



---

# 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.

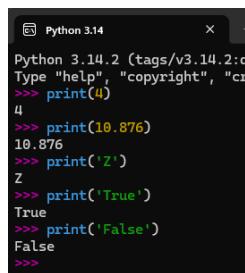
NIM	71251182
Nama Lengkap	Michael Dylan
Minggu ke / Materi	02 / Variable Expression & Statement

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

## BAGIAN 1: MATERI MINGGU 1 (40%)

### 2.3.1 Values dan type

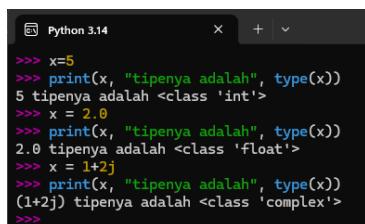
Value adalah nilai dasar dalam program seperti angka dan huruf, contohnya 1, 2, 'a', atau "Hello World". Setiap value memiliki type atau tipe data, misalnya 2 bertipe integer dan "Hello World" bertipe string. Interpreter mengenali string karena ditulis dengan tanda petik. Values dan type dapat dipahami dengan mencoba langsung di mode interaktif Python. Untuk lebih mengenal Values dan type bisa menggunakan python interactive seperti code dibawah ini.



```
Python 3.14.2 (tags/v3.14.2:6d8bf5e, Jun 29 2023, 11:14:37) [GCC 11.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information
>>> print(4)
4
>>> print(10.876)
10.876
>>> print('Z')
Z
>>> print(True)
True
>>> print(False)
False
>>>
```

Gambar 2.1: Code values dan type dari Python Interactive v3.14

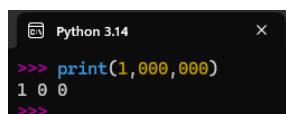
Perintah print dapat digunakan untuk menampilkan berbagai jenis value seperti integer, float, character,dan boolean. Setiap value memiliki **type** (tipe data). Python menyediakan fungsi bawaan type() untuk mengetahui tipe data dari suatu value. Seperti dibawah ini.



```
Python 3.14
>>> 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'>
>>>
```

Gambar 2.2: Fungsi type() bisa mengetahui tipe data dari value dari Python Interactive 3.14

Dalam penulisan bilangan besar sering digunakan pemisah tiga digit seperti 1.000.000. Namun di Python, penulisan 1,000,000 tidak dianggap satu bilangan utuh, melainkan dibaca sebagai beberapa parameter (1, 0, 0) pada fungsi print, bukan sebagai satu integer.

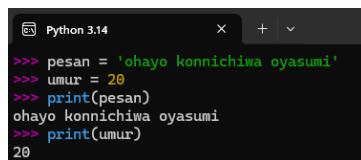


```
Python 3.14
>>> print(1,000,000)
1 0 0
>>>
```

Gambar 2.3: 1,000,000 tidak dianggap satu bilangan utuh. Sumber Python Interactive 3.14

## 2.3.2 Variabel

Variabel merupakan tempat di memori untuk menyimpan nilai yang digunakan selama program berjalan, dan nilainya dapat berubah. Variabel dapat menyimpan berbagai tipe data. Dalam Python, variabel bersifat dinamis, sehingga tidak perlu mendeklarasikan tipe data dan nilainya bisa berubah saat program dijalankan. Untuk menampilkan nilai variabel digunakan perintah print. Contohnya variable pesan menyimpan string dan umur menyimpan integer 20 seperti gambar dibawah ini.



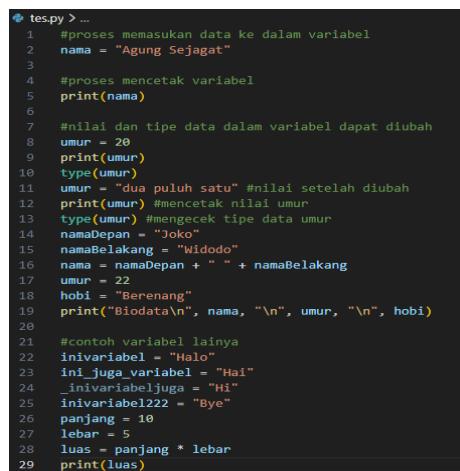
```
Python 3.14
>>> pesan = 'ohayo konnichiwa oyasumi'
>>> umur = 20
>>> print(pesan)
ohayo konnichiwa oyasumi
>>> print(umur)
20
```

Gambar 2.4: mengprint variabel yang menyimpan tipe data. Sumber Python Interative 3.14

## 2.3.3 Nama Variabel dan Keywords

Nama variabel harus diawali huruf atau garis bawah (\_), diikuti huruf, angka, atau garis bawah. Nama variabel bersifat case-sensitive sehingga huruf besar dan kecil dibedakan. Selain itu, nama variabel tidak boleh menggunakan keyword Python seperti if, for, dan while.

Python sendiri memiliki 35 keyword yang tidak boleh digunakan untuk memberi nama variable: **and, del, from, None, True, as, elif, global, nonlocal, try, assert, else, if, not, while, break, except, import, or, width, class, False, in, pass, yield, continue, finally, is, raise, async, def, for, lambda, return, wait.**

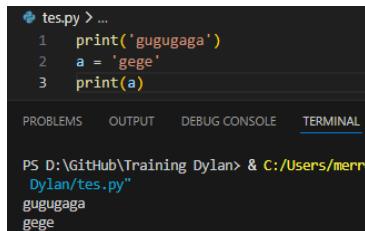


```
tes.py > ...
1 #proses memasukan data ke dalam variabel
2 nama = "Agung Sejagat"
3
4 #proses mencetak variabel
5 print(nama)
6
7 #nilai dan tipe data dalam variabel dapat diubah
8 umur = 20
9 print(umur)
10 type(umur)
11 umur = "dua puluh satu" #nilai setelah diubah
12 print(umur) #mencetak nilai umur
13 type(umur) #mengecek tipe data umur
14 namaDepan = "Joko"
15 namaBelakang = "Widodo"
16 nama = namaDepan + " " + namaBelakang
17 umur = 22
18 hobi = "Berenang"
19 print("Biodata\n", nama, "\n", umur, "\n", hobi)
20
21 #contoh variabel lainnya
22 inivariabel = "Halo"
23 ini_juga_variabel = "Hai"
24 _inivariabeljuga = "Hi"
25 inivariabel1222 = "Bye"
26 panjang = 10
27 lebar = 5
28 luas = panjang * lebar
29 print(luas)
```

Gambar 2.5: penggunaan variable dalam bahasa pemrograman python. Sumber Vscode

### 2.3.4 Statements

Statement adalah bagian kode Python yang dapat dieksekusi, seperti print, expression, dan assignment. Pada mode interaktif, statement langsung dijalankan dan hasilnya ditampilkan. Sedangkan pada script, statement dijalankan secara berurutan (sekuensial) dalam satu program.



```
tes.py > ...
1  print('gugugaga')
2  a = 'gege'
3  print(a)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS D:\GitHub\Training Dylan> & C:/Users/merrin/Dylan/tes.py
gugugaga
gege
```

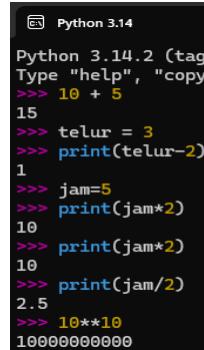
Gambar 2.6: Contoh statement dan outputnya. Sumber Vscode Python

### 2.3.5 Operator dan Operand

Operator ialah simbol tertentu yang digunakan untuk melakukan operasi aritmatika maupun logika. Nilai yang padanya dilakukan operasi disebut operand. Misalnya adalah  $2 + 3$ . Di sini tanda  $+$  adalah operator penjumlahan.  $2$  dan  $3$  adalah operand. Pada bagian ini secara khusus akan membahas operator aritmatika pada Python. Operator aritmatika adalah operator yang digunakan untuk melakukan operasi matematika, seperti penjumlahan, pengurangan, perkalian, pembagian, dan sebagainya.

Operator	Nama dan fungsi	Contoh
$+$	Penjumlahan, menjumlahkan 2 buah operand	$x + y$
$-$	Pengurangan, mengurangkan 2 buah operand	$x - y$
$*$	Perkalian, mengalikan 2 buah operand	$x * y$
$/$	Pembagian, membagi 2 buah operand	$x / y$
$**$	Pemangkatan, memangkatkan bilangan	$x^{**}y$

Tabel 2.1: Fungsi operator pada Python

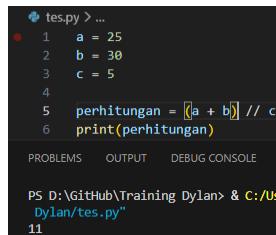


```
Python 3.14.2 (tags/v3.14.2:4e9d93e, Dec 17 2023, 13:46:32)
Type "help", "copyright", "credits" or "license" for more information
>>> 10 + 5
15
>>> telur = 3
>>> print(telur-2)
1
>>> jam=5
>>> print(jam*2)
10
>>> print(jam*2)
10
>>> print(jam/2)
2.5
>>> 10**10
10000000000
```

Gambar 2.7: Contoh penggunaan operator aritmatika dari Python Interactive 3.14

### 2.3.6 Expressions

Expression adalah gabungan dari value, variabel, dan operator yang merepresentasikan suatu nilai. Value atau variabel sendiri juga dapat disebut expression. Pada mode interaktif, expression akan langsung dievaluasi oleh interpreter dan hasilnya ditampilkan.



```
tes.py > ...
1 a = 25
2 b = 30
3 c = 5
4
5 perhitungan = [a + b] // c
6 print(perhitungan)

PROBLEMS OUTPUT DEBUG CONSOLE
```

PS D:\GitHub\Training Dylan> & C:/Users/Dylan/tes.py"
11

Gambar 2.8: Contoh Expressions pada Python di Vscode

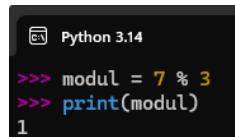
### 2.3.7 Urutan Operasi

Urutan operasi digunakan ketika dalam satu expression terdapat lebih dari satu operator. Python mengikuti aturan **PEMDAS**, yaitu tanda kurung (parentheses) dikerjakan terlebih dahulu, kemudian pangkat (exponentiation), lalu perkalian dan pembagian, dan terakhir penjumlahan serta pengurangan. Operator yang memiliki prioritas sama akan dihitung dari kiri ke kanan. Contohnya  $2*(3-1)$  hasilnya 4 karena di dalam kurung dihitung dulu,  $2^{**}1+1$  hasilnya 3 karena pangkat didahulukan,  $6+4/2$  hasilnya 8 karena pembagian lebih dulu, dan  $5-3-1$  hasilnya 1 karena dihitung dari kiri ke kanan. Untuk menghindari kesalahan, tanda kurung bisa digunakan agar urutan perhitungan sesuai yang diinginkan.

### 2.3.8 Operator Modulus dan String

#### Modulus

Operator modulus itu operator untuk mencari sisa hasil bagi dari pembagian dua bilangan dan hanya berlaku pada tipe data integer. Dalam Python, operator modulus dilambangkan dengan tanda persen (`%`). Contohnya, `7 % 3` menghasilkan 1 karena 7 dibagi 3 hasilnya 2 dengan sisa 1. Operator ini sering digunakan untuk mengecek apakah suatu bilangan habis dibagi bilangan lain, misalnya jika `x % y = 0` maka x habis dibagi y, serta untuk mengambil digit terakhir dari suatu angka, seperti `x % 10` untuk digit paling kanan atau `x % 100` untuk dua digit terakhir.

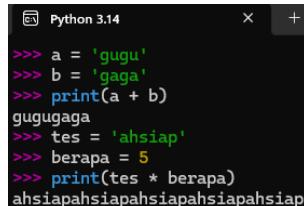


```
Python 3.14
>>> modul = 7 % 3
>>> print(modul)
1
```

Gambar 2.9: Contoh modul pada Python Interactive 3.14

### String

Pada tipe data string, operator + digunakan untuk menggabungkan string, bukan penjumlahan angka, sedangkan operator \* digunakan untuk mengulang string sebanyak bilangan integer tertentu. Contohnya "A" + "B" menjadi "AB" dan "Hi" \* 3 menjadi "HiHiHi".



```
Python 3.14
>>> a = 'gugu'
>>> b = 'gaga'
>>> print(a + b)
gugugaga
>>> tes = 'ahsiap'
>>> berapa = 5
>>> print(tes * berapa)
ahsiapahsiapahsiapahsiapahsiap
```

Gambar 3.0: Contoh string penggabungan dan pengulangan kata dari Python Interactive 3.14

### 2.3.9 Menangani Input dari Pengguna

Input ialah data yang dimasukkan agar program dapat berjalan, proses adalah langkah-langkah yang dilakukan untuk mengolah data tersebut, dan output adalah hasil yang diperoleh. Contoh login akun web dengan memasukan email dan password (input) lalu web loading memeriksa data(proses) jika benar maka berhasil masuk akun pada web(output).

Python dapat menerima input dari pengguna melalui fungsi bawaan `input()`. Saat fungsi ini dipanggil, program akan berhenti sementara dan menunggu pengguna mengetik sesuatu. Sebelum menerima input, sebaiknya menampilkan prompt agar pengguna tahu apa yang harus diisi dengan menuliskan string di dalam fungsi `input()`. Tanda `\n` digunakan untuk membuat baris baru

sehingga input muncul di bawah prompt. Jika input berupa bilangan, maka dapat dikonversi ke integer menggunakan fungsi int(), namun akan terjadi error jika pengguna memasukkan data selain angka. Seperti di Gambar 3.1

The screenshot shows a Python 3.14 interactive session. The user inputs a string 'prompt = 'Tanggal berapa sekarang\n' followed by 'tgl = input(prompt)'. A yellow circle highlights the '\n' character, with a callout pointing to it stating 'fungsi \n untuk membuat garis baru'. The next line 'Tanggal berapa sekarang?' is shown. The user then inputs '13' followed by 'int(tgl)'. A blue box highlights the 'int(tgl)' line, with a callout pointing to it stating 'variable string untuk input agar pengguna tahu apa yang harus diisi saat input'. The output shows '13'. The user then inputs 'gugu' followed by 'int(tgl)'. A green box highlights the 'int(tgl)' line, with a callout pointing to it stating 'Input() akan menerima dari pengguna yang diketik, jika pengguna mengentik bilangan bulat maka menggunakan int.'. A red bracket on the right side of the screen groups the last two callouts and points to the error message 'Akan terjadi error jika pengguna memasukkan data selain angka/int'. The error message 'ValueError: invalid literal for int() with base 10: 'gugu'' is displayed at the bottom.

Gambar 3.1: Contoh persoalan input. Menggunakan Python Interactive 3.14

## 2.4 Komentar

Tanda # pada python merupakan komentar/catatan untuk suatu kode. Komentar tidak akan diproses oleh Interpreter Python seperti pada Gambar 3.2.

The screenshot shows a Python file named 'tes.py' in the VS Code editor. The code contains two lines: '1 a = 'halo' #variabel pertama (komentar)' and '2 print (a) #print untuk menghasilkan output dari variabel a'. The '#variabel pertama (komentar)' and '#print untuk menghasilkan output dari variabel a' are comments. Below the code, the terminal window shows the command 'PS D:\GitHub\Training Dylan> & C:/Users/merro/AppData/Local/Python/pythonco' followed by 'halo' and then 'PS D:\GitHub\Training Dylan>'. The terminal tabs are labeled PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS.

Gambar 3.2: Contoh komentar enggunakan Python Vscode

Python tidak memiliki komentar multibaris khusus, sehingga setiap baris komentar harus ditulis satu per satu menggunakan tanda #.

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

## **BAGIAN 2: LATIHAN MANDIRI (60%)**

**Link Github** = [https://github.com/MichaelDylan-ti/71251182\\_michaeldylan/tree/main](https://github.com/MichaelDylan-ti/71251182_michaeldylan/tree/main)

LATIHAN 2.1

Buatlah program yang dapat menghitung berat badan yang diperlukan, jika diketahui tinggi badan dan nilai Body Mass Index (BMI) yang diharapkan! Body Mass Index dihitung dengan cara:  $BMI = \frac{\text{berat}}{\text{tinggi}^2}$ . Perhatikan, berat badan dalam satuan kilogram (kg) dan tinggi badan dalam satuan meter (m).

## INPUT

```
Soal01.py > ...
1 tinggi = float(input("masukkan tinggi badan:"))
2 berat = float(input("masukkan berat badan:"))
3 bmi = berat / (tinggi**2)
4 print("Berat yang diperlukan adalah", bmi, "kg")
```

Gambar 3.3: Source code untuk Latihan 2.1. Menggunakan Python Vscode

Program meminta pengguna untuk memasukkan tinggi badan dan berat badan melalui kotak input. Pengguna menuliskan tinggi badan dalam satuan meter, misalnya 1.71, dan berat badan dalam kilogram, misalnya 55. Program kemudian menghitung nilai BMI (Body Mass Index) dengan rumus berat dibagi tinggi kuadrat.

## OUTPUT

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\GitHub\Training Dylan> & C:/Users/merro/AppData/L
masukkan tinggi badan:1.71
masukkan berat badan:55
Berat yang diperlukan adalah 18.80920625149619 kg
```

Gambar 3.4: Output dari Gambar 3.3. Menggunakan Python Vscode

Setelah perhitungan selesai, program menampilkan hasilnya ke layar dalam bentuk teks, misalnya Berat yang diperlukan adalah 18.80920625149619 kg, yang menunjukkan nilai BMI dari data yang dimasukkan. Output ini adalah angka desimal yang merepresentasikan rasio berat terhadap tinggi badan pengguna.

## LATIHAN 2.2

Buatlah program yang dapat menghitung hasil dari fungsi  $f(x) = 2x^3 + 2x + \frac{15}{x}$ , di mana x merupakan bilangan bulat yang dimasukkan oleh pengguna.

INPUT

```
tes.py > ...
1 x = int(input("Masukkan nilai x:"))
2 f = (2*(x**3) + 2*x + 15/x)
3 print("hasil f(x)",f)
```

Gambar 3.5: Source code untuk Latihan 2.2. Menggunakan Python Vscode

Input pada program ini adalah nilai **x** yang dimasukkan oleh pengguna melalui keyboard menggunakan fungsi `input()`, kemudian dikonversi ke tipe **integer** dengan `int()`. Nilai x merupakan bilangan bulat yang akan digunakan sebagai variabel dalam perhitungan fungsi.

OUTPUT

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\GitHub\Training Dylan> & C:/Users/merro/AppData/Masukkan nilai x:10
hasil f(x) 2021.5
```

Gambar 3.6: Output dari Gambar 3.5. Menggunakan Python Vscode

Output dari program ini adalah hasil perhitungan nilai **f(x)** berdasarkan rumus  $f(x) = 2x^3 + 2x + \frac{15}{x}$ , yang ditampilkan ke layar menggunakan perintah `print()`. Hasil ini menunjukkan nilai fungsi sesuai dengan input x yang diberikan pengguna.

## LATIHAN 2.3

Budi tertarik untuk melamar pekerjaan pada liburan semester yang akan berlangsung selama 5 minggu. Gaji yang diberikan adalah gaji per jam. Total pajak yang harus budi bayarkan dari penghasilannya selama bekerja adalah 14%. Setelah membayar pajak, budi menghabiskan 10% dari pendapatan bersihnya untuk membeli baju dan aksesoris yang akan digunakan pada semester baru, dan 1% untuk membeli alat tulis. Setelah membeli baju, aksesoris dan alat tulis, Budi menggunakan 25% dari sisa uangnya untuk disedekahkan. Setiap Rp.1000 yang Budi

sedekahkan 30% nya akan diserahkan kepada anak yatim, dan sisanya akan diserahkan ke kaum dhuafa. Buatlah sebuah program, dengan input:

1. Gaji per jam yang anda inginkan
2. Jumlah jam kerja yang akan dilakukan dalam 1 minggu

Output dari program adalah sebagai berikut :

1. Pendapatan Budi selama libur musim panas sebelum melakukan pembayaran pajak.
2. Pendapatan Budi selama libur musim panas setelah melakukan pembayaran pajak.
3. Jumlah uang yang akan Budi habiskan untuk membeli pakaian dan aksesoris.
4. Jumlah uang yang akan Budi habiskan untuk membeli alat tulis.
5. Jumlah uang yang akan Budi sedekahkan.
6. Jumlah uang yang akan diterima anak yatim.
7. Jumlah uang yang akan diterima kaum dhuafa.

#### INPUT

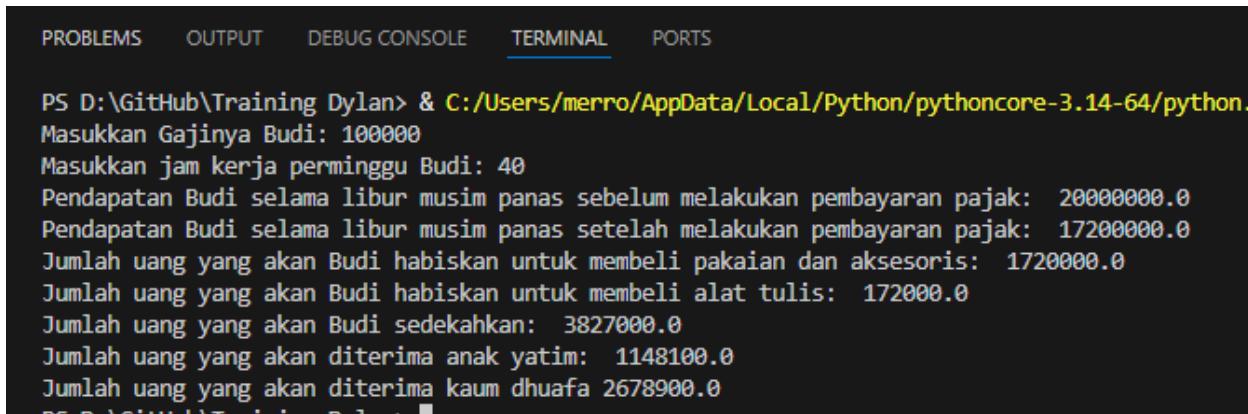
```
❶ Soal03.py > ...
1  gaji_perjam = int(input("Masukkan Gajinya Budi: "))
2  jam_kerja = float(input("Masukkan jam kerja perminggu Budi: "))
3  minggu = 5
4
5  pendapatan_sebelum = gaji_perjam * jam_kerja * minggu
6
7  pendapatan_sesudah_pajak = pendapatan_sebelum - ((pendapatan_sebelum * 14) / 100)
8  uang_pakaian_aksesoris = 0.10 * pendapatan_sesudah_pajak
9  uang_alattulis = 0.01 * pendapatan_sesudah_pajak
10 total_pengeluaran = pendapatan_sesudah_pajak - uang_pakaian_aksesoris - uang_alattulis
11
12 uang_sedekah = 0.25 * total_pengeluaran
13 uang_anakyatim = 0.30 * uang_sedekah
14 uang_kaumdhufa = uang_sedekah - uang_anakyatim
15
16 print("Pendapatan Budi selama libur musim panas sebelum melakukan pembayaran pajak: ", pendapatan_sebelum)
17 print("Pendapatan Budi selama libur musim panas setelah melakukan pembayaran pajak: ", pendapatan_sesudah_pajak)
18 print("Jumlah uang yang akan Budi habiskan untuk membeli pakaian dan aksesoris: ", uang_pakaian_aksesoris)
19 print("Jumlah uang yang akan Budi habiskan untuk membeli alat tulis: ", uang_alattulis)
20 print("Jumlah uang yang akan Budi sedekahkan: ", uang_sedekah)
21 print("Jumlah uang yang akan diterima anak yatim: ", uang_anakyatim)
22 print("Jumlah uang yang akan diterima kaum dhuafa", uang_kaumdhufa)
```

Gambar 3.7: Source code untuk Latihan 2.3. Menggunakan Python Vscode

Program ini meminta Budi untuk memasukkan dua data penting sebagai input. Pertama, Budi diminta memasukkan gaji per jam dalam bentuk bilangan bulat, yang akan menjadi dasar perhitungan pendapatan total. Kedua, Budi diminta memasukkan jumlah jam kerja per minggu,

yang dapat berupa angka desimal, sehingga menghitung total jam kerja secara lebih fleksibel. Kedua input ini kemudian digunakan bersama dengan jumlah minggu libur musim panas, yaitu lima minggu, untuk menghitung pendapatan Budi sebelum dan sesudah pajak. Dari pendapatan yang sudah dipotong pajak, program akan menghitung alokasi uang untuk membeli pakaian dan alat tulis, serta menentukan besarnya uang yang disedekahkan, yang sebagian diberikan kepada anak yatim dan sebagian lagi kepada kaum dhuafa. Dengan begitu, input yang dimasukkan Budi menjadi dasar seluruh perhitungan keuangan dan pembagian dana dalam program ini.

## OUTPUT



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\GitHub\Training Dylan> & C:/Users/merro/AppData/Local/Python/pythoncore-3.14-64/python.
Masukkan Gajinya Budi: 100000
Masukkan jam kerja perminggu Budi: 40
Pendapatan Budi selama libur musim panas sebelum melakukan pembayaran pajak: 20000000.0
Pendapatan Budi selama libur musim panas setelah melakukan pembayaran pajak: 17200000.0
Jumlah uang yang akan Budi habiskan untuk membeli pakaian dan aksesoris: 1720000.0
Jumlah uang yang akan Budi habiskan untuk membeli alat tulis: 172000.0
Jumlah uang yang akan Budi sedekahkan: 3827000.0
Jumlah uang yang akan diterima anak yatim: 1148100.0
Jumlah uang yang akan diterima kaum dhuafa 2678900.0
PS D:\GitHub\Training Dylan>
```

Gambar 3.8: Output dari Gambar 3.7. Menggunakan Python Vscode

Output tersebut menunjukkan perhitungan keuangan Budi selama liburan musim panas. Pertama, program meminta pengguna untuk memasukkan gaji Budi per minggu dan jumlah jam kerja per minggu. Berdasarkan input gaji 100.000 dan jam kerja 40, program menghitung total pendapatan Budi selama musim panas sebelum pajak sebesar 20.000.000. Setelah dikurangi pajak, pendapatan yang tersisa menjadi 17.200.000. Selanjutnya, program membagi pengeluaran Budi untuk beberapa keperluan, yaitu 1.720.000 untuk pakaian dan aksesoris, 172.000 untuk alat tulis, serta sisa uang digunakan untuk sedekah, yang dibagi kepada anak yatim sebesar 1.148.100 dan kaum dhuafa sebesar 2.678.900. Dengan begitu itulah rincian pengeluaran Budi.