



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	71251188
Nama Lengkap	ANGELA REVALINE SETIAWAN
Minggu ke / Materi	02 / Variable, Expression, dan Statements

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

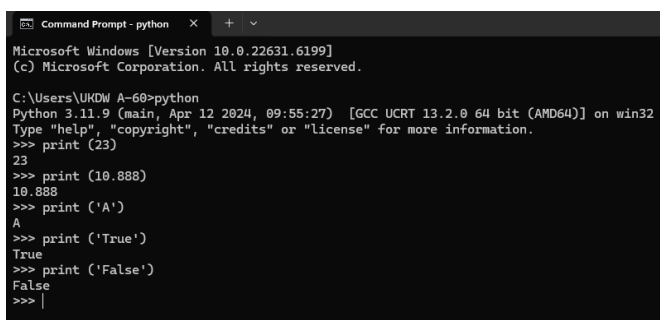
BAGIAN 1: MATERI MINGGU INI (40%)

Link Github : https://github.com/angelarevalines/71251188_valine.git

1. Values dan Type

Pada materi 1 minggu kedua ini menjelaskan tentang Value dan Type. Value adalah komponen utama dari sebuah program. Value dapat berupa huruf ataupun angka. Value dapat dibedakan menjadi beberapa tipe yang berbeda, contohnya “2” untuk sebuah nilai integer dan “Hello World” untuk sebuah nilai *string*. Interpreter dapat melakukan identifikasi terhadap *string* karena setiap penulisannya ditutup menggunakan tanda petik (*quotation mark*).

Berikut beberapa contoh baris kode untuk memahami values dan type.

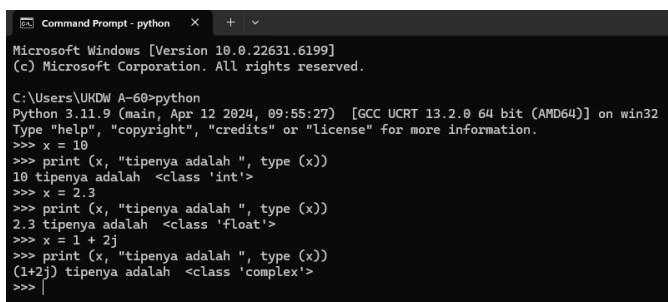


```
Microsoft Windows [Version 10.0.22631.6199]
(c) Microsoft Corporation. All rights reserved.

C:\Users\UKDW A-60>python
Python 3.11.9 (main, Apr 12 2024, 09:55:27) [GCC UCRT 13.2.0 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print(23)
23
>>> print(10.888)
10.888
>>> print('A')
A
>>> print('True')
True
>>> print('False')
False
>>> |
```

Gambar 1.1 : Contoh baris kode untuk memahami perbedaan values dan type.

Perintah print tidak hanya bekerja untuk value *string* dan *integer*, tetapi juga *float* (bilangan pecahan), *character* (huruf), atau *bool* (benar/salah). Setiap value yang pasti memiliki type untuk mengetahui tipe data tersebut. Python menyediakan fungsi built-in untuk melakukan pengecekan tipe data pada value dengan menggunakan fungsi *type()*.

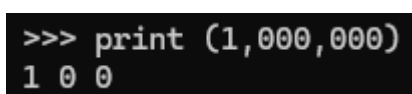


```
Microsoft Windows [Version 10.0.22631.6199]
(c) Microsoft Corporation. All rights reserved.

C:\Users\UKDW A-60>python
Python 3.11.9 (main, Apr 12 2024, 09:55:27) [GCC UCRT 13.2.0 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 10
>>> print(x, "tipenya adalah ", type(x))
10 tipenya adalah <class 'int'>
>>> x = 2.3
>>> print(x, "tipenya adalah ", type(x))
2.3 tipenya adalah <class 'float'>
>>> x = 1 + 2j
>>> print(x, "tipenya adalah ", type(x))
(1+2j) tipenya adalah <class 'complex'>
>>> |
```

Gambar 1.2 : Mengecek tipe data menggunakan python mode interaktif.

Ketika sedang menggunakan bilangan bulat besar, beberapa model penulisan akan menggunakan tanda koma (,) diantara kelompok tiga digit. Contohnya pada penulisan 1.000.000, Python akan menganggapnya sebagai bilangan bulat.



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

Gambar 1.3 : Mengecek bilangan bulat besar tiga digit.

Hal ini dikarenakan Python menganggap 1.000.000 adalah kiriman parameter sebanyak 3 parameter pada fungsi print, yaitu 1, 0, dan 0.

2. Variabel

Variable adalah lokasi memori yang dicadangkan untuk menyimpan nilai-nilai. Ini maksudnya ketika anda membuat sebuah *variable* berarti anda memesan beberapa ruang di memori. *Variable* menyimpan data yang dilakukan selama program dieksekusi. Nantinya isi *variable* dapat diubah oleh operasi-operasi tertentu pada program yang menggunakan *variable*. *Variable* dapat menyimpan berbagai macam tipe data. Dalam bahasa pemrograman Python, kita tidak perlu mendeklarasikan tipe data tertentu sebelum menjalankan sebuah program. Hal ini berarti menunjukkan sifat *variable* yang dinamis.

Berikut adalah contoh penggunaan *variable*.

1. *Variable* pesan yang berisi string.
2. *Variable* n yang berisi nilai integer 23.
3. *Variable* ipk yang berisi nilai float 3,63.

```
>>> pesan = 'halo! selamat pagi semua'
>>> n = 23
>>> ipk = 3.63
```

Gambar 2.1 : Contoh penggunaan *variable*.

Untuk bisa menampilkan nilai dari setiap *variable*, kita dapat menggunakan perintah print.

```
>>> print (n)
23
>>> print (ipk)
3.63
```

Gambar 2.2 : Menampilkan nilai tiap *variable* menggunakan perintah print.

3. Nama Variabel dan Keywords

Panduan memberi nama pada variabel, antara lain :

1. Nama *variable* boleh diawali dengan menggunakan huruf atau garis bawah (_).
Contoh :
 - keyboard
 - _keyboard
 - keyboardKu
 - keyboard_ku
2. Kemudian karakter selanjutnya dapat berupa huruf, garis bawah (_), atau angka.
Contoh :
 - _keyboard
 - keyboard2
 - k2
3. Karakter pada nama *variable* bersifat sensitif (*case-sensitif*). Maksudnya adalah huruf besar dan kecil dibedakan. Contoh :
 - keyboard_ku
 - keyboard_Ku

Dua *variable* tersebut adalah *variable* yang berbeda.

4. Nama *variable* tidak boleh menggunakan kata kunci yang sudah ada dalam python. (if, while, for, dll).

Dalam bahasa pemrograman Python terdapat 35 keyword yang tidak boleh digunakan sebagai nama *variable*.

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

Gambar 3.1 : 35 keyword yang tidak boleh digunakan sebagai nama *variable*.

```
#proses memasukkan data ke dalam variabel
nama = "Valine"
#proses mencetak variabel
print(nama)
#nilai dan tipe data dalam variabel dapat diubah
umur = 18
print(umur)
type(umur)
umur = "delapan belas" #nilai setelah diubah
print(umur) #mencetak nilai umur
type(umur) #mengecek tipe data umur
namaDepan = "Angela"
namaBelakang = "Setiawan"
nama = namaDepan + " " + namaBelakang
umur = 18
hobi = "Menonton Drama China"
print("Biodata\n", nama, "\n", umur, "\n", hobi)
```

Gambar 3.2 : Penggunaan variabel dalam Python.

4. Statements

Statements adalah bagian dari perintah Python yang dapat dieksekusi, contohnya seperti print, *expression statements*, dan *assignment*. Jika menggunakan Python mode interaktif, Python akan langsung mengeksekusi dan menampilkan hasilnya. Sedangkan bila menggunakan *script mode*, *script* biasanya berisi *statements* yang saling berhubungan.

```
3 print(2)
4 x = 3
5 print(x)
```

PROBLEMS OUTPUT DEBUG CONSOLE

PS C:\prak_alpro\praktikum_alpro> python.exe "c:/prak_alpro/praktik

2

3

Gambar 4.1 : Contoh statements print dan outputnya.

5. Operator dan Operand

Operator adalah simbol tertentu untuk melakukan operasi aritmatika maupun logika. Sedangkan nilai yang dilakukan operasi disebut *operand*. Contohnya adalah 2 + 3. Tanda + disini berarti sebagai operator penjumlahan. 2 dan 3 sebagai operand. Pada materi kali ini akan membahas operator aritmatika pada Python. Operator aritmatika yang digunakan dalam Python adalah operasi matematika seperti penjumlahan, pengurangan, perkalian, dan pembagian. Tabel 5.1 : Operator pada Python.

Berikut adalah beberapa contoh penggunaan operator aritmatika 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	$x ** y$

Tabel 5.1 : Operator aritmatika.

```
>>> 23 + 32
55
>>> hour = 5
>>> print (hour - 1)
4
>>> minute = 60
>>> print (hour * 2 + minute)
70
>>> print (minute / 60)
1.0
>>> 5**2
25
>>> (2 + 3) * (7 - 2)
25
```

Gambar 5.2 : Penggunaan operator aritmatika menggunakan mode interaktif.

6. Expressions

Expression adalah representasi dari nilai dan dapat terdiri dari gabungan antara *values*, *variable*, dan *operator*. *Values* dapat dianggap sebagai *expression* dan variabel. Namun secara umum, semua dapat disebut dengan *expression*.

17

x

x+17

Namun apabila menggunakan *expression* dalam mode interaktif, interpreter akan melakukan evaluasi dan menampilkan hasilnya.

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

Gambar 6.1 : *Expression* dalam mode interaktif.

7. Urutan Operasi

Urutan operasi berlaku bila ada lebih dari satu operator dalam *expression* dan bergantung pada aturan prioritas. Urutan operasi sering disingkat dengan PEMDAS (*Parantheses, Exponentiation, Multiplication, and Division, Operator*).

- Parantheses* (Tanda kurung) adalah prioritas tertinggi dan digunakan untuk memaksa *expression* dalam urutan yang sesuai. Contohnya adalah $2 * (3 - 1) = 4$ dan $(1 + 1) ** (5 - 2) = 8$.
- Exponentiation* (Eksponensial / Pemangkatan) adalah prioritas tertinggi berikutnya, contohnya adalah $2 ** 1 + 1 = 3$, bukan 4. Kemudian $3 * 1 ** 3 = 3$, bukan 27.
- Multiplication and Division* (Perkalian dan Pembagian) memiliki prioritas yang sama namun lebih tinggi dari penjumlahan dan pengurangan. Penjumlahan dan

pengurangan memiliki prioritas yang sama juga. Contohnya adalah $2*3-1 = 5$, bukan 4 dan $6 + 4/2 = 8$, bukan 5.

- d) *Operators* (operator) memiliki prioritas yang sama, dibaca dari kiri ke kanan. Contohnya $5-3-1 = 1$, bukan 3 karena yang dikerjakan adalah operasi pengurangan $5-3$ terlebih dahulu lalu hasilnya dikurangi dengan 1.

Apabila terjadi keraguan. Dapat gunakan tanda kurung di dalam ekspresi untuk memastikan bahwa komposisinya sesuai dengan yang diinginkan.

8. Operator Modulus dan String

a) Modulus

Operator Modulus adalah 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 (%).

```
>>> quotient = 7 // 3
>>> print (quotient)
2
>>> oprmomulus = 7 % 3
>>> print (oprmomulus)
1
```

Gambar 8.1 : Contoh penggunaan modulus.

b) String

Operator '+' apabila bekerja dengan string tidak berarti penjumlahan secara matematika, tetapi berupa penggabungan antar string. Berikut adalah contohnya.

```
>>> first = 10
>>> second = 15
>>> print (first + second)
25
>>> first = '100'
>>> second = '150'
>>> print (first + second)
100150
```

Gambar 8.2 : Perbedaan hasil penggunaan operator apabila tipe data integer dan string.

Operator '*' juga dapat digunakan dengan string, dengan melakukan perkalian antara string dan integer.

```
>>> first = 'Test'
>>> second = 3
>>> print (first * second)
TestTestTest
```

Gambar 8.3 : Penggunaan operator '*' dalam perkalian antara string dan integer.

9. Menangani Input dari Pengguna

Sebuah program pada dasarnya memiliki alur kerja Input - Proses - Output.



Gambar 9.1 : Input - Proses - Output.

Input merupakan data yang dibutuhkan agar program dapat berjalan. Proses adalah langkah-langkah yang dikerjakan oleh program untuk memecahkan masalah. Output adalah hasil yang didapat dari proses tersebut. Contohnya mengambil uang lewat ATM. Alur kerjanya dibagi menjadi 3, yaitu Input-Proses-Output.

- 1) Input = masukkan kartu ATM, pin, dan memilih menu pengambilan uang.
- 2) Proses = mesin ATM akan memproses transaksi dengan menghubungi server Bank, lalu dilakukan berbagai macam pengecekan (contoh : apakah saldonya cukup? apakah kartunya masih berlaku? apakah ada blokir?).
- 3) Output = ATM mengeluarkan uang, bukti pengambilan uang, dan kartu.

Dalam Python, input dari pengguna dapat diambil dari fungsi bawaan `input()`. Apabila fungsi ini dipanggil, program akan berhenti dan menunggu pengguna mengetik sesuatu hingga menekan 'Enter'. Kemudian mengembalikan apa yang diketik oleh pengguna sebagai string.

```

>>> inp = input()
Hari minggu adalah hari libur
>>> print(inp)
Hari minggu adalah hari libur
  
```

Gambar 9.2 : Contoh Input-Proses-Output dalam Python.

Tanda `\n` pada akhir prompt mewakili garis 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?
22
>>> int(suhu)
22
>>> int(suhu) + 3
25
  
```

Gambar 9.3 : Contoh penggunaan tanda `\n`.

10. Komentar

Tanda pagar (`#`) digunakan untuk menandai komentar di Python. Komentar tidak akan diproses oleh interpreter Python.

```

trial.py
1  #komentarpertama
2  print("Halo Dunia!") #komentarkedua
  
```

Gambar 10.1 : Contoh kode untuk memahami komentar.

```

Halo Dunia!
  
```

Gambar 10.2 : Output yang dihasilkan.

Sumber Materi : Modul 02 Praktikum Algoritma dan Pemrograman ([02 - Variable, Expression, dan Statements](#))

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.

Link Github : https://github.com/angelarevalines/71251188_valine.git

SOAL 1

Source Code :

```
#tinggi badan harus dalam meter
```

```
tinggibadan = float (input("Tinggi badan (m) = "))
```

```
#BMI harus dalam kg/m^2
```

```
BMI = float (input("Nilai BMI = "))
```

```
#menghitung berat badan
```

```
beratbadan = round(BMI * (tinggibadan**2),2)
```

```
print (f"Berat badanmu = {beratbadan} kg")
```


Output :

```
soal01.py X
C: > Users > LENOVO > OneDrive > valine > Prak Alpro > minggu02 > soal01.py > ...
1  #tinggi badan harus dalam meter
2  tinggibadan = float (input("Tinggi badan (m) = "))
3
4  #BMI harus dalam kg/m^2
5  BMI = float (input("Nilai BMI = "))
6
7  #menghitung berat badan
8  beratbadan = round(BMI * (tinggibadan**2),2)
9
10 print (f"Berat badanmu = {beratbadan} kg")

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\LENOVO> python -u "c:\Users\LENOVO\OneDrive\valine\Prak Al
Tinggi badan (m) = 1.65
Nilai BMI = 18.5
Berat badanmu = 50.37 kg
```

Langkah-langkah :

1. Soal diminta untuk menghitung berat badan, pengguna cukup input tinggi badan (m) dan nilai BMI (kg/m²).
2. Langkah pertama kita akan membuat variabel tinggi badan dan BMI yang menggunakan fungsi input dan float karena nilai bisa berupa bilangan desimal.
3. Rumus mencari berat badan adalah tinggi badan ² x BMI dengan menggunakan round untuk pembulatan.
4. Lalu gunakan fungsi print untuk menampilkan output yang diinginkan.

SOAL 2

Source code :

#x adalah bilangan bulat

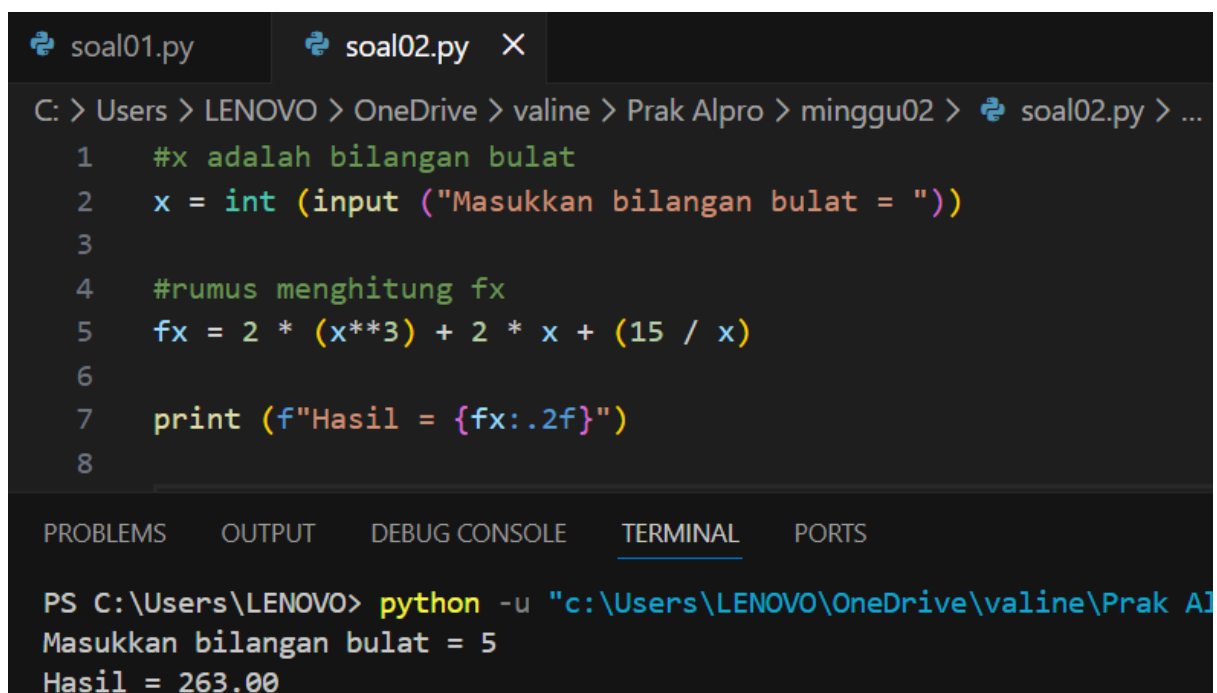
```
x = int (input ("Masukkan bilangan bulat = "))
```

#rumus menghitung fx

```
fx = 2 * (x**3) + 2 * x + (15 / x)
```

```
print (f"Hasil = {fx:.2f}")
```

Output :



```
soal01.py  soal02.py X
C: > Users > LENOVO > OneDrive > valine > Prak Alpro > minggu02 > soal02.py > ...
1  #x adalah bilangan bulat
2  x = int (input ("Masukkan bilangan bulat = "))
3
4  #rumus menghitung fx
5  fx = 2 * (x**3) + 2 * x + (15 / x)
6
7  print (f"Hasil = {fx:.2f}")
8

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\LENOVO> python -u "c:\Users\LENOVO\OneDrive\valine\Prak Al
Masukkan bilangan bulat = 5
Hasil = 263.00
```

Langkah-langkah :

1. Soal meminta agar pengguna dapat menginput nilai x dan program akan otomatis menghitung hasil dari nilai x yang telah di input oleh pengguna.
2. Dibutuhkan *variable* x dengan value input dari pengguna. Dalam value tersebut tertulis tipe data *integer* yang bertujuan agar pengguna memasukkan bilangan bulat.
3. Lalu buat *variable* baru dengan value operasi hitung fungsinya.
4. Lalu gunakan fungsi print dengan variasi .2f untuk menampilkan hasil yang berupa bilangan float dengan 2 angka di belakang koma.

SOAL 3

Source code :

```
#masukkan gaji per jam dan jumlah jam kerja selama 1 minggu
gajiperjam = int (input("Berapa gaji per jam Budi? Rp "))
jumlahjamkerja1minggu = int (input("Berapa jam kerja Budi selama 1 minggu? "))
jumlahminggu = 5

#menghitung total gaji selama kerja 5 minggu
totalgaji = round((gajiperjam * jumlahjamkerja1minggu * jumlahminggu),2)

#pendapatan bersih setelah potong pajak
pendapatanbersih = round(totalgaji - (totalgaji * (14/100)),2)

#menghitung uang belanja dan alat tulis
uangbelanja = round(pendapatanbersih * (10/100),2)
uangalattulis = round(pendapatanbersih * (1/100),2)

#menghitung uang setelah belanja
uangsetelahbelanja = pendapatanbersih - (uangbelanja + uangalattulis)

#menghitung uang yang akan disedekahkan, diterima anak yatim, dan kaum dhuafa
uangsedekah = round(uangsetelahbelanja * (25/100),2)
uanganakyatim = round(uangsedekah * (30/100),2)
uangkaumdhuafa = round((uangsedekah - uanganakyatim),2)

print (f"Pendapatan Budi selama libur musim panas sebelum melakukan pembayaran pajak adalah Rp {totalgaji}")
```

```

print (f"Pendapatan Budi selama libur musim panas setelah melakukan
pembayaran pajak adalah Rp {pendapatanbersih}")

print (f"Jumlah uang yang Budi habiskan untuk membeli pakaian dan
aksesoris adalah Rp {uangbelanja}")

print (f"Jumlah uang yang Budi habiskan untuk membeli alat tulis adalah
Rp {uungalattulis}")

print (f"Jumlah uang yang Budi sedekahkan adalah Rp {uangsedekah}")

print (f"Jumlah uang yang akan diterima anak yatim adalah Rp
{uanganakyatim}")

print (f"Jumlah uang yang akan diterima kaum dhuafa adalah Rp
{uangkaumdhuafa}")

```

Output :

The screenshot shows a Python IDE with three tabs: soal01.py, soal02.py, and soal03.py. The active tab is soal03.py, which contains the following code:

```

1  #masukkan gaji per jam dan jumlah jam kerja selama 1 minggu
2  gajiperjam = float (input("Berapa gaji per jam Budi? Rp "))
3  jumlahjamkerja1minggu = float (input("Berapa jam kerja Budi selama 1 minggu? "))
4  jumlahminggu = 5
5
6  #menghitung total gaji selama kerja 5 minggu
7  totalgaji = round((gajiperjam * jumlahjamkerja1minggu * jumlahminggu),2)
8
9  #pendapatan bersih setelah potong pajak
10 pendapatanbersih = round(totalgaji - (totalgaji * (14/100)),2)
11
12 #menghitung uang belanja dan alat tulis
13 uangbelanja = round(pendapatanbersih * (10/100),2)
14 uungalattulis = round(pendapatanbersih * (1/100),2)
15
16 #menghitung uang setelah belanja
17 uangsetelahbelanja = pendapatanbersih - (uangbelanja + uungalattulis)
18
19 #menghitung uang yang akan disedekahkan, diterima anak yatim, dan kaum dhuafa
20 uangsedekah = round(uangsetelahbelanja * (25/100),2)
21 uanganakyatim = round(uangsedekah * (30/100),2)
22 uangkaumdhuafa = round((uangsedekah - uanganakyatim),2)

```

Below the code editor, the output of the program is displayed:

```

24 print (f"Pendapatan Budi selama libur musim panas sebelum melakukan pembayaran pajak adalah Rp {totalgaji}")
25 print (f"Pendapatan Budi selama libur musim panas setelah melakukan pembayaran pajak adalah Rp {pendapatanbersih}")
26 print (f"Jumlah uang yang Budi habiskan untuk membeli pakaian dan aksesoris adalah Rp {uangbelanja}")
27 print (f"Jumlah uang yang Budi habiskan untuk membeli alat tulis adalah Rp {uungalattulis}")
28 print (f"Jumlah uang yang Budi sedekahkan adalah Rp {uangsedekah}")
29 print (f"Jumlah uang yang akan diterima anak yatim adalah Rp {uanganakyatim}")
30 print (f"Jumlah uang yang akan diterima kaum dhuafa adalah Rp {uangkaumdhuafa}")
31

```

The terminal output shows the following results:

```

PS C:\Users\LENOVO> python -u "c:\Users\LENOVO\OneDrive\valine\Prak Alpro\minggu02\soal03.py"
Berapa gaji per jam Budi? Rp 95000
Berapa jam kerja Budi selama 1 minggu? 35
Pendapatan Budi selama libur musim panas sebelum melakukan pembayaran pajak adalah Rp 16625000.0
Pendapatan Budi selama libur musim panas setelah melakukan pembayaran pajak adalah Rp 14297500.0
Jumlah uang yang Budi habiskan untuk membeli pakaian dan aksesoris adalah Rp 1429750.0
Jumlah uang yang Budi habiskan untuk membeli alat tulis adalah Rp 142975.0
Jumlah uang yang Budi sedekahkan adalah Rp 3181193.75
Jumlah uang yang akan diterima anak yatim adalah Rp 954358.12
Jumlah uang yang akan diterima kaum dhuafa adalah Rp 2226835.63

```

Langkah-langkah :

1. Soal meminta agar pengguna dapat menginput gaji per jam dan jumlah jam kerja yang dilakukan dalam waktu 1 minggu.
2. Buat *variable* gaji per jam dan jumlah jam kerja 1 minggu dengan menggunakan fungsi input dan float agar nilai yang diinput bisa berupa bilangan desimal.
3. Buat juga *variable* jumlah minggu dengan value 5 (minggu).
4. Hitung total gaji selama kerja 5 minggu dengan mengalikan gaji per jam, jumlah jam kerja selama 1 minggu, dan jumlah minggu dengan menggunakan round untuk pembulatan.
5. Kemudian hitung pendapatan bersih setelah dipotong pajak 14% dengan menggunakan round untuk pembulatan.
6. Hitung uang yang diperlukan untuk belanja baju dan aksesoris dengan mengalikan pendapatan bersih dan 10% menggunakan round untuk pembulatan.
7. Hitung uang yang diperlukan untuk beli alat tulis dengan mengalikan pendapatan bersih dan 1% menggunakan round untuk pembulatan.
8. Hitung sisa uang setelah belanja baju, aksesoris, dan alat tulis dengan mengurangi pendapatan bersih dan uang belanja.
9. Untuk menghitung besar uang yang akan disedekahkan yaitu dengan mengalikan sisa uang setelah belanja dengan 25% menggunakan round untuk pembulatan.
10. Kemudian untuk menghitung uang yang akan disedekahkan ke anak yatim dengan mengalikan 30% dan uang sedekah menggunakan round untuk pembulatan.
11. Untuk mencari uang yang akan disedekahkan ke kaum dhuafa dapat langsung mengurangi total uang sedekah dan uang untuk anak yatim menggunakan round untuk pembulatan.
12. Lalu gunakan fungsi print untuk menampilkan pendapatan sebelum dan setelah membayar pajak, uang untuk membeli pakaian dan aksesoris, alat tulis, uang untuk sedekah, uang untuk anak yatim, dan uang untuk kaum dhuafa.