



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231005
Nama Lengkap	Josephine Marcelia
Minggu ke / Materi	09 / 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%)

Sifat-sifat List

List di Python adalah kumpulan nilai yang bisa diakses dengan satu nama. Berbeda dengan String yang hanya berisi karakter, **List** bisa berisi berbagai macam tipe data, seperti karakter, angka (integer dan float), dan bahkan **List** lain. Penulisan **List** menggunakan tanda kurung siku [] seperti contoh berikut ini :

```
Kota = ["Jogja", "Jakarta", "Bandung"]
nilai = [80, 90, 100]
campur = ["data 1", 100, 2.75, True]
list_dalam_list = [10, [15, 20], 25]
```

Perbedaan berikutnya antara list dan string adalah, list memiliki sifat yang dapat diubah (mutable), sementara string tidak dapat diubah (immutable).

Contohnya ditunjukkan pada program berikut ini :

```
angka = [10, 20, 30, 40, 50]
angka[1] = 60
print(angka)
```

Output :

```
[10, 60, 30, 40, 50]
```

Pada contoh diatas, kita mengganti indeks ke-0 yaitu "beras" dengan "ayam". Lalu kita menambahkan "nugget" ke daftar_belanja dengan fungsi append. Kemudian kita menghapus list "tempe" dengan fungsi remove. Sehingga hasil outputnya adalah 'ayam', 'bakso', 'nugget'.

Perbedaan lainnya adalah, jika ada dua variabel string dengan nilai yang sama, keduanya akan merujuk ke objek yang sama. Namun, jika ada dua list dengan elemen yang sama, keduanya akan merujuk pada objek yang berbeda.

Berikut ini merupakan contohnya :

```
nama = "Jose"
name = "Jose"
print(nama is name)

angka = [10, 20, 30]
number = [10, 20, 30]
print(angka is number)
```

Output :

```
True
False
```

Operasi isi List

Beberapa operasi yang dapat kita gunakan dalam list yaitu :

1. Operator Penambahan (+) untuk menambah elemen

```
negara_asia = ['Jepang', 'China', 'Korea']
negara_eropa = ['Perancis', 'Jerman', 'Italia']
nama_negara = negara_asia + negara_eropa
print(nama_negara)
```

Output :

```
['Jepang', 'China', 'Korea', 'Perancis', 'Jerman', 'Italia']
```

2. Operator (*) untuk perulangan elemen

```
list = [10, 20, 30]
ulang = list * 3
print(ulang)
```

Output :

```
[10, 20, 30, 10, 20, 30, 10, 20, 30]
```

3. Elemen pada list dapat diakses secara individu atau berkelompok dengan menggunakan indeks. Indeks list dimulai dari 0.

```
hewan = ['anjing', 'kucing', 'kelinci', 'babi']
print(hewan[0])
print(hewan[1])
print(hewan[2])
print(hewan[-1])
print(hewan[-3])
```

Output :

```
anjing
kucing
kelinci
babi
kucing
```

Dapat kita lihat contoh diatas memiliki 4 buah elemen dengan indeks 0 sampai 3. Jika kita memasukkan indeks yang tidak ada dalam list, maka akan muncul pesan error seperti berikut :

```
Traceback (most recent call last):
  File "c:\Users\josep\OneDrive\Documents\71231005-jose\laprakyok.py", line 9, in <module>
    print(hewan[4])
    ~~~~~^
IndexError: list index out of range
```

4. Kita juga bisa mengakses list untuk banyak indeks

```
hewan = ['anjing', 'kucing', 'kelinci', 'babi']  
print(hewan[:])  
print(hewan[1:3])  
print(hewan[:3])
```

Output :

```
['anjing', 'kucing', 'kelinci', 'babi']  
['kucing', 'kelinci']  
['anjing', 'kucing', 'kelinci']
```

5. Dengan mengakses banyak elemen, kita bisa mengganti nilai pada list seperti contoh berikut :

```
hewan = ['anjing', 'kucing', 'kelinci', 'babi']  
hewan [1:3] = ['singa', 'sapi']  
print(hewan)
```

Output :

```
['anjing', 'singa', 'sapi', 'babi']
```

Metode dan Fungsi untuk List

Berikut ini merupakan beberapa metode yang dapat kita gunakan untuk mengoperasikan sebuah list :

1. Append : untuk menambah elemen baru

```
jajan = ['seblak', 'cimol', 'cilok']  
jajan.append ('cireng')  
print(jajan)
```

Output :

```
['seblak', 'cimol', 'cilok', 'cireng']
```

2. Extend : Metode ini memungkinkan kita untuk menambahkan elemen-elemen baru dari iterable (seperti list lain atau tuple) ke list yang sudah ada. Setiap elemen dalam iterable akan ditambahkan ke list sebagai elemen individu, memperluas list tersebut dengan elemen-elemen baru.

```
warna = ["purple", "blue"]  
warna_lain = ["red", "green", "blue"]  
  
warna.extend(warna_lain)  
print(warna)
```

Output :

```
['purple', 'blue', 'red', 'green', 'blue']
```

3. Sort : Untuk mengurutkan elemen dari yang terkecil hingga terbesar

```
lst1 = [1,3,6,2,4,5]
lst1.sort()
print(lst1)
```

Output :

```
[1, 2, 3, 4, 5, 6]
```

Untuk menghapus elemen pada list, kita bisa menggunakan beberapa metode berikut :

1. Pop : Untuk menghapus elemen pada indeks tertentu

```
lst1 = [0,1,2,3,4,5]
lst1.pop(3)
print(lst1)
```

Output :

```
[0, 1, 2, 4, 5]
```

2. Del : Untuk menghapus list atau elemen. Berbeda dengan metode pop(), del tidak mengembalikan elemen yang dihapus.

```
lst2 = [10, 15, 20, 25, 30]
del lst2[3]
print(lst2)
```

Output :

```
[10, 15, 20, 30]
```

3. Remove : Untuk menghapus elemen pertama dari list yang memiliki nilai yang sama.

```
lst1 = [1,2,3,4,5,3,7,5,3]
for i in range(lst1.count(3)):
    lst1.remove(3)
print(lst1)
```

Output :

```
[1, 2, 4, 5, 7, 5]
```

List juga mempunyai fungsi built-in(dapat langsung digunakan), yaitu :

```
num = [5, 10, 20, 15, 30]
print(len(num))
print(max(num))
print(min(num))
print(sum(num))
```

Output :

```
5
30
5
80
```

len → menghitung Panjang elemen pada list
min → mendapat nilai minimum pada list
max → mendapat nilai maksimum pada list
sum → menghitung jumlah total elemen list

Perbedaan List dan String

String adalah deretan karakter yang diisi dalam setiap indeks, sementara list adalah deretan nilai yang bisa berupa berbagai tipe data. List karakter belum tentu dianggap sebagai string. Kita dapat mengubah string menjadi list karakter dengan menggunakan:

```
x = "Jerman"  
y = list(x)  
print(y)
```

Output :

```
['J', 'e', 'r', 'm', 'a', 'n']
```

Kita bisa menggunakan split untuk memecah kalimat menjadi beberapa kata dengan menggunakan fungsi split seperti berikut :

```
kalimat = "saya suka makan sate"  
kata = kalimat.split()  
print(kata)
```

Output :

```
['saya', 'suka', 'makan', 'sate']
```

Dengan menggunakan fungsi Join, kita dapat mengembalikan list pada contoh diatas yang berupa kata menjadi string Kembali seperti berikut :

```
space = " "  
a = space.join(kata)  
print(a)
```

Output :

```
saya suka makan sate
```

List Sebagai Parameter Fungsi

Saat list digunakan sebagai argumen fungsi, fungsi tersebut bisa menerima dan mengolah data dalam bentuk list. Hal ini memberi fungsi kemampuan untuk bekerja dengan kumpulan data yang panjangnya bisa berbeda-beda, serta memungkinkan operasi seperti menambah, menghapus, atau mengedit elemen dalam list.

Berikut merupakan contoh programnya:

```
num = [20, 80, 30, 60, 50]

def rata2(nilai_list):
    if len(nilai_list) == 0:
        return 0

    jumlah = sum(nilai_list)
    rata_rata = jumlah / len(nilai_list)
    return rata_rata

hasil= rata2(num)
print(f"Rata-rata dari list num {num} adalah: {hasil}")
```

Output :

```
Rata-rata dari list num [20, 80, 30, 60, 50] adalah: 48.0
```

Pada program diatas fungsi rata2 menerima list angka (nilai_list) sebagai parameter. Kemudian fungsi tersebut akan menghitung rata-rata angka dalam nilai_list.

if len(nilai_list) == 0:kita gunakan untuk memeriksa apakah nilai_list kosong (tidak ada elemen). Jika kosong, maka fungsi tersebut akan mengembalikan 0 sebagai rata-rata.

Dengan menggunakan fungsi bawaan sum kita akan menghitung jumlah semua angka dalam nilai_list. Kemudian kita menghitung rata-rata dengan membagi jumlah angka dengan jumlah elemen dalam nilai_list.

Lalu kita akan mengembalikan nilai rata-rata dengan fungsi return. Program diatas kemudian akan mencetak hasil rata_rata dari list num.

BAGIAN 2 : LATIHAN MANDIRI

Soal 9.1

```
def tiga_nilai_tertinggi(listnya):  
    listnya.sort(reverse = True)  
    print(listnya)  
  
    tiga_terbesar = listnya[:3]  
  
    return tiga_terbesar  
  
listnya = [80,30,60,50,70]  
  
tiga_terbesar = tiga_nilai_tertinggi(listnya)  
  
print("tiga nilai terbesar(dimulai dari yang paling tinggi) adalah: ", tiga_terbesar)
```

Output :

```
[80, 70, 60, 50, 30]  
tiga nilai terbesar(dimulai dari yang paling tinggi) adalah: [80, 70, 60]
```

Penjelasan :

Pada program diatas saya menggunakan fungsi sort untuk mengurutkan elemen list dari yang terkecil dan terbesar. Lalu saya menggunakan fungsi reverse agar nilai yang tercetak diurutkan dari nilai yang terbesar ke nilai yang terkecil. Kode ('listnya[:3]') saya gunakan untuk mengambil tiga nilai teratas yaitu indeks ke 0 hingga 2 sehingga menghasilkan output seperti yang tercetak pada gambar diatas.

Soal 9.2

```
deretan_angka = []  
  
while True:  
    input_angka = input("Masukkan angka (atau ketik 'done' untuk selesai): ")  
    if input_angka.lower() == 'done':  
        break  
    else:  
        if input_angka.isdigit():  
            deretan_angka.append(int(input_angka))  
        else:  
            print("Input tidak valid. Silakan masukkan angka atau 'done' untuk selesai.")  
  
print("List angka yang dimasukkan:", deretan_angka)  
  
if deretan_angka:  
    print("Nilai maksimum:", max(deretan_angka))  
    print("Nilai minimum:", min(deretan_angka))  
else:  
    print("List kosong, tidak ada nilai maksimum dan minimum.")
```


Output :

```
Masukkan angka (atau ketik 'done' untuk selesai): 2
Masukkan angka (atau ketik 'done' untuk selesai): 6
Masukkan angka (atau ketik 'done' untuk selesai): 20
Masukkan angka (atau ketik 'done' untuk selesai): 30
Masukkan angka (atau ketik 'done' untuk selesai): done
List angka yang dimasukkan: [2, 6, 20, 30]
Nilai maksimum: 30
Nilai minimum: 2
```

Penjelasan :

Pada program diatas saya menggunakan loop while True Dimana program akan terus berjalan hingga pengguna mengetik 'done' untuk berhenti.

Fungsi lower saya gunakan untuk membuat inputan done pengguna sebagai huruf kecil(case-insensitive).

Fungsi break saya gunakan untuk menghentikan while loop Ketika pengguna mengetikkan done.

Fungsi isdigit saya gunakan untuk memeriksa apakah inputan adalah angka (bukan teks atau symbol).Jika inputan bukan angka maka akan menampilkan pesan 'input tidak valid'

deretan_angka.append(int(input_angka)) artinya jika input_angka adalah angka, maka program akan mengubah ke tipe data integer (int(input_angka)) dan menambahkan angka tersebut ke dalam daftar deretan_angka.

Dengan fungsi min() dan max() program diatas kemudian akan menampilkan nilai terbesar dan terkecil dari semua inputan angka yang telah dimasukkan oleh user.

Soal 9.3

```
print("\n\n==== Kata Unik =====\n")
teks = []
with open("berita.txt", "r", encoding="utf8") as file:
    for line in file:
        if line.startswith("=" or " "):
            continue
        else:
            a = line.split(" " or "\n")
            print(a)
```

Output :

```
==== Kata Unik =====

['SERANG', 'KOMPAS.com', '-', 'Pemberlakuan', 'aturan', 'ganjil', 'genap', '(gag)', 'bagi', 'kendaraan', 'bermotor', 'di', 'wilayah', 'Tangerang', 'kan', 'diterapkan.\n']
['Pelaksanaan', 'gag', 'dalam', 'rangka', 'mengatasi', 'polusi', 'udara', 'yang', 'salah', 'satu', 'faktor', 'penyebabnya', 'dari', 'gas', 'karbon', 'kendaraan.\n']
["Kaitan", 'dengan', 'aglomerasi', 'dilakukan', 'penguatan', 'dan', 'perluasan', 'ganjil-genap', 'di', 'wilayah', 'Tangerang,', 'kata', 'Penjabat', 'r', 'Banten', 'Al', 'Muktabar', 'melalui', 'keterangan', 'tertulis', 'yang', 'diterima', 'Kompas.com,', 'Selasa', '(29/8/2023).\n']
['Al', 'Muktabar', 'mengatakan', 'pemberlakuan', 'ganjil', 'genap', 'akan', 'diterapkan', 'di', 'ruas', 'jalan', 'yang', 'menjadi', 'akses', 'utama', 'DKI', 'Jakarta.\n']
['Namun', 'untuk', 'lokasi', 'penerapan', 'gag', 'akan', 'dikoordinasikan', 'terlebih', 'dahulu', 'dengan', 'Pemerintah', 'Kota', 'Tangerang', 'Kot
```

Penjelasan :

Program diatas akan membuka file berita.txt dalam mode baca ("r").

Kode `if line.startswith("=" or "")`:: saya gunakan untuk memeriksa apakah baris dimulai dengan tanda = atau baris kosong (""). Jika ya, maka program akan melanjutkan ke iterasi berikutnya (`continue`).

Bagian `else` akan dijalankan jika kondisi `if` tidak terpenuhi, yaitu Ketika baris tidak dimulai dengan = dan tidak kosong.

Pada kode `a = line.split(" " or "\n")` saya menggunakan metode `split()` pada objek string line untuk memecah string line berdasarkan pembatas argumen (" " or "\n"), dan hasilnya adalah daftar list dari bagian-bagian string yang telah dipisahkan.

Kemudian dengan fungsi `print(a)` program diatas akan mencetak daftar list yang berisi kata-kata yang telah dipisahkan berdasarkan spasi.

Link Github:

<https://github.com/Josephinemrc/Tugas9-PrakAlpro.git>