

**LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGRAMAN
KOMPUTER
MODUL 3
OPERASI DASAR INPUT DAN OUTPUT**

Laporan ini disusun untuk memenuhi Tugas Mata Kuliah
Praktikum Basis Data



Disusun Oleh :

AHSAN MAULANA RIZQI

104062400071

S1 BD-04-B

**PROGRAM STUDI S1 BISNIS DIGITAL
TELKOM UNIVERSITY PURWOKERTO
2025**

1. Dasar Teori

1.1. Indentasi

Indentasi mengacu pada spasi atau tab yang digunakan di awal baris kode untuk menunjukkan tingkat penumpukannya dalam blok. Dalam Python, indentasi bukan hanya masalah gaya atau preferensi, tetapi merupakan persyaratan sintaksis. Tidak seperti bahasa pemrograman lain yang menggunakan kurung kurawal atau kata kunci untuk mendefinisikan blok kode.

Aturan Dasar Indentasi: Dalam Python, setiap baris kode dalam satu blok harus diindentasi dengan jumlah spasi atau tab yang sama. Konvensi standar adalah menggunakan empat spasi untuk indentasi. Sebaiknya hindari penggunaan tab untuk indentasi guna mencegah tercampurnya tab dan spasi.

Tingkat Indentasi dan Nesting: Python menggunakan indentasi untuk menunjukkan tingkat nesting dalam sebuah blok. Setiap blok nesting harus diindentasi lebih jauh dari blok induknya. Ini membantu dalam merepresentasikan hierarki blok kode secara visual.

1.2. Komentar

Komentar adalah kode yang ada pada *script* python dan tidak kode tersebut tidak akan dieksekusi. Komentar biasanya digunakan untuk memberikan informasi tambahan terhadap script yang ditulis. Untuk menuliskan komentar diawali dengan menuliskan tanda pagar (#).

1.3. Python *Input* dan *Output*

Input dan *output* merupakan bagian dalam bahasa pemrograman yang sangat penting. *Input* adalah masukan yang kita berikan kepada program lalu akan muncul outputnya. *Output* merupakan hasil keluaran dari sesuatu yang telah kita input sebelumnya. salah satu fungsi tersebut yaitu fungsi `print()`, fungsi *print()* pada python digunakan untuk melakukan operasi output sedangkan untuk operasi input pada python menggunakan fungsi *input()*.

Catatan: Segala bentuk inputan yang ditambahkan menggunakan fungsi `input()` tipe datanya akan dianggap sebagai string. Jadi ketika kita inputkan angka maka angka tersebut akan tetap dianggap sebagai string, untuk itu kita perlu melakukan yang namanya casting tipe data atau mengkonversi data yang diinputkan sesuai dengan apa yang kita harapkan

1.4. *Casting* Tipe Data

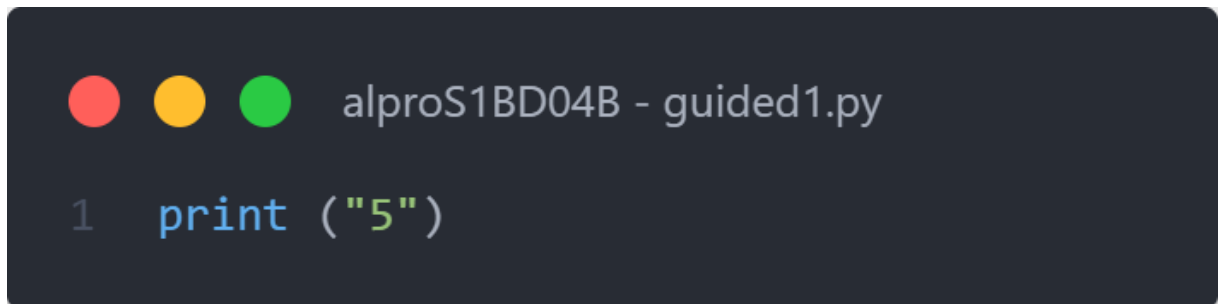
Casting tipe data adalah mengubah tipe data tertentu menjadi tipe data lainnya. Contoh kita mengubah tipe data integer menjadi tipe data float, supaya bisa berubah maka perlu menggunakan casting. Berikut fungsi casting yang bisa digunakan

Tabel 1. Fungsi-fungsi casting dan kegunaannya

Fungsi Casting	Keterangan
int()	Mengubah menjadi integer
float()	Mengubah menjadi float
bool()	Mengubah menjadi Boolean
chr()	Mengubah menjadi karakter
str()	Mengubah menjadi string
bin()	Mengubah menjadi bilangan biner
hex()	Mengubah menjadi bilangan heksadesimal
oct()	Mengubah menjadi bilangan oktal

2. Pembahasan Tugas Guided

2.1. Indentasi



```

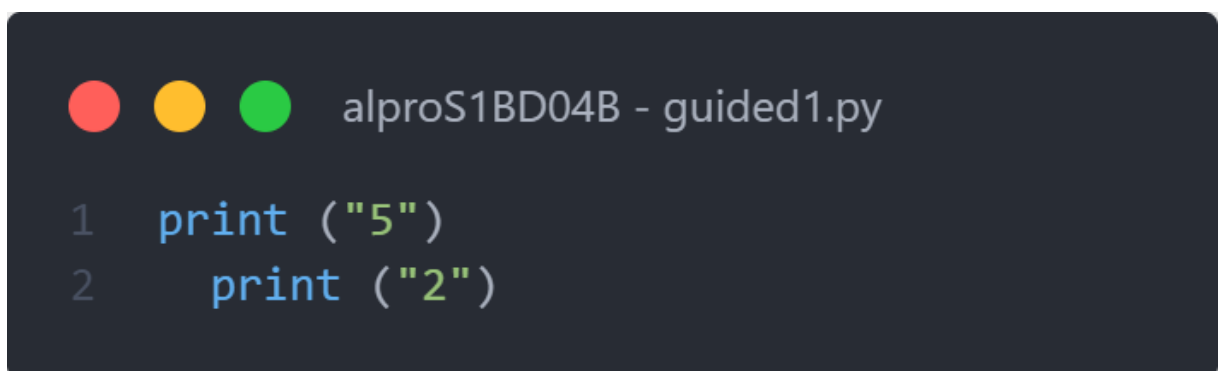
alproS1BD04B - guided1.py
1  print ("5")

```

Gambar 1. Print angka 5

Gambar 2. Hasil output

Perintah `print ("5")` pada gambar 1 di baris 1 akan menghasilkan output gambar 2.

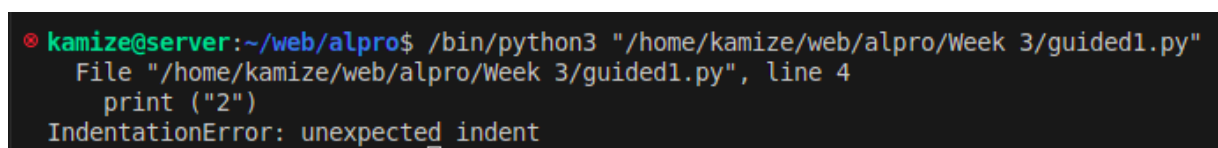


```

alproS1BD04B - guided1.py
1  print ("5")
2    print ("2")

```

Gambar 3. Print angka menggunakan indentasi



```

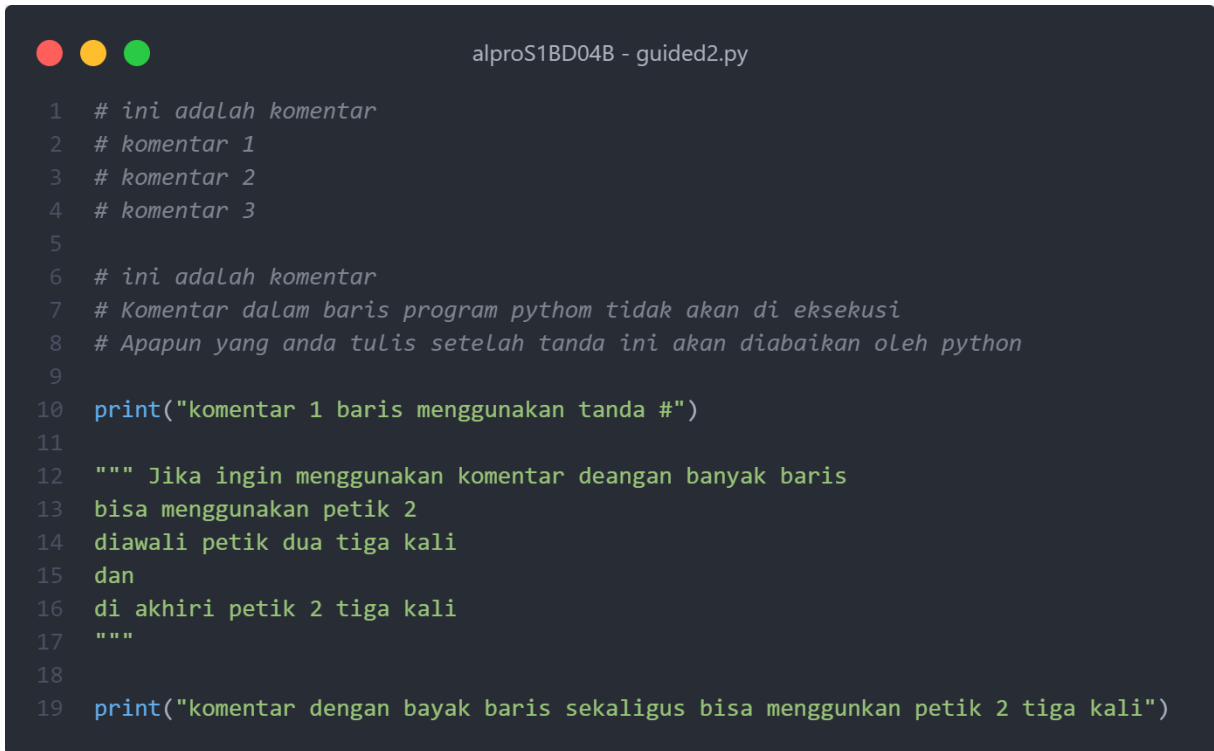
kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 3/guided1.py"
File "/home/kamize/web/alpro/Week 3/guided1.py", line 4
    print ("2")
IndentationError: unexpected indent

```

Gambar 4. output error

Penggunaan indentasi yang salah seperti pada contoh gambar 3 yang akan menghasilkan output error seperti pada gambar 4. Pesan *syntax error: unexpected indent* yang artinya terjadi error yang berupa kesalahan dalam pemberian indentasi.

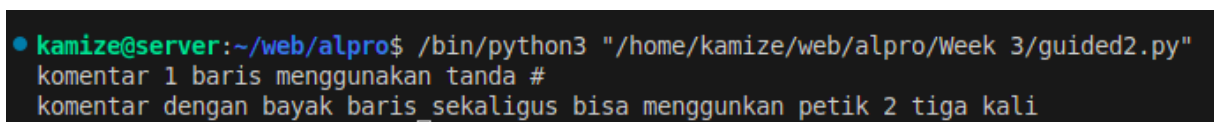
2.2. Comment



```
alproS1BD04B - guided2.py
1  # ini adalah komentar
2  # komentar 1
3  # komentar 2
4  # komentar 3
5
6  # ini adalah komentar
7  # Komentar dalam baris program pythom tidak akan di eksekusi
8  # Apapun yang anda tulis setelah tanda ini akan diabaikan oleh python
9
10 print("komentar 1 baris menggunakan tanda #")
11
12 """ Jika ingin menggunakan komentar deangan banyak baris
13 bisa menggunakan petik 2
14 diawali petik dua tiga kali
15 dan
16 di akhiri petik 2 tiga kali
17 """
18
19 print("komentar dengan bayak baris sekaligus bisa menggunakan petik 2 tiga kali")
```

Gambar 5. Contoh pemggunaan komentar

Baris kode yang diawali dengan tanda pagar (#) menunjukkan komentar untuk satu baris. Sedangkan baris yang diawali dan diakhiri dengan tanda perik dua (""" ... """) tiga kali menunjukkan komentar dengan banya baris.



```
kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 3/guided2.py"
komentar 1 baris menggunakan tanda #
komentar dengan bayak baris sekaligus bisa menggunakan petik 2 tiga kali
```

Gambar 6. Output komentar

kode program pada gambar 5 akan menghasilkan output pada gambar 6. Yang dimana komenter tidak di eksekusi dan hanya dijadikan keterangan pada *source code*.

2.3. Output



```

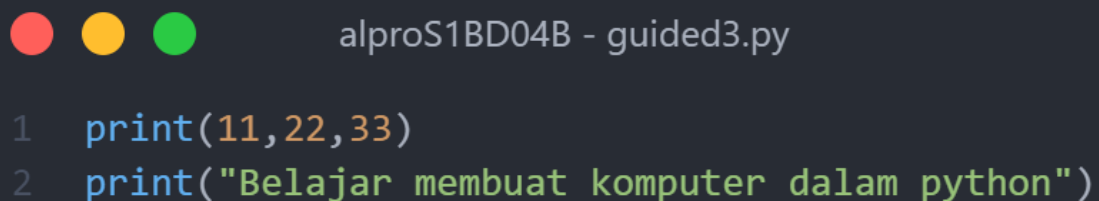
alproS1BD04B - guided3.py

1  # Latihan Output
2  print(11,22,33)
3  print("Belajar membuat komputer dalam python")
4
5  prodi = "S1 Bisnis Digital"
6  kampus = "Tel-U Purwokerto"
7  print(f"program studi : {prodi},{kampus}")
8
9  # Latihan output menggunakan parameter sep dan end
10 # menggunakan parameter sep= " " dan end= " "
11
12 print ("Purwokerto","Banyumas","Jawa Tengah",sep="-")
13 print ("30","maret","1950", sep="/")
14 print ("Purwokerto","Banyumas","Jawa Tengah",end="****")
15 print ("Republik", end= " ")
16 print ("Indonesia")

```

Gambar 7. Penggunaan fungsi print()

Kode di gambar 7 adalah latihan penggunaan fungsi `print()` dalam Python untuk menampilkan output ke layar dengan berbagai format. Fungsi `print()` memiliki beberapa parameter opsional seperti `sep` (separator) dan `end`, yang digunakan untuk mengatur cara elemen-elemen dalam output dipisahkan dan bagaimana baris output diakhiri.



```

alproS1BD04B - guided3.py

1  print(11,22,33)
2  print("Belajar membuat komputer dalam python")

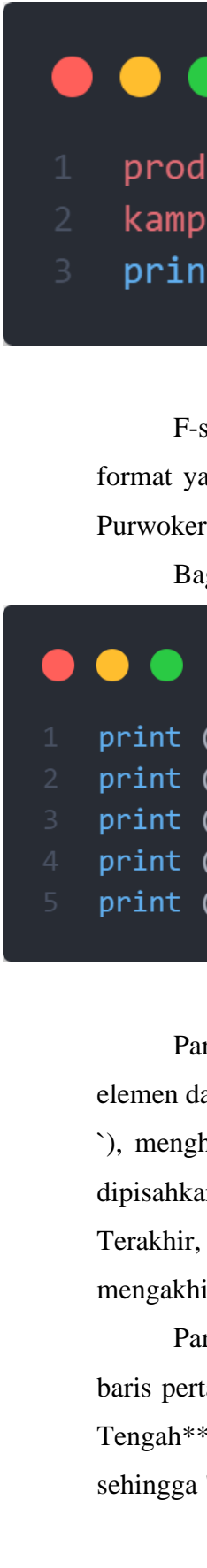
```

Gambar 8. fungsi print() angka dan teks

Pada gambar 8, kode menampilkan angka dan teks sederhana:

Baris pertama mencetak tiga angka (11, 22, 33) yang secara default dipisahkan oleh spasi karena parameter `sep` tidak diubah. Baris kedua mencetak string "Belajar membuat komputer dalam python".

Selanjutnya, kode menggunakan string interpolasi (f-string) untuk mencetak variabel `prodi` dan `kampus`:

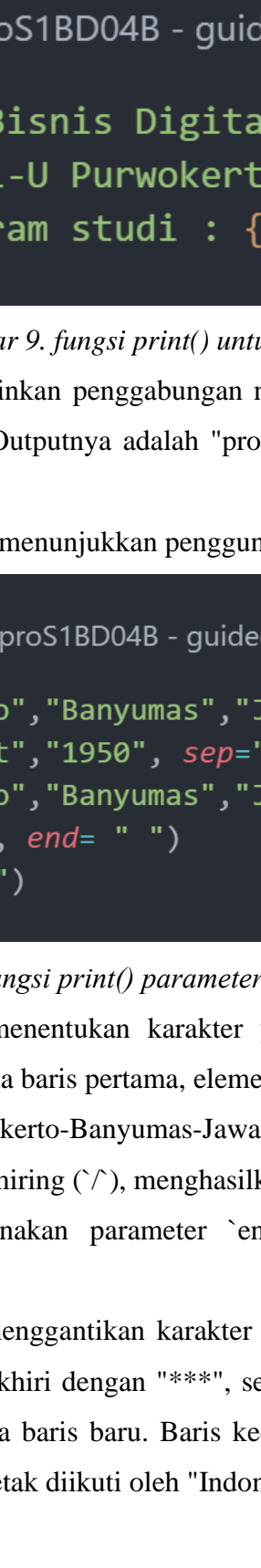


```
1 prodi = "S1 Bisnis Digital"
2 kampus = "Tel-U Purwokerto"
3 print(f"program studi : {prodi},{kampus}")
```

Gambar 9. fungsi print() untuk variabel

F-string memungkinkan penggabungan nilai variabel ke dalam string dengan format yang lebih rapi. Outputnya adalah "program studi : S1 Bisnis Digital,Tel-U Purwokerto".

Bagian berikutnya menunjukkan penggunaan parameter `sep` dan `end`:



```
1 print ("Purwokerto", "Banyumas", "Jawa Tengah", sep="-")
2 print ("30", "maret", "1950", sep="/")
3 print ("Purwokerto", "Banyumas", "Jawa Tengah", end="***")
4 print ("Republik", end=" ")
5 print ("Indonesia")
```

Gambar 10. fungsi print() parameter "sep" dan "end"

Parameter `sep` menentukan karakter yang digunakan untuk memisahkan elemen dalam output. Pada baris pertama, elemen dipisahkan dengan tanda hubung (`-`), menghasilkan "Purwokerto-Banyumas-Jawa Tengah". Pada baris kedua, elemen dipisahkan dengan garis miring (`/`), menghasilkan "30/maret/1950". Terakhir, kode menggunakan parameter `end` untuk mengatur karakter yang mengakhiri baris output:

Parameter `end` menggantikan karakter akhir default (`\n`, baris baru). Pada baris pertama, output diakhiri dengan "***", sehingga "Purwokerto-Banyumas-Jawa Tengah***" dicetak tanpa baris baru. Baris kedua mengakhiri output dengan spasi, sehingga "Republik " dicetak diikuti oleh "Indonesia" pada baris yang sama.

```
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 3/guided3.py"
11 22 33
Belajar membuat komputer dalam python
program studi : S1 Bisnis Digital,Tel-U Purwokerto
Purwokerto-Banyumas-Jawa Tengah
30/maret/1950
Purwokerto Banyumas Jawa Tengah***Republik Indonesia
```

Gambar 11. Hasil Output

2.4. Input



```
alproS1BD04B - guided4.py

1  # variabel input
2  kota = input()
3  alamat = input("Masukan Alamat domisili : ")
4
5  # output
6  print(f"nama kota : {kota}")
7  print("Alamat domisili saya adalah : ", alamat)
```

Gambar 12. Fungsi `input()` dan `print()`

Kode gambar 12 adalah contoh sederhana dari penggunaan fungsi bawaan Python, yaitu `input()` dan `print()`, untuk menerima masukan dari pengguna dan menampilkan hasilnya ke layar.

`kota = input()`: Fungsi `input()` dipanggil tanpa argumen, sehingga tidak ada pesan prompt yang ditampilkan. Program akan menunggu pengguna memasukkan teks, yang kemudian disimpan dalam variabel `kota`.

`alamat = input("Masukan Alamat domisili : ")`: Fungsi `input()` dipanggil dengan argumen *string* "Masukan Alamat domisili : ". *String* ini ditampilkan sebagai prompt untuk memberi tahu pengguna apa yang harus dimasukkan. Masukan pengguna kemudian disimpan dalam variabel `alamat`.

Fungsi `input()` selalu mengembalikan data dalam bentuk *string*, sehingga masukan pengguna akan disimpan sebagai teks, terlepas dari apa yang dimasukkan.

`print(f"nama kota : {kota}")`: Fungsi `print()` digunakan untuk mencetak teks ke layar. Dalam hal ini, *f-string* (`f"..."`) digunakan untuk menyisipkan nilai variabel `kota` ke dalam *string*. Misalnya, jika pengguna memasukkan "Purwokerto" untuk `kota`, outputnya akan menjadi `nama kota : Purwokerto`.

`print("Alamat domisili saya adalah : ", alamat)` : Fungsi `print()` mencetak teks statis "Alamat domisili saya adalah : " diikuti oleh nilai variabel `alamat`. Fungsi ini secara default memisahkan argumen dengan spasi (`sep=" "`).

```
kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 3/guided4.py"
Purwokerto
Masukan Alamat domisili : JL. Panjaitan no. 128
nama kota : Purwokerto
Alamat domisili saya adalah : JL. Panjaitan no. 128
```

Gambar 13. Hasil Output

2.5. Casting Type Data

Kode gambar 14 menunjukkan penggunaan *casting* (konversi tipe data) dalam Python, serta perbedaan hasil operasi sebelum dan sesudah casting. Berikut adalah penjelasan rinci:

```
alproS1BD04B - guided5.py

1  # casting
2  print (float(60)) #casting float
3  print (int (5.8967)) #casting integer
```

Gambar 14. Casting float dan integer

`float(60)` : Fungsi `float()` mengonversi bilangan bulat 60 menjadi bilangan desimal (*floating-point*). Outputnya adalah 60.0. `int(5.8967)`: Fungsi `int()` mengonversi bilangan desimal 5.8967 menjadi bilangan bulat dengan membuang bagian desimalnya. Outputnya adalah 5.

```
kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 3/guided5.py"
60.0
5
kamize@server:~/web/alpro$
```

Gambar 15. Hasil Output

```
alproS1BD04B - guided5.py

1  # variabel sebelum casting tipe data
2  angka1 = input("Masukan Angka pertama : ")
3  angka2 = input("Masukan Angka kedua : ")
4  hasil = angka1+angka2 # operasi penjumlahan
5
6  print ("Hasil penjumlahan", angka1, "dan", angka2, "adalah", hasil) # output string
```


Gambar 16. Sebelum casting integer

Fungsi `input()` menerima masukan dari pengguna dalam bentuk *string*. Variabel `angka1` dan `angka2` menyimpan nilai *string* yang dimasukkan. Operasi `angka1 + angka2` adalah penggabungan *string* (*concatenation*), bukan penjumlahan numerik. Misalnya, jika pengguna memasukkan 234 untuk `angka1` dan 966 untuk `angka2`, hasilnya adalah *string* 234966. Fungsi `print()` kemudian menampilkan hasil penggabungan *string* tersebut.

```
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 3/guided5.py"
Masukan Angka pertama : 234
Masukan Angka kedua   : 966
Hasil penjumlahan 234 dan 966 adalah 234966
○ kamize@server:~/web/alpro$
```

Gambar 17. Hasil Output

```
alproS1BD04B - guided5.py
1 # variabel setelah casting tipe data
2 angka1 = int(input("Masukan Angka pertama : "))
3 angka2 = int(input("Masukan Angka kedua   : "))
4 hasil = angka1+angka2 # operasi penjumlahan
5
6 print ("Hasil penjumlahan", angka1, "dan", angka2, "adalah", hasil) # output integer
```

Gambar18. Setelah casting integer

Fungsi `int()` digunakan untuk mengonversi masukan *string* dari `input()` menjadi bilangan bulat (*integer*). Dengan demikian, variabel `angka1` dan `angka2` sekarang menyimpan nilai numerik. Operasi `angka1 + angka2` adalah penjumlahan numerik. Misalnya, jika pengguna memasukkan 234 untuk `angka1` dan 966 untuk `angka2`, hasilnya adalah bilangan bulat 1200. Fungsi `print()` menampilkan hasil penjumlahan numerik tersebut.

```
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 3/guided5.py"
Masukan Angka pertama : 234
Masukan Angka kedua   : 966
Hasil penjumlahan 234 dan 966 adalah 1200
○ kamize@server:~/web/alpro$
```

Gambar 19. Hasil Output

3. Pembahasan Tugas Unguided

3.1. Tuliskan kode di dalamnya sehingga hasil yang didapatkan seperti berikut :

```

      *
    * *
  *   *
 *     *
***   ***
 *     *
 *     *
*****
```

```
alpro - unguided1.2.py

1  # inisialisasi variabel
2  sisi = 9
3  spasi = int(sisi/2) # spasi awal
4  # loop untuk pola segitiga
5  for count in range(1, sisi + 1,2):
6      if count == 1:
7          print(" " * spasi + "*")
8      elif count == sisi:
9          sisi /= 3
10         count /= 3
11         print("*" * int(sisi) + " " * int(count) + "*" * int(sisi))
12     else:
13         print(" " * spasi + "*" + " " * (count - 2) + "*")
14         spasi -= 1
15 # loop untuk pola persegi panjang
16 for i in range(int(sisi)):
17     print(" " * int(sisi/2), '*', " " * int(sisi/2), "*")
18 print(" " * int(sisi/2), '*' * int(sisi+2))
```

Gambar 19. Code isialisasi unguided 1

Inisialisasi variabel sisi dengan nilai 9 sebagai panjang sisi segitiga dan spasi yang dihitung dari setengah nilai sisi untuk menentukan nilai spasi awal sebelum simbol * pada setiap baris segitiga.

For *count in range (1,sisi +1 ,2)* untuk menghasilkan *looping* dari 1 sampai variabel sisi+1 dengan kenaikan 2 yang akan menghasilkan angka ganjil. Jika variabel *count* sama dengan 1 maka cetak * dengan spasi awal sebanyak variabel spasi. Jika variabel *count* sama dengan variabel sisi maka jalankan proses variabel sisi dan count dibagi dengan 3 untuk menentukan jumlah * dan spasi untuk membuat baris terakhir segitiga. Jika kondisi sebelumnya tidak ada yang terpenuhi maka akan mencetak dua simbol * dengan spasi diataranya. Jumlah spasi sebanyak variabel count – 2. Variabel spasi akan dikurangi 1 disetiap barisnya untuk menciptakan pola runcing.

For *count in range (sisi)* untuk menghasilkan *looping* dari 1 sampai variabel sisi dengan output spasi sejumlah setengah dari variabel sisi dan 2 simbol * yang diataran keduanya terdapat spasi sebanyak setengah dari variabel sisi. Baris terakhir akan dicetak penuh.

```
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 3/unguided1.2.py"
*
* *
* *
* *
*** ***
* *
* *
* *
*****
○ kamize@server:~/web/alpro$
```

Gambar 20. hasil output unguided 1

3.2. Gunakan fungsi *input()* untuk memasukkan biodata dan gunakan casting tipe data yang benar. Data dari biodata sebagai berikut :

- Nama
- NIM
- Alamat
- Tanggal Lahir
- Jenis Kelamin
- Umur
- Berat Badan
- Tinggi Badan
- No. HP

```
alpro - unguided2.py

1  # program yang meminta inputan dari user berupa
2  nama = str(input("Masukkan nama Anda: "))          # input nama
3  NIM = int(input("Masukkan NIM Anda: "))             # input NIM
4  alamat = str(input("Masukkan alamat Anda: "))      # input alamat
5  tanggal_lahir = str(input("Masukkan tanggal lahir Anda: ")) # input tanggal lahir
6  jenis_kelamin = str(input("Masukkan jenis kelamin Anda: ")) # input jenis kelamin
7  umur = int(input("Masukkan umur Anda: "))          # input umur
8  tinggi = float(input("Masukkan tinggi Anda: "))    # input tinggi
9  berat = float(input("Masukkan berat Anda: "))       # input berat
10
11  display = f"""
12  {"="*10} Data Mahasiswa {"="*10}
13  nama          : {nama}
14  NIM           : {NIM}
15  alamat        : {alamat}
16  tanggal lahir : {tanggal_lahir}
17  jenis kelamin : {jenis_kelamin}
18  umur          : {umur}   tahun
19  tinggi        : {tinggi} cm
20  berat         : {berat}  kg
21  """
22  print(display) # print data mahasiswa
```

Gambar 21. input biodata unguided 2

Inisialisasi variabel input menggunakan casting *str()*, *int()*, dan *float()*. Casting *str()* untuk variabel nama, Alamat, tanggal_lahir, dan jenis_kelamin. Casting *int()* untuk variabel NIM dan umur. Casting *float()* untuk variabel tinggi dan berat. Variabel *display* sebagai template untuk menampilkan data mahasiswa sesuai *input user*.

```
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 3/unguided2.py"
Masukkan nama Anda: AHSAN MAULANA RIZQI
Masukkan NIM Anda: 104062400071
Masukkan alamat Anda: SUMPIUH, BANYUMAS, JAWA TENGAH
Masukkan tanggal lahir Anda: 15 APRIL 2003
Masukkan jenis kelamin Anda: LAKI LAKI
Masukkan umur Anda: 21
Masukkan tinggi Anda: 166
Masukkan berat Anda: 60

===== Data Mahasiswa =====
nama          : AHSAN MAULANA RIZQI
NIM           : 104062400071
alamat        : SUMPIUH, BANYUMAS, JAWA TENGAH
tanggal lahir : 15 APRIL 2003
jenis kelamin : LAKI LAKI
umur          : 21 tahun
tinggi        : 166.0 cm
berat         : 60.0 kg

○ kamize@server:~/web/alpro$
```

Gambar 22. hasil output unguided 2

3.3. Diketahui kurs rupiah dalam dollar sebagai berikut: 1 dollar = 15000 rupiah Buatlah sebuah program konversi dari rupiah menjadi dollar, gunakan fungsi *input()* dan casting.

```
● ● ● alpro - unguided3.py

1  rupiah = int(input("Masukan Nominal : Rp. "))
2  kurs = 15000
3  dolar = rupiah / kurs
4  print(f"Rp. {rupiah} to ${dolar:.2f} USD")
```

Gambar 23. konversi dolar ke rupiah unguided 3

Inisialisasi variabel rupiah dengan fungsi input yang di casting *int()*. Variabel kurs dengan nilai 15000. Variabel dolar untuk menyimpan rumus dasar konversi rupiah ke dolar yaitu variabel rupiah dibagi dengan variabel kurs. Hasil outputnya pada gambar 24.

```

● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 3/unguided3.py"
Masukan Nominal : Rp. 12000
Rp. 12000 to $0.80 USD
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 3/unguided3.py"
Masukan Nominal : Rp. 15000
Rp. 15000 to $1.00 USD
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 3/unguided3.py"
Masukan Nominal : Rp. 97500
Rp. 97500 to $6.50 USD
○ kamize@server:~/web/alpro$ █

```

Gambar 24. hasil output *unguided 3*

4. Ringkasan

Laporan ini menjelaskan tentang konsep-konsep dasar bahasa pemrograman Python, di dalam modul ini menjelaskan tentang indentasi, komentar, input-output, dan konversi tipe data. Indentasi digunakan untuk tanda pada blok kode yang harus digunakan dengan benar agar program yang dijalankan tidak mengalami pesan *syntax error: unexpected indent*. Komentar dalam bahasa pemrograman Python ditulis menggunakan simbol/tanda pagar untuk satu baris sedangkan tanda kutip tiga kali (""" ... """) untuk komentar lebih dari satu baris, yang biasanya digunakan sebagai dokumentasi kode.

Dalam hal input dan output, Bahasa pemrograman Python menggunakan fungsi *print()* untuk menampilkan hasil output ke terminal atau layer, sedangkan *input()* untuk menerima masukan pengguna dari terminal. Fungsi *input()* selalu memperoleh data yang berupa tipe data *string*, sehingga jika memerlukan tipe data selain *string*, maka harus dikonversi menggunakan fungsi *casting* seperti *int()* untuk tipe data integer, *float()* untuk tipe data *float*, atau *bool()* untuk tipe data *boolean*.

Laporan ini juga berisi tugas praktik, seperti penggunaan indentasi, komentar, dan fungsi *print()* serta *input()*. Salah satu contoh latihan adalah penggunaan parameter *sep* dan *end* dalam fungsi *print()* untuk mengatur pemisah elemen *output* dan cara mengakhiri baris. Selain itu, dicontohkan pula penggunaan *casting* untuk mengubah *input string* menjadi angka, agar dapat digunakan dalam operasi aritmatika.

Dalam bagian tugas mandiri (*unguided*), diberikan studi kasus seperti membuat segitiga dengan simbol *, memasukkan biodata dengan *casting* tipe data yang benar, serta konversi nilai mata uang dari Rupiah ke Dollar menggunakan fungsi *input()* dan operasi matematika sederhana.

Dalam laporan ini, dapat disimpulkan bahwa pentingnya memahami indentasi, komentar, input-output, dan konversi tipe data dalam bahasa pemrograman Python. Menggunakan aturan ini dengan benar akan membantu dalam menulis kode yang lebih rapi, mudah dibaca, serta menghindari kesalahan yang dapat menghambat jalannya program.