



Laporan Praktikum Algoritma & Pemrograman

Semester Genap 2025/2026

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUMINI 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	71251235
Nama Lengkap	NATALIA GRECIA PUTRI
Minggu ke / Materi	02 / Variable, Expression dan Statements

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

BAGIAN 1: MATERI MINGGUINI (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.

Values dan type

Value (nilai) adalah elemen utama dalam sebuah program, yang dapat berupa huruf, angka, atau karakter lain. Contoh nilai yang umum digunakan meliputi bilangan seperti 1, 2, 3, atau teks seperti "Hello World". Setiap nilai memiliki tipe data tertentu, misalnya angka 2 bertipe integer, sedangkan "Hello World" bertipe string. Python interpreter secara otomatis mendeteksi tipe data untuk memproses ilia dengan tepat

Python mendukung beragam tipe data termasuk string (teks), integer (bilangan bulat), float (bilangan decimal), character (karakter tunggal), dan Boolean(nilai benar/salah).

Untuk memeriksa tipe data suatu nilai, gunakan fungsi type().

```
1 x = 20
2 print(x, "adalah", type(x))
3 y = 23.4
4 print(y, "adalah", type(y))
5 z = 7 + 8j
6 print(z, "adalah", type(z))
7
PROBLEMS 3 OUTPUT TERMINAL ...
PS E:\Praktek AlPro & C:/Users/Administr
/python.exe "e:/Praktek AlPro/python.py"
20 adalah <class 'int'>
23.4 adalah <class 'float'>
(7+8j) adalah <class 'complex'>
PS E:\Praktek AlPro>
```

Gambar 1.2: Contoh penggunaan fungsi type()

Saat menangani bilangan besar, Python memperbolehkan penggunaan underscore (_) sebagai pemisah ribuan untuk meningkatkan keterbacaan. Namun perlu hati-hati, kesalahan penulisan dapat menyebabkan Python menginterpretasikan angka tersebut sebagai beberapa parameter terpisah.

```
1 print(1_000_000) #Menampilkan angka 1000000 dengan pemisah agar lebih mudah dibaca
```

Gambar 1.3: Contoh penggunaan underscore (_)

Variabel

Variabel adalah konsep fundamental dalam pemrograman yang berfungsi sebagai wadah penyimpanan data. Secara teknis, variabel merupakan lokasi memori komputer yang dialokasikan untuk menyimpan berbagai tipe nilai. Ketika membuat variabel, Python secara otomatis mengalokasikan ruang memori yang diperlukan. Nilai yang tersimpan dalam variabel bersifat dinamis dan dapat dimodifikasi sesuai kebutuhan program.

Python menggunakan sistem tipe dinamis, artinya programmer tidak perlu mendeklarasikan tipe data variable secara eksplisit. Python akan secara otomatis mengenali tipe berdasarkan nilai yang diberikan. Untuk menampilkan isi variable ke layer, gunakan fungsi `print()`.

Nama Variabel dan Keywords

Penamaan variable di Python harus mengikuti konvensi tertentu agar program dapat berfungsi dengan baik. Berikut yang harus diperhatikan:

1. Karakter pertama harus berupa huruf atau underscore (_), contoh: `nama`, `_nama`, `namaSaya`, `data_variabel`.
2. Setelah karakter awal, variabel dapat mengandung kombinasi huruf, angka, atau underscore (_), misalnya: `_nilai`, `data1`, `angka_terakhir`.
3. Python bersifat case-sensitive dalam penamaan variabel, sehingga `VariabelKu` dan `variabel_ku` dianggap sebagai dua variabel yang berbeda.
4. Hindari penggunaan kata kunci (keyword) Python yang telah memiliki fungsi spesifik, seperti `if`, `while`, `for`, `return`, dan lainnya.

Python memiliki 35 kata kunci yang tidak dapat digunakan sebagai nama variabel karena telah direservasi untuk keperluan bahasa pemrograman.

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

Gambar 1.5: Nama variabel yang tidak dapat digunakan

Contoh penggunaan yang benar dan salah

```
1 #BENAR
2 nama_lengkap = "Katarina Elena"
3 usia1 = 18
4 _data_user = "Aktif"
5 nilaiX = 3.17
6
7 #SALAH
8 1nama = "Katarina"    #Tidak boleh diawali dengan angka
9 for = 17             #Tidak boleh menggunakan kata kunci Python
10 nama_lengkap = "Giselle" #Tidak boleh ada spasi
```

Gambar 1.6: Contoh nama variabel yang benar dan salah

Menggunakan kata kunci Python sebagai nama variabel akan menghasilkan error.

Oleh karena itu, penting memilih nama variabel yang sesuai dengan aturan agar kode dapat berjalan tanpa masalah

Statements

Statement dalam Python adalah unit kode yang dapat dieksekusi oleh interpreter.

Print merupakan salah satu contoh statement yang berfungsi menampilkan output.

Statement dapat berupa expression statement (pernyataan ekspresi) maupun assignment statement (pernyataan penugasan).

Ketika menggunakan mode interaktif Python, interpreter akan segera menjalankan perintah yang diinput dan menampilkan hasilnya. Sebaliknya, dalam mode skrip, program terdiri dari serangkaian pernyataan yang dieksekusi secara berurutan untuk membangun program yang lebih kompleks.

```
1 a = 7
2 b = 3
3 hasil = a + b
4 print("Hasil penjumlahan: ", hasil)
5

PROBLEMS 4 OUTPUT TERMINAL ...
PS E:\Praktek AlPro> & C:/Users/Administrator/AppData/Local/Programs/Python/Python38/python.exe "e:/Praktek AlPro/python.py"
Hasil penjumlahan: 10
PS E:\Praktek AlPro>
```

Gambar 1.7: Contoh statement dan outputnya

Operator dan Operand

Dalam Python, operator merupakan simbol yang digunakan untuk melakukan operasi aritmatika atau logika, sedangkan operand adalah nilai atau variabel yang menjadi subjek operasi. Sebagai ilustrasi, dalam operasi $2 + 3$, simbol $+$ adalah operator penjumlahan, sementara 2 dan 3 adalah operand.

Pembahasan ini secara khusus fokus pada operator aritmatika di Python yang digunakan untuk operasi matematika seperti penjumlahan, pengurangan, perkalian, pembagian, dan pemangkatan.

Expressions

Ekspresi dalam Python adalah gabungan dari nilai (values), variabel (variables), dan operator (operators) yang dievaluasi untuk menghasilkan suatu nilai. Sebuah nilai tunggal dapat dianggap sebagai ekspresi, begitu pula dengan variabel.

```
1  17      # Nilai tunggal, ini juga merupakan ekspresi
2  x      # Variabel juga merupakan ekspresi
3  x + 17  # Kombinasi variabel dan operator membentuk ekspresi
```

Gambar 1.8: Contoh penggunaan expressions (ekspresi)

Ketika sebuah ekspresi dijalankan dalam mode interaktif (interactive mode), Python akan langsung mengevaluasi dan menampilkan hasilnya. Contohnya:

Urutan Operasi

Dalam Python, urutan operasi diterapkan ketika suatu ekspresi mengandung lebih dari satu operator. Urutan ini mengikuti hirarki prioritas operator yang dikenal dengan akronim PEMDAS (Parentheses, Exponentiation, Multiplication and Division, Addition and Subtraction).

Hierarki operator dalam Python:

1. Parentheses (Tanda Kurung)

Memiliki prioritas tertinggi dan berfungsi untuk mengelompokkan ekspresi agar dievaluasi terlebih dahulu.

2. Exponentiation (Pemangkatan)

Memiliki prioritas tertinggi kedua setelah tanda kurung.

3. Multiplication and Division (Perkalian dan Pembagian)

Memiliki prioritas lebih tinggi dibanding penjumlahan dan pengurangan. Ketika terdapat beberapa operator dengan prioritas sama, eksekusi dilakukan dari kiri ke kanan.

4. Addition and Subtraction (Penjumlahan dan Pengurangan)

Operator ini memiliki prioritas yang setara dan dieksekusi dari kiri ke kanan.

Apabila terjadi ketidakjelasan dalam perhitungan, gunakan tanda kurung untuk memastikan urutan operasi sesuai dengan yang diinginkan.

Operator Modulus dan String Modulus

Modulus

Operator modulus (%) berfungsi untuk mendapatkan sisa pembagian dari dua bilangan. Operator ini khusus digunakan untuk tipe data integer.

Beberapa contoh penggunaan modulus meliputi: menentukan kelipatan (jika $x \% y == 0$, berarti x habis dibagi y) dan mengekstrak digit terakhir dari suatu bilangan.

String

Operator + dan * memiliki perilaku khusus ketika diterapkan pada string.

Operator + (Penggabungan String)

Operator + tidak melakukan penjumlahan matematis pada string, melainkan menggabungkannya.

Menangani input dan Pengguna

Dalam pemrograman, alur kerja fundamental mengikuti pola Input - Proses - Output:

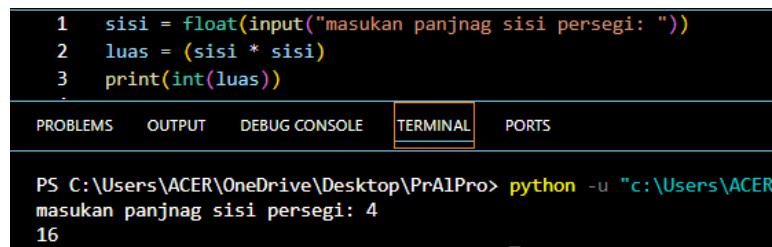
- Input: Data atau masukan yang diperlukan untuk menjalankan program.
- Proses: Serangkaian langkah yang dilakukan program untuk menyelesaikan masalah.
- Output: Hasil akhir setelah menjalankan proses.

Contoh: Menghitung Luas Persegi

Sistem sederhana yang mengikuti pola Input - Proses - Output adalah perhitungan luas persegi.

- Input
Pengguna memasukkan panjang sisi persegi.
- Proses
Sistem menghitung luas dengan rumus: luas = sisi × sisi.
- Output
Sistem menampilkan hasil luas persegi.

Contoh Program Python:



```

1 sisi = float(input("masukan panjang sisi persegi: "))
2 luas = (sisi * sisi)
3 print(int(luas))

```

The screenshot shows a terminal window with the following content:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ACER\OneDrive\Desktop\PrA1Pro> python -u "c:\Users\ACER\OneDrive\Desktop\PrA1Pro\persegipanjang.py"
masukan panjang sisi persegi: 4
16

```

Gambar 1.11:

Penjelasan:

- Pengguna memasukkan panjang sisi (Input).
- Program menghitung luas dengan rumus sisi × sisi (Proses).
- Program menampilkan hasil luas (Output).

Komentar

Dalam bahasa Python, simbol pagar (#) digunakan untuk menandai komentar dalam kode. Komentar tidak akan dieksekusi oleh interpreter Python dan berfungsi sebagai catatan atau dokumentasi untuk membantu programmer memahami kode dengan lebih mudah.

Python tidak memiliki sintaks khusus untuk komentar multi-baris seperti beberapa bahasa pemrograman lain. Namun, kita dapat menggunakan simbol pagar pada setiap baris komentar.

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.

SOAL 1

https://github.com/grec12/71251235_nataliagrecia.git

Program yang dapat menghitung berat badan yang diperlukan, jika diketahui tinggi badan dan nilai Body Mass Index (BMI) yang diharapkan.

```
1 def hitung_berat_badan(tinggi, bmi_target):
2     berat_diperlukan = bmi_target * (tinggi ** 2)
3     return berat_diperlukan
4
5 tinggi = float(input("Masukkan tinggi badan (m): "))
6 bmi_target = float(input("Masukkan nilai BMI yang diharapkan: "))
7
8 berat_diperlukan = hitung_berat_badan(tinggi, bmi_target)
9
10 print(f"Berat badan yang diperlukan agar mencapai BMI {bmi_target} adalah {berat_diperlukan: .2f} kg")
11
```

PROBLEMS 27 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 27
PS E:\PrAIPro\Winggu2> & C:/Users/Administrator/AppData/Local/Programs/Python/Python314/python.exe e:/PrAIPro/Minggu2/LAT
Masukkan tinggi badan (m): 1.7
Masukkan nilai BMI yang diharapkan: 20
Berat badan yang diperlukan agar mencapai BMI 20.0 adalah 57.80 kg
PS E:\PrAIPro\Winggu2>

Gambar 1.12: Body Mass Index (BMI)

Penjelasan kode:

1. Fungsi untuk Menghitung Berat Badan

```
1 def hitung_berat_badan(tinggi, bmi_target):
2     berat_diperlukan = bmi_target * (tinggi ** 2)
3     return berat_diperlukan
```

Gambar 1.13: Fungsi menghitung berat badan

- Fungsi `hitung_berat_badan()` menerima dua parameter:
 1. Tinggi: badan dalam meter.
 2. `bmi_target`: nilai BMI yang diharapkan.
- Menggunakan rumus BMI untuk mengitung berat badan yang diperlukan.
- Hasilnya dikembalikan (return).

2. Menerima Input dari Pengguna

```
5 tinggi = float(input("Masukkan tinggi badan (m): "))
6 bmi_target = float(input("Masukkan nilai BMI yang diharapkan: "))
```

Gambar 1.14: Input

- Program meminta pengguna untuk memasukkan tinggi badan dalam satuan cm, kemudian mengonversinya ke meter dengan membagi 100.

- Program juga meminta input nilai BMI yang diharapkan dari pengguna.

3. Pemanggilan Fungsi Perhitungan

```
8     berat_diperlukan = hitung_berat_badan(tinggi, bmi_target)
```

Gambar 1.15: Memanggil fungsi

- Program memanggil fungsi hitung_berat_badan() dengan menggunakan nilai tinggi badan dan BMI yang telah diinput pengguna.
- Hasil perhitungan disimpan dalam variabel berat_diperlukan.

4. Menampilkan Hasil

```
10    print(f"Berat badan yang diperlukan agar mencapai BMI {bmi_target} adalah {berat_diperlukan: .2f} kg")
```

Gambar 1.16: menampilkan hasil

Hasil ditampilkan dengan format dua angka di belakang koma (.2f) untuk presisi yang baik.

SOAL 2

Membuat program untuk menghitung hasil dari fungsi $f(x) = 2x^3 + 2x + 15/x$, dengan x merupakan bilangan bulat yang diinput oleh pengguna.

```
1 x = int(input("Masukkan nilai x: "))
2
3 if x == 0:
4     print("Error: x tidak boleh 0 karena akan menyebabkan pembagian dengan nol.")
5 else:
6     f_x = 2 * x**3 + 2 * x + (15 / x)
7
8     print(f"Hasil f({x}) = {f_x:.2f}")

PROBLEMS 36 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 36

PS E:\PrAlPro\Minggu2> & C:/Users/Administrator/AppData/Local/Programs/Python/Python314/python.exe
Masukkan nilai x: 7
Hasil f(7) = 702.14
PS E:\PrAlPro\Minggu2>
```

Gambar 1.17: menghitung hasil dari fungsi $f(x) = 2x^3 + 2x + 15/x$

Penjelasan code:

1. Input dari Pengguna

```
1 x = int(input("Masukkan nilai x: "))
2
```

Gambar 1.18: input

Program meminta pengguna untuk memasukkan bilangan bulat x.

2. Variabel Input (x tidak boleh 0)

```
2
3 if x == 0:
4     print("Error: x tidak boleh 0 karena akan menyebabkan pembagian dengan nol.")
```

Gambar 1.19: variabel input

Karena terdapat operasi pembagian $15/x$ dalam rumus, nilai x tidak boleh nol untuk menghindari error pembagian dengan nol.

3. Perhitungan Nilai $f(x)$

```
5 else:
6     f_x = 2 * x**3 + 2 * x + (15 / x)
7
```

Gambar 1.20: menghitung nilai $f(x)$

Mengimplementasikan rumus yang diberikan untuk menghitung nilai fungsi.

4. Menampilkan Hasil $f(x)$

```
7
8     print(f"Hasil f({x}) = {f_x:.2f}")
9
```

Gambar 1.21: menampilkan hasil

Format dengan dua angka desimal (:.2f) digunakan agar output lebih rapi dan mudah dibaca.

SOAL 3

Membuat program untuk menghitung distribusi keuangan Budi berdasarkan gaji per jam dan jumlah jam kerja per minggu.

```
1 gaji_per_jam = float(input("Masukkan gaji per jam: "))
2 jam_per_minggu = int(input("Masukkan jumlah jam kerja per minggu: "))
3
4 pendapatan_kotor = gaji_per_jam * jam_per_minggu * 4
5
6 pajak = pendapatan_kotor * 0.14
7 pendapatan_bersih = pendapatan_kotor - pajak
8
9 biaya_pakaian = pendapatan_bersih * 0.10
10 biaya_alat_tulis = pendapatan_bersih * 0.01
11
12 sisa_uang = pendapatan_bersih - (biaya_pakaian + biaya_alat_tulis)
13
14 sedekah = sisa_uang * 0.25
15 anak_yatim = sedekah * 0.30
16 kaum_dhuafa = sedekah * 0.70
17
18 print("\nHasil Perhitungan Keuangan Budi:")
19 print(f"1. Pendapatan sebelum pajak: Rp{pendapatan_kotor:.2f}")
20 print(f"2. Pendapatan setelah pajak: Rp{pendapatan_bersih:.2f}")
21 print(f"3. Biaya pakaian dan aksesoris: Rp{biaya_pakaian:.2f}")
22 print(f"4. Biaya alat tulis: Rp{biaya_alat_tulis:.2f}")
23 print(f"5. Total sedekah: Rp{sedekah:.2f}")
24 print(f"6. Uang untuk anak yatim: Rp{anak_yatim:.2f}")
25 print(f"7. Uang untuk kaum dhuafa: Rp{kaum_dhuafa:.2f}")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

PS C:\Users\ACER\OneDrive\Desktop\PrAIPro> python -u "c:\Users\ACER\OneDrive\Desktop\PrAIPro\Soal3.py"
Masukkan gaji per jam: 50.000
Masukkan jumlah jam kerja per minggu: 40

Hasil Perhitungan Keuangan Budi:
1. Pendapatan sebelum pajak: Rp8,000.00
2. Pendapatan setelah pajak: Rp6,880.00
3. Biaya pakaian dan aksesoris: Rp688.00
4. Biaya alat tulis: Rp68.00
5. Total sedekah: Rp1,530.00
6. Uang untuk anak yatim: Rp459.24
7. Uang untuk kaum dhuafa: Rp1,071.56

Gambar 1.22: menghitung distribusi keuangan Budi

Penjelasan Code:

1. Input dari Pengguna

```
1 gaji_per_jam = float(input("Masukkan gaji per jam: "))
2 jam_per_minggu = float(input("Masukkan jumlah jam kerja per minggu: "))
3
```

Gambar 1.23: Input

Program meminta pengguna untuk memasukkan gaji per jam dan total jam kerja dalam seminggu.

Variabel gaji_per_jam menggunakan tipe float karena gaji dapat berupa bilangan desimal. Variabel jam_per_minggu menggunakan tipe int karena jam kerja umumnya bilangan bulat.

2. Perhitungan Pendapatan Kotor

```
4 pendapatan_kotor = gaji_per_jam * jam_per_minggu * 4
```

Gambar 1.24: Hitung pendapatan kotor

Pendapatan kotor diperoleh dengan mengalikan gaji per jam dengan jumlah jam kerja per minggu, kemudian dikalikan 4 (asumsi satu bulan = 4 minggu).

3. Perhitungan Pajak dan Pendapatan Bersih

```
5  
6 pajak = pendapatan_kotor * 0.14  
7 pendapatan_bersih = pendapatan_kotor - pajak  
8
```

Gambar 1.25: Hitung pajak dan pendapatan kotor

Pajak dihitung sebesar 14% dari pendapatan kotor. Pendapatan bersih diperoleh dengan mengurangkan pajak dari pendapatan kotor.

4. Perhitungan Pengeluaran Kebutuhan

```
8  
9 biaya_pakaian = pendapatan_bersih * 0.10  
10 biaya_alat_tulis = pendapatan_bersih * 0.01  
11
```

Gambar 1.26: hitung pengeluaran kebutuhan

Biaya pakaian dan aksesoris: 10% dari pendapatan bersih.

Biaya alat tulis: 1% dari pendapatan bersih.

5. Perhitungan Sisa Uang

```
12 sisa_uang = pendapatan_bersih - (biaya_pakaian + biaya_alat_tulis)  
13
```

Gambar 1.27: hitung sisa

Sisa uang dihitung setelah dikurangi pengeluaran untuk pakaian dan alat tulis.

6. Perhitungan Sedekah dan Distribusinya

```
13  
14 sedekah = sisa_uang * 0.25  
15 anak_yatim = sedekah * 0.30  
16 kaum_dhuafa = sedekah * 0.70  
17
```

Gambar 1.28: hitung sedekah dan distribusi

25% dari sisa uang dialokasikan untuk sedekah.

Dari total sedekah, 30% diberikan kepada anak yatim dan 70% kepada kaum dhuafa.

7. Menampilkan Hasil Perhitungan

```
18 print("\nHasil Perhitungan Keuangan Budi: ")
19 print(f"1. Pendapatan sebelum pajak: Rp{pendapatan_kotor:,.2f}")
20 print(f"2. Pendapatan setelah pajak: Rp{pendapatan_bersih:,.2f}")
21 print(f"3. Biaya pakaian dan aksesoris: Rp{biaya_pakaian:,.2f}")
22 print(f"4. Biaya alat tulis: Rp{biaya_alat_tulis:,.2f}")
23 print(f"5. Total sedekah: Rp{sedekah:,.2f}")
24 print(f"6. Uang untuk anak yatim: Rp{anak_yatim:,.2f}")
25 print(f"7. Uang untuk kaum dhuafa: Rp{kaum_dhuafa:,.2f}")
```

Gambar 1.29: menampilkan hasil perhitungan

Program mencetak semua hasil perhitungan dengan format dua desimal (2f), memastikan angka ditampilkan dengan dua digit di belakang koma untuk presisi yang konsisten.