



Laporan Praktikum Algoritma dan Pemrograman

NIM	71220933
Nama Lengkap	Yosep Yoga Jalu Pamungkas
Minggu ke / Materi	02 / Variable, Expression dan Statements

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 I

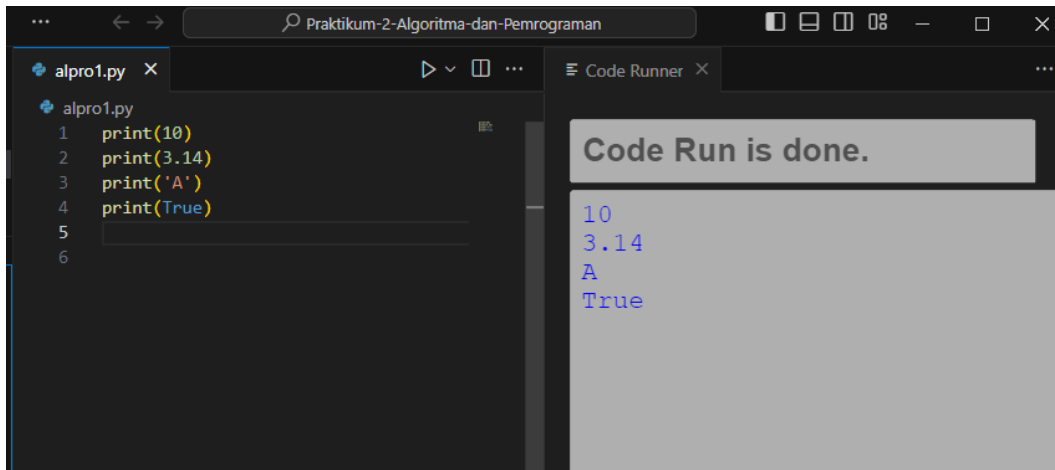
MATERI 1

Value merupakan komponen seperti huruf dan angka, value biasanya menggunakan 1,2,'a','z', dan "Hello world" value dibagi menjadi beberapa tipe misalnya 2 untuk nilai integer dan "Hello World" untuk sebuah nilai string.

Jenis nilai yang disimpan, seperti integer,float,string,boolean,dll.

Contoh lain komponen:

- integer: '2', '10', '100'
- Float: '3.14', '2.5', '10.0'

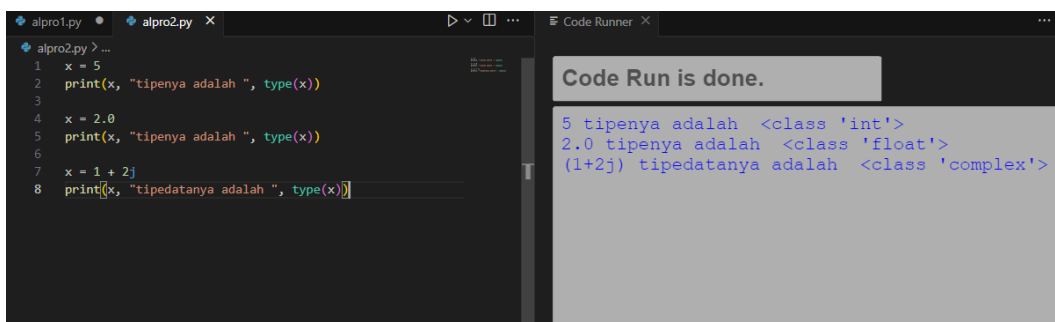


```
alpro1.py
1 print(10)
2 print(3.14)
3 print('A')
4 print(True)
5
6
```

Code Run is done.

```
10
3.14
A
True
```

Contoh perintah 'print' digunakan untuk menampilkan nilai dari berbagai tipe data, termasuk integer, float, karakter, dan boolean. Outputnya akan sesuai dengan nilai yang diberikan dalam argumen print.



```
alpro2.py
1 x = 5
2 print(x, "tipenya adalah ", type(x))
3
4 x = 2.0
5 print(x, "tipenya adalah ", type(x))
6
7 x = 1 + 2j
8 print(x, "tipenya adalah ", type(x))
```

Code Run is done.

```
5 tipenya adalah <class 'int'>
2.0 tipenya adalah <class 'float'>
(1+2j) tipenya adalah <class 'complex'>
```

Contoh di atas, fungsi 'type()' digunakan untuk mengetahui tipe data dari masing-masing nilai. Hasilnya akan mencetak jenis tipe data dari nilai tersebut.

```
alpro3_bilangan_bulat.py > ...
1 large_number_without_commas = 1000000000
2 print(large_number_without_commas)
3
4 large_number_with_commas = 1_000_000_000
5 print(large_number_with_commas)
6
```

Code Run is done.

```
1000000000
1000000000
```

Contoh variable 'large_number_without_commas' dan 'large_number_with_commas' memiliki nilai yang sama, yaitu 1 miliar.

Python mengabaikan Tanda koma saat mengevaluasi ekspresi yang melibatkan bilangan bulat.

MATERI 2

Python, variabel dapat dianggap sebagai "label" yang terhubung dengan lokasi memori yang berisi nilai tertentu. Tipe data ke variabel berdasarkan nilai yang diberikan, sehingga tidak perlu mendeklarasikan tipe data secara eksplisit.

```
alpro4_Variabel_Jelbel.py > ...
1 # Variabel x bertipe integer
2 x = 10
3 print(x) # Output: 10
4
5 # Mengubah tipe data variabel x menjadi string
6 x = "Hello"
7 print(x) # Output: Hello
8
9 # Variabel y bertipe float
10 y = 3.14
11 print(y) # Output: 3.14
12
13 # Mengubah nilai variabel y menjadi boolean
14 y = True
15 print(y) # Output: True
16
```

Code Run is done.

```
10
Hello
3.14
True
```

MATERI 3

Penamaan variabel dengan Python mengikuti pedoman berikut:

1. Nama variabel boleh diawali dengan huruf atau garis bawah (_).
2. Karakter selanjutnya dapat berupa huruf, garis bawah (_), atau angka.
3. Nama variabel peka huruf besar-kecil.
4. Hindari menggunakan kata kunci Python seperti if, while, for, dll

```
penamaan_Variabel.py > jumlah_mahasiswa
1 # Contoh penggunaan variabel yang sesuai dengan pedoman penamaan
2 nama = "Yoga" # variabel diawali dengan huruf
3 umur = 20 # variabel diawali dengan garis bawah
4 alamat_rumah = "Jalan jalan3" # variabel terdiri dari huruf
5 jumlah_mahasiswa = 30 # variabel dapat terdiri dari huruf dan angka
6 total_Pendapatan = 5000.50 # variabel sensitif terhadap huruf
7
8 print(nama)
9 print(umur)
10 print(alamat_rumah)
11 print(jumlah_mahasiswa)
12 print(total_Pendapatan)
13
```

Code Run is done.

```
Yoga
20
Jalan jalan3
30
5000.5
```

Contoh penggunaan variabel dengan Python.

MATERI 4

nilai ke variabel ($x = 5$, $y = 3$), menghitung ekspresi ($\text{jumlah} = x + y$), dan menampilkan hasilnya ke layar (`print`). Semua pernyataan ini dapat dieksekusi secara berurutan ketika program dijalankan.

```
alpro.py > ...
1 x = 5 # Ini adalah pernyataan penugasan, mengatur nilai variabel x menjadi 5
2 y = 3 # Pernyataan penugasan lainnya, mengatur nilai variabel y menjadi 3
3
4 # Pernyataan ekspresi, menghitung hasil penjumlahan variabel x dan y
5 jumlah = x + y
6
7 # Pernyataan print, menampilkan hasil penjumlahan ke layar
8 print("Hasil penjumlahan x dan y adalah:", jumlah)
9
```

Code Run is done.

```
Hasil penjumlahan x dan y adalah: 8
```

MATERI 5

+ (penjumlahan): Menjumlahkan dua nilai.

Contoh: $x + y$

- (pengurangan): Mengurangkan nilai kedua dari nilai pertama.

Contoh: $x - y$

* (perkalian): Mengalikan dua nilai.

Contoh: $x * y$

/ (pembagian): Membagi nilai pertama dengan nilai kedua.

Contoh: x / y

** (pangkat): Menaikkan nilai pertama ke pangkat nilai kedua.

Contoh: $x ** y$

Dengan menggunakan operator-operator ini, kita dapat melakukan berbagai operasi matematika dalam bahasa pemrograman Python. Misalnya, untuk menambahkan dua bilangan, kita menggunakan operator +, untuk mengalikan dua bilangan, kita menggunakan operator *, dan seterusnya.

MATERI 6

Dalam mode interaktif seperti interpreter Python, penggunaan ekspresi melibatkan penilaian oleh interpreter yang kemudian menampilkan hasilnya. Sebagai contoh, ekspresi matematika seperti $1 + 1$ akan dievaluasi menjadi 2, dan ekspresi $3 + 2$ akan dievaluasi menjadi 5.

Contohnya:

```
alpro1.py > ...
1  x = 1 # Variabel x diberi nilai 1
2  y = 2 # Variabel y diberi nilai 2
3
4  # Contoh ekspresi
5  hasil1 = x + y # Ekspresi penjumlahan, hasilnya adalah 3
6  hasil2 = x * y # Ekspresi perkalian, hasilnya adalah 2
7
8  print(hasil1) # Output: 3
9  print(hasil2) # Output: 2
10 |
```

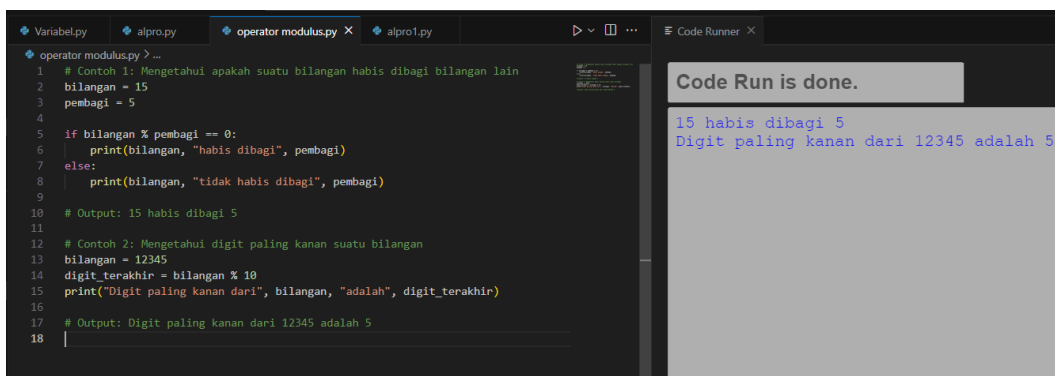
MATERI 7

1. Tanda Kurung (Parentheses): Operasi di dalam tanda kurung dievaluasi terlebih dahulu. Ini memungkinkan mengatur urutan operasi dengan jelas. Misalnya, dalam $(2*(3-1))$, operasi dalam tanda kurung $3-1$ dievaluasi terlebih dahulu sebelum hasilnya dikalikan dengan 2.
2. Eksponensial (Exponents): Operasi pangkat dievaluasi setelah operasi dalam tanda kurung. Ini berarti jika memiliki ekspresi seperti $2**3$, itu akan dievaluasi menjadi 8, karena 2 dipangkatkan dengan 3.
3. Perkalian dan Pembagian (Multiplication and Division): Operasi perkalian dan pembagian memiliki prioritas yang sama dan dievaluasi sebelum penjumlahan dan pengurangan. Misalnya, dalam ekspresi $2*3$, perkalian dilakukan sebelum operasi penjumlahan atau pengurangan lainnya.

4. Penjumlahan dan Pengurangan (Addition and Subtraction): Operasi penjumlahan dan pengurangan memiliki prioritas yang sama, tetapi dievaluasi setelah operasi perkalian dan pembagian. Misalnya, dalam ekspresi `5-3+1`, pengurangan dilakukan sebelum penjumlahan, sehingga hasilnya adalah 3-1 yang kemudian menjadi 2.

MATERI 8

Pada tipe data integer, operator modulus menghasilkan sisa pembagian dua bilangan. Ilustrasi penggunaan operator modulus di Python:

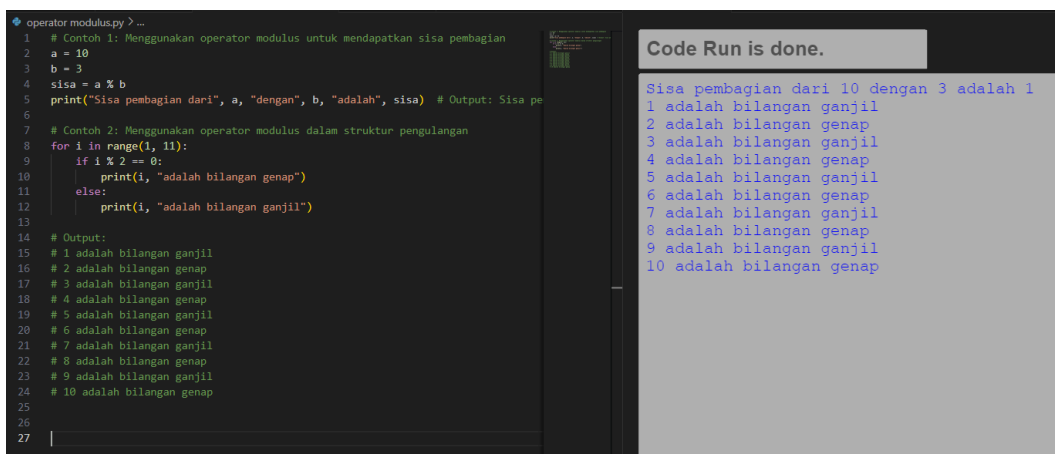


```
1 # Contoh 1: Mengetahui apakah suatu bilangan habis dibagi bilangan lain
2 bilangan = 15
3 pembagi = 5
4
5 if bilangan % pembagi == 0:
6     print(bilangan, "habis dibagi", pembagi)
7 else:
8     print(bilangan, "tidak habis dibagi", pembagi)
9
10 # Output: 15 habis dibagi 5
11
12 # Contoh 2: Mengetahui digit paling kanan suatu bilangan
13 bilangan = 12345
14 digit_terakhir = bilangan % 10
15 print("Digit paling kanan dari", bilangan, "adalah", digit_terakhir)
16
17 # Output: Digit paling kanan dari 12345 adalah 5
18 |
```

Code Run is done.

15 habis dibagi 5
Digit paling kanan dari 12345 adalah 5

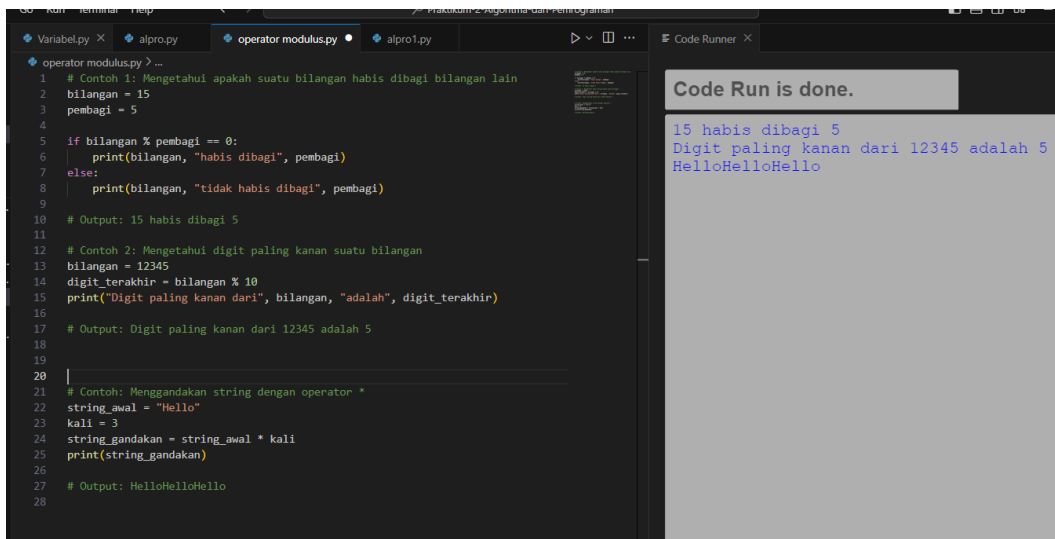
Untuk mengetahui apakah suatu bilangan habis dibagi bilangan lain atau untuk mengetahui digit paling kanan suatu bilangan, Anda dapat menggunakan contoh penggunaan operator modulus. Alih-alih menggabungkan string, operator + pada string tidak melakukan penjumlahan matematika. Misalnya:



```
1 # Contoh 1: Menggunakan operator modulus untuk mendapatkan sisa pembagian
2 a = 10
3 b = 3
4 sisa = a % b
5 print("Sisa pembagian dari", a, "dengan", b, "adalah", sisa) # Output: Sisa pe
6
7 # Contoh 2: Menggunakan operator modulus dalam struktur pengulangan
8 for i in range(1, 11):
9     if i % 2 == 0:
10         print(i, "adalah bilangan genap")
11     else:
12         print(i, "adalah bilangan ganjil")
13
14 # Output:
15 # 1 adalah bilangan ganjil
16 # 2 adalah bilangan genap
17 # 3 adalah bilangan ganjil
18 # 4 adalah bilangan genap
19 # 5 adalah bilangan ganjil
20 # 6 adalah bilangan genap
21 # 7 adalah bilangan ganjil
22 # 8 adalah bilangan genap
23 # 9 adalah bilangan ganjil
24 # 10 adalah bilangan genap
25
26
27 |
```

Code Run is done.

Sisa pembagian dari 10 dengan 3 adalah 1
1 adalah bilangan ganjil
2 adalah bilangan genap
3 adalah bilangan ganjil
4 adalah bilangan genap
5 adalah bilangan ganjil
6 adalah bilangan genap
7 adalah bilangan ganjil
8 adalah bilangan genap
9 adalah bilangan ganjil
10 adalah bilangan genap



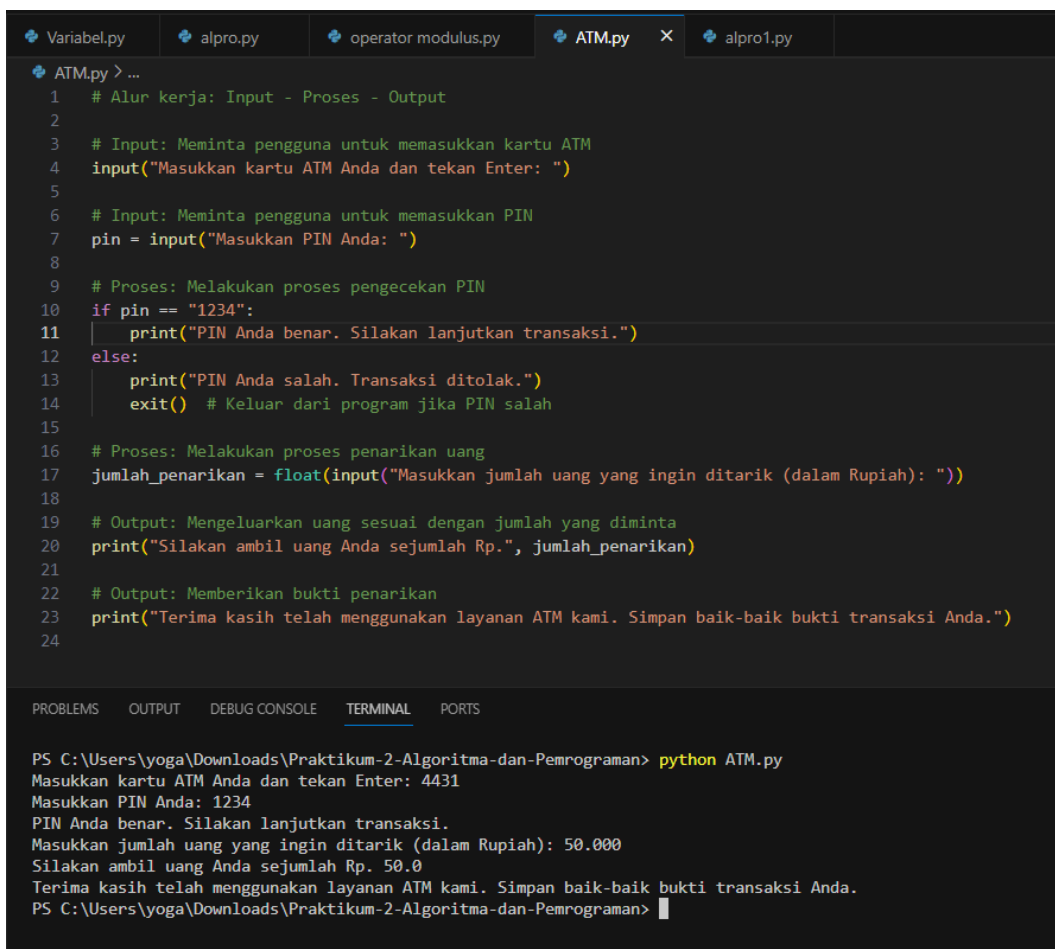
```
1 # Contoh 1: Mengetahui apakah suatu bilangan habis dibagi bilangan lain
2 bilangan = 15
3 pembagi = 5
4
5 if bilangan % pembagi == 0:
6     print(bilangan, "habis dibagi", pembagi)
7 else:
8     print(bilangan, "tidak habis dibagi", pembagi)
9
10 # Output: 15 habis dibagi 5
11
12 # Contoh 2: Mengetahui digit paling kanan suatu bilangan
13 bilangan = 12345
14 digit_terakhir = bilangan % 10
15 print("Digit paling kanan dari", bilangan, "adalah", digit_terakhir)
16
17 # Output: Digit paling kanan dari 12345 adalah 5
18
19
20
21 # Contoh: Menggandakan string dengan operator *
22 string_awal = "Hello"
23 kali = 3
24 string_gandakan = string_awal * kali
25 print(string_gandakan)
26
27 # Output: HelloHelloHello
28
```

Code Run is done.

15 habis dibagi 5
Digit paling kanan dari 12345 adalah 5
HelloHelloHello

Selain itu, string dapat digandakan menggunakan operator *, yang melakukan perkalian antara string dan bilangan bulat. Misalnya:

MATERI 9



```
1 # Alur kerja: Input - Proses - Output
2
3 # Input: Meminta pengguna untuk memasukkan kartu ATM
4 input("Masukkan kartu ATM Anda dan tekan Enter: ")
5
6 # Input: Meminta pengguna untuk memasukkan PIN
7 pin = input("Masukkan PIN Anda: ")
8
9 # Proses: Melakukan proses pengecekan PIN
10 if pin == "1234":
11     print("PIN Anda benar. Silakan lanjutkan transaksi.")
12 else:
13     print("PIN Anda salah. Transaksi ditolak.")
14     exit() # Keluar dari program jika PIN salah
15
16 # Proses: Melakukan proses penarikan uang
17 jumlah_penarikan = float(input("Masukkan jumlah uang yang ingin ditarik (dalam Rupiah): "))
18
19 # Output: Mengeluarkan uang sesuai dengan jumlah yang diminta
20 print("Silakan ambil uang Anda sejumlah Rp.", jumlah_penarikan)
21
22 # Output: Memberikan bukti penarikan
23 print("Terima kasih telah menggunakan layanan ATM kami. Simpan baik-baik bukti transaksi Anda.")
24
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\yoga\Downloads\Praktikum-2-Algoritma-dan-Pemrograman> python ATM.py
Masukkan kartu ATM Anda dan tekan Enter: 4431
Masukkan PIN Anda: 1234
PIN Anda benar. Silakan lanjutkan transaksi.
Masukkan jumlah uang yang ingin ditarik (dalam Rupiah): 50.000
Silakan ambil uang Anda sejumlah Rp. 50.0
Terima kasih telah menggunakan layanan ATM kami. Simpan baik-baik bukti transaksi Anda.
PS C:\Users\yoga\Downloads\Praktikum-2-Algoritma-dan-Pemrograman>

Program ini memiliki alur kerja Input - Proses - Output. Data yang diperlukan dimasukkan, langkah pemecahan masalah dilakukan, dan hasilnya dikeluarkan. Misalnya penarikan uang melalui ATM dibagi menjadi Input (kartu ATM, PIN), Proses (proses transaksi dengan pengecekan), dan Output (uang keluar, bukti penarikan). Fungsi input() digunakan dalam Python untuk menangani input pengguna. Sampai saat itu tiba, pencetakan cepat akan membantu pengguna mengetahui apa yang harus dimasukkan. Sebagai contoh:

BAGIAN 2

SOAL 1

Contoh program Python sederhana yang menghitung berat badan yang dibutuhkan berdasarkan tinggi badan dan nilai Indeks Massa Tubuh (BMI)

```
Menghitung_Berat_badan.py > ...
1  def hitung_berat_badan(tinggi_badan, bmi_harapan):
2      """
3      Fungsi untuk menghitung berat badan yang dibutuhkan berdasarkan tinggi badan
4      dan nilai Indeks Massa Tubuh (BMI) yang diharapkan.
5
6      Parameters:
7      tinggi_badan (float): Tinggi badan dalam satuan meter.
8      bmi_harapan (float): Nilai BMI yang diharapkan.
9
10     Returns:
11     float: Berat badan yang dibutuhkan untuk mencapai BMI yang diharapkan.
12     """
13     berat_badan = bmi_harapan * tinggi_badan**2
14     return berat_badan
15
16 def main():
17     # Input tinggi badan dari pengguna
18     tinggi_badan = float(input("Masukkan tinggi badan Anda (dalam meter): "))
19
20     # Input nilai BMI yang diharapkan
21     bmi_harapan = float(input("Masukkan nilai Indeks Massa Tubuh (BMI) yang diharapkan: "))
22
23     # Hitung berat badan yang dibutuhkan
24     berat_badan_dibutuhkan = hitung_berat_badan(tinggi_badan, bmi_harapan)
25
26     # Output hasil
27     print("Berat badan yang dibutuhkan untuk mencapai BMI", bmi_harapan, "adalah:", berat_badan_dibutuhkan, "kg")
28
29 if __name__ == "__main__":
30     main()

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\yoga\Downloads\Praktikum-2-Algoritma-dan-Pemrograman> python Menghitung_Berat_badan.py
Masukkan tinggi badan Anda (dalam meter): 160
Masukkan nilai Indeks Massa Tubuh (BMI) yang diharapkan: 30
Berat badan yang dibutuhkan untuk mencapai BMI 30.0 adalah: 768000.0 kg
PS C:\Users\yoga\Downloads\Praktikum-2-Algoritma-dan-Pemrograman> 
```

Fungsi hitung_berat_tubuh digunakan oleh program untuk menghitung tinggi badan pengguna dan nilai BMI yang diharapkan sebagai parameter. Selanjutnya,

program meminta tinggi badan pengguna dan nilai BMI yang diharapkan, lalu menghitung dan menampilkan berat badan yang diperlukan

LINK GITHUB: <https://github.com/Yoga4431/Praktikum-2-Algoritma-dan-Pemrograman.git>