

**MODUL MATA KULIAH**

# **BAHASA PEMROGRAMAN DASAR**

**PG168 – 3 SKS**



**FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BUDI LUHUR**

**JAKARTA  
SEPTEMBER 2021**

**TIM Penyusun :**

Agus Umar Hamdani, M.Kom  
Tri Ika Jaya Kusumawati, M.Kom



## MODUL PERKULIAHAN #2

### STANDAR TIPE DATA DALAM BAHASA PYTHON

|                      |   |   |
|----------------------|---|---|
| Capaian Pembelajaran | : | <b>Mahasiswa Mampu:</b> <ol style="list-style-type: none"><li>1. Mengerti dan memahami tentang tipe data standar yang digunakan dalam bahasa pemrograman Python.</li><li>2. Mengerti dan memahami penggunaan jenis-jenis tipe data dalam bahasa pemrograman Python.</li><li>3. Mengerti dan memahami komponen dasar dalam pemrograman Python.</li></ol>   |
| Sub Pokok Bahasan    | : | <ol style="list-style-type: none"><li>1. String</li><li>2. Numeric (Integer, Float, Complex Number)</li><li>3. Boolean</li><li>4. String</li><li>5. List</li><li>6. Set</li><li>7. Tuple</li><li>8. Dictionary</li></ol>  |
| Daftar Pustaka       | : | <ol style="list-style-type: none"><li>1. Zarman, Wendi dan Wicaksono, Mochamad Fajar. <i>"Implementasi Algoritma dalam bahasa Python"</i>. Edisi Pertama. Bandung : Penerbit Informatika. 2020.</li><li>2. Kurniawati, Arik. <i>"Algoritma dan Pemrograman menggunakan Python"</i>. Edisi Pertama. Yogyakarta : Depublish. 2016.</li><li>3. Ngodingdata.com. Python Pandas: Manipulasi Date dan Time (<a href="http://ngodingdata.com/python-pandas-manipulasi-date_time/">http://ngodingdata.com/python-pandas-manipulasi-date_time/</a> diakses pada 08 Oktober 2021, pukul 21.55 WIB)</li><li>4. Belajarpython. Tanggal dan Waktu (<a href="http://belajarpython.com/tutorial/tanggal-waktu-python">http://belajarpython.com/tutorial/tanggal-waktu-python</a> diakses pada 08 Oktober 2021, pukul : 22:00 WIB)</li><li>5. W3schools.com Python Datetime. (<a href="http://w3schools.com/python/python_datetime.asp">http://w3schools.com/python/python_datetime.asp</a> diakses pada 08 Oktober 2021 pukul 22:05 WIB)</li><li>6. Pythonindo.com. String. (<a href="http://pythonindo.com/string">http://pythonindo.com/string</a> diakses pada 08 Oktober 2021 pukul : 22:05 WIB)</li><li>7. Jagongoding.com. Python Dasar: Tipe Data List (<a href="http://jagongoding.com/python/dasar/list">http://jagongoding.com/python/dasar/list</a> diakses pada 08 Oktober 2021 pukul:22.10 WIB)</li></ol> |

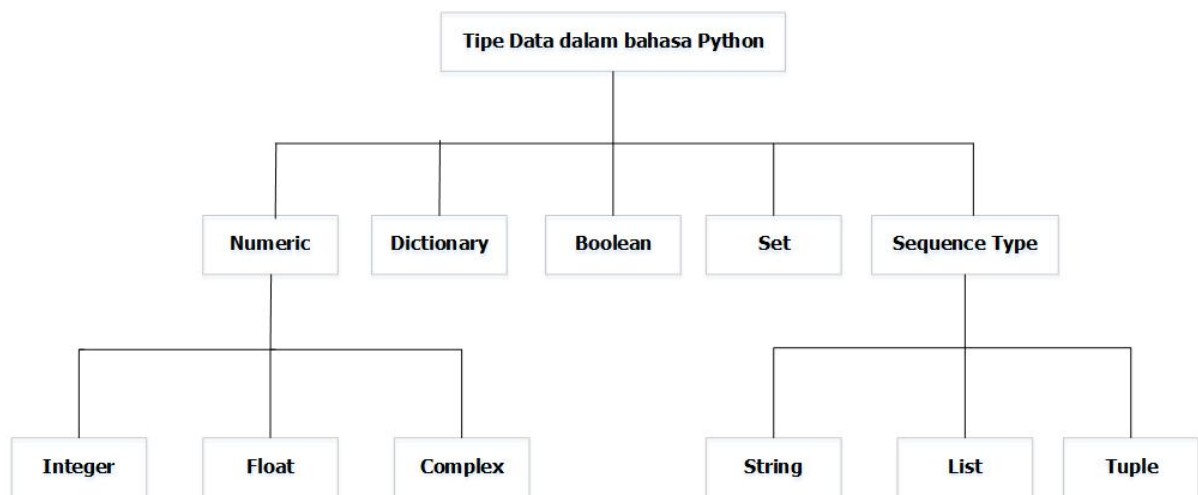
## PRAKTIKUM 2

### STANDAR TIPE DATA DALAM BAHASA PYTHON

#### 1.1. Teori

##### Tipe Data (Data Types)

Tipe Data atau biasanya disebut dengan Tipe saja adalah sebuah pengelompokan data untuk memberitahu Compiler atau Interpreter tentang bagaimana programmer ingin mengolah data [Shaffer, C.A. 2011]. Data Structures and Algorithm Analysis in C++. Edisi 3. New York : Dover]. Standar tipe data yang ada di dalam bahasa Python dapat dilihat pada gambar 2.1.



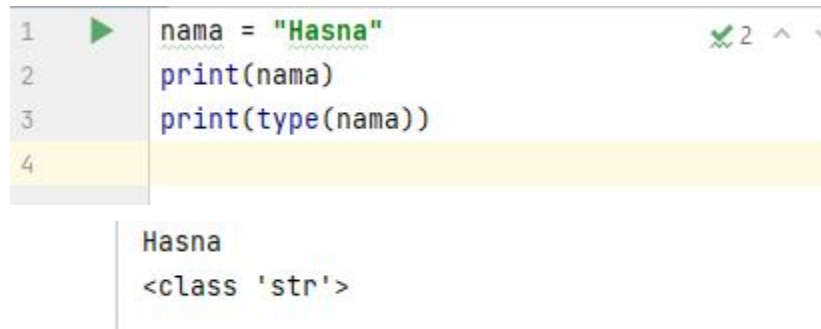
Gambar 2.1 Tipe Data dalam Bahasa Python

Python memiliki tipe data standar atau tipe data yang paling banyak digunakan dalam pemrograman [Zarman dan Wicaksono, 2020], yaitu :

##### a. String

Menyatakan karakter / kalimat, bisa berupa huruf angka, dll (diapit dengan tanda " " atau ' '). String adalah tipe data yang anggotanya berurutan dan memiliki index. Index dimulai dari angka 0, bila dimulai dari depan dan index bernilai -1, bila diindex dari belakang. Tiap karakter bisa diakses menggunakan index-nya dengan perintah : **namastring[index]**. Pada string juga bisa dilakukan Slicing atau mengakses sekelompok substring dengan perintah : **namastring[awal:akhir]**.

Contoh 1 Tipe String :

A screenshot of a Python IDE. The code editor shows four lines: 1. `nama = "Hasna"`, 2. `print(nama)`, 3. `print(type(nama))`, and 4. (empty). The output console shows the results: "Hasna" and "<class 'str'>".

```
1 nama = "Hasna"
2 print(nama)
3 print(type(nama))
4
```

Hasna  
<class 'str'>

Gambar 2.2 Contoh Penggunaan Tipe String

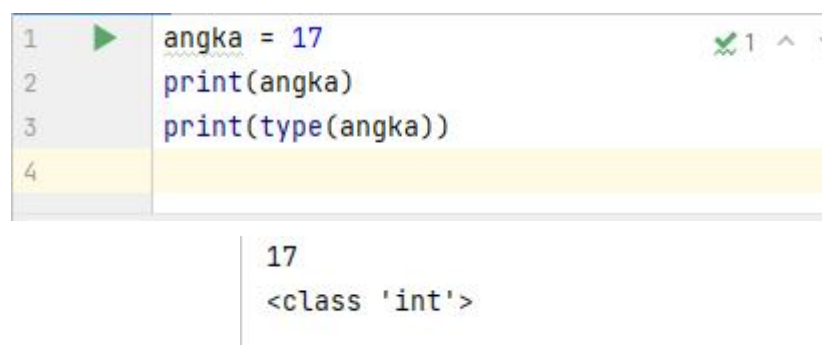
## b. Bilangan (*Numeric*)

Menyatakan data berupa angka atau Numeric. Tipe bilangan ini terdiri dari : *Integer*, *Float* dan *Complex Number*.

### 1. Tipe Data Integer

Tipe data Integer digunakan untuk menyatakan bilangan bulat. Dalam bahasa Python, tipe data Integer memiliki panjangnya bisa berapa saja dan panjangnya hanya dibatasi oleh besar memori yang tersedia [Zarman dan Wicaksono, 2020].

Contoh 1 Tipe Data Integer :

A screenshot of a Python IDE. The code editor shows four lines: 1. `angka = 17`, 2. `print(angka)`, 3. `print(type(angka))`, and 4. (empty). The output console shows the results: 17 and "<class 'int'>".

```
1 angka = 17
2 print(angka)
3 print(type(angka))
4
```

17  
<class 'int'>

Gambar 2.3 Contoh Penggunaan Tipe Integer

### 2. Tipe Data Float

Tipe data Float digunakan untuk menyatakan bilangan yang mempunyai koma (bilangan pecahan, desimal). Tipe data Float memiliki panjang akurat sampai 17 angka di belakang koma [Zarman dan Wicaksono, 2020].

Contoh 1 Tipe Data Float :

```
1  ▶ angka = 17.355
2    print(angka)
3    print(type(angka))
4
```

```
17.355
<class 'float'>
```

Gambar 2.4 Contoh Penggunaan Tipe Float

### 3. Tipe Data Complex Number

Tipe data Complex digunakan untuk menyatakan pasangan angka real dan imajiner [Zarman dan Wicaksono, 2020].

Contoh 1 Tipe Data Complex Number :

```
1  ▶ a = 5j
2    b = 10j
3    c = a + b
4    print(a, "+", b, "=", c)
5    print(type(c))
6
```

```
5j + 10j = 15j
<class 'complex'>
```

Gambar 2.5 Contoh Penggunaan Tipe Complex Number

#### c. Boolean

Tipe data ini memberikan nilai **TRUE**, apabila pernyataan yang diuji adalah benar dan memberikan nilai **FALSE**, apabila pernyataan yang diuji adalah salah [Zarman dan Wicaksono, 2020].

Contoh Tipe Data Boolean :

```
1 Saya_manusia = True
2 Saya_robot = False
3
4 print("Tipe dari Saya_manusia", type(Saya_manusia))
5 print("Tipe dari Saya_robot", type(Saya_robot))
6
```

```
Tipe dari Saya_manusia <class 'bool'>
Tipe dari Saya_robot <class 'bool'>
```

Gambar 2.6 Contoh Penggunaan Tipe Boolean

#### d. Date

Tipe data Date dalam bahasa Python bukanlah tipe datanya sendiri, tetapi kita perlu mengimpor modul bernama **datetime** untuk bekerja dengan tipe data Date sebagai objek tanggal [w3shools.com, 2021]. Untuk membuat tanggal, kita dapat menggunakan kelas **`datetime()`** dari modul **`datetime`**. **Kelas `datetime()`** membutuhkan tiga parameter untuk membuat tanggal: tahun, bulan, hari.

Contoh 1 Tipe Data Date :

```
1 from datetime import datetime
2
3 sekarang = datetime.now()
4 print('Sekarang : ', sekarang)
5 print(type(sekarang))
6 print(sekarang.day)
7 print(sekarang.month)
8 print(sekarang.year)
9 print(sekarang.hour)
10 print(sekarang.minute)
```

```
Sekarang : 2021-10-09 10:44:02.571987
<class 'datetime.datetime'>
9
10
2021
10
44
```

Gambar 2.7 Contoh Penggunaan Tipe Date

Objek datetime memiliki metode untuk memformat objek tanggal menjadi string yang dapat dibaca. Metode ini disebut **strftime()**, dan mengambil satu parameter, format, untuk menentukan format string yang dikembalikan.

Berikut ini adalah referensi semua kode format resmi yang ada di bahasa Python [ngodingdata.com, 2021]:

Tabel 2.2 Kode Format untuk Tipe Date

| Format | Deskripsi   | Contoh                      |
|--------|---|-----------------------------|
| %a     | Menyatakan nama hari (short version).               | Wed                         |
| %A     | Menyatakan nama hari (full version).                | Wednesday                   |
| %w     | Menyatakan nomor urut hari, 0 adalah Sunday         | 3 = Tuesday                 |
| %d     | Menyatakan urutan tanggal dalam satu bulan (0 – 31) | 31                          |
| %b     | Menyatakan nama bulan (short version)               | Dec                         |
| %B     | Menyatakan nama bulan (full version)                | December                    |
| %m     | Menyatakan nomor urut bulan (1-12)                  | 12                          |
| %y     | Menyatakan tahun (pendek)                           | 18                          |
| %Y     | Menyatakan tahun (lengkap)                          | 2018                        |
| %H     | Menyatakan jam (00 – 23)                            | 17                          |
| %I     | Menyatakan jam (00 – 12)                            | 05                          |
| %M     | Menyatakan menit (00 – 59)                          | 41                          |
| %S     | Menyatakan detik (00 – 59)                          | 08                          |
| %c     | Menyatakan format tanggal dan waktu versi local     | Mon Dec 31<br>17:41:00 2018 |
| %x     | Menyatakan tanggal versi local                      | 12/31/18                    |
| %X     | Menyatakan waktu dalam versi local.                 | 17:41:00                    |

#### e. List

Tipe data ini berisi item data yang berurutan. Seperti halnya tipe data string, setiap item (anggota) list memiliki index sesuai dengan urutannya. Index dimulai dari 0 (bukan dari 1). List bisa berisi anggota dan tipe yang sama atau berbeda. List



bersifat *Mutable*, dimana anggotanya bisa diubah. **Untuk mendeklarasikan tipe data List, maka gunakanlah tanda kurung [ ] dan masing-masing anggotanya dipisahkan oleh tanda koma** [Zarman dan Wicaksono, 2020].

Sintak penulisan : **data = [nilai1, nilai2]**

Untuk mengakses item yang ada di List, caranya adalah dengan memanggil nama list diikuti index dari item yang bersangkutan, yaitu dengan perintah **namalist[index]**. Selain itu, bisa dilakukan pengaksesan terhadap sejumlah item dari index ke index. Index dimulai dari 0 untuk anggota yang pertama, Sedangkan selanjutnya diikuti berdasarkan banyaknya anggota. Selain itu, bisa juga dengan menggunakan nilai min(-). Contoh index ke -1, maka index tersebut akan mengakses nilai paling akhir.

Contoh 1 Tipe Data List :



```
1 list1 = ['fisika', 'kimia', 1993, 2017]
2
3 print(list1)
4 print(type(list1))
5 print(list1[2])
6 print(list1[:2])
7 print(list1[2:])
8 print(list1[1:3])
9
```

---

```
['fisika', 'kimia', 1993, 2017]
<class 'list'>
1993
['fisika', 'kimia']
[1993, 2017]
['kimia', 1993]
```

Gambar 2.9 Contoh Penggunaan Tipe List

#### f. Tuple

Tipe data ini mirip dengan List. Perbedaanya adalah anggotanya tidak bisa diubah (*immutable*). Sekali tuple dibuat, maka isinya tidak dapat dimodifikasi lagi. **Untuk mengenali tipe data Tuple ini ditandai dengan tanda kurun ( ) dan dan**



masing-masing anggotanya dipisahkan oleh tanda koma [Zarman dan Wicaksono, 2020].

Sintak penulisan : **data = (nilai1, nilai2)**

### Cara Mengakses Anggota Tuple :

Tuple berbentuk menyerupai tipe List, maka kita bisa mengakses anggota tuple menggunakan index. Sintak penulisannya adalah : **nama\_tuple[index-ke]**.

Index dimulai dari 0 untuk anggota yang pertama, Sedangkan selanjutnya diikuti berdasarkan banyaknya anggota. Selain itu, bisa juga dengan menggunakan nilai min(-).

Contoh Tipe Data Tuple :



```
1 tuple1 = ('fisika', 'kimia', 1993, 2017)
2
3 print(tuple1)
4 print(type(tuple1))
5 print(tuple1[2])
6 print(tuple1[:2])
7 print(tuple1[2:])
8 print(tuple1[1:3])
9
```

```
('fisika', 'kimia', 1993, 2017)
<class 'tuple'>
1993
('fisika', 'kimia')
(1993, 2017)
('kimia', 1993)
```

Gambar 2.10 Contoh Penggunaan Tipe Tuple

### g. Set

Set dalam bahasa pemrograman python adalah tipe data kolektif yang digunakan untuk menyimpan banyak nilai dalam satu variabel dengan ketentuan:

- nilai anggota yang disimpan harus unik (tidak duplikat)
- nilai anggota yang sudah dimasukkan tidak bisa diubah lagi
- set bersifat *unordered* atau tidak berurut, artinya anggotanya tidak bisa diakses dengan index.

Set bisa digunakan untuk melakukan operasi himpunan matematika, seperti : gabungan, irisan, selisih dan komplemen [Zarman dan Wicaksono, 2020].

### Cara Mengakses Tipe Set :

Secara umum kita bisa membuat set dengan 2 cara: dengan kurung kurawal { }, atau dengan sebuah list yang kita passing ke dalam fungsi : **set([nilai])**.

Contoh Tipe Data Set :



```
1 set1 = {'fisika', 'kimia', 1993, 2017}
2
3 print(set1)
4 print(type(set1))
5
6
```

```
{1993, 'fisika', 2017, 'kimia'}
<class 'set'>
```

Gambar 2.11 Contoh Penggunaan Tipe Set

### h. Dictionary

Tipe data ini memiliki anggota yang terdiri dari pasangan kunci-nilai (key-value). Mirip seperti kamus, dimana terdapat kata dan ada arti. Dictionary umumnya dipakai untuk data yang besar dan untuk mengakses anggota data secara acak. Anggota Dictionary tidak memiliki indeks. Dictionary dideklarasikan menggunakan tanda kurung kurawal { }, dimana anggotanya memiliki bentuk kunci : nilai atau key-value, dan setiap anggota dipisahkan dengan tanda koma. Selain itu, bisa juga diakses dengan menggunakan fungsi **get()**. Dengan menggunakan fungsi **get()**, bila kunci tidak ada di dalam dictionary, maka akan dikembalikan None. Bila tidak menggunakan fungsi **get()**, maka akan terjadi error **KeyError** bila kunci yang hendak diakses tidak ada di dalam dictionary [Zarman dan Wicaksono, 2020].

Sintak penulisan tipe data Dictionary adalah

**nama\_variabel = {"Key1": "value", "Key2": "value"}**

Contoh Tipe Data Dictionary :

```

1  ▶
2  #tipe data Dictionary
3  data_mhs = {'Name1': 'Zara', 'Age1': 17, 'Class1': '1A',
4             'Name2': 'Joko', 'Age2': 19, 'Class2': 'B1',
5             'Name3': 'Wawan', 'Age3': 21, 'Class3': 'C3'}
6  print ("data_mhs['Name2']: ", data_mhs['Name2'])
7  print ("data_mhs['Age2']: ", data_mhs['Age2'])
8
data_mhs['Name2']:  Joko
data_mhs['Age2']:  19

```

Gambar 2.12 Contoh Penggunaan Tipe Dictionary

### i. Fungsi Manipulasi String

Tipe data String pada bahasa pemrograman Python dapat dimanipulasi sebagaimana pada bahasa pemrograman lainnya [Zarman dan Wicaksono, 2020]. Berikut ini adalah beberapa fungsi yang digunakan untuk manipulasi tipe String yang sering digunakan dapat dilihat pada tabel 2.3.

Tabel 2.3 Fungsi Manipulasi String

| Nama Fungsi  | Deskripsi  | Contoh Penggunaan  |
|--------------|--|--|
| Capitalize() | Fungsi ini digunakan untuk mengembalikan string, dimana huruf awal dari string akan berubah menjadi huruf kapital. Sintak penulisan : tuliskan string atau variabel dengan menambahkan fungsi <b>".capitalize"</b> . | Nama = "wahid hasyim"<br>Print( <b>Nama.capitalize()</b> )<br><u>Output</u> : Wahid Hasyim |
| Lower()      | Fungsi ini digunakan untuk mengubah semua string menjadi huruf kecil.  | Nama = "WAHID HASYIM"<br>print( <b>Nama.lower()</b> )<br><u>Output</u> : 'wahid hasyim     |

|           |   |  |
|-----------|---|--|
|           | Sintak penulisan : tuliskan string atau variabel diikuti dengan fungsi <b>".lower"</b> .  |  |
| Upper()   | Fungsi ini digunakan untuk mengubah semua string menjadi huruf besar.<br>Sintak penulisan : tuliskan string atau variabel diikuti dengan fungsi <b>".upper"</b> .   | Nama = "wahid hasyim"<br>print( <b>Nama.upper()</b> )<br><u>Output</u> : WAHID HASYIM                                  |
| Find()    | Fungsi ini digunakan untuk mencari posisi huruf yang dicari dan mengembalikan nilai index posisi huruf yang dicari.<br>Sintak penulisan : tuliskan string atau variabel diikuti fungsi <b>".find"</b> serta menuliskan huruf yang dicari. | Nama = "wahid hasyim"<br>Print( <b>Nama.find('d')</b> )<br><u>Output</u> : '4  |
| replace() | Fungsi ini digunakan untuk mengganti suatu string tertentu.<br>Sintak penulisan :<br><b>variabel.replace("huruf yang akan diganti", "huruf pengganti")</b>  | kata = "selamat"<br>print( <b>kata.replace("t", "nya")</b> )<br><u>Output</u> : selamanya                              |
| Strip()   | Fungsi ini digunakan untuk menghilangkan spasi pada kata ataupun kalimat dan spasi selanjutnya yang berada di dalam kalimat akan tetap ada.<br>Sintak penulisan :<br><b>variabel.strip()</b>  | sapaKalimat = "Teknik Komputer"<br>print(sapaKalimat)<br><b>sapaKalimat.strip()</b><br><u>Output</u> : Teknik Komputer |

|         |  |  |
|---------|--|--|
| Split() | <p>Fungsi ini digunakan untuk memisahkan kumpulan string sesuai dengan pembatas yang diujikan.</p> <p>Sintak penulisan :</p> <p><b>variabel.split(pembatas, max_split)</b></p>   | <pre>varku = "var1 var2 var3" varku.split(" ") a, b, c = <b>varku.split(" ")</b> print("a = ", a) print("b = ", b) print("c = ", c)</pre> <p>Output :</p> <pre>a = var1 b = var2 c = var3</pre>  |
| In()    | <p>Fungsi ini digunakan untuk mencari karakter string yang terdapat pada kumpulan string yang diberikan. Jika string yang dicari ada, maka akan dikembalikan nilai TRUE dan jika string yang dicari tidak ada, maka akan dikembalikan nilai FALSE.</p> <p>Sintak penulisan :</p> <p><b>"huruf_yang_dicari" in variabel</b></p> | <pre>Kalimat = "Algoritma, Pemrograman, Python" Print("p" in Kalimat)</pre> <p><u>Output</u> : False</p> <pre>Kalimat = "Algoritma, Pemrograman, Python" Print("P" in kalimat)</pre> <p><u>Output</u> : True</p> <pre>Kalimat = "Algoritma, Pemrograman, Python" Print("Teknik" in kalimat)</pre> <p><u>Output</u> : False</p> |

## j. Penggabungan (*Concatenation*)

### Operator + pada tipe String

Operator + digunakan untuk menggabungkan 2 atau lebih string, dimana proses penggabungan string ini berdasarkan urutan string yang akan ditambahkan [Pythonindo.com, 2021].

Contoh Penggunaan Operator +:

```
1 ▶ s1="Welcome"
2   s2="to"
3   s3="python"
4   s4=s1+" " + s2+" "+ s3
5   print (s4)
6
```

Welcome to python

Gambar 2.13 Contoh Penggunaan Operator + pada tipe String

Penggabungan tipe data yang berbeda akan menghasilkan kesalahan, sehingga salah satu dari tipe data yang akan digabungkan tersebut diubah ke tipe lainnya.

### Operator + pada tipe Tuple

Operator + digunakan untuk menggabungkan 2 atau lebih pada Tuple, dimana proses penggabungan string ini berdasarkan urutan string yang akan ditambahkan [Jagongoding.com, 2021].

Contoh Penggunaan Operator +:

```
1 ▶ t1=(1,2)
2   t2=(3,4)
3   print ("t1 + t2 = ", t1+t2)
4
```

t1 + t2 = (1, 2, 3, 4)

Gambar 2.14 Contoh Penggunaan Operator + pada tipe Tuple

### Operator + pada tipe List

Operator + digunakan untuk menggabungkan 2 atau lebih pada List, dimana proses penggabungan string ini berdasarkan urutan string yang akan ditambahkan [jagongoding.com, 2021].

Contoh Penggunaan Operator + :

```
1 ▶ l1=[1,2]
2   l2=[3,4]
3   print ("l1 + l2 = ", l1+l2)
4
```

```
l1 + l2 = [1, 2, 3, 4]
```

Gambar 2.15 Contoh Penggunaan Operator + pada tipe List

## Operator |

Operator | digunakan untuk menggabungkan 2 atau lebih bilangan (Integer saja yang didukung) dengan hasil berupa penjumlahan atas kedua bilangan tersebut

Contoh Penggunaan Operator | :

```
1 x = 5
2 y = 20
3 print(" x | y = ", x|y)
4
```

x | y = 21

Gambar 2.16 Contoh Penggunaan Operator | pada tipe Integer

## 1.2. Praktikum

1. Tuliskan kode program 2.1 berikut ini pada editor Pycharm Edu 2021.2 atau editor Python lainnya seperti terlihat pada gambar 2.17.

```
1 kalimat = 'Nama saya Umar'
2
3
4 print(kalimat)      #mencetak string lengkap
5 print(kalimat[0])   #mencetak karakter pertama
6 print(kalimat[-1])  #mencetak karakter terakhir
7 print(kalimat[5:9]) #mencetak karakter dari index ke-5 sampai ke-9
8 print(kalimat[:4])  #mencetak karakter dari index 0 sampai ke-3
9
```

Gambar 2.17 Praktikum Penggunaan Tipe String

Kemudian jalankan program tersebut dengan menggunakan perintah **RUN** dan buatlah analisis terhadap kode program tersebut dan sebutkan keluaran (output) yang dihasilkan!



2. Tuliskan kode program 2.2 berikut ini pada editor Pycharm Edu 2021.2 atau editor Python lainnya seperti terlihat pada gambar 2.18.

```
1  ▶  bil1 = 5.15
2      bil2 = 25
3      print('nilai bil1 adalah ', bil1)
4      print(type(bil1))
5      print('nilai bil2 adalah ', bil2)
6      print(type(bil2))
7      print('nilai bil1 + bil 2 adalah ', bil1 + bil2)
8      print(type(bil1+bil2))
9
```

Gambar 2.18 Praktikum Penggunaan Tipe Integer dan Float

Kemudian jalankan program tersebut dengan menggunakan perintah **RUN** dan buatlah analisis terhadap kode program tersebut dan sebutkan keluaran (output) yang dihasilkan!

3. Tuliskan kode program 2.3 berikut ini pada editor Pycharm Edu 2021.2 atau editor Python lainnya seperti terlihat pada gambar 2.19.

```
1  ▶  a1 = 5
2      b1 = 10j
3      b2 = 35j
4
5      c1 = a1 + b1
6      c2 = b1 + b2
7      print(a1, '+', b1, '=', c1)
8      print(b1, '+', b2, '=', c1)
9
```

Gambar 2.19 Praktikum Penggunaan Tipe Integer dan Complex Number

Kemudian jalankan program tersebut dengan menggunakan perintah **RUN** dan buatlah analisis terhadap kode program tersebut dan sebutkan keluaran (output) yang dihasilkan!

4. Tuliskan kode program 2.5 berikut ini pada editor Pycharm Edu 2021.2 atau editor Python lainnya seperti terlihat pada gambar 2.20.

```

1  import datetime
2  from datetime import timedelta
3
4  t1 = datetime.date(2018, 6, 1)
5  print(t1)
6  print('t1 = ', t1.strftime("%A, %d-%B-%Y"))
7
8  t2 = t1 + timedelta(days=5)
9  print('\n5 hari setelah tanggal : ', t1, ' adalah : ', t2)
10 print(t2.strftime("t2 = %A, %d-%B-%Y"))
11

```

Gambar 2.20 Praktikum Penggunaan Tipe Date

Kemudian jalankan program tersebut dengan menggunakan perintah **RUN** dan buatlah analisis terhadap kode program tersebut dan sebutkan keluaran (output) yang dihasilkan!

5. Tuliskan kode program 2.6 berikut ini pada editor Pycharm Edu 2021.2 atau editor Python lainnya seperti terlihat pada gambar 2.21.

```

1  nilai = 5 > 10
2  nilai2 = 10 > 5
3  huruf = "Hallo" == "hallo"
4
5  print(nilai)
6  print(nilai2)
7  print(huruf)
8

```

Gambar 2.21 Praktikum Penggunaan Tipe Boolean

Kemudian jalankan program tersebut dengan menggunakan perintah **RUN** dan buatlah analisis terhadap kode program tersebut dan sebutkan keluaran (output) yang dihasilkan!

6. Tuliskan kode program 2.7 berikut ini pada editor Pycharm Edu 2021.2 atau editor lainnya seperti terlihat pada gambar 2.22.

```
1 ▶ a = [5,10,15,20,25,30,35,40]
2
3 #mencetak item dari index ke-2
4 print("a[2] = ", a[2])
5
6 #mencetak item dari index ke-0 sampai ke-2
7 print("a[0:3] = ", a[0:3])
8
9 #mencetak item mulai dari index ke-5 sampai akhir
10 print("a[5:] = ", a[5:])
11
12 #mencetak item mulai dari index ke-0 sampai index ke-4
13 print("a[:5] = ", a[:5])
14
15 print(len((a)))
16
```

Gambar 2.22 Praktikum Penggunaan Tipe List

Kemudian jalankan program tersebut dengan menggunakan perintah **RUN** dan buatlah analisis terhadap kode program tersebut dan sebutkan keluaran (output) yang dihasilkan!

7. Tuliskan kode program 2.8 berikut ini pada editor Pycharm Edu 2021.2 atau editor lainnya seperti terlihat pada gambar 2.23.

```

1  ▶ data = ('a','b','c','d',1,2,3,4)
2  #mengakses data paling pertama
3  print ("data[0] = ", data[0])
4
5  #mengakses nilai dari index ke-3 sampai index terakhir
6  print ("data[3:] = ", data[3:])
7
8  #mengakses nilai dari index ke-0 sampai index ke-2
9  print ("data[:3] = ", data[:3])
10
11 #mengakses nilai dari index ke-2 sampai index ke-6
12 print ("data[2:6] = ", data[2:6])
13
14 #mengakses data paling terakhir
15 print ("data[-1] = ", data[-1])
16

```

Gambar 2.23 Praktikum Penggunaan Tipe Tuple

Kemudian jalankan program tersebut dengan menggunakan perintah **RUN** dan buatlah analisis terhadap kode program tersebut dan sebutkan keluaran (output) yang dihasilkan!

8. Tuliskan kode program 2.9 berikut ini pada editor Pycharm Edu 2021.2 atau editor lainnya seperti terlihat pada gambar 2.24.

```

1 # set dengan nilai campuran
2 data = {'a', 'b', 'c', 'd', 1, 2, 3, 4}
3 print (data)
4
5 # set tidak menampung nilai yang sama
6 x = {1, 2, 3, 4, 5, 6, 1, 2, 3, 4}
7 print (x)
8 # set dengan nilai campuran
9 data = {'a', 'b', 'c', 'd', 1, 2, 3, 4}
10 print (data)
11
12 # set tidak menampung nilai yang sama
13 x = {1, 2, 3, 4, 5, 6, 1, 2, 3, 4}
14 print (x)

```

Gambar 2.24 Praktikum Penggunaan Tipe Set

Kemudian jalankan program tersebut dengan menggunakan perintah **RUN** dan buatlah analisis terhadap kode program tersebut dan sebutkan keluaran (output) yang dihasilkan!

9. Tuliskan kode program 2.10 berikut ini pada editor Pycharm Edu 2021.2 atau editor lainnya seperti terlihat pada gambar 2.25.

```

1 #tipe data Dictionary
2 data_mhs = {'Name1': 'Zara', 'Age1': 17, 'Class1': '1A',
3             'Name2': 'Joko', 'Age2': 19, 'Class2': 'B1',
4             'Name3': 'Wawan', 'Age3': 21, 'Class3': 'C3'}
5 print ("data_mhs['Name2'] + data_mhs['Class1'] ", data_mhs['Name2'] + data_mhs['Class1'])
6 print ("data_mhs['Age2']: ", data_mhs['Age2'])
7

```

Gambar 2.25 Praktikum Penggunaan Tipe Dictionary

Kemudian jalankan program tersebut dengan menggunakan perintah **RUN** dan buatlah analisis terhadap kode program tersebut dan sebutkan keluaran (output) yang dihasilkan!

10. Tuliskan kode program 2.1 berikut ini pada editor Pycharm Edu 2021.2 atau editor lainnya seperti terlihat pada gambar 2.26.

```
1 ▶ var1 = 'Hello Python!'
2   var2 = "I love Python"
3   print("var1[0]", var1[0])
4   print("var2[2:6]:", var2[2:6])
5   print("String Update: - ", var1[:6] + 'World')
6
```

Gambar 2.26 Praktikum Penggunaan Operator Penggabungan pada Tipe Integer

Kemudian jalankan program tersebut dengan menggunakan perintah **RUN** dan buatlah analisis terhadap kode program tersebut dan sebutkan keluaran (output) yang dihasilkan!

### 1.3. Latihan

1. Tuliskan kode program Python menggunakan tipe data String seperti terlihat pada gambar 2.27. Kemudian simpan file tersebut dengan nama "Praktikum21.py".

```
1 ▶ nama = 'John Doe'
2   pesan = "John Doe belajar bahasa Python di Kelas Python"
3   print ("nama[0]: ", nama[0])
4   print ("pesan[0:8]: ", pesan[0:8])
5   print ("pesan[:30]: ", pesan[:30])
6   print ("pesan[31:]: ", pesan[31:])
7
```

Gambar 2.27. Kode Program Praktikum Tipe String

Kemudian jalankan program tersebut dengan menggunakan perintah **RUN** dan buatlah analisis terhadap kode program tersebut dan sebutkan keluaran (output) yang dihasilkan!



2. Tuliskan kode program Python menggunakan tipe data String, Date dan Numeric seperti terlihat pada gambar 2.28. Kemudian simpan file tersebut dengan nama "Praktikum22.py".

```
1 import datetime
2
3 nama_depan = 'Wahid'
4 nama_belakang = 'Hasyim'
5 #simbol + " variabel atau nilai" + digunakan untuk menggabungkan dua string
6 nama_lengkap = nama_depan + ' ' + nama_belakang
7 tgl_lahir = datetime.datetime(1990, 5, 10)
8 usia = 31
9 alamat = 'Jakarta Selatan'
10 kata_mutiara = "Don't judge a book by it's cover"
11
12 print(nama_lengkap, tgl_lahir, 'berusia ', usia, 'tahun dari', alamat, 'kata mutiara:', kata_mutiara)
13
14 print('\nTipe dari nama_lengkap:', type(nama_lengkap))
15 print('Tipe dari tanggal lahir:', type(tgl_lahir))
16 print('Tipe dari usia', type(usia))
17 print('Tipe dari alamat:', type(alamat))
18 print('Tipe dari kata_mutiara:', type(kata_mutiara))
19
```

Gambar 2.28. Kode Program Praktikum Tipe String, Numeric dan Date

Kemudian jalankan program tersebut dengan menggunakan perintah **RUN** dan buatlah analisis terhadap kode program tersebut dan sebutkan keluaran (output) yang dihasilkan!

3. Tuliskan kode program Python menggunakan tipe data List seperti terlihat pada gambar 2.29. Kemudian simpan file tersebut dengan nama "Praktikum23.py".


```
1 list_buah = ['Pisang', 'Nanas', 'Melon', 'Durian']
2 list_angka = [1, 2, 3, 4, 5, 6, 7]
3
4
5 print(list_buah[0:1])
6 print(list_angka[0:2])
7 print(list_buah[1:3])
8 print(list_angka[0:-1])
9 print(list_buah[-1:-3])
10 print(list_angka[-1:3])
11 print(list_buah[-3:-1])
12
```

Gambar 2.29. Kode Program Praktikum Tipe List



Kemudian jalankan program tersebut dengan menggunakan perintah **RUN** dan buatlah analisis terhadap kode program tersebut dan sebutkan keluaran (output) yang dihasilkan!

4. Tuliskan kode program Python menggunakan tipe data Tuple seperti terlihat pada gambar 2.30. Kemudian simpan file tersebut dengan nama "Praktikum24.py".

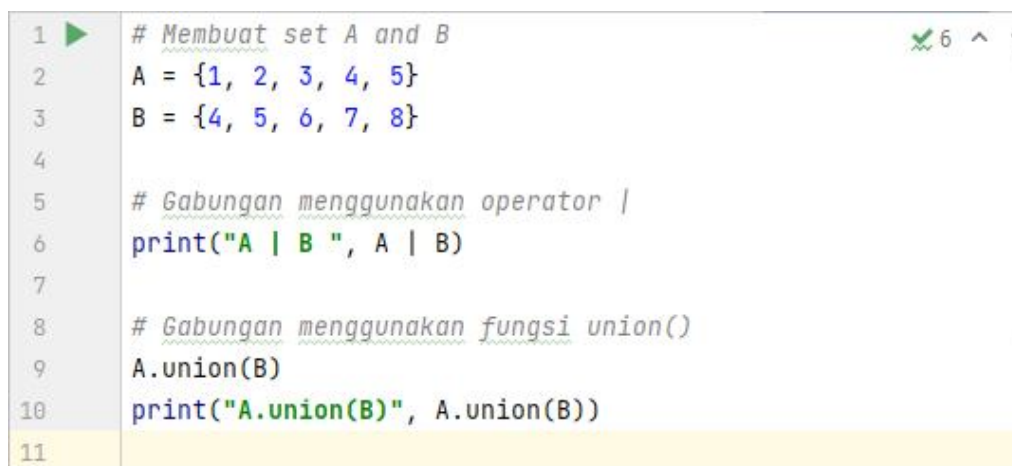
A screenshot of a Python IDE with a light blue background. The code is as follows:

```
1  ▶ 2  tup1 = ('Fisika', 'Kimia', 1993, 2018, 1, 2, 3, 4)
3
4  print(type(tup1))
5  print(tup1[0:2])
6  print(tup1[:3])
7  print(tup1[5:])
8  print(tup1[-1:3])
9  print(tup1[-3:-1])
10
```

Gambar 2.30. Kode Program Praktikum Tipe Tuple

Kemudian jalankan program tersebut dengan menggunakan perintah **RUN** dan buatlah analisis terhadap kode program tersebut dan sebutkan keluaran (output) yang dihasilkan!

5. Tuliskan kode program Python menggunakan tipe data Set seperti terlihat pada gambar 2.31. Kemudian simpan file tersebut dengan nama "Praktikum26.py".

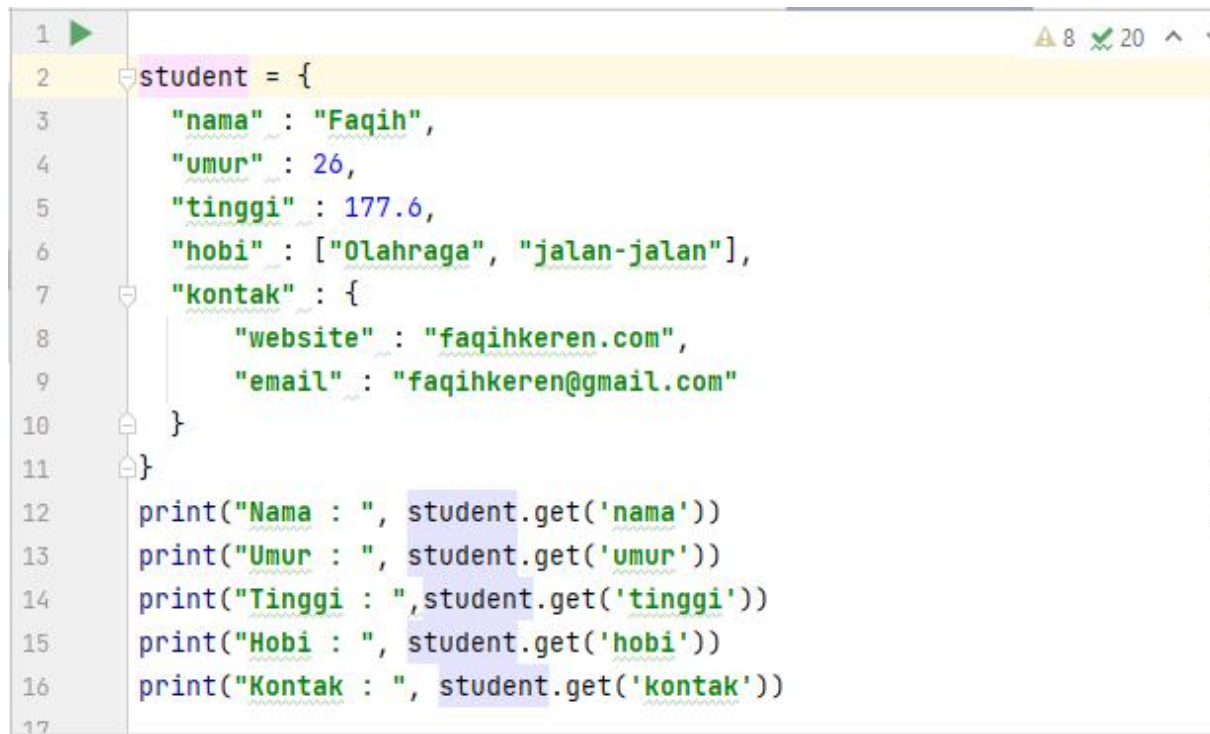
A screenshot of a Python IDE with a light blue background. The code is as follows:

```
1  ▶  # Membuat set A and B
2  A = {1, 2, 3, 4, 5}
3  B = {4, 5, 6, 7, 8}
4
5  # Gabungan menggunakan operator |
6  print("A | B ", A | B)
7
8  # Gabungan menggunakan fungsi union()
9  A.union(B)
10 print("A.union(B)", A.union(B))
11
```

Gambar 2.31. Kode Program Praktikum Tipe Set

Kemudian jalankan kode program tersebut menggunakan perintah **RUN** dan buatlah analisis dari hasil keluaran program tersebut!

6. Tuliskan kode program Python menggunakan tipe data Dictionary seperti terlihat pada gambar 2.32. Kemudian simpan file tersebut dengan nama "Praktikum26.py".

The image shows a code editor window with a Python script. The script defines a dictionary named 'student' with keys for 'nama', 'umur', 'tinggi', 'hobi', and 'kontak'. The 'kontak' key has a nested dictionary with 'website' and 'email'. Below the dictionary definition, there are five print statements that use the .get() method to retrieve and display the values for each key. The code is as follows:

```
1 student = {
2     "nama": "Faqih",
3     "umur": 26,
4     "tinggi": 177.6,
5     "hobi": ["Olahraga", "jalan-jalan"],
6     "kontak": {
7         "website": "faqihkeren.com",
8         "email": "faqihkeren@gmail.com"
9     }
10 }
11
12 print("Nama : ", student.get('nama'))
13 print("Umur : ", student.get('umur'))
14 print("Tinggi : ", student.get('tinggi'))
15 print("Hobi : ", student.get('hobi'))
16 print("Kontak : ", student.get('kontak'))
17
```

Gambar 2.32. Kode Program Praktikum Tipe Dictionary

Kemudian jalankan program tersebut dengan menggunakan perintah **RUN** dan buatlah analisis terhadap kode program tersebut dan sebutkan keluaran (output) yang dihasilkan!

#### 1.4. Rangkuman

1. Bahasa Python memiliki standar tipe data yang sering digunakan oleh programmer untuk menangani nilai, seperti : String, Integer, Float, Complex Number, Date, List, Tuple, Set dan Dictionary.
2. Tipe data String, Integer, Float, Complex Number dan Date digunakan untuk menangani nilai tunggal (*Single-Value*).

3. Tipe data List, Tuple, Set dan Dictionary digunakan untuk menangani banyak nilai tunggal (*Multi-Value*).
4. Kita dapat melakukan merubah tampilan String dengan menggunakan fungsi manipulasi String.
5. Operator + digunakan untuk menggabungkan dua string, sedangkan operatot | digunakan untuk menggabungkan dua bilangan (Integer saja).
6. Input Function digunakan agar pengguna dapat berinteraksi dengan program.

### 1.5. Tugas Mandiri

1. Jelaskan perbedaan tipe data Integer, Float dan Complex?
2. Jelaskan perbedaan antara tipe data List dan Tuple?
3. Jelaskan apa yang disebut dengan *Object Immutable* pada tipe data Set dan Tuple?
4. Jelaskan apa yang disebut dengan *Object Mutable* pada tipe data List dan Dictionary?
5. Buatlah kode program menggunakan tipe List dengan keluaran (output) seperti terlihat pada gambar 2.33.

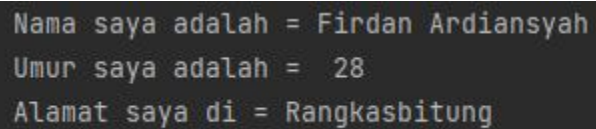
```
=====
                        PROGRAM LIST
                        nilaiTK = [90, 75, 95, 85, 80]
=====
Mencetak semua isi list :
[90, 75, 95, 85, 80]
=====
Mencetak isi list index ke-1
75
=====
Mencetak isi list mulai dari index ke-1 sampai index ke-2
[75, 95]
=====
Mengambil isi list index ke-2 ke variabel nilaiKU
nilaiKU = 95
=====
```

Gambar 2.33. Hasil Eksekusi Kode Program

6. Diketahui variabel Dictionary sebagai berikut :

- Key1 = "nama"
- Key2 = "umur"
- Key3 = "alamat"
- Value1 = "Firdan Ardiansyah"
- Value2 = 28
- Value3 = "Rangkasbitung"

Eksekusilah kode program yang anda buat dengan memanggil key name 'Firdan Ardiansyah' sehingga menghasilkan keluaran (output) seperti gambar 2.34.



```
Nama saya adalah = Firdan Ardiansyah
Umur saya adalah = 28
Alamat saya di = Rangkasbitung
```

Gambar 2.34. Hasil Eksekusi Kode Program



**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS BUDI LUHUR**

Jl. Raya Ciledug, Petukangan Utara, Pesanggrahan  
Jakarta Selatan, 12260  
Telp: 021-5853753 Fax : 021-5853752  
<http://fti.budiluhur.ac.id>