



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230992
Nama Lengkap	Andriano Kurniawan Ladjeba
Minggu ke / Materi	09 / Tipe Data List

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2024

BAGIAN 1: MATERI MINGGU INI (40%)

Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

MATERI 1

Sifat-sifat List

List pada bahasa pemrograman Python adalah sebuah rangkaian nilai-nilai yang dapat diakses menggunakan nama tunggal. Perbedaan list dan string? String ialah rangkaian karakter-karakter, sedangkan list dapat berisi sebuah karakter, integer dan float ataupun tipe data lainnya. List sendiri juga bisa berisi list lainnya. Rangkaian nilai-nilai tersebut di dalam [] seperti contoh berikut:

```
nilai_ujian = [80,75,70,90,81,84,92,71,65,80,70]
nama_pahlawan = ['Sukarno', 'Diponegoro', 'Jend. Sudirman', 'Cut Nya
Dhien']
nilai_campuran = ['Javascript', 20, 34.4, True]
list_dalam_list = [23, [22, 20], 45]
```

Dibandingkan String, list adalah mutable, sedangkan String tidak. Mutable berarti nilainya dapat diubah secara langsung, seperti yang ditunjukkan dalam kode program berikut:

```
#definisikan list berisi 4 nilai
data = [10,20,30,40]
#ubah nilai index ke-0 menjadi 50
data[0] = 50
#isinya sekarang: 50, 20, 30, 40
print (data)
#definisikan sebuah string
nama = 'Yuan Lukito'
#ubah karakter index ke-0 menjadi Z
nama [0] = 'Z'
# tidak akan mencapai baris ini karena muncul error
# TypeError: 'str' object does not support item assignment
print (nama)
```

Ketika dua string memiliki isi yang sama, keduanya menunjuk (meru-juk/mengacu) pada objek yang sama, tetapi ketika dua list memiliki isi yang sama, keduanya menunjuk pada objek yang berbeda. Ini adalah contoh Python shell:

```
>>> a = 'banana'
>>> b='banana'
>>> a is b
True
>>> a = [1, 2, 3]
>>> b = [1, 2, 3]
>>> a is b
False
```

Operasi isi List

Beberapa operasi yang dapat digunakan dalam memproses sebuah list, ialah:

1. Penambahan elemen pada list dengan operator +:

```
>>> bil1 = [1,2,3,4]
>>> bil2 = [5,6,7]
>>> bilTotal = bil1+bil2
>>> print (bilTotal)
[1, 2, 3, 4, 5, 6, 7]
```

2. Perulangan elemen pada list dengan operator *:

```
>>> bil1 = [1,2,3,4]
>>> bilTotal = [1,2,3,4]*2
>>> print(bilTotal)
[1, 2, 3, 4, 1, 2, 3, 4]
```

3. Pengaksesan elemen list: Anda dapat mengakses elemen list baik secara individual maupun dalam kelompok elemen dengan menggunakan indeks elemen tersebut. Indeks list dimulai dari indeks 0.

```
>>> nama = ["hidenori","yoshitake", "karasawa","motoharu"]
>>> nama [0]
'hidenori'
>>> nama [3]
'motoharu'
#indeks akses dihitung dari belakang
>>> nama [-2]
'karasawa'
```

```
>>> nama [-3]
'yoshitake'
>>> nama [-4]
'hidenori'
```

Contoh di atas menunjukkan sebuah daftar yang mengandung empat elemen nama. Variabel `list[indeksElemen]` digunakan untuk mengakses. Jika indeks elemen yang dimasukkan tidak ada dalam list, Python akan mengeluarkan error seperti berikut :

```
>>> nama [4]
Traceback (most recent call last):
File "<stdin>", line 1, in <module> IndexError:
list index out of range
```

Karena nomor indeks pada list nama dapat dimulai dari 0 hingga 3, elemen `nama[4]` tidak ada pada list tersebut.

```
# Seluruh elemen diakses
>>> nama [:]
['hidenori', 'yoshitake', 'karasawa', 'motoharu']
# elemen diakses pada indeks 1-2
>>> nama [1:3]
['yoshitake', 'karasawa']
# elemen diakses dari indeks awal hingga 2
>>> nama [:3]
['hidenori', 'yoshitake', 'karasawa']
# elemen diakses dari indeks 1 hingga akhir
>>> nama [1:]
['yoshitake', 'karasawa', 'motoharu']
```

4. Penggantian nilai pada elemen list: Penggantian nilai pada sebuah list juga dapat dilakukan dengan mengakses banyak elemen.

```
>>> nama [1:3] = ["felix", "ryan"]
>>> nama
['hidenori', 'felix', 'ryan', 'motoharu']
```

Metode dan Fungsi Untuk List

Python memberikan beberapa metode yang dapat digunakan untuk melakukan operasi pada sebuah list. Beberapa metode tersebut ialah:

1. **Append** : Metode yang digunakan untuk menambahkan elemen baru dan dianggap sebagai kesatuan objek pada bagian akhir list

```
>>> nama
['hidenori', 'felix', 'ryan', 'motoharu']
>>> nama.append(['bejo', 'tejo'])
>>> nama
['hidenori', 'felix', 'ryan', 'motoharu', ['bejo', 'tejo']]
```

2. **Extend** : Metode yang digunakan untuk menambahkan elemen pada sebuah list, dan memper- lakukan setiap element baru sebagai elemen list secara individual.

```
>>> nama
['hidenori', 'felix', 'ryan', 'motoharu', 'bejo']
>>> nama.extend(["tejo", "ujo"])
>>> nama
['hidenori', 'felix', 'ryan', 'motoharu', 'bejo', 'tejo', 'ujo']
```

3. **Sort** : Metode yang digunakan untuk mengurutkan elemen pada sebuah list dimulai dari nilai terkecil hingga terbesar.

```
>>> nama.sort()
>>> nama
['bejo', 'felix', 'hidenori', 'ryan', 'tejo', 'ujo', 'motoharu']
```

Dengan menggunakan metode bertipe void, beberapa metode di atas akan mengubah list dan memiliki nilai balik None. Dengan kata lain, jika Anda secara tidak sengaja mengetikkan nama = nama.sort(), Anda akan menerima hasil None, dan isi elemen pada variabel list nama akan hilang.

Untuk menghapus elemen dari sebuah daftar, ada beberapa cara yang sering digunakan, seperti:

1. **Pop** : Jika indeks elemen sudah diketahui, dan ingin mendapatkan nilai elemen yang dihapus.

```
>>> huruf = ['a', 'b', 'c', 'd']
>>> huruf.pop(3)
'd'
>>> huruf
['a', 'b', 'c']
```

2. **Del** : Jika indeks elemen sudah diketahui, dan tidak memerlukan nilai elemen yang dihapus.

```
>>> huruf = ['a', 'b', 'c', 'd']
>>> del huruf [2]
>>> huruf
['a', 'b', 'd']
>>> huruf = ['a', 'b', 'c', 'd']
>>> del huruf [1:3]
>>> huruf
['a', 'd']
```

3. **Remove** : Jika yang diketahui ialah nilai elemen yang akan dihapus.

```
>>> huruf = ['a', 'b', 'c', 'd']
>>> huruf.remove("c")
>>> huruf
['a', 'b', 'd']
```

Penghapusan dengan metode remove akan menghasilkan nilai balik None.

Selain itu, List menawarkan fungsi yang sudah terintegrasi yang dapat digunakan secara cepat tanpa harus menulis program perulangan.

```
>>> angka = [10,20,30]
>>> len (angka)
3
>>> max (angka)
30
>>> min (angka)
10
>>> sum(angka)
60
```

1. len() -> untuk mendapatkan banyaknya elemen pada list
2. max()-> untuk mendapatkan nilai maksimum elemen pada list
3. min()-> untuk mendapatkan nilai minimum elemen pada list
4. sum()-> untuk mendapatkan total nilai elemen pada list. Metode ini hanya bisa digunakan pada list dengan tipe elemennya ialah sebuah angka.

Fungsi len, max, min hanya dapat digunakan pada list dengan nilai elemennya yang dapat dibandingkan satu sama lainnya.

Perbedaan List dan String

Berbeda dengan list, string terdiri dari deretan karakter yang mengisi setiap indeks, sedangkan string hanya terdiri dari sebuah karakter, tetapi list dari sebuah karakter tidak sama dengan string. Nilai dari setiap elemen list dapat diisi dengan berbagai tipe data. Salah satu cara untuk mengubah string menjadi list karakter adalah dengan menggunakan:

```
>>> a = "banana"
>>> b = list(a)
>>> print(b)
['b', 'a', 'n', 'a', 'n', 'a']
```

Dalam list, fungsi split dapat digunakan untuk memecah sebuah kalimat menjadi beberapa kata terpisah, dengan pemisah yang dipilih secara default adalah spasi.

```
>>> kalimat = "saya sedang makan"
>>> kata2 = kalimat.split()
>>> print (kata2)
['saya', 'sedang', 'makan']
>>> kalimat = "saya-sedang-makan"
>>> kata2 = kalimat.split('-')
>>> print (kata2)
['saya', 'sedang', 'makan']
```

Untuk mengembalikan sebuah list yang berisi kata menjadi sebuah string, dapat kita gunakan fungsi join.

```
>>> print (kata2)
['saya', 'sedang', 'makan']
>>> delimiter = ' '
>>> t = delimiter.join(kata2)
>>> type (t)
<class 'str'>
>>> print(t)
saya sedang makan
```

Pada sebuah daftar, aliasing menghasilkan dua variabel yang mengacu pada objek yang sama jika nilai variabel list a dipisahkan dari variabel list b. Oleh karena itu, jika terjadi perubahan pada elemen di list b, perubahan tersebut juga akan berdampak pada elemen di list a.

```
>>> a = [1,2,3]
>>> b=a
>>> b is a
True
```

```
>>> b[0] 17
>>> print(a)
[17, 2, 3]
```

Tidak sama jika kita membuat dua variabel dan memberi mereka nilai masing-masing. Akibatnya, kedua variabel ini tidak akan mengacu pada objek yang sama pada akhirnya.

```
>>> a = [1,2,3]
>>> b = [1,2,3]
>>> b is a
False
>>> b[0] 17
>>> print(a)
[1, 2, 3]
>>>
```

List Sebagai Parameter Fungsi

Pada parameter fungsi, Anda juga dapat menggunakan tipe data list. Dalam menggunakan tipe data list, ada beberapa operasi penting yang harus dipahami. Operasi seperti append mengubah sebuah list dan operasi + membuat variabel list baru.

```
>>> t1 = [1, 2]
>>> t2=t1.append(3)
>>> print(t1)
[1, 2, 3]
>>> print(t2)
None
>>> t3=t1 + [3]
>>> print(t3)
[1, 2, 3]
>>> t2 is t3
False
```



```
def hapus (inputList):  
    return inputList [1:]  
  
huruf = ['a', 'b', 'c']  
hasil = hapus (huruf)  
  
>>>print (hasil)  
['b', 'c']
```

SOAL 1

Latihan 9.1

```
1 nilai = [10, 5, 22, 4, 18, 11]
2 nilai = sorted(nilai, reverse=True)
3 tiga_terbaik = nilai[:3]
4 print(f"Tiga nilai terbaik:{tiga_terbaik}")
5
6
7
```

SOAL 2

Latihan 9.2

```
Latihan 9.2.py > ...
1  angka = []
2
3  print("Masukkan angka satu per satu (ketik 'done' untuk selesai):")
4  while True:
5      angka_input = input("Angka: ")
6      if angka_input == 'done':
7          break
8      try:
9          list_angka = float(angka_input)
10         angka.append(list_angka)
11     except ValueError:
12         print("Format Nilai salah. Silahkan masukan nilai atau 'done'.")
13
14 maxNum = max(angka)
15 minNum = min(angka)
16
17 if angka:
18     print(f"Nilai maksimum: {maxNum}")
19     print(f"Nilai minimum: {minNum}")
20 else:
21     print("Nilai kosong, Silahkan masukan nilai!")
22
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

Angka: 30
Angka: 44
Angka: 20
Angka: 10
Angka: 5
Angka: done
Nilai maksimum: 44.0
Nilai minimum: 5.0
PS D:\A-Kuliah\Semester 2\Prakkrikum Algoritma dan Pemograman\tugas 9\latihan>

SOAL 3

Latihan 9.3

```
Latihan 9.3.py X
Latihan 9.3.py > ...
1  nama_file = "berita.txt"
2  try:
3      with open(nama_file, "r", encoding="utf-8") as file:
4          for line in file:
5              kata = line.split()
6              print(f"Kata unik dalam kalimat: {kata}")
7  except FileNotFoundError:
8      print("File tidak ditemukan.")
9
10
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

PS D:\A-Kuliah\Semester 2\Prakkrikum Algoritma dan Pemograman\tugas 9\latihan> & C:\Users\acer\AppData\Local\Programs\Python\Python110\python.exe "d:\A-Kuliah\Semester 2\Prakkrikum Algoritma dan Pemograman\tugas 9\latihan\latihan 9.3.py"

Kata unik dalam kalimat: ['Profesor', 'Senior', 'bidang', 'Klasik', 'di', 'Universitas', 'Nasional', 'Victoria', 'Samuel', 'Seuss', 'pernah', 'menyamakan', 'perpustakaan', 'Sir', 'Cassegrain', 'yang', 'luas', 'dengan', 'pohon', 'hobby', 'di', 'pinggiran', 'Londonium', 'dalam', 'esensi', 'pengetahuan', 'ini', 'bayang-bayang', 'sejarah', 'yang', 'panjang', 'ini', 'kata', 'tidak', 'pernah', 'gaga', 'untuk', 'mereka', 'kesesjajaran', 'dengan', 'bagaimana', 'kita', 'sendiri', 'telah', 'melangkah', 'dan', 'dengan', 'bagaimana', 'kita', 'akan', 'selangkah', 'tercatat', 'epos', 'dan', 'filosofia', 'satu', 'demi', 'satu', 'tersiapan', 'mitologi', 'dan', 'teori', 'Para', 'sarjana', 'yang', 'rendah', 'hati', 'mengisi', 'halaman', 'demi', 'halaman', 'dengan', 'kebijaksanaan', 'dan', 'ketekunan', 'mereka', 'masing-masing', 'penyair', 'pemenang', 'sebuah', 'rak', 'buku', 'untuk', 'diri', 'mereka', 'sendiri."]

PS D:\A-Kuliah\Semester 2\Prakkrikum Algoritma dan Pemograman\tugas 9\latihan>

Link GitHub : <https://github.com/EchoGinDev/tugasAlpro9.git>