



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	71251210
Nama Lengkap	Gabriel Ekklesia
Minggu ke / Materi	02 / Variable Expression & Statement

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

BAGIAN 1: MATERI MINGGU INI (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 adalah komponen utama dari program, salah satunya huruf atau angka.

Value terbagi menjadi beberapa tipe yang berbeda, 2 adalah sebuah nilai integer dan "Hallo Word" adalah sebuah nilai string. Interpreter dapat melakukan identifikasi terhadap string karena pada penulisannya ditutup menggunakan tanda petik (quotation mark). 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. Python menyediakan fungsi built-in untuk melakukan pengecekan tipe data pada value dengan menggunakan fungsi `type()`.

Ketika kita menggunakan bilangan bulat besar, beberapa model penulisan akan menggunakan tanda koma (,) diantara kelompok tiga digit.

```
>>> 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">
```

Variabel

Salah satu fitur powerfull dalam bahasa pemrograman adalah kemampuannya yang berfungsi untuk manipulasi variable. Variable merupakan lokasi memori yang dicadangkan untuk menyimpan nilai-nilai. Variable menyimpan data yang dilakukan selama program dieksekusi, yang nantinya isi dari variabel tersebut dapat diubah oleh operasi - operasi tertentu pada program yang menggunakan variable. Selain itu, penggunaan variabel yang tepat akan

memudahkan pengelolaan data, meningkatkan fleksibilitas program, dan membuat kode lebih mudah dipahami serta dikembangkan.

Variable dapat menyimpan berbagai macam tipe data. Variable mempunyai sifat yang dinamis, artinya variable Python tidak perlu dideklarasikan tipe data dan variable Python dapat diubah saat program dijalankan.

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

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

Nama Variabel dan Keywords

Pemberian nama pada variabel berikut ini:

1. Nama variable boleh diawali menggunakan huruf atau garis bawah (_).
2. Karakter selanjutnya dapat berupa huruf, garis bawah (_) atau angka
3. Karakter pada nama variable bersifat sensitif (case-sensitif). Artinya huruf besar dan kecil dibedakan.
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.

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

Statements merupakan bagian dari code interpreter Python yang dapat dieksekusi. 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$ █

```

Operator dan Operand

Operator banyak digunakan dalam pemrograman untuk memproses data, seperti penjumlahan, pengurangan, perbandingan nilai, hingga pengambilan keputusan dalam alur program. Dengan adanya operator, program dapat melakukan perhitungan dan menghasilkan keluaran sesuai dengan logika yang telah ditentukan. Operator aritmatika adalah operator yang digunakan untuk melakukan operasi matematika, seperti penjumlahan, pengurangan, perkalian, pembagian, dan sebagainya. Tabel berikut menunjukkan jenis operator aritmatika.

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
**	Pemangkat, memangkatkan bilangan	X ** y

Expressions

Expression merupakan representasi dari suatu nilai dalam pemrograman yang dapat tersusun dari kombinasi value (nilai), variabel, dan operator. Sebuah value secara mandiri sudah dapat disebut sebagai expression, demikian pula variabel yang menyimpan nilai tertentu. Secara umum, setiap bentuk yang menghasilkan nilai ketika dievaluasi oleh program dapat dikategorikan sebagai expression. Dengan demikian, expression berperan penting dalam proses pengolahan data karena digunakan untuk melakukan perhitungan, perbandingan, maupun operasi logika dalam suatu program.

Urutan Operasi

Urutan operasi digunakan ketika sebuah expression memiliki lebih dari satu operator. Python mengikuti aturan prioritas matematika yang dikenal dengan PEMDAS, yaitu Paratheses (tanda kurung), Exponentiation (pemangkatan), Multiplication and Division (perkalian dan pembagian), serta Addition and Subtraction (penjumlahan dan pengurangan). Tanda kurung memiliki prioritas tertinggi dan digunakan untuk menentukan urutan perhitungan secara eksplisit serta mempermudah pembacaan expression. Pemangkatan diproses setelah tanda kurung, diikuti oleh perkalian dan pembagian yang memiliki tingkat prioritas sama, kemudian penjumlahan dan pengurangan. Jika operator memiliki prioritas yang sama, maka perhitungan dilakukan dari kiri ke kanan. Aturan ini memastikan setiap expression dievaluasi secara konsisten dan menghasilkan nilai yang tepat.

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 (%).

2. String

Operator + ketika bekerja dengan string tidak berarti penjumlahan secara matematika, melainkan penggabungan antar string.

Menangani Input dari Pengguna

Python menyediakan fasilitas untuk menerima masukan dari pengguna melalui fungsi bawaan `input()`. Fungsi ini digunakan untuk mengambil data berupa teks yang diketik melalui keyboard. Ketika `input()` dijalankan, program akan berhenti sementara dan menunggu pengguna memasukkan data. Setelah pengguna menekan tombol Enter, program akan melanjutkan eksekusi dan nilai yang dimasukkan akan dikembalikan dalam bentuk string. Fitur ini memungkinkan program berinteraksi langsung dengan pengguna dalam proses pengolahan data.

Sebelum menerima masukan dari pengguna, sebaiknya program menampilkan prompt yang berfungsi sebagai petunjuk mengenai data yang harus diinput. Prompt tersebut dapat diberikan sebagai argumen pada fungsi `input()` sehingga akan ditampilkan terlebih dahulu sebelum program menunggu masukan dari pengguna. Penambahan karakter `\n` pada akhir prompt digunakan untuk membuat baris baru, sehingga teks yang diketik oleh pengguna akan muncul pada baris di bawah pesan petunjuk. Hal ini membantu meningkatkan kejelasan dan kenyamanan dalam interaksi antara pengguna dan program.

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.

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

```
#latihan 2.1

#mengambil input
tinggi_m = float(input("masukan tinggi badan (meter): "))
target_bmi = float(input("masukan nilai bmi: "))

#perhitungan
berat_yang_diperlukan = target_bmi * (tinggi_m ** 2)

# hasil
print("-" * 35)
print(f"mencari BMI {target_bmi},")
print(f"berat badan yang diperlukan adalah: {berat_yang_diperlukan:.2f} kg")
```

Bagian mengambil input digunakan untuk meminta data dari pengguna.

`tinggi_m = float(input("masukan tinggi badan (meter): "))` meminta pengguna memasukkan tinggi badan dalam meter, lalu mengubahnya menjadi tipe data desimal (`float`).

`target_bmi = float(input("masukan nilai bmi: "))` meminta nilai BMI yang diinginkan dan juga mengubahnya menjadi bilangan desimal.

Bagian perhitungan melakukan proses utama program.

`berat_yang_diperlukan = target_bmi * (tinggi_m ** 2)` menghitung berat badan menggunakan rumus BMI. Tinggi badan dipangkatkan dua terlebih dahulu, kemudian dikalikan dengan nilai BMI target.

Bagian hasil menampilkan output kepada pengguna.

`print("-" * 35)` mencetak garis pemisah agar tampilan lebih rapi.

`print(f"mencari BMI {target_bmi},")` menampilkan nilai BMI yang ingin dicapai.

`print(f"berat badan yang diperlukan adalah {berat_yang_diperlukan:.2f} kg")` menampilkan berat badan yang harus dimiliki. Format `:.2f` membuat hasil ditampilkan dengan dua angka di belakang koma.

SOAL 2

```
#latihan 2.2

#mengambil input
x = int(input("masukan nilai x (bilangan bulat): "))

#perhitungan
if x != 0:
    hasil = (2 * (x ** 3)) + (2*x) + (15 / x)

#hasil
    print(f"hasil dari f({x}) adalah {hasil}")
else:
    print("error: nilai x tidak boleh 0 karena akan terjadi pembagian dari nol")
```

Program tersebut digunakan untuk menghitung nilai suatu fungsi matematika berdasarkan nilai x yang dimasukkan pengguna, dengan syarat x tidak boleh bernilai 0.

Pada bagian mengambil input, program meminta pengguna memasukkan nilai x berupa bilangan bulat melalui keyboard. Input tersebut kemudian diubah menjadi tipe data integer dengan fungsi `int()` agar dapat digunakan dalam perhitungan matematika.

Pada bagian perhitungan, program menggunakan percabangan `if`. Kondisi `if x != 0` digunakan untuk memastikan bahwa nilai x bukan nol. Hal ini penting karena pada rumus terdapat operasi pembagian $15 / x$, yang tidak dapat dilakukan jika x bernilai 0. Jika syarat terpenuhi, program menghitung nilai fungsi: $2 \times x^3 + 2x + 15/x$

Hasil perhitungan disimpan dalam variabel `hasil`.

Pada bagian hasil, program menampilkan nilai fungsi yang telah dihitung dalam bentuk kalimat. Jika pengguna memasukkan nilai `x = 0`, program tidak melakukan perhitungan, melainkan menampilkan pesan kesalahan yang menjelaskan bahwa pembagian dengan nol tidak diperbolehkan.

Secara keseluruhan, program ini menerima input, memeriksa validitas nilai, melakukan perhitungan fungsi matematika, dan menampilkan hasil atau pesan kesalahan sesuai kondisi input.

SOAL 3

```
#latihan 2.3

#mengambil input
gaji_per_jam = float(input("masukan gaji per jam yang diinginkan: "))
jam_kerja_minggu = float(input("masukan jam kerja per minggu: "))

#perhitungan
pendapatan_bruto = gaji_per_jam * jam_kerja_minggu * 5

#pendapatan netto
pajak = 0.14 * pendapatan_bruto
pendapatan_bersih = pendapatan_bruto - pajak

#belanja baju dan aksesoris (10% dari pendapatan bersih)
uang_baju = 0.01 * pendapatan_bersih

#belanja alat tulis (1% dari pendapatan bersih)
uang_belanja_alat_tulis = 0.01 * pendapatan_bersih

#sedekah (25% dari sisa uang belanja)
sisa_uang_belanja = pendapatan_bersih - uang_baju - uang_belanja_alat_tulis
total_sedekah = 0.25 * sisa_uang_belanja

#pembagian sedekah
uang_anak_yatim = 0.30 * total_sedekah
uang_kaum_dhuafa = total_sedekah - uang_anak_yatim
```

Program ini digunakan untuk menghitung pengelolaan pendapatan seseorang berdasarkan gaji per jam, kemudian membaginya ke beberapa kebutuhan dan sedekah.

Pada bagian mengambil input, program meminta pengguna memasukkan gaji per jam dan jumlah jam kerja per minggu. Nilai yang dimasukkan diubah menjadi tipe data desimal (float) agar dapat digunakan dalam perhitungan.

Pada bagian perhitungan pendapatan bruto, program menghitung total pendapatan kotor dengan mengalikan gaji per jam, jam kerja per minggu, dan 5. Angka 5 menunjukkan perhitungan pendapatan selama lima minggu.

Pada bagian pendapatan netto, program menghitung pajak sebesar 14% dari pendapatan bruto. Setelah itu, pendapatan bersih diperoleh dengan mengurangi pendapatan bruto dengan pajak.

Selanjutnya program mengatur alokasi pengeluaran dari pendapatan bersih. Sebagian pendapatan digunakan untuk belanja baju dan aksesoris, serta untuk membeli alat tulis. Masing-masing dihitung sebagai persentase dari pendapatan bersih. Setelah kedua pengeluaran tersebut dikurangi, diperoleh sisa uang belanja.

Pada bagian sedekah, program menghitung jumlah sedekah sebesar 25% dari sisa uang belanja. Nilai sedekah kemudian dibagi menjadi dua bagian, yaitu untuk anak yatim sebesar 30% dari total sedekah, dan sisanya untuk kaum dhuafa.

Secara keseluruhan, program ini menerima data pendapatan, menghitung pajak, mengatur alokasi pengeluaran, menentukan jumlah sedekah, dan membagi sedekah sesuai proporsi yang telah ditentukan.

```
print("\n" + "="*40)
print(f" pendapatan bruto budi : Rp {pendapatan_bruto:,.2f}")
print(f" pendapatan netto setelah pajak : Rp {pendapatan_bersih:,.2f}")
print(f" uang untuk belanja untuk membeli pakaian : Rp {uang_baju:,.2f}")
print(f" uang untuk membeli alat tulis : Rp {uang_belanja_alat_tulis:,.2f}")
print(f" total uang yang disedekahkan : RP {total_sedekah:,.2f}")
print(f" total sedekah untuk anak yatim : Rp {uang_anak_yatim:,.2f}")
print(f" uang sedekah untuk kaum dhufan : Rp {uang_kaum_dhuafa:,.2f}")
print("="*40)
```

Baris `print("\n" + "="*40)` mencetak baris kosong lalu menampilkan garis pemisah sepanjang 40 karakter tanda sama dengan. Tujuannya agar tampilan output lebih jelas dan terpisah dari proses sebelumnya.

Baris-baris berikutnya menampilkan rincian hasil perhitungan pendapatan dan alokasi uang. Program menampilkan pendapatan bruto, pendapatan bersih setelah pajak, jumlah uang untuk membeli pakaian, uang untuk membeli alat tulis, total sedekah, serta pembagian sedekah untuk anak yatim dan kaum dhuafa.

Pada setiap baris digunakan format `:.2f` yang berfungsi menampilkan angka dengan dua digit di belakang koma. Penulisan `Rp` menunjukkan bahwa nilai yang ditampilkan menggunakan satuan mata uang rupiah.

Baris terakhir `print("="*40)` kembali mencetak garis pemisah sebagai penutup tampilan hasil. Secara keseluruhan, bagian kode ini hanya berfungsi untuk menampilkan hasil perhitungan dengan format yang terstruktur dan mudah dibaca.

LINK GITHUB: https://github.com/gilbetjuliesa-crypto/71251210_Gabriel_Ekklesia.git