



Modul Praktikum

Algoritma dan Pemrograman

Jan Wantoro, S.T., M.Eng.

Irma Yuliana, S.T., M.M., M.Eng.

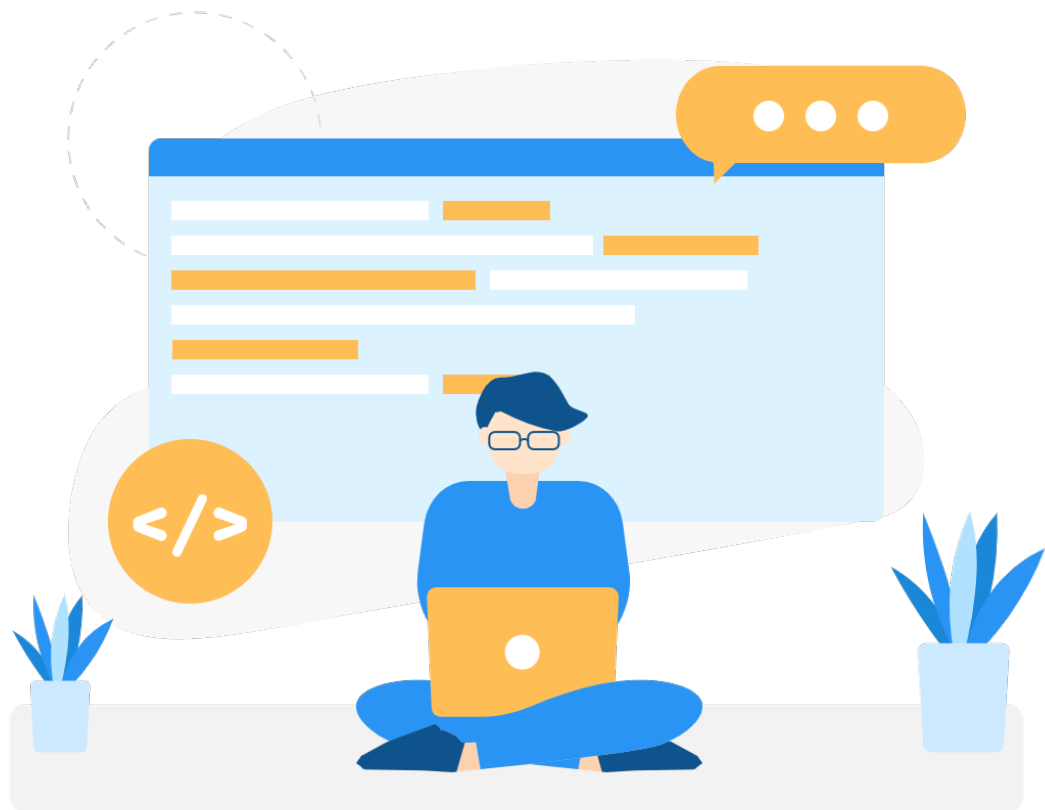
Arif Setiawan, S.Kom., M.Eng.

Program Studi Pendidikan Teknik Informatika

Fakultas Keguruan dan Ilmu Pendidikan Universitas Muhammadiyah Surakarta

BAB 6

Struktur Data pada Python



6.1 Tujuan

1. Dapat menjelaskan list, tuple, set dan dictionary pada bahasa python.
2. Dapat mengimplementasikan list, tuple, set dan dictionary pada pemrograman

6.2 Pengantar

Struktur data merupakan cara yang digunakan untuk menangani suatu data pada bahasa pemrograman. Pada bab ini kita akan belajar empat struktur data yang disediakan oleh python yaitu list, tuple, set dan dictionary.

6.2.1 List

List merupakan sekumpulan data yang terdapat dalam sebuah variabel. Penggunaan list hampir sama dengan fungsi array pada bahasa pemrograman yang lain. Setiap elemen anggota pada list diakses menggunakan indeks yang dimulai dari 0. Data yang tersimpan pada list diapit dengan tanda kurung siku [].

Berikut contoh penggunaan list.

```
1. #deklarasi list
2. pti = ['fkip', 'ums', 2014, True]
3.
4. #akses list menggunakan indeks 0
5. print(pti[0])
6.
7. #akses list menggunakan indeks 2
8. print(pti[2])
```

Baris 1 merupakan deklarasi list yang memiliki 4 element yang berbeda tipe yaitu string, integer dan boolean. Indeks elemen ini diawali dari angka 0. Sehingga ketika program tersebut dijalankan akan menghasilkan output.

```
fkip
2014
```

Indeks pada list dapat bernilai negatif. Indeks -1 akan menunjuk elemen terakhir pada list. Indeks -2 akan menunjukkan elemen kedua dari akhir. Kode dibawah ini akan mengakses elemen terakhir dari list dan elemen ketiga dari akhir.

```
1. #deklarasi list
2. pti = ['fkip', 'ums', 2014, True]
3.
```

```

4. #akses elemen terakhir menggunakan indeks -1
5. print(pti[-1])
6.
7. #akses elemen ketiga dari akhir menggunakan indeks -3
8. print(pti[-3])

```

Kita juga dapat mengakses elemen yang berada pada rentang indeks tertentu menggunakan slicing operator atau tanda titik dua (:). Perhatikan kode berikut ini untuk mengetahui penggunaan slicing operator.

```

1. #deklarasi list
2. pti = ['p','t','i','u','m','s']
3.
4. #akses elemen ketiga hingga kelima
5. print(pti[2:5])
6.
7. #akses elemen pertama hingga ke empat
8. print(pti[:4])
9.
10. #akses elemen keempat hingga ke terakhir
11. print(pti[3:])
12.
13. #akses elemen awal hingga akhir
14. print(pti[:])

```

List merupakan tipe data yang bersifat mutable. Artinya data yang berada di dalam list dapat diubah. Kita dapat menggunakan tanda assignment (=) untuk mengubah data yang berada dalam list.

```

1. #deklarasi list
2. kabupaten = ['sukoharjo', 'sragen', 'wonogiri', 'karanganyar', 'klaten']
3.
4. #ubah element pertama
5. kabupaten[0] = 'sleman'
6.

```

```

7. #cetak list
8. print(kabupaten)
9.
10. #ubah element kedua hingga ke empat
11. kabupaten[1:4] = ['kulonprogo', 'bantul', 'gunungkidul']
12.
13. #cetak list
14. print(kabupaten)

```

kode diatas akan menghasilkan output sebagai berikut

```

['sleman', 'sragen', 'wonogiri', 'karanganyar', 'klaten']
['sleman', 'kulonprogo', 'bantul', 'gunungkidul', 'klaten']

```

Untuk menambahkan data ke dalam list kita dapat menggunakan fungsi **append()**, sedangkan jika data yang ditambahkan berjumlah lebih dari satu maka dapat menggunakan fungsi **extend()**. Fungsi **insert()** digunakan jika kita ingin menambahkan data pada indeks tertentu pada list.

```

1. #deklarasi list
2. kabupaten = ['sukoharjo', 'sragen']
3.
4. #menambah 1 elemen
5. kabupaten.append('wonogiri')
6.
7. #cetak list
8. print(kabupaten)
9.
10. #menambah 2 elemen
11. kabupaten.extend(['karanganyar', 'klaten'])
12.
13. #cetak list
14. print(kabupaten)
15.
16. #insert elemen pada indeks ke 1

```

```
17. kabupaten.insert(1, 'boyolali')
18.
19. #cetak list
20. print(kabupaten)
```

kode diatas akan menghasilkan output seperti berikut

```
['sukoharjo', 'sragen', 'wonogiri']
['sukoharjo', 'sragen', 'wonogiri', 'karanganyar', 'klaten']
['sukoharjo', 'boyolali', 'sragen', 'wonogiri', 'karanganyar', 'klaten']
```

Fungsi **del** kita gunakan jika ingin menghapus salah satu atau keseluruhan elemen yang ada pada list. Sedangkan fungsi **remove()** dapat kita gunakan untuk menghapus elemen berdasar nilai yang diberikan. Fungsi **pop()** digunakan untuk menghapus indeks terakhir dari list. Fungsi terakhir yaitu **clear()** digunakan untuk mengosongkan isi list.

```
1. #deklarasi list
2. kabupaten = ['sukoharjo', 'sragen', 'wonogiri', 'karanganyar', 'klaten']
3.
4. #hapus indeks ke 2
5. del kabupaten[2]
6.
7. #hapus beberapa element
8. del kabupaten[1:3]
9.
10. #hapus seluruh list
11. del kabupaten
12.
13. #deklarasi list
14. pti = ['p', 't', 'i', 'u', 'm', 's']
15.
16. #hapus huruf t
17. pti.remove('t')
18.
```

```

19. #hapus elemen terakhir
20. pti.pop()
21.
22. #kosongkan list
23. pti.clear()

```

Beberapa fungsi lain yang dapat kita gunakan dalam list yaitu

Tabel 6. 1 Fungsi lain yang tersedia pada list

Fungsi	Keterangan
len()	Mengetahui panjang list
max()	Mengembalikan nilai tertinggi pada list
min()	Mengembalikan nilai terendah pada list
sorted()	Mengurutkan list dari terendah ke besar
index()	Mengembalikan nilai index yang sesuai
count()	Menghitung jumlah pada element berdasar angka yang diinput
reverse()	Membalikkan urutan list
copy()	Membuat duplikat list

6.2.2 Tuple

Tuple merupakan bentuk lain daripada list yang memiliki perbedaan pada sifatnya. Jika list bersifat mutable maka tuple bersifat immutable. Artinya data yang ada pada tuple tidak dapat ditambah, diubah atau dihapus. Penulisan tuple diapit dengan tanda kurung (). Penulisan tanda kurung ini bersifat optional namun disarankan untuk tetap digunakan. Perhatikan kode berikut untuk contoh penggunaan tuple.

```

1. koordinat= (13.4125, 103.866667)
2. print("Latitude:", koordinat[0])
3. print("Longitude:", koordinat[1])

```

Output diatas akan menghasilkan seperti berikut

```
Latitude: 13.4125
Longitude: 103.866667
```

Data pada tuple dapat kita assign ke variabel yang berbeda. Konsep ini dinamakan tuple unpacking. Perhatikan baris ke dua pada contoh kode berikut ini

```
1. ukuran = 20,30,40
2. panjang, lebar, tinggi = ukuran
3. print('Buku ini memiliki dimensi {} x {} x {}'.format(panjang,lebar,tinggi))
```

pada baris kedua, nilai dari tuple ukuran diassign ke dalam tiga variabel yang berbeda yaitu panjang, lebar dan tinggi.

6.2.3 Set

Set merupakan sekumpulan data dengan elemen-elemen yang berbeda (unique) sehingga salah satu fungsi set digunakan untuk menghapus elemen yang sama / ganda. Terdapat dua jenis set yaitu set dinamis yang bisa diubah data dan ukurannya, ditandai dengan kata set dan set statis yang isinya tidak dapat diubah, ditandai dengan kata frozenset.

```
1. #set dinamis
2. pti =set('Pendidikan Teknik Informatika')
3. print(type(pti))
4. print(pti)
5.
6. #set statis
7. kip = frozenset('Keguruan Ilmu Pendidikan')
8. print(type(kip))
9. print(kip)
```

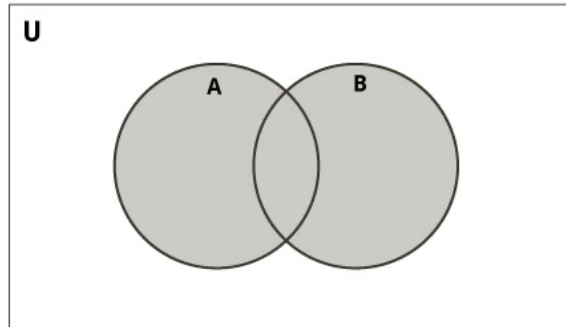
Kode diatas akan menghasilkan output

```
<class 'set'>
{'o', 'd', 'e', 'f', 'i', 'I', ' ', 'P', 'k', 'r', 't', 'T', 'a', 'n', 'm'}
```



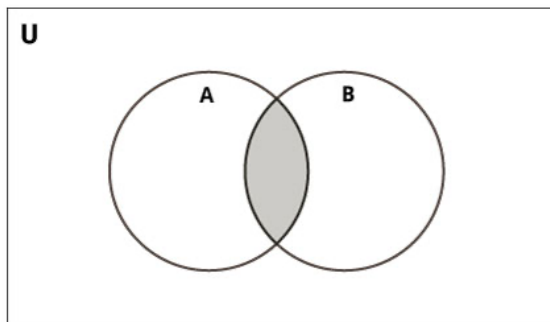
```
<class 'frozenset'>
frozenset({'g', 'd', 'e', 'l', 'i', 'I', 'r', ' ', 'P', 'u', 'k', 'K', 'a',
'n', 'm'})
```

Set dalam python merupakan penerapan set pada ilmu Matematika. Jadi di dalam set kita dapat melakukan operasi keanggotaan seperti union dan intersection.



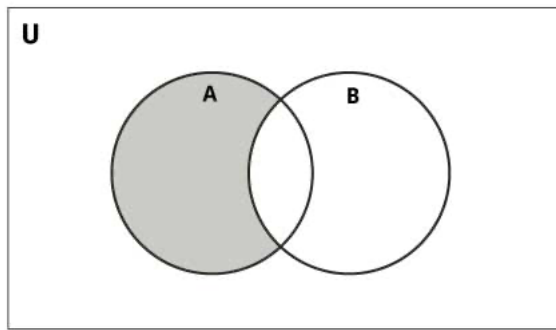
Gambar 6. 1 Operasi Union

Union merupakan gabungan dari semua elemen dari kedua set. Union dapat digunakan menggunakan tanda | atau fungsi **union()**.



Gambar 6. 2 Operasi Intersection

Intersection merupakan operasi untuk menampilkan elemen-elemen yang menjadi irisan antara dua set. Intersection dapat digunakan dengan tanda &.



Gambar 6. 3 Operasi Difference

Difference (komplemen) merupakan elemen-elemen yang hanya terdapat pada satu set saja dan tidak terdapat pada set yang lain. Difference dapat digunakan dengan tanda minus (-). Perhatikan kode berikut untuk contoh penggunaan operasi himpunan dalam set.

```
1. #deklarasi set
2. A = {1,2,3,4,5}
3. B = {4,5,6,7,8}
4.
5. #union
6. print (A|B)
7.
8. #intersect
9. print (A&B)
10.
11. #difference
12. print (A-B)
```

Kode diatas akan menghasilkan output

```
{1, 2, 3, 4, 5, 6, 7, 8}
{4, 5}
{1, 2, 3}
```

Tabel 6.2 berikut ini merupakan fungsi-fungsi yang dapat digunakan dalam set

Tabel 6. 2 Fungsi yang ada dalam Set

Fungsi	Keterangan
add()	Menambah satu elemen baru pada set
update()	Menambah elemen baru pada set
remove()	Menghapus elemen pada set
clear()	Mengosongkan isi set
pop()	Menghapus elemen pertama pada set

6.2.4 Dictionaries

Dictionaries merupakan kumpulan data yang terdiri dari sepasang kunci dan nilai. Dictionaries mendukung semua tipe data yang ada pada python. Sehingga elemen-elemen yang ada dalam dictionaries bisa berupa elemen yang berbeda tipe. Penulisan dictionaries ditandai dengan tanda kurung kurawal {}. Perhatikan contoh kode berikut ini untuk penulisan dan pengaksesan dictionaries.

```

1. #dictionaries tanpa isian
2. lantai = {}
3.
4. #deklarasi dictionaries
5. lantai = {'lobby':1,'kantor':2,'kantin':3,'parkir':'rooftop'}
6.
7. #akses dictionaries
8. print(lantai['lobby'])
9. print(lantai.get('parkir'))

```

kode diatas akan menghasilkan output

```

1
rooftop

```

Karena data pada dictionaries bersifat mutable. Maka kita dapat melakukan operasi penambahan, update dan hapus data.

```

1. #deklarasi dictionaries

```

```

2. lantai = {'lobby':1,'kantor':2,'kantin':3,'parkir':'rooftop'}
3.
4. #akses dictionaries
5. lantai['parkir'] = 'basement'
6.
7. #menambah elemen
8. lantai['labkom'] = 4
9.
10. #hapus elemen berdasar key menggunakan pop
11. lantai.pop('lobby')
12.
13. #hapus semua elemen
14. lantai.clear()

```

Tabel 6.3 berikut ini merupakan fungsi-fungsi yang ada pada dictionaries

Tabel 6. 3 Fungsi pada dictionaries

Fungsi	Keterangan
clear()	Menghapus seluruh elemen pada dictionaries
item()	Menampilkan seluruh elemen
keys()	Menampilkan seluruh kunci
values()	Menampilkan seluruh nilai yang terdapat dalam dictionaries
pop(key)	Menghapus elemen dengan kunci tertentu
get()	Mengembalikan nilai balik berupa nilai dari kunci

6.3 Kegiatan Praktikum

6.3.1 Kegiatan Praktikum 1 : Bekerja dengan list

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini

```

1. bulan1 = ['januari', 'februari', 'maret', 'april', 'mei', 'juni']
2. bulan2 = ['juli', 'agustus', 'september', 'oktober', 'november', 'desember']
3. print('Jumlah elemen pada list bulan1 : ', len(bulan1))

```

```

4. tahun = bulan1 + bulan2
5. print('Jumlah elemen pada list tahun : ', len(tahun))
6. print(tahun)
7. print(tahun[2:5])
8. print(tahun[:6])
9. print(tahun[8:])
10. del tahun[2]
11. tahun.remove('desember')
12. print(tahun)
13. tahun.insert(2, 'maret')
14. tahun.append('desember')
15. print(tahun)
16. tahun.reverse()
17. print(tahun)

```

2. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 1 ini

6.3.2 Kegiatan Praktikum 2 : Bekerja dengan Tuple

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini

```

1. truk = ('hino', 3000, 2.5 , 130 )
2. merk, cc , berat , top_speed = truk
3. print(truk[0])
4. print(truk[:2])
5. print(truk[2:])
6. print(truk.index(3000))
7. print(2.5 in truk)
8. print('truk {} memiliki berat {} ton, kapasitas {} cc dan top speed {} km/jam'.
      format(merk,berat,cc,top_speed))

```

2. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 2 ini

6.3.3 Kegiatan Praktikum 3 : Bekerja dengan Set

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini

```

1. set_satu = {1,2,3}

```

```

2. set_dua = {4,5,6}
3. set_satu.add(4)
4. set_dua.add(7)
5. print(set_satu)
6. print(set_dua)
7. set_satu.update([5,6])
8. set_dua.update([8,9])
9. print(set_satu)
10. print(set_dua)
11. set_satu.discard(6)
12. set_dua.remove(4)
13. print(set_satu | set_dua)
14. print(set_satu & set_dua)
15. print(1 in set_satu)

```

2. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 3 ini

6.3.4 Kegiatan Praktikum 4 : Bekerja dengan Dictionaries

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini

```

1. mahasiswa = {'nama': 'Andika', 'umur': 21}
2. mahasiswa['umur'] = 19
3. mahasiswa['alamat'] = 'Sragen'
4. mahasiswa['angkatan'] = 2020
5. print(mahasiswa)
6. print(mahasiswa.pop('angkatan'))
7. print(mahasiswa)
8. print('nama' in mahasiswa)
9. print(len(mahasiswa))
10. print(sorted(mahasiswa))
11. mahasiswa.clear()
12. print(mahasiswa)

```

2. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 3 ini

6.4 Tugas

1. Buat program python yang menerima inputan dari user berupa
 - Daftar nama 5 teman beserta no handphone
2. Simpan kedalam tipe data yang sesuai
3. Tampilkan output dalam bentuk berikut

Phone Book

1. [Nama 1] = [no hp 1]
2. [Nama 2] = [no hp 2]
3. [Nama 3] = [no hp 3]
4. [Nama 4] = [no hp 4]
5. [Nama 5] = [no hp 5]

DAFTAR PUSTAKA

- Hunt, J. (2019). *A Beginners Guide to Python 3 Programming*. In Springer
- Romano, Fabrizio. (2015). *Learning Python*. Packt Publishing.
- Swastika, W. (2019). *Pengantar Algoritma dan Penerapannya pada Python*. Ma Chung Press.
- Wadi, H. *Pemrograman Python untuk Mahasiswa dan Pelajar*. TR Publisher
- Python File Handling Tutorial: How to Create, Open, Read, Write, Append*. (n.d.). Retrieved September 7, 2020, from <https://www.softwaretestinghelp.com/python/python-file-reading-writing/>
- Learn Python Programming*. (n.d.). Retrieved September 7, 2020, from <https://www.programiz.com/python-programming>
- Tutorial Pemrograman Python*. (n.d.). Retrieved September 5 , 2020, from <https://www.petanikode.com/tutorial/python/>
- PY4E - Python for Everybody*. (n.d.). Retrieved September 2, 2020, from <https://www.py4e.com/>
- Google's Python Class | Python Education | Google Developers*. (n.d.). Retrieved September 9, 2020, from <https://developers.google.com/edu/python/>
- Learn Python the Hard Way*. (n.d.). Retrieved September 2, 2020, from <https://learnpythonthehardway.org/book/>
- Python Programming Tutorials*. (n.d.). Retrieved September 9, 2020, from <https://pythonprogramming.net/python-fundamental-tutorials/>

LAMPIRAN

Laporan Sementara

Laporan sementara dilakukan dengan langkah sebagai berikut:

1. Buatlah screenshot atau gambar dari hasil kegiatan yang dilakukan.
2. Ganti nama file dengan format: bab-kegiatan-no.gambar-nim.jpg Misalnya file adalah gambar ke 1 pada Bab 1, Kegiatan 1, dan nim anda adalah A40009001, maka nama filenya adalah **1-1-1-A40009001.jpg**
3. Simpan file tersebut kemudian tempatkan pada folder sesuai instruksi dosen/asisten praktikum.

Laporan Praktikum

Laporan ditulis dalam kertas putih ukuran A4. Sedangkan urutan susunan laporan adalah sebagai berikut:

1. Cover depan: Berwarna sama dengan cover modul praktikum
2. Halaman Cover: Contoh dapat di-download di <http://bit.ly/lap-alpro>
3. Kata Pengantar
4. Daftar isi
5. Laporan tiap modul (1-10) sesuai dengan format terlampir
6. Penulis: berisi biodata penulis (disertai foto), pesan dan kesan, kritik dan saran demi kemajuan praktikum berikutnya.

Format Laporan Tiap Bab

Matakuliah : Algoritma dan Pemrograman	Acc
NIM :	
Nama :	
Tgl. Prakt.:	Tgl:

BAB I

Judul

1. Dasar Teori

300 sampai dengan 350 kata

2. Tujuan

3. Analisa Hasil

3.1. Kegiatan 1: ...

Tampilkan hasil praktikum berupa kode yang dibuat atau hasil output, kemudian berikan analisisnya. Jika terdapat gambar, berikan juga nomor gambar.

3.2. Kegiatan 2: ...

4. Penyelesaian Tugas

Jika terdapat tugas yang dikerjakan, tuliskan disini langkah pengerjaan dan hasilnya.

5. Kesimpulan

Berikan kesimpulan yang didapatkan setelah anda menyelesaikan praktikum