



ALGORITMA DAN STRUKTUR DATA

INF1083

LAPORAN PRAKTIKUM 2 : Built-in Data & Input-Output

Oleh :

Akhmad Qasim

2211102441237

Teknik Informatika
Fakultas Sains & Teknologi
Universitas Muhammadiyah Kalimantan Timur

Samarinda, 2023

Laporan Praktikum 2:

Built-in Data & Input-Output

Pokok Bahasan:

- ❖ Built-in Data
- ❖ Input-Output

Tujuan Pembelajaran:

- ✓ Memahami jenis-jenis Built-in Data & Input-Output dari Python 3.

Percobaan & Latihan:

1. Numerik

a) Berikan tampilan output dan keterangan penjelasan!

Tampilan Input:

```
1 a = 12
2 b = 37
3
4 print(a + b * 4) # Data type: Integer
5 print((a + b) * 4) # Data type: Integer
6 print(a ** 10) # Data type: Integer
7 print(a / 3) # Data type: Float
8 print(b / 3) # Data type: Float
9 print(a // b) # Data type: Integer
10 print(a % b) # Data type: Integer
11 print(a / b) # Data type: Float
12 print(a // b) # Data type: Integer
13 print(a % b) # Data type: Integer
14 print(b ** 100) # Data type: Integer
```

Tampilan Output:

```
160
196
61917364224
4.0
12.333333333333334
0
12
0.32432432432432434
0
12
66095578288438667743482968577936153209860683
```

1. `a + b * 4`

Variable a memiliki nilai 12 dan variable b memiliki nilai 37, lalu mencetak hasil ekspresi `a + b * 4` ke konsol. Ekspresi `b * 4` mengalikan nilai b dengan 4, menghasilkan nilai 148. Kemudian, `a + 148` menambahkan nilai a (yaitu 12) menjadi 148, menghasilkan hasil akhir 160.

2. `(a + b) * 4`

Variable a memiliki nilai 12 dan variable b memiliki nilai 37, dan kemudian menghitung ekspresi `(a + b) * 4`. Pertama, `a + b` menambahkan nilai a (yaitu 12) ke nilai b (yaitu 37), menghasilkan nilai 49. Kemudian, `49 * 4` mengalikan hasilnya dengan 4, menghasilkan hasil akhir 196.

3. `a ** 10`

Variable a memiliki nilai 12, lalu menghitung nilai a pangkat 10 menggunakan operator asterisk ganda `**`. Ekspresi `a ** 10` melakukan operasi 12 dipangkatkan 10, menghasilkan nilai 61917364224.

4. `a / 3`

Variable a memiliki nilai 12, lalu melakukan operasi pembagian dengan angka 3. Ekspresi `a / 3` menghasilkan nilai 4.

5. `b / 3`

Variable b memiliki nilai 37, lalu melakukan operasi pembagian dengan angka 3. Hasil dari ekspresi `b / 3` menampilkan output 12.33 dari hasil operasi `b / 3`.

6. `a // b`

Variable a memiliki nilai 12 dan variable b memiliki nilai 37, lalu melakukan operasi pembagian bilangan bulat dengan membagi nilai a dengan nilai b. Ekspresi `a // b` melakukan operasi pembagian bilangan bulat, yang membagi nilai a (yaitu 12) dengan nilai b (yaitu 37), menghasilkan nilai 0.

Ini karena hasil pembagiannya lebih kecil dari 1, dan operator // membuang bagian pecahan dari hasil dan hanya mengembalikan bagian bilangan bulat.

7. **a % b**

Variable a memiliki nilai 12 dan variable b memiliki nilai 37, lalu melakukan operasi modulus menggunakan operator %, yang menghitung sisa setelah membagi nilai a dengan nilai b. Ekspresi **a % b** menghitung sisa bagi setelah membagi nilai a (yaitu 12) dengan nilai b (yaitu 37), menghasilkan nilai 12. Ini karena 12 adalah bilangan bulat terbesar yang dapat dikurangkan dari 12 tanpa membuat hasilnya negatif, saat mengurangi kelipatan.

8. **a / b**

Variable a memiliki nilai 12, variable b memiliki nilai 37, ekspresi **a / b** akan melakukan pembagian menggunakan operasi pembagian (/). Hasil ekspresi dari **12 / 37** akan menampilkan output 0,32.

9. **a // b**

Variable a memiliki nilai 12 dan variable b memiliki nilai 37, lalu melakukan operasi pembagian bilangan bulat dengan membagi nilai a dengan nilai b. Ekspresi **a // b** melakukan operasi pembagian bilangan bulat, yang membagi nilai a (yaitu 12) dengan nilai b (yaitu 37), menghasilkan nilai 0. Ini karena hasil pembagiannya lebih kecil dari 1, dan operator // membuang bagian pecahan dari hasil dan hanya mengembalikan bagian bilangan bulat.

10. **a % b**

Variable a memiliki nilai 12 dan variable b memiliki nilai 37, lalu melakukan operasi modulus menggunakan operator %, yang menghitung sisa setelah membagi nilai a dengan nilai b. Ekspresi **a % b** menghitung sisa bagi setelah membagi nilai a (yaitu 12) dengan nilai b (yaitu 37), menghasilkan nilai 12. Ini karena 12 adalah bilangan bulat terbesar yang dapat dikurangkan dari 12 tanpa membuat hasilnya negatif, saat mengurangi kelipatan.

11. **b ** 100**

Variable b memiliki nilai 37 dan kemudian menghitung nilai b pangkat 100 menggunakan operator asterisk ganda **. Ekspresi **b ** 100** menaikkan nilai b pangkat 100, menghasilkan angka yang sangat besar yang memiliki hasil 157 digit dengan hasil 66095578288438667.

b) Tentukan tipe data numerik dari setiap output diatas!

1. Integer
2. Integer
3. Integer
4. Float
5. Float
6. Integer

7. Integer
8. Float
9. Integer
10. Integer
11. Integer

2. Boolean

a) Berikan tampilan output dan keterangan penjelasan!

Syntax:

```

1  a = 12
2  b = 37
3
4  print(a == b)
5  print(b > a)
6  print((b >= 1) and (b <= 10))
7
8  print(3 != 4)
9  print(1 == 1 and 2 < 3)
10 print(1 > 10 or 2 < 3)
11 print(not 1 > 10)
12

```

Tampilan Output:

```

False
True
False

```

1. `a == b`

Variable a memiliki nilai 12 dan variable b memiliki nilai 37. Operator perbandingan *equal* (`==`) akan mengembalikan nilai boolean true atau false. Ekspresi `a == b` akan membandingkan nilai a dan nilai b yaitu `12 == 37` yang akan mengembalikan nilai *false* dikarenakan 27 tidak sama dengan 37.

2. `b > a`

Variable a memiliki nilai 12 dan variable b memiliki nilai 37. Operator lebih dari (`>`) akan mengembalikan nilai Boolean. Ekspresi `b > a` akan mengembalikan nilai *true* dikarenakan 37 lebih besar dibandingkan 12.

3. `(b >= 1) and (b <= 10)`

Variable b memiliki nilai 37. Operator *and* akan membandingkan kedua nilai, yang keduanya harus bernilai *true* untuk mengembalikan nilai *true*. Jika ada nilai *false*, maka akan mengembalikan nilai *false*. Ekspresi `b >= 1`

mengembalikan Boolean *true* dan $b \leq 10$ mengembalikan Boolean *false*, oleh karena itu operator *and* mengembalikan *false*, karena ada salah satu nilai Boolean *false*.

b) Berikan contoh lain menggunakan operator:

1. **!=**

```
print(3 != 4)
```

```
True
```

2. **and**

```
print(1 == 1 and 2 < 3)
```

```
True
```

3. **or**

```
print(1 > 10 or 2 < 3)
```

```
True
```

4. **not**

```
print(not 1 > 10)
```

```
True
```

3. Variable

a) Berikan tampilan output dan keterangan penjelasan!

Syntax:

```
1 MyNim = str(2211102441237) # Merubah tipe data integer ke string
2
3 MyNim_10th = MyNim[9] # Mengambil karakter ke-10 dari string MyNim
4 MyNim_11th = MyNim[-3] # Mengambil karakter ke-3 dari belakang (11) string MyNim
5 i = 11
6 MyNim_12th = MyNim[i] # Mengambil karakter ke-12 dari string MyNim
7 MyNim_13th = MyNim[i+1] # Mengambil karakter ke-13 dari string MyNim
8 a = int(MyNim_10th + MyNim_11th) # Merubah tipe data string ke integer
9 b = int(MyNim_12th + MyNim_13th) # Merubah tipe data string ke integer
10
11 print(MyNim) # Menampilkan isi variabel MyNim
12 print(MyNim_10th) # Menampilkan isi variabel MyNim_10th
13 print(MyNim_11th) # Menampilkan isi variabel MyNim_11th
14 print(MyNim_12th) # Menampilkan isi variabel MyNim_12th
15 print(MyNim_13th) # Menampilkan isi variabel MyNim_13th
16 print(a) # Menampilkan isi variabel a, yaitu 24
17 print("nilai float b = ", b) # Menampilkan isi variabel b, yaitu 37
```

Tampilan Output:

```
2211102441237
1
2
3
7
12
nilai float b = 37
```

b) Apa yang dimaksud dengan variable Python? Apa perbedaan variable dan objek?

Dalam Python, variabel adalah nama yang digunakan untuk menyimpan nilai atau objek di dalam memori komputer, sehingga nilai tersebut dapat diakses dan dimanipulasi selama program berjalan. Variabel dalam Python dapat menunjuk ke objek apa pun seperti angka, string, list, tuple, dictionary, fungsi, dan sebagainya. Variabel dapat dinyatakan dengan nama apapun, tetapi ada beberapa aturan dan konvensi yang harus diikuti. Sebagai contoh, nama variabel harus dimulai dengan huruf atau garis bawah, tidak boleh diawali dengan angka, dan tidak boleh menggunakan spasi.

Sementara itu, objek adalah sesuatu yang menyimpan nilai dan memiliki tipe data tertentu, seperti angka, string, list, tuple, dan sebagainya. Objek dapat dibuat dalam Python menggunakan konstruktor yang sesuai untuk tipe data yang diinginkan. Contohnya, kita dapat membuat objek bilangan bulat (integer) dengan mengetik `x = 5` dan membuat objek string dengan mengetik

```
nama = "hendy"
```

Perbedaan antara variabel dan objek di Python adalah bahwa variabel hanya menunjuk pada objek atau nilai tertentu yang disimpan dalam memori, sementara objek adalah sesuatu yang memiliki nilai dan tipe data tertentu. Variabel adalah nama yang digunakan untuk merujuk pada objek atau nilai tertentu, dan kita dapat mengubah nilai yang ditunjuk oleh variabel tersebut atau bahkan mengubah objek yang ditunjuk oleh variabel, tetapi objek tetap ada di memori sampai objek tersebut tidak digunakan lagi dan dihapus dari memori oleh *garbage collector* Python.

4. List

a) Berikan tampilan output dan keterangan penjelasan!

Syntax:

```
1 myList = [1, 3, True, 6.5] # List berisi integer, boolean, dan float
2 print(MyList) # Menampilkan isi variabel MyList
3 A = [MyList] * 3 # Membuat list A yang berisi 3 kalinya list MyList
4 print(A) # Menampilkan isi variabel A
5 myList[2] = 45 # Mengubah isi list MyList pada indeks ke-2 menjadi 45
6 print(A) # Menampilkan isi variabel A
7
8 myList_A = [1024, 3, True, 6.5] # List berisi integer, boolean, dan float
9 myList_A.append(False) # Menambahkan elemen False ke list myList_A
10 print(myList_A) # Menampilkan isi variabel myList_A
11 myList_A.insert(2, 4.5) # Menambahkan elemen 4.5 ke list myList_A pada indeks ke-2
12 print(myList_A) # Menampilkan isi variabel myList_A
13 print(myList_A.pop()) # Menghapus elemen terakhir dari list myList_A
14 print(myList_A) # Menampilkan isi variabel myList_A
15 print(myList_A.pop(1)) # Menghapus elemen pada indeks ke-1 dari list myList_A
16 print(myList_A) # Menampilkan isi variabel myList_A
17 myList_A.pop(2) # Menghapus elemen pada indeks ke-2 dari list myList_A
18 print(myList_A) # Menampilkan isi variabel myList_A
19 myList_A.sort() # Mengurutkan elemen list myList_A
20 print(myList_A) # Menampilkan isi variabel myList_A
21 myList_A.reverse() # Membalik urutan elemen list myList_A
22 print(myList_A) # Menampilkan isi variabel myList_A
23 print(myList_A.count(6.5)) # Menghitung jumlah kemunculan elemen 6.5 pada list myList_A
24 print(myList_A.index(4.5)) # Menampilkan indeks elemen 4.5 pada list myList_A
25 myList_A.remove(6.5) # Menghapus elemen 6.5 pada list myList_A
26 print(myList_A) # Menampilkan isi variabel myList_A
27 del myList_A[0] # Menghapus elemen pada indeks ke-0 dari list myList_A
28 print(myList_A) # Menampilkan isi variabel myList_A
```

Tampilan Output:

```
[1, 3, True, 6.5]
[[1, 3, True, 6.5], [1, 3, True, 6.5], [1, 3, True, 6.5]]
[[1, 3, 45, 6.5], [1, 3, 45, 6.5], [1, 3, 45, 6.5]]
[1024, 3, True, 6.5, False]
[1024, 3, 4.5, True, 6.5, False]
False
[1024, 3, 4.5, True, 6.5]
3
[1024, 4.5, True, 6.5]
[1024, 4.5, 6.5]
[4.5, 6.5, 1024]
[1024, 6.5, 4.5]
1
2
[1024, 4.5]
[4.5]
[1, 2, 3, 4, 5, 6]
3
[2, 3]
```


- b) Dibaris ke berapa terjadi kesalahan? Berikan alasannya dan perbaiki baris tersebut sehingga perintah tersebut dapat dijalankan!

```

1 myList = [1, 3, True, 6.5] # List berisi integer, boolean, dan float
2 print(myList) # Menampilkan isi variabel myList
3 A = [myList] * 3 # Membuat list A yang berisi 3 kali list myList
4 print(A) # Menampilkan isi variabel A
5 myList[2] = 45 # Mengubah isi list myList pada indeks ke-2 menjadi 45
6 print(A) # Menampilkan isi variabel A
7

```

Kesalahan terjadi pada baris ke-3 dan ke-5 dikarenakan nama variabel yang belum ada (salah dalam penetikkan). Dalam mengatasi permasalahan tersebut, nama variabel akan diganti menjadi myList karena sudah didefinisikan pada baris pertama.

```

1 myList = [1, 3, True, 6.5] # List berisi integer, boolean, dan float
2 print(myList) # Menampilkan isi variabel myList
3 A = [myList] * 3 # Membuat list A yang berisi 3 kali list myList
4 print(A) # Menampilkan isi variabel A
5 myList[2] = 45 # Mengubah isi list myList pada indeks ke-2 menjadi 45
6 print(A) # Menampilkan isi variabel A

```

- c) Berikan contoh penggunaan operator lain dibawah ini untuk list diperintah tersebut:

1. +

```

30 # Contoh "+" dalam penggunaan list
31 myList_A = [1, 2, 3]
32 myList_B = [4, 5, 6]
33 print(myList_A + myList_B)

```

Menampilkan hasil penggabungan list myList_A dan myList_B

2. Len

```

35 # Contoh "len" dalam penggunaan list
36 myList_A = [1, 2, 3]
37 print(len(myList_A))

```

Menampilkan panjang list myList_A

3. [:]

```

39 # Contoh "[:]" dalam penggunaan list
40 myList_A = [1, 2, 3]
41 print(myList_A[1:])

```

Menampilkan list myList_A mulai dari indeks ke-1 sampai akhir

5. List

- a) Berikan tampilan output dan keterangan penjelasan!

Syntax:

```

1  # Menampilkan hasil dari beberapa contoh method
2  print("Berikut ini adalah hasil contoh fragment dari (54).__ad__(21) : \n", (54).__add__(21))
3
4  print(range(10)) # Menampilkan range(0, 10)
5  print(list(range(10))) # Menampilkan list dari range 0 sampai 9
6  print(list(range(5, 10))) # Menampilkan list dari range 5 sampai 9
7  print(list(range(5, 15, 3))) # Menampilkan list dari range 5 sampai 14 dengan interval 3
8  print(list(range(10, 1, -1))) # Menampilkan list dari range 10 sampai 2 dengan interval -1
9
10 myName = "Akhmad Qasim" # String berisi nama "Akhmad Qasim"
11 print(len(myName)) # Menampilkan panjang string myName
12 print(myName[3]) # Menampilkan karakter ke-4 dari string myName
13 print(list(myName)) # Menampilkan list dari string myName
14 print(myName.upper()) # Menampilkan string myName dalam huruf kapital
15 print(myName.center(10)) # Menampilkan string myName menjadi yang di tengah dengan lebar 10
16 print(myName.find("a")) # Menampilkan indeks dari karakter "a" pada string myName
17 print(myName.split("a")) # Menampilkan list dari string myName yang dipisahkan oleh karakter "a"

```

Tampilan Output:

```

Berikut ini adalah hasil contoh fragment dari (54).__ad__(21) :
 75
range(0, 10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[5, 6, 7, 8, 9]
[5, 8, 11, 14]
[10, 9, 8, 7, 6, 5, 4, 3, 2]
12
m
['A', 'k', 'h', 'm', 'a', 'd', ' ', 'Q', 'a', 's', 'i', 'm']
AKHMAD QASIM
Akhmad Qasim
4
['Akhm', 'd Q', 'sim']
2
Akhmad Qasim
    Akhmad Qasim

```

b) Berikan contoh penggunaan metode lain dibawah ini untuk myName diperintah tersebut:

1. `count()`

```

19 # Contoh count() pada myName
20 print(myName.count("a"))

```

Menampilkan jumlah kemunculan karakter "a" pada string myName

2. **ljust()**

```
22 # Contoh ljust() pada myName
23 print(myName.ljust(20))
```

Menampilkan string myName menjadi yang di kiri dengan lebar 20

3. **rjust()**

```
25 # Contoh rjust() pada myName
26 print(myName.rjust(20))
```

Menampilkan string myName menjadi yang di kanan dengan lebar 20

c) **Apakah variabel myName dapat diganti dengan tipe data integer?**

Dalam Python mendeklarasikan ulang sebuah variable dengan tipe data yang berbeda diperbolehkan. Perubahan boleh dilakukan jika tidak menyebabkan masalah. Tetapi sebaiknya untuk mengikuti kaidah konsistensi disarankan untuk tetap menggunakan satu tipe data saja dalam sebuah variable. Python memiliki beberapa fungsi bawaan (built-in functions) yang dapat digunakan untuk mengubah tipe data dari satu tipe ke tipe lainnya. Salah satu fungsi yang dapat digunakan untuk mengubah tipe data string menjadi tipe data integer adalah fungsi int(). Fungsi ini mengonversi string yang berisi representasi bilangan bulat menjadi bilangan bulat (integer). Namun, perlu diingat bahwa string yang dikonversi harus berisi hanya angka atau karakter khusus seperti tanda minus (-) untuk bilangan negatif, dan tidak boleh ada karakter atau spasi di antaranya.

d) **Apa yang terjadi jika ditambahkan perintah berikut? Jelaskan!**

```
myName[0] = 'x'
```

Jika sebuah variabel myName dideklarasikan sebagai `myName = "Akhamad Qasim"`, dan kemudian dijalankan perintah `myName[0] = 'x'`, maka akan terjadi error. Hal ini terjadi karena string di Python bersifat immutable, artinya kita tidak dapat mengubah karakter-karakter dalam string tersebut setelah string itu dibuat.

6. Tuple

a) **Berikan tampilan output dan keterangan penjelasan!**

Syntax:

```
1 myTuple = (2, True, 4.96) # Tuple berisi integer, boolean, dan float
2 print(myTuple) # Menampilkan isi variabel myTuple
3 print(len(myTuple)) # Menampilkan panjang tuple myTuple
4 print(myTuple[0]) # Menampilkan elemen tuple myTuple pada indeks ke-0
5 print(myTuple * 3) # Menampilkan 3 kali isi tuple myTuple
6 print(myTuple[0:2]) # Menampilkan elemen tuple myTuple pada indeks ke-0 dan ke-1
```

Tampilan Output:

```
(2, True, 4.96)
3
2
(2, True, 4.96, 2, True, 4.96, 2, True, 4.96)
(2, True)
```

b) Tambahkan perintah dibawah ini dan berikan penjelasan:

```
myTuple[1] = False
```

Mengubah isi tuple myTuple pada indeks ke-1 menjadi False tetapi akan menghasilkan error karena tuple bersifat immutable.

7. Set

a) Berikan tampilan output dan keterangan penjelasan!

Syntax:

```
1 mySet = {3, 6, "kucing", 4.5, False} # Set berisi integer, string, dan boolean
2 print(mySet) # Menampilkan isi variabel mySet
3 print(len(mySet)) # Menampilkan panjang set mySet
4 print(False in mySet) # Menampilkan apakah elemen False ada di dalam set mySet
5 print("cat" in mySet) # Menampilkan apakah elemen "cat" ada di dalam set mySet
6
7 yourSet = {99, 3, 100, "kuda"} # Set berisi integer, string, dan boolean
8 print(mySet.union(yourSet)) # Menampilkan gabungan dari set mySet dan yourSet
9 print(mySet | yourSet) # Menampilkan gabungan dari set mySet dan yourSet
10 print(mySet.intersection(yourSet)) # Menampilkan irisan dari set mySet dan yourSet
11 print(mySet & yourSet) # Menampilkan irisan dari set mySet dan yourSet
12 print(mySet.difference(yourSet)) # Menampilkan selisih dari set mySet dan yourSet
13 print(mySet - yourSet) # Menampilkan selisih dari set mySet dan yourSet
14 print({3100}.issubset(yourSet)) # Menampilkan apakah set {3100} merupakan subset dari yourSet
15 print({3100} <= yourSet) # Menampilkan apakah set {3100} merupakan subset dari yourSet
16
17 mySet.add("rumah") # Menambahkan elemen "rumah" ke dalam set mySet
18 print(mySet) # Menampilkan isi variabel mySet
19 mySet.remove(4.5) # Menghapus elemen 4.5 dari set mySet
20 print(mySet) # Menampilkan isi variabel mySet
21 print(mySet.pop()) # Menghapus elemen acak dari set mySet
22 print(mySet) # Menampilkan isi variabel mySet
23
24 mySet.clear() # Menghapus semua elemen dari set mySet
25 print(mySet) # Menampilkan isi variabel mySet
```

Tampilan Output:

```
{False, 3, 4.5, 6, 'kucing'}
5
True
False
{False, 3, 4.5, 99, 6, 100, 'kucing', 'kuda'}
{False, 3, 4.5, 99, 6, 100, 'kucing', 'kuda'}
{3}
{3}
{False, 'kucing', 4.5, 6}
{False, 'kucing', 4.5, 6}
False
False
{False, 'rumah', 3, 4.5, 6, 'kucing'}
{False, 'rumah', 3, 6, 'kucing'}
False
{'rumah', 3, 6, 'kucing'}
set()
```

8. Dictionary

a) Berikan tampilan output dan keterangan penjelasan!

Syntax:

```
1 capitals = {"KalTim": "Samarinda", "Bali": "Denpasar"} # variabel capitals berisi dictionary
2 print(capitals["KalTim"]) # Menampilkan isi variabel capitals pada key "KalTim"
3 capitals["JaTim"] = "Surabaya" # Menambahkan key "JaTim" dengan value "Surabaya" ke dictionary capitals
4 print(capitals) # Menampilkan isi variabel capitals
5 capitals["KalSel"] = "Banjarmasin" # Menambahkan key "KalSel" dengan value "Banjarmasin" ke dictionary capitals
6 print(len(capitals)) # Menampilkan panjang dictionary capitals
7 for k in capitals: # Menampilkan isi dictionary capitals
8     print(capitals[k], "adalah ibukota dari ", k) # Menampilkan isi dictionary capitals
```

Tampilan Output:

```
Samarinda
{'KalTim': 'Samarinda', 'Bali': 'Denpasar', 'JaTim': 'Surabaya'}
4
Samarinda adalah ibukota dari KalTim
Denpasar adalah ibukota dari Bali
Surabaya adalah ibukota dari JaTim
Banjarmasin adalah ibukota dari KalSel
```

b) Berikan contoh dictionary yang berisi tipe data integer dan float!

```
10 # contoh dictionary yang berisi tipe data integer dan float
11 contoh = {"a": 1, "b": 2, "c": 3, "d": 4.5}
```

9. Input & Output 1

a) Berikan tampilan output dan keterangan penjelasan!

Syntax:

```
1 aName = input("Nama lengkap saya adalah ") # Input nama
2 print("Huruf kapital untuk nama anda ", aName.upper(), # Menampilkan nama dalam huruf kapital
3      " dan mempunyai karakter sebanyak ", len(aName)) # Menampilkan panjang nama
4
```

Tampilan Output:

```
Nama lengkap saya adalah Akhmad Qasim
Huruf kapital untuk nama anda AKHMAD QASIM dan mempunyai karakter sebanyak 12
Process finished with exit code 0
```

b) Apakah input dapat berupa sebuah bilangan? Berikan penjelasan!

Input yang dimasukkan juga dapat berupa bilangan, tetapi fungsi

`aName.upper()` tidak dapat digunakan, dikarenakan fungsi tersebut hanya dapat mengubah string alphabet menjadi huruf besar. Tidak terjadi error dikarenakan number yang dimasukkan kedalam `aName.upper()` akan diubah terlebih dahulu menjadi string dan akan menampilkan hasilnya seperti gambar di bawah ini.

```
Nama lengkap saya adalah 123
Huruf kapital untuk nama anda 123 dan mempunyai karakter sebanyak 3
Process finished with exit code 0
```

10. Input & Output 2

a) Berikan tampilan output dan keterangan penjelasan!

Syntax:

```
1 sisiKubus = input("Input nilai sisi persegi ") # Input nilai sisi
2 sisi = float(sisiKubus) # Konversi ke tipe data float
3 luas = 2 * sisi # Menghitung luas dengan rumus 2 x sisi
4 print("Luas persegi yaitu ", luas) # Menampilkan hasil
```

Tampilan Output:

```
Input nilai sisi persegi 5
Luas persegi yaitu 10.0
```

b) Apakah input dapat berupa tipe data string? Berikan penjelasan!

Jika tipe data string dimasukkan ke dalam input akan terjadi error pada saat konversi tipe data ke float, karena string tidak dapat dikonversi menjadi tipe data float seperti gambar dibawah.

```
Input nilai sisi persegi masalah
Traceback (most recent call last):
  File "C:\Users\PC-LAB-2BA3\Documents\Mata Kuliah\Algoritma dan Struktur Data\Laporan 02\Source Code\2.10.py", line 2, in <module>
    sisi = float(sisiKubus) # Konversi ke tipe data float
ValueError: could not convert string to float: 'masalah'

Process finished with exit code 1
```

11. Input & Output 3

a) Berikan tampilan output dan keterangan penjelasan!

Syntax:

```
1  aName = input("Nama: ") # Input nama
2  myNim = input("Nim: ") # Input nim
3
4  print(aName, myNim) # Menampilkan nama dan nim
5  print(aName, myNim, sep="-", end=".")
6  # Menampilkan nama dan nim dengan separator - dan diakhiri dengan titik
```

Tampilan Output:

```
Nama: Akhmad Qasim
Nim: 2211102441237
Akhmad Qasim 2211102441237
Akhmad Qasim-2211102441237.
```

b) Tambahkan perintah berikut dibaris ke-6 di kode program anda dan berikan hasil output!

```
print("%s mempunyai NIM %i" % (aName, myNim))
```

Tampilan Output:

```
Nama: Akhmad Qasim
Nim: 2211102441237
Akhmad Qasim 2211102441237
Akhmad Qasim-2211102441237.Traceback (most recent call last):
  File "C:\Users\PC-LAB-2BA3\Documents\Mata Kuliah\Algoritma dan Struktur Data\Laporan 02\Source Code\2.11.py", line 6, in <module>
    print("%s mempunyai NIM %i" % (aName, myNim))
TypeError: %i format: a real number is required, not str

Process finished with exit code 1
```

c) Jelaskan mengapa baris ke-6 terdapat error!

Terjadi error dikarenakan tipe data yang dimasukkan ke dalam format string merupakan sebuah string, bukan sebuah integer.

- d) Ubahlah kode program agar dapat menggunakan output yang terdapat di soal b!

```

1  aName = input("Nama: ") # Input nama
2  myNim = int(input("Nim: ")) # Input nim
3
4  print(aName, myNim) # Menampilkan nama dan nim
5  print(aName, myNim, sep="-", end=".")
6  print("%s mempunyai NIM %i" % (aName, myNim))

```

Kode diubah dengan menambahkan int() pada baris ke-2 bagian input untuk merubah tipe data string yang dimasukkan menjadi integer agar tidak terjadi error.

12. Input & Output 4

- a) Berikan tampilan output dan keterangan penjelasan!

Syntax:

```

1  price = 5800 # Variable price bertipe integer dengan nilai 5800
2  item = "pisang" # Variable item bertipe string dengan nilai "pisang"
3  print("Harga sebuah %s adalah %d rupiah" % (item, price)) # Menampilkan hasil dengan format string
4  print("Harga sebuah %+10s adalah %5.2f rupiah" % (item, price)) # Menampilkan hasil dengan format string dan padding
5  print("Harga sebuah %+10s adalah %10.2f rupiah" % (item, price)) # Menampilkan hasil dengan format string dan padding
6  itemdict = {"item": "pisang", "harga": 6800} # Variable itemdict bertipe dictionary
7  print("Harga sebuah %(item)s adalah %(cost)7.1f rupiah" % itemdict) # Menampilkan hasil format string dan padding
8
9  # %+10s adalah padding untuk menambahkan spasi sebanyak 10 karakter
10 # %5.2f adalah format float dengan 5 karakter total dan 2 karakter dibelakang koma
11

```

Tampilan Output:

```

Harga sebuah pisang adalah 5800 rupiah
Harga sebuah      pisang adalah 5800.00 rupiah
Harga sebuah      pisang adalah      5800.00 rupiah
Traceback (most recent call last):
  File "C:\Users\PC-LAB-2BA3\Documents\Mata Kuliah\Algoritma dan Struktur Data\Laporan 02\Source Code\2.12.py", line 7, in <module>
    print("Harga sebuah %(item)s adalah %(cost)7.1f rupiah" % itemdict)
  KeyError: 'cost'
Process finished with exit code 1

```

- b) Jelaskan mengapa terjadi error!

Karena key pada string format tidak menemukan key yang ada di dictionary itemdict, hanya terdapat key item dan key harga pada dictionary.

c) Ubahlan kode program tidak terjadi error!

```

1 price = 5800 # Variable price bertipe integer dengan nilai 5800
2 item = "pisang" # Variable item bertipe string dengan nilai "pisang"
3 print("Harga sebuah %s adalah %d rupiah" % (item, price)) # Menampilkan hasil dengan format string
4 print("Harga sebuah %+10s adalah %5.2f rupiah" % (item, price)) # Menampilkan hasil dengan format string dan padding
5 print("Harga sebuah %+10s adalah %10.2f rupiah" % (item, price)) # Menampilkan hasil dengan format string dan padding
6 itemdict = {"item": "pisang", "harga": 6800} # Variable itemdict bertipe dictionary
7 print("Harga sebuah %(item)s adalah %(harga)7.1f rupiah" % itemdict) # Menampilkan hasil format string dan padding
8
9 # %+10s adalah padding untuk menambahkan spasi sebanyak 10 karakter
10 # %5.2f adalah format float dengan 5 karakter total dan 2 karakter dibelakang koma

```

Agar tidak terjadi error, maka key cost yang terdapat pada string template harus diubah menjadi key harga agar dapat menampilkan harga.

Kesimpulan:

Tipe data built-in pada Python adalah tipe data yang sudah tersedia di dalam bahasa pemrograman Python tanpa perlu melakukan instalasi atau pustaka tambahan. Beberapa contoh tipe data built-in pada Python adalah sebagai berikut:

1. Integer (int) - tipe data bilangan bulat seperti 1, 2, 3, dst.
2. Float - tipe data bilangan pecahan seperti 3.14, 1.5, dst.
3. String (str) - tipe data untuk teks atau karakter, seperti "Halo", "Belajar Python", dst.
4. Boolean (bool) - tipe data yang hanya memiliki nilai True atau False.
5. List - tipe data untuk menyimpan kumpulan data yang berurutan, seperti [1, 2, 3, 4], ["apel", "jeruk", "pisang"], dst.
6. Tuple - mirip dengan list, tetapi bersifat tidak dapat diubah setelah dibuat.
7. Dictionary - tipe data yang berisi pasangan key-value, seperti {'nama': Fauzan, 'umur': 30}, {'apel': 10, 'jeruk': 5}.

Input pada Python bisa dilakukan menggunakan fungsi input(). Fungsi ini memungkinkan pengguna untuk memberikan input dari keyboard dan menyimpannya ke dalam variabel. Contoh:

```

nama = input("Masukkan nama Anda: ")
print("Halo, " + nama + "!")

```

Output pada Python bisa dilakukan menggunakan fungsi print(). Fungsi ini digunakan untuk menampilkan output ke layar. Contoh:

```

print("Halo, Fauzan!")

```

Output yang dihasilkan adalah "Halo, Fauzan!" yang ditampilkan di layar. Fungsi print() juga dapat digunakan untuk menampilkan nilai variabel, seperti:

```

nama = "Fauzan"
print("Halo, " + nama + "!")

```

Output yang dihasilkan adalah "Halo, Fauzan!" yang ditampilkan di layar.