



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>71230982</b>
<b>Nama Lengkap</b>	<b>Rainie Fanita Chrisabel Hadisantoso</b>
<b>Minggu ke / Materi</b>	<b>09 / Tipe Data List</b>

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

## MATERI : Percabangan dan Perulangan Kompleks

Tujuan dari pembelajaran Percabangan dan Perulangan Kompleks adalah: 1. Dapat mendefinisikan list serta memahami sifat-sifat list, 2. Menjalankan operasi Create, Read, Update, dan Delete pada list, 3. Mengaplikasikan fungsi-fungsi Python untuk list, 4. Menggunakan list sebagai salah satu parameter fungsi.

### Sifat-sifat List

List adalah data-data individu yang menjadi satu dalam sebuah variabel. List dan String hampir mirip, keduanya sama-sama menyimpan data-data individu dalam sebuah variabel dan dapat diakses melalui indeks. Perbedaan utama antara string dan list adalah: String merupakan rangkaian data karakter-karakter. Sedangkan list dapat berisi data berupa karakter, integer, float dan lain-lain. Selain itu, list juga bisa menampung list-list di dalamnya. List dituliskan dengan tanda [].

```
nilai_kelas = [34,56,98,0,89,87,100]
nama_teman = ["Galih","Jose","Amel","Ria","Mekel"]
data_campuran = ["Python",3.11,True]
nested_list = [0,100,[34,45,62],23,[65,98,{1,45},98]]
```

Perbedaan lain antara string dan list adalah, list bersifat mutable dan string bersifat immutable. Mutable adalah data dapat diubah secara langsung. Sehingga string bersifat tetap, sedangkan list dapat diubah-ubah dengan program.

```
list = ["r","a","i","n"]
list[0]= "R"
#sekarang list berisi ["R","a","i","n"]

string = "rain"
string[0] = "R"
#program akan menampilkan error
```

Perbedaan terakhir, string yang berisi data karakter sama, jika keduanya dibandingkan maka akan dianggap benar karena program mengacu pada data yang sama. Sedangkan jika dua list dengan isi yang sama dibandingkan maka hasilnya adalah false. Hal tersebut disebabkan karena keduanya mengacu pada data yang berbeda.

```
string1 = "rain"
string2 = "rain"
string1 is string2 #True

list1 = ["r","a","i","n"]
list2 = ["r","a","i","n"]
list1 is list2 #False
```

### Operasi Isi List

Ada banyak operasi yang dapat digunakan untuk mengakses sebuah list, berikut beberapa contoh operasi-operasi yang paling umum digunakan untuk sekarang.

1. Operasi tambah (+):

Tanda operasi tambah (+) dapat digunakan untuk menambahkan atau menggabungkan 2 list atau lebih menjadi sebuah list.

```
bil1 = [1,2,3,4,5]
bil2 = [6,7,8,9,10]
total = bil1 + bil2
print(total)
#output : [1,2,3,4,5,6,7,8,9,10]
```

2. Operasi perkalian (\*):

List dapat diolah dengan perkalian, konsepnya \* digunakan untuk mengkalikan atau mengulang data-data dalam sebuah list. Jadi data dalam list akan dikalikan berdasarkan pengali.

```
bil1 = [1,2,3,4,5]
total = bil1 * 3
print(total)
#output : [1,2,3,4,5,1,2,3,4,5,1,2,3,4,5]
```

3. Indeks:

Indeks digunakan untuk mengakses elemen dalam sebuah list. Seperti string, indeks list berawal dari 0 hingga panjang isi list dikurangi 1.

```
nama_teman = ["Blob", "Keran", "Dundun", "Susq"]
nama_teman[0] # "Blob"
nama_teman[3] # "Susq"
```

Indeks negatif juga dapat digunakan untuk mengakses list. Indeks negatif adalah indeks yang ditandai negatif untuk mengakses data list dari urutan paling belakang. Indeks negatif diawali dari -1 hingga – jumlah data dalam list.

```
nama_teman[-2] # "Dundun"
nama_teman[-3] # "Keran"
```

Ketika sebuah indeks tidak ada dalam sebuah list maka program akan menampilkan error. Seperti contoh dibawah, data list “nama\_teman” hanya berisi 4 data jadi, indkes biasa = 0 hingga 3 dan indeks negatif = -1 hingga -4.

```
nama_teman[4] #error
nama_teman[-5] #error
```

4. Banyak elemen indeks [:]:

Pengaksesan list menggunakan indke sjuga dapat menggunakan banyak elemen dengan [:], dengan format [indeks awal : indeks berhenti].

```
nama_teman = ["Blob", "Keran", "Dundun", "Susq"]
nama_teman[:] # ["Blob", "Keran", "Dundun", "Susq"]
nama_teman[1:3] # ["Keran", "Dundun"]
nama_teman[1:] # ["Keran", "Dundun", "Susq"]
nama_teman[:3] # ["Blob", "Keran", "Dundun"]
```

Mengakses banyak elemen juga dapat digunakan untuk mengganti elemen dalam sebuah list. Dengan ini kita mengganti dengan cara memberikan list data pengganti.

```
nama_teman[1:3] = ["Suv", "Glycerin"]
print(nama_teman)
#output : ["Blob", "Suv", "Glycerin", "Susq"]
```

## Metode dan Fungsi untuk List

Python memiliki banyak metode bawaan yang dapat digunakan untuk mengolah data dalam sebuah list. Berikut beberapa metode yang dapat digunakan untuk menambahkan data dalam sebuah list:

### 1. Append():

Append digunakan untuk menambahkan elemen baru pada bagian akhir list. Semua data yang ditambahkan akan dianggap sebagai satu kesatuan.

```
data = ["a", "b", "c", "d", "e"]
data.append("f")
#output: ["a", "b", "c", "d", "e", "f"]
data.append(["g", "h"])
#output: ["a", "b", "c", "d", "e", "f", ["g", "h"]]
```

### 2. Extend():

Extend digunakan untuk menambahkan tiap elemen baru yang dianggap sebagai data individual pada bagian akhir list. Sehingga jika yang diextend adalah list, list tersebut tidak dimasukkan sebagai satu kesatuan, namun elemen dalam list dipisahkan menjadi individual baru ditambahkan pada bagian akhir list utama.

```
data = ["a", "b", "c", "d", "e"]
data.extend("f")
#output: ["a", "b", "c", "d", "e", "f"]
data.extend(["g", "h"])
#output: ["a", "b", "c", "d", "e", "f", "g", "h"]
```

Berikut beberapa metode yang digunakan untuk menghilangkan data dalam sebuah list:

### 1. Pop():

Menghapus elemen berdasarkan indeks elemen serta mengambil nilai element yang dihapus. Jadi indeks elemen sudah harus diketahui untuk menggunakan pop untuk menghilangkan elemen tersebut.

```
data = ["a", "b", "c", "d", "e"]
data.pop(2)
#output: "c"
data
#output: ["a", "b", "d", "e"]
```

### 2. Del():

Hanya menghapus elemen berdasarkan indeks tanpa mengambil data elemen yang dihapus.

```
data = ["a", "b", "c", "d", "e"]
del data[2]
data
#output: ["a", "b", "d", "e"]
```

### 3. Remove():

Menghilangkan elemen dengan nilai elemen yang ingin dihapus.

```
data = ["a", "b", "c", "d", "e"]
data.remove("c")
data
#output: ["a", "b", "d", "e"]
```

Berikut beberapa metode yang dapat digunakan untuk mendata elemen-elemen sebuah list.

1. Len(): Menghitung jumlah data dalam list
2. Max(): Mencari data terbesar dalam sebuah list
3. Min(): Mencari data terkecil dalam sebuah list
4. Sum(): Menjumlahkan data-data interger dalam sebuah list
5. Sort(): Mengurutkan data-data dalam sebuah list dari terkecil ke terbesar

```
data = [1, 2, 4, 3, 5]
data.sort()
data
#output: [1, 2, 3, 4, 5]
```

### 6. Reverse(): Membalik urutan data-data dalam sebuah list.

```
data = [1, 2, 4, 3, 5]
data.sort()
#output: [1, 2, 3, 4, 5]
data.reverse()
#output: [5, 4, 3, 2, 1]
```

Beberapa metode tersebut adalah metode dengan tipe kosong atau void. Sehingga ketika dimodifikasi dengan metode tersebut, list akan bernilai kosong atau None. Sebagai contoh `nama_teman = nama_teman.sort()`, maka hasilnya akan menjadi None dan semua elemen dalam list akan hilang. Sehingga perlu diingat untuk langsung menggunakan `sort()` tanpa menggunakan variabel penampung.

```
data = [1, 2, 4, 3, 5]
data = data.sort()
data
#output: []
```

## Perbedaan List dan String

String adalah deretan karakter berindeks, sedangkan list deretan data tidak hanya data karakter berindeks. Meskipun berbeda string dapat diubah menjadi list, dan demikian juga sebaliknya. Berikut fungsi-fungsi untuk memproses list dan string.

String dapat diubah menjadi list berisi karakter-karakter dengan menggunakan perintah `list()`.

```
string = "rain"
list = list(string)
print(list)
#output: ["r", "a", "i", "n"]
```

Dapat juga digunakan fungsi `split()` untuk memisahkan karakter berdasarkan karakter tertentu dalam sebuah deret string. Pemisah karakter default adalah spasi atau " ". Tapi dapat digunakan karakter lain untuk memisahkan string.

```
string = "rainie fanita chrisabel hadisantoso"
list = string.split()
print(list)
#output: ["rainie", "fanita", "chrisabel", "hadisantoso"]
```

List dapat diubah menjadi string dengan fungsi `join()`. Join menggabungkan elemen list menjadi kesatuan dengan penyambung spasi " ". Namun dapat digunakan penyambung karakter lainnya.

```
list = ["rainie", "fanita", "chrisabel", "hadisantoso"]
string = " ".join(list)
print(string)
#output: "rainie fanita chrisabel hadisantoso"
```

### List sebagai Parameter Fungsi

Seperti tipe data lainnya, list juga dapat digunakan sebagai parameter dalam sebuah fungsi. Namun harus diperhatikan operasi apa saja yang memodifikasi data utama dan operasi apa saja yang menciptakan data baru. Contohnya fungsi `append()` memodifikasi data list, sedangkan operasi `+` membuat variabel baru. Fungsi seperti `append()` tidak perlu didefinisikan dalam variabel baru, sedangkan operasi seperti `+` perlu didefinisikan dalam sebuah variabel baru.

Append:

```
data1 = [1,2,3]
data2 = data1.append(4)
print(data1)    #[1,2,3,4]
print(data2)    #None
```

Tambah:

```
data3 = data1 + [4]
print(data3)    #[1,2,3,4]
data2 is data3  #False
```

## LATIHAN MANDIRI 9

### SOAL 9.1

Membuat program untuk menampilkan 3 data nilai terbaik dalam sebuah list.

a. Source code

```
nilai = [20,80,100,29,1,99,20,40,19,1,0]
nilai.sort()
nilai.reverse()
for i in range (3):
    print(nilai[i])
```

b. Output

```
PS C:\Users\asus\Downloads\File> python -u "c:\Users\asus\Downloads\File
erFile.py"
100
99
80
```

c. Penjelasan

Pertama, didefinisikan list “nilai” berisi kumpulan nilai dengan range dari 0 hingga 100. Lalu list “nilai” di urutkan datanya menggunakan perintah sort(). Sort() mengurutkan data dari kecil terbesar, sehingga digunakan everse() untuk membalik urutan data list menjadi dari besar ke kecil. Kemudian digunakan perulangan 3 kali untuk menampilkan data list dengan indeks dari 0 hingga 2. Sehingga terambil 3 nilai terbaik dalam list “nilai”.

### SOAL 9.2

Meminta input int dari user. Input dimasukkan dalam list. Kemudian ketika user menginput “done” akan menampilkan nilai max dan min dari list. Gunakan min() dan max()

a. Source code

```
nilai = []
print("Masukkan nilai-nilai (kalau sudah ketik 'done')")
masuk = False
i = 1
while masuk == False:
    hasil = input(f"Nilai {i}: ")
    i += 1
    try:
        hasil = int(hasil)
        nilai.append(hasil)
    except:
        if hasil.lower() == "done":
            masuk = True
```

```

else:
    print("Input salah!")

print(f"Nilai minimum: {min(nilai)}")
print(f"Nilai maximum: {max(nilai)}")

```

b. Output

```

PS C:\Users\asus\Downloads\File> python -u "c:\Users\asus\Downloads\File\erFile.py"
Masukkan nilai-nilai (kalau sudah ketik 'done')
Nilai 1: 24
Nilai 2: 1
Nilai 3: 35
Nilai 4: 90
Nilai 5: 98
Nilai 6: 68
Nilai 7: 79
Nilai 8: 0
Nilai 9: 100
Nilai 10: 45
Nilai 11: done
Nilai minimum: 0
Nilai maximum: 100

```

c. Penjelasan

Siapkan variabel “nilai” sebagai list. Ditampilkan perintah untuk memasukan data nilai-nilai. Didefinisikan variabel “masuk” berisi False, variabel ini akan digunakan untuk menentukan kapan berhentinya perulangan. Dan variabel i bernilai 1 untuk memberi nomor nilai yang diinput. Menggunakan perulangan while selama masuk masih bernilai False.

- Dalam perulangan jika yang diinput pengguna adalah data interger maka data akan dimasukkan kedalam list “nilai” menggunakan append().
- Jika data bukan int maka akan dicek jika yang diinput pengguna adalah kata “done” (input pengguna di lower() agar sesuai kriteria if ), nilai masuk akan menjadi True. Karena nilai masuk true maka perulangan akan berhenti dan program akan berhenti dan menampilkan nilai terkecil dan nilai terbesar.
- Jika yang dimasukkan bukan “done” maka program akan menampilkan “Input salah!” dan tetap meneruskan perulangan.

### SOAL 9.3

Cari artikel berita, simpan dalam bentuk .txt. Pecah berita jadi kata-kata dengan split(), dan disimpan dalam list kata.

a. Source code



```
kata2 = []
with open("berita.txt", "r") as f:
    for w in f:
        kata = w.split()
        kata2.extend(kata)

print(kata2)
```

b. File

Cari artikel berita, simpan dalam bentuk .txt. Pecah berita jadi kata-kata dengan split(), dan disimpan dalam list kata.

c. Output

```
PS C:\Users\asus\Downloads\File> python -u "c:\Users\asus\Downloads\File\Latihan_9_71230982\tempCodeRunn
erFile.py"
['Cari', 'artikel', 'berita,', 'simpan', 'dalam', 'bentuk', '.txt.', 'Pecah', 'berita', 'jadi', 'kata-ka
ta', 'dengan', 'split()', 'dan', 'disimpan', 'dalam', 'list', 'kata.']
```

d. Penjelasan

Didefinisikan variabel kata2 sebagai list. Menggunakan perintah with open untuk membuka file berita.txt, dengan mode read atau membaca, karena program hanya perlu melihat isi dan tidak mengubahnya. Isi berita.txt disimpan dengan nama f oleh program. Kemudian dalam perulangan untuk setiap kata dalam file:

- Variabel kata untuk menyimpan kata yang di pisahkan dari file dengan split()
- Kemudian kata disimpan dalam kata2 dengan extend()

Kemudian isi berita ditampilkan dengan perintah print()

Link Github :

[https://github.com/rainiefch/Praktikum-Algoritma-Pemrograman\\_71230982/tree/fc3c47a245a163bbd54e99158585a9c3599e24ed/Latihan%209](https://github.com/rainiefch/Praktikum-Algoritma-Pemrograman_71230982/tree/fc3c47a245a163bbd54e99158585a9c3599e24ed/Latihan%209)