C:\Tugas> █
```

Dalam contoh ini, operator + digunakan untuk menggabungkan string nama\_depan, spasi, dan string nama\_belakang menjadi satu string nama\_lengkap. Output dari program ini adalah "Nama lengkap: John Doe".

## Expressions

Expression adalah representasi dari nilai yang dapat berupa nilai konstan (values), variabel, operator, atau gabungan dari semuanya. Secara umum, expression dapat mencakup nilai apa pun yang dapat dievaluasi menjadi suatu nilai tunggal.

Berikut adalah contoh dari Expression beserta source codenya dan output, antara lain :

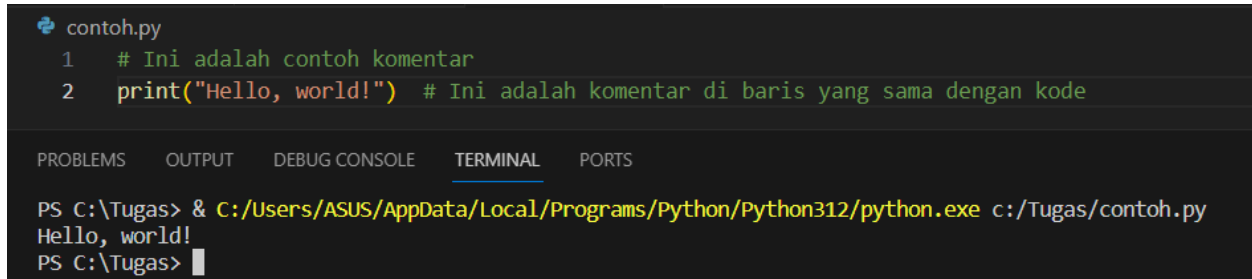
```
contoh.py > ...
1  nilai = 42
2  x = 10
3  y = 5
4  total = x + y
5  hasil = total * 2
6  print("Nilai total:", total)
7  print("Hasil:", hasil)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe c:/Tugas/contoh.py
Nilai total: 15
Hasil: 30
PS C:\Tugas> █
```

## Komentar

Tanda pagar (#) digunakan untuk menandai komentar di Python. Komentar adalah bagian dari kode yang tidak akan dieksekusi oleh interpreter Python dan hanya berguna untuk memberikan penjelasan atau catatan kepada programmer. Contoh penggunaan komentar:



```
contoh.py
1  # Ini adalah contoh komentar
2  print("Hello, world!") # Ini adalah komentar di baris yang sama dengan kode
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe c:/Tugas/contoh.py
Hello, world!
PS C:\Tugas> 
```

Komentar digunakan untuk memberikan penjelasan tentang kode, menyertakan dokumentasi, atau menonaktifkan sementara bagian dari kode.

## BAGIAN 2: LATIHAN MANDIRI

### SOAL 1.1

- Source Code :

```
Latihan 2,1.py > ...
1 def hitung_berat_badan(tinggi_badan, bmi_harap):
2     berat_badan = bmi_harap * (tinggi_badan**2)
3     return berat_badan
4
5 tinggi_badan = float(input("Masukkan tinggi badan (dalam meter): "))
6 bmi_harap = float(input("Masukkan nilai BMI yang diharapkan: "))
7
8 berat_badan_diperlukan = hitung_berat_badan(tinggi_badan, bmi_harap)
9 print(f"Berat badan yang diperlukan untuk tinggi badan {tinggi_badan} m dan BMI {bmi_harap} adalah {berat_badan_diperlukan:.2f} kg")
```

- Output :

```
PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe "c:/Tugas/Latihan 2,1.py"
Masukkan tinggi badan (dalam meter): 1
Masukkan nilai BMI yang diharapkan: 75
Berat badan yang diperlukan untuk tinggi badan 1.0 m dan BMI 75.0 adalah 75.00 kg
```

- Penjelasan :

Fungsi `hitung_berat_badan` menghitung berat badan yang diperlukan berdasarkan tinggi badan (`tinggi_badan`) dan nilai BMI yang diharapkan (`bmi_harap`). Kemudian, program meminta input dari pengguna untuk tinggi badan dan nilai BMI yang diharapkan. Nilai-nilai tersebut digunakan sebagai argumen untuk memanggil fungsi `hitung_berat_badan`, dan hasilnya diprint dalam satu baris menggunakan f-string.

## SOAL 1.2

- Source Code :

```
Latihan 2,2.py > ...
1  def hitung_fungsi(x):
2      return (2 * x**3 + 2 * x + 15) / x
3
4  try:
5      x = int(input("Masukkan bilangan bulat x: "))
6      hasil = hitung_fungsi(x)
7      print(f"Hasil fungsi f({x}) adalah: {hasil}")
8  except ValueError:
9      print("Input yang dimasukkan bukan bilangan bulat.")
10 except ZeroDivisionError:
11     print("Input yang dimasukkan tidak boleh nol.")
```

- Output :

```
PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe "c:/Tugas/Latihan 2,2.py"
Masukkan bilangan bulat x: 4
Hasil fungsi f(4) adalah: 37.75
```

- Penjelasan :

Fungsi `hitung_fungsi` menghitung nilai fungsi matematika Program meminta input dari pengguna untuk nilai `x` dalam bentuk bilangan bulat. Nilai tersebut digunakan sebagai argumen untuk memanggil fungsi `hitung_fungsi`. Hasil perhitungan kemudian diprint menggunakan f-string.



### SOAL 1.3

- Source Code :

```
Latihan 2,3.py > hitung_pendapatan
1 def hitung_pendapatan(g, j):
2     k = j * 5
3     p = g * k
4     pa = 0.14 * p
5     ps = p - pa
6     bba = 0.10 * ps
7     bat = 0.01 * ps
8     sub = ps - bba - bat
9     s = 0.25 * sub
10    say = 0.30 * s
11    sd = 0.70 * s
12    return (p, ps, bba, bat, s, say, sd)
13
14 g = float(input("Masukkan gaji per jam yang Anda inginkan: "))
15 j = int(input("Masukkan jumlah jam kerja per minggu: "))
16
17 p, ps, bba, bat, s, say, sd = hitung_pendapatan(g, j)
18
19 print(f"1. Pendapatan Budi sebelum pajak: Rp {p}")
20 print(f"2. Pendapatan Budi setelah pajak: Rp {ps}")
21 print(f"3. Belanja baju dan aksesoris: Rp {bba}")
22 print(f"4. Belanja alat tulis: Rp {bat}")
23 print(f"5. Sedekah: Rp {s}")
24 print(f"6. Sedekah untuk anak yatim: Rp {say}")
25 print(f"7. Sedekah untuk kaum dhuafa: Rp {sd}")
```

- Output :

```
PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe "c:/Tugas/Latihan 2,3.py"
Masukkan gaji per jam yang Anda inginkan: 2
Masukkan jumlah jam kerja per minggu: 14
1. Pendapatan Budi sebelum pajak: Rp 140.0
2. Pendapatan Budi setelah pajak: Rp 120.4
3. Belanja baju dan aksesoris: Rp 12.040000000000001
4. Belanja alat tulis: Rp 1.2040000000000002
5. Sedekah: Rp 26.789
6. Sedekah untuk anak yatim: Rp 8.0367
7. Sedekah untuk kaum dhuafa: Rp 18.752299999999998
```

- Penjelasan :

Fungsi `hitung_pendapatan` menghitung pendapatan Budi selama libur musim panas berdasarkan gaji per jam (`gaji_per_jam`) yang diinginkan dan jumlah jam kerja per minggu (`jam_kerja_per_minggu`). Program kemudian meminta input dari pengguna untuk nilai gaji per jam dan jumlah jam kerja per minggu. Hasil perhitungan kemudian diprint dalam bentuk kalimat yang menjelaskan masing-masing komponen pendapatan dan pengeluaran Budi selama libur musim panas.

- Link GitHub :

<https://github.com/Jamesmarvins/Tugas-Alpro-2.git>