



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	71251223
Nama Lengkap	Banny Priskila haba
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

Values dan type value merupakan komponen fundamental dalam pemrograman yang mempresentasikan data yang akan diproses oleh program. Values dapat berupa data seperti angka, huruf, atau teks. Setiap value memiliki tipe tertentu yang menentukan bagaimana data tersebut disimpan dan dioperasikan dalam program. Contohnya integer untuk menyimpan data berupa angka dan string untuk menyimpan data berupa teks. Contoh dalam bentuk code:

```
>>> 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). Untuk mencobanya kita dapat menggunakan perintah python untuk menjalankan interpreter.

Python menyediakan fungsi built-in untuk melakukan pengecekan 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">
```

Bilangan bukat besar dalam Python harus ditulis tanpa pemisah ribuan, atau menggunakan underscore(_) sebagai pemisah agar lebih mudah dibaca. Penggunaan tanda koma(,) dan titik(.) akan menyebabkan interpreter salah menginterpretasikan nilai tersebut.

VARIABEL

Variabel adalah tempat penyimpanan data di memori computer. Ketika membuat variabel, kita memesan ruang di memori untuk menyimpan nilai yang dapat diubah selama program berjalan.

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

Di dalam pemrograman Python variabel bersifat dinamis sehingga tidak perlu menentukan tipe data di awal, dan tipe data dapat berubah sesuai kebutuhan. Pada contoh diatas variabel pesan berisi string, contoh kedua berisi variabel n yang berisi nilai integer 17 dan contoh ketiga merupakan nilai dari pi (π). Untuk menampilkan nilai dari variabel, dapat menggunakan perintah print.

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

NAMA VARIABEL DAN KEYWORDS

Pemberian nama pada variabel mengacu pada panduan berikut ini.

1. Awalan menggunakan huruf atau garis bawah (contoh: nama, _nama, namaKu, nama_variable).
2. Karakter selanjutnya menggunakan huruf, angka dan garis bawah (contoh: nilai1, buku_2).
3. Huruf besar dan kecil dibedakan (misal: variabel Nama dan nama adalah dua variabel yang berbeda).
4. Tidak boleh menggunakan keyword Python.

Berikut merupakan 35 *keyword* Python yang tidak boleh digunakan untuk nama variabel.

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	wait

STATEMENTS

Statement adalah instruksi yang dapat di eksekusi oleh Python. Contohnya statement print, dapat berupa *expression statement* dan *assignment*. Pada mode interactive statement

langsung dieksekusi satu persatu dan hasil langsung ditampilkan. Hal ini tentu saja berbeda ketika menggunakan script mode. Script biasanya berisi statements yang saling berhubungan secara sekuensial.

```
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 2.1: Contoh statement dan outputnya

OPERATOR DAN OPERAND

Operator adalah simbol untuk melakukan operasi. Operand adalah nilai yang dioperasikan. pada bagian ini akan membahas operator aritmatika pada Python. Operator aritmatika adalah operator yang digunakan untuk melakukan operasi matematika, seperti penjumlahan, pengurangan, perkalian, pembagian, dan sebagainya. Tabel berikut menunjukkan jenis operator aritmatika.

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	$x ** y$

EXPRESIONS

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.

```
17
x
x+17
```

Ketika menggunakan expression dalam model intercatave, interpreter akan melakukan evaluasi dan menampilkan hasilnya.

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

URUTAN OPERASI

Ketika ada beberapa operator dalam satu ekspresi, Python mengikuti aturan prioritas PEMDAS:

1. Parentheses (tanda kurung) – prioritas tertinggi
2. Exponentiation (pangkat) – Prioritas kedua
3. Multiplication & Division – prioritas sama, kiri ke kanan
4. Addition & Subtraction – prioritas sama dari kiri ke kanan

Gunakan tanda kurung jika ragu untuk memastikan urutan yang diinginkan.

OPERATOR MODULUS DAN STRING

Modulus

Operator modulus adalah sisa hasil bagi dari pembagian dua bilangan. Operator ini hanya berlaku pada tipe data integer. Operator modulus dilambangkan 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.

Contoh penggunaan operator modulus:

- Memeriksa habis dibagi. Misal jika $x \% y$ adalah 0, maka x dapat dibagi oleh y .
- Mengekstrak digit. Misal $x \% 10$ menghasilkan digit x paling kanan (dalam basis 10).

String

Dalam string operator $+$ bekerja untuk menggabungkan 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

Input adalah data yang diperlukan agar program dapat berjalan. Proses adalah Langkah-langkah yang dilakukan program untuk mengolah input menjadi output. Output adalah hasil dari proses.

Python menyediakan fungsi bawaan (built-in function) bernama `input()` yang digunakan untuk menerima data/masukan dari pengguna melalui keyboard. Ketika pengguna menekan tombol Enter, program akan dilanjutkan dan input akan mengembalikan apa yang diketik oleh pengguna sebagai string.

```
>>> inp = input()
Pada hari minggu kuturut ayah ke kota
>>> print(inp)
Pada hari minggu kuturut ayah ke kota
```

Sebelum meminta input dari pengguna, sebaiknya tampilkan prompt yang menjelaskan apa yang harus diinput.

```
>>> 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
```

Akan terjadi error jika pengguna memasukkan data selain angka.

```
>>> prompt = 'Berapa suhu ruangan sekarang?\n'
>>> suhu = input(prompt)
Berapa suhu ruangan sekarang?
Ruangan depan atau belakang ?
>>> int(suhu)
ValueError: invalid literal for int() with base 10:
```

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.

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

Outputnya:

```
hai dunia
```

Python tidak memiliki fitur komentar multibaris. Kita harus mengomentari satu persatu baris seperti berikut:

```
# Ini komentar
# Ini juga adalah komentar
# Ini juga masih komentar
```

BAGIAN 2: LATIHAN MANDIRI (60%)

Link github: https://github.com/banny05/71251223_priskilahaba.git

SOAL 2.1

Pada soal ini diminta untuk membuat program yang dapat menghitung berat badan yang diperlukan, jika diketahui tinggi badan dan nilai Body Mass Index (BMI) yang diharapkan.

1. Langkah pertama adalah meminta pengguna memasukkan tinggi badan dalam satuan cm. Program menggunakan fungsi `input()` untuk menampilkan pesan "Tinggi(cm) : " dan menerima masukan dari pengguna. Input yang diterima berupa string kemudian dikonversi menjadi tipe data float menggunakan fungsi `float()` agar dapat digunakan dalam perhitungan matematika. Setelah pengguna mengetik dan menekan Enter, nilai Tinggi akan disimpan dalam variabel `tinggi`.

Code yang dibuat seperti ini:

```
soal2.1.py > ...  
1  tinggi = float(input("Tinggi(cm) : "))
```

Pengguna memasukkan nilai pada inputan dibawah ini:

```
• Tinggi(cm) : 150
```

2. Langkah kedua adalah meminta input nilai BMI yang diharapkan oleh pengguna. Program menggunakan fungsi `input()` dengan pesan "BMI yang diharapkan : " untuk menerima masukan. Sama seperti sebelumnya, input string dikonversi menjadi float agar dapat diproses dalam operasi perhitungan. Nilai BMI ini disimpan dalam variabel `bmi`.

Code yang dibuat seperti ini:

```
2  bmi = float(input("BMI yang diharapkan : "))
```

Pengguna memasukkan nilai pada inputan dibawah ini:

```
BMI yang diharapkan : 19.1
```

3. Langkah ketiga adalah melakukan perhitungan berat badan yang diperlukan berdasarkan rumus BMI. Karena rumus BMI = berat/ tinggi kuadrat, maka untuk mencari berat badan, rumus dibalik menjadi berat = BMI * tinggi kuadrat. Dalam perhitungan ini, tinggi yang awalnya dalam centimeter dibagi 100 untuk dikonversi

ke meter, kemudian dipangkatkan dua, dan hasilnya dikalikan dengan nilai BMI. Hasil perhitungan disimpan dalam variabel berat.

Code yang dibuat seperti ini:

```
3 berat = (tinggi/100)**2 * bmi
```

- Langkah keempat adalah menampilkan hasil perhitungan. Untuk menampilkan hasil perhitungan menggunakan fungsi print() dengan format f-string untuk menampilkan pesan berisi berat badan. Pada program ini menggunakan fungsi round untuk membulatkan hasil perhitungan menjadi satu angka di belakang koma.

```
4 print (f"Berat yang diperlukan {round(berat,1)} kg")
```

Output yang ditampilkan adalah "berat yang diperlukan" diikuti dengan nilai berat dalam satuan kilogram.

```
Berat yang diperlukan 43.0 kg
```

Berikut adalah source code lengkap:

```
soal2.1.py > ...
1 tinggi = float(input("Tinggi(cm) : "))
2 bmi = float(input("BMI yang diharapkan : "))
3 berat = (tinggi/100)**2 * bmi
4 print (f"Berat yang diperlukan {round(berat,1)} kg")
```

Output:

```
PS C:\71251223_Semester2\71251223_priskila\minggu02> & C:\Users\USER\AppData\Local\Programs\Python\Python314\python.exe c:\71251223_Semester2\71251223_priskila\minggu02\soal2.1.py
Tinggi(cm) : 150
BMI yang diharapkan : 19.1
Berat yang diperlukan 43.0 kg
PS C:\71251223_Semester2\71251223_priskila\minggu02>
```

SOAL 2.2

- Langkah pertama adalah meminta input dari pengguna dengan menggunakan fungsi input() untuk menampilkan pesan "x=" dan menerima masukan dari pengguna. Input yang diterima berupa string kemudian dikonversi menjadi tipe data integer menggunakan fungsi int() karena soal menyebutkan bahwa x merupakan bilangan bulat. Nilai x ini disimpan dalam variabel bernama x. code yang dibuat seperti ini:

```
soal2.2.py > ...
1 x = int(input("x = "))
```

Pengguna memasukkan nilai pada pesan berikut.

```
x = 
```

- Langkah kedua adalah menghitung nilai $f(x)$ berdasarkan rumus yang diberikan yaitu $2x^3 + 2x + 15/x$. Dalam perhitungan ini, program menggunakan operator matematika Python di mana `**` digunakan untuk pangkat, `*` untuk perkalian, `+` untuk penjumlahan, dan `/` untuk pembagian. Perhitungan dilakukan secara berurutan sesuai aturan matematika, yaitu $2*x**3$ untuk menghitung $2x^3$, kemudian ditambah $2*x$, dan terakhir ditambah $15/x$. Hasil perhitungan fungsi ini disimpan dalam variabel `fungsi_x`. code yang dibuat seperti ini :

```
2 fungsi_x = 2*x**3 + 2*x + 15/x
```

- Langkah ketiga adalah menampilkan hasil perhitungan. Program menggunakan fungsi `print()` untuk menampilkan teks `"f(x) = "` diikuti dengan nilai hasil perhitungan fungsi yang tersimpan dalam variabel `fungsi_x`. Output akan menampilkan nilai fungsi dalam bentuk desimal jika hasilnya bukan bilangan bulat.

```
f(x) = 263.0
```

Berikut adalah bentuk source code lengkap:

```
soal2.2.py > ...
1 x = int(input("x = "))
2 fungsi_x = 2*x**3 + 2*x + 15/x
3
4 print("f(x) = ",fungsi_x)
```

Output:

```
PS C:\71251223_Semester2\71251223_priskila\minggu02> & C:\Users\USER\AppData\Local\Programs\Python\Python314\python.exe c:
/71251223_Semester2/71251223_priskila/minggu02/soal2.2.py
x = 5
f(x) = 263.0
PS C:\71251223_Semester2\71251223_priskila\minggu02>
```

SOAL 2.3

diminta untuk menghitung rincian penghasilan dan pengeluaran Budi selama liburan semester 5 minggu, dengan berbagai potongan dan alokasi dana untuk pajak, belanja kebutuhan, dan sedekah.

1. Langkah pertama adalah membuat 2 input yaitu gaji per jam yang diinginkan yang disimpan dalam variabel `gaji_perjam` dan jumlah jam kerja yang akan dilakukan dalam 1 minggu yang disimpan dalam variabel `jam_kerja_yang_diinginkan`. Program menggunakan fungsi `input()` untuk menampilkan pesan "gaji per jam yang diinginkan : Rp." Pada inputan pertama dan pesan "jumlah jam kerja 1 minggu : " pada inputan kedua.

```
soal2.3.py > ...  
1  gaji_perjam = int(input("gaji per jam yang diinginkan : Rp."))  
2  jam_kerja_perminggu = int(input("jumlah jam kerja 1 minggu : "))
```

2. Langkah kedua adalah menghitung total penghasilan Budi sebelum dipotong pajak. Perhitungan dilakukan dengan mengalikan gaji per jam dengan jumlah jam kerja per minggu, kemudian dikalikan 5 karena Budi bekerja selama 5 minggu. Hasil perhitungan disimpan dalam variabel `total_gaji_awal`.

```
4  total_gaji_awal = gaji_perjam * jam_kerja_perminggu * 5
```

3. Langkah ketiga adalah menghitung jumlah pajak yang harus dibayar Budi sebesar 14% dari total gaji awal. Perhitungan dilakukan dengan mengalikan total gaji awal dengan 0.14 (yang merupakan representasi desimal dari 14%). Nilai pajak disimpan dalam variabel `pajak`.

```
5  pajak = total_gaji_awal * 0.14
```

4. Langkah keempat adalah menghitung gaji bersih Budi setelah dipotong pajak dengan mengurangi jumlah pajak dari total gaji awal. Hasil perhitungan disimpan dalam variabel `gaji_setelah_pajak`.

```
6  gaji_setelah_pajak = total_gaji_awal - pajak
```

5. Langkah kelima adalah menghitung jumlah uang yang dihabiskan untuk membeli pakaian dan aksesoris sebesar 10% dari gaji setelah pajak. Perhitungan dilakukan

dengan mengalikan gaji setelah pajak dengan 0.10. Hasil disimpan dalam variabel `pengeluaran_aksesoris`.

```
7 pengeluaran_aksesoris = (gaji_setelah_pajak) * 0.10
```

6. Langkah keenam adalah menghitung jumlah uang yang dihabiskan untuk membeli alat tulis sebesar 1% dari gaji setelah pajak dengan mengalikan gaji setelah pajak dengan 0.01. Hasil disimpan dalam variabel `pengeluaran_alat_tulis`.

```
8 pengeluaran_alat_tulis = (gaji_setelah_pajak) * 0.01
```

7. Langkah ketujuh adalah menghitung total pengeluaran untuk belanja pakaian, aksesoris, dan alat tulis dengan menjumlahkan kedua variabel pengeluaran sebelumnya. Hasil disimpan dalam variabel `pengeluaran_tersier`.

```
9 pengeluaran_tersier = pengeluaran_aksesoris + pengeluaran_alat_tulis
```

8. Langkah kedelapan adalah menghitung jumlah uang yang akan disedekahkan sebesar 25% dari sisa uang setelah membeli pakaian, aksesoris, dan alat tulis. Perhitungan dilakukan dengan mengurangi pengeluaran tersier dari gaji setelah pajak, kemudian hasilnya dikalikan dengan 0.25. Nilai sedekah disimpan dalam variabel `sedekah`.

```
10 sedekah = (gaji_setelah_pajak - pengeluaran_tersier) * 0.25
```

9. Langkah kesembilan adalah menghitung jumlah uang sedekah yang akan diterima anak yatim sebesar 30% dari total sedekah. Perhitungan dilakukan dengan mengalikan jumlah sedekah dengan 0.30. Hasil disimpan dalam variabel `untuk_anak_yatim`.

```
11 untuk_anak_yatim = sedekah * 0.30
```

10. Langkah kesepuluh adalah menghitung jumlah uang sedekah yang akan diterima kaum dhuafa. Perhitungan dilakukan dengan mengurangi jumlah sedekah dengan jumlah sedekah untuk anak yatim. Hasil disimpan dalam variabel `untuk_kaum_dhuafa`.

```
12 untuk_kaum_dhuafa = sedekah - untuk_anak_yatim
```

11. Langkah kesebelas adalah menampilkan semua hasil perhitungan kepada pengguna.

Program menggunakan fungsi print() sebanyak tujuh kali untuk menampilkan informasi secara berurutan. Seperti pada gambar dibawah ini.

```
15 print("1. gaji sebelum pajak: Rp.",total_gaji_awal)
16 print("2. gaji setelah pajak: Rp.",gaji_setelah_pajak)
17 print("3. jumlah uang untuk membeli pakaian dan aksesoris: Rp.",pengeluaran_aksesoris)
18 print("4. jumlah uang untuk membeli alat tulis: Rp.",pengeluaran_alat_tulis)
19 print("5. jumlah uang sedekah: Rp.",sedekah)
20 print("6. jumlah uang yang akan diterima anak yatim: Rp.",untuk_anak_yatim)
21 print("7. Jumlah uang yang akan diterima kaum dhuafa: Rp.",untuk_kaum_dhuafa)]
```

Berikut adalah bentuk source code yang lengkap.

```
soal2.3.py > ...
1 gaji_perjam = int(input("gaji per jam yang diinginkan : Rp.))
2 jam_kerja_perminggu = int(input("jumlah jam kerja 1 minggu : "))
3
4 total_gaji_awal = gaji_perjam * jam_kerja_perminggu * 5
5 pajak = total_gaji_awal * 0.14
6 gaji_setelah_pajak = total_gaji_awal - pajak
7 pengeluaran_aksesoris = (gaji_setelah_pajak) * 0.10
8 pengeluaran_alat_tulis = (gaji_setelah_pajak) * 0.01
9 pengeluaran_tersier = pengeluaran_aksesoris + pengeluaran_alat_tulis
10 sedekah = (gaji_setelah_pajak - pengeluaran_tersier) * 0.25
11 untuk_anak_yatim = sedekah * 0.30
12 untuk_kaum_dhuafa = sedekah - untuk_anak_yatim
13
14
15 print("1. gaji sebelum pajak: Rp.",total_gaji_awal)
16 print("2. gaji setelah pajak: Rp.",gaji_setelah_pajak)
17 print("3. jumlah uang untuk membeli pakaian dan aksesoris: Rp.",pengeluaran_aksesoris)
18 print("4. jumlah uang untuk membeli alat tulis: Rp.",pengeluaran_alat_tulis)
19 print("5. jumlah uang sedekah: Rp.",sedekah)
20 print("6. jumlah uang yang akan diterima anak yatim: Rp.",untuk_anak_yatim)
21 print("7. Jumlah uang yang akan diterima kaum dhuafa: Rp.",untuk_kaum_dhuafa)]
```

Output:

```
PS C:\71251223_Semester2\71251223_priskila\minggu02> & C:\Users\USER\AppData\Local\Programs\Python\Python314\python.exe c:/71251223_Semester2/71251223_priskila/minggu02/soal2.3.py
gaji per jam yang diinginkan : Rp.50000
jumlah jam kerja 1 minggu : 35
1. gaji sebelum pajak: Rp. 8750000
2. gaji setelah pajak: Rp. 7525000.0
3. jumlah uang untuk membeli pakaian dan aksesoris: Rp. 752500.0
4. jumlah uang untuk membeli alat tulis: Rp. 75250.0
5. jumlah uang sedekah: Rp. 1674312.5
6. jumlah uang yang akan diterima anak yatim: Rp. 502293.75
7. Jumlah uang yang akan diterima kaum dhuafa: Rp. 1172018.75
PS C:\71251223_Semester2\71251223_priskila\minggu02>
```