



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	<I71230972>
Nama Lengkap	<Oktavian Christ Putranto>
Minggu ke / Materi	11 / 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

Secara sederhana list dapat diartikan sebagai Kumpulan data/value dalam satu list yang dapat diakses satu persatu, dan setiap value tersebut dapat memiliki tipe yang berbeda (bisa berisi Boolean, string, integer, dll) di tiap value nya bahkan hingga menampung list lain didalamnya (disebut sebagai list didalam list).

Bentuk penulisan list memakai symbol [] contoh dari list :

```
list_1=[1, 2, 3, 4]
list_2=["ayam", "sapi", "babi", "Domba", "ayam"]
list_3=[9, True, "suiiiiii", 14, 14, "DHUARRR"]
list_4=["ayam", 12, 14, [145, "domba"], 12, "SUII"]
```

dari ketiga list dapat kita perhatikan bahwa terdapat banyak value di dalam satu list dan dapat berisi value yang memiliki tipe data yang berbeda beda seperti dalam list_3 maupun list_4, dalam list_4 pun dapat kita lihat bahwa di dalam list kita dapat memasukan list lain di dalam nya yang dihitung menjadi satu value list dalam di index 3.

Beberapa hal penting yang harus diperhatikan dalam list:

- Indeks list selalu dimulai dari 0
- Apabila indeks yang kita miinta dari list tidak ada maka akan terjadi eror value
- Isi list dapat diubah, ditambah, maupun digabungkan dengan list lain nya
- List dapat menjadi sebuah parameter fungsi
- List dapat di isi beragam tipe data

Oprasi List

Tiap tiap value di dalam list dapat diakses masing masing melalui indeksnya, selain itu list juga dapat dioprasikan maupun di urutkan memakai beragam built in untuk list, dan kita akan mempelajari beberapa diantaranya, untuk yang pertama kita akan membahas cara mengakses list tersebut mulai dari indeks nya.

Cara mengakses list:

```
list_1=[1, 2, 3, 4]
list_2=["ayam","sapi","babi","Domba","ayam"]
list_3=[9,True,"suiiiiii",14,14,"DHUARRR"]
list_4=["ayam",12,14,[145,"domba"],12,"SUII"]
print (list_4[3][1])
print (list_3[0])
print (list_4[2])
print (list_1[1::])
print (list_2[-1])
print (list_2[0:3])
```

Output :

```
PS C:\Users\ASUS VivoBook\Downloads\list_Python> & "C:/Users/ASUS VivoBook/Downloads/list_Python/extend.py"
domba
9
14
[2, 3, 4]
ayam
['ayam', 'sapi', 'babi']
PS C:\Users\ASUS VivoBook\Downloads\list_Python>
```

Dari yang kita ketahui indeks list dimulai dari 0 oleh karena itu kita harus memperhatikan indeks dari listnya untuk dapat menampilkan atau memakai value yang sesuai agar tidak terjadi error seperti di bawah ini.

```
list_1=[1, 2, 3, 4]
list_2=["ayam","sapi","babi","Domba","ayam"]
list_3=[9,True,"suiiiiii",14,14,"DHUARRR"]
list_4=["ayam",12,14,[145,"domba"],12,"SUII"]
print (list_1[4])
```

output :

```
PS C:\Users\ASUS VivoBook\Downloads\list_Python> & "C:/Users/ASUS VivoBook/Documents/kuliah/msys64/mingw64/python.exe" "c:/Users/ASUS VivoBook/Downloads/list_Python/extend.py"
Traceback (most recent call last):
  File "c:\Users\ASUS VivoBook\Downloads\list_Python\extend.py", line 5, in <module>
    print (list_1[4])
IndexError: list index out of range
```

Cara menggabungkan list :

Cara penjumlahan (+):

```
bil_1=[1,2,3,4]
bil_2=[1,2,3,4]
bil_3=bil_1+bil_2
print (bil_3)
print(bil_3*3)
```

Output :

```
thon.exe "c:/Users/ASUS_VivoBook/Downloads/list_Python/gabung.py"
[1, 2, 3, 4, 1, 2, 3, 4]
[1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]
```

Selain berlaku disatukan dengan symbol +, list juga bisa di tampilkan berkali kali memakai perkalian

Cara extend :

```
list_1=[1, 2, 3, 4]
list_2=["ayam", "sapi", "babi", "Domba", "ayam"]
list_3=[9, True, "suiiiiii", 14, 14, "DHUARRR"]
list_4=["ayam", 12, 14, [145, "domba"], 12, "SUII"]

list_1.extend(list_2)
list_1.extend(list_3)

print (list_1)
```

Output :

```
thon.exe "c:/Users/ASUS_VivoBook/Downloads/list_Python/extend.py"
[1, 2, 3, 4, 'ayam', 'sapi', 'babi', 'Domba', 'ayam', 9, True, 'suiiiiii', 14, 14, 'DHUARRR']
```

Cara extend hamper sama dengan cara (+) namun letak perbedaan nya ialah ketika memakai + maka kita perlu menambah variable baru untuk menampung hasil penjumlahan nya sedangkan extend langsung mengubah nilai list_1

append :

```
list_1 =[1,5,2]
list_1.append("dhuar")
list_1.append(34)
list_1.append(145)
list_1.append([123,1,4,5,'SUII'])
list_1.append(True)

print (list_1)
```

Output :

```
● thon.exe" "c:/Users/ASUS VivoBook/Downloads/list_Python/baru.py"
[1, 5, 2, 'dhuar', 34, 145, [123, 1, 4, 5, 'SUII'], True]
```

Append akan menambahkan 1 value di akhir list value tersebut dapat berupa data apapun bahkan list

Error ketika dimasukan lebih dari 1 value :

```
list_1 =[1,5,2]
list_1.append("dhuar")
list_1.append(34)
list_1.append(145)
list_1.append([123,1,4,5,'SUII'])
list_1.append(True)
list_1.append(1,2,3)

print (list_1)
```

Output :

```
thon.exe" "c:/Users/ASUS VivoBook/Downloads/list_Python/baru.py"
Traceback (most recent call last):
  File "c:/Users/ASUS VivoBook/Downloads/list_Python/baru.py", line 7, in <module>
    list_1.append(1,2,3)
TypeError: list.append() takes exactly one argument (3 given)
```

Ketika dua list di append :

```
list_1 =[1,5,2]
list_2 =["ayam",12,14]
list_1.append(list_2)
print (list_1)
thon.exe" "c:/Users/ASUS VivoBook/Down
[1, 5, 2, ['ayam', 12, 14]]
```

Beberapa cara untuk mengolah list :

- Sorting list :

```
list1=[1,5,2,3,90,123,12,1456,12]
list1.sort()
print (list1)

list2=["apple","zombie","gooa","goob","k","f"]
list2.sort()
print (list2)
```

Output :

[1, 2, 3, 5, 12, 12, 90, 123, 1456]

['apple', 'f', 'gooa', 'goob', 'k', 'zombie']

Ketika perintah sort diberi maka akan langsung mengurutkan dari nilai terkecil dan untuk data string maka sesuai alphabet

```
list1=[1,5,2,3,90,123,12,1456,12]
list1.sort(reverse=True)
print (list1)
```

Output :

[1456, 123, 90, 12, 12, 5, 3, 2, 1]

sort juga dapat membalik urutan agar mengurutkan dari nilai terbesar ke kecil dengan mengaktifkan reverse menjadi true

- POP :

```
list1=[0,1,2,3,4,5]
terpop=list1.pop(4)
print (list1)
print (terpop)
```

output :

[0, 1, 2, 3, 5]

4

Fungsi pop berguna untuk mengambil salah satu value dari dalam list dan dapat di recall Kembali

BAGIAN 2: LATIHAN MANDIRI (60%)

Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

SOAL 1

Mencari 3 nilai terbaik dari atas

Input :

```
#mencari 3 nilai terbaik dari yang paling tinggi
def urutkan(list_angka):
    list_angka.sort()
    ilang=[]
    for i in list_angka:
        if i not in ilang:
            ilang.append(i)
    balik=ilang[::-1]
    return(balik[0:3])

print(urutkan([1,2,3,4,5,5,5,5,5]))
```

Output :

[5, 4, 3]

Penjelasan:

Pertama tama saya membuat function dengan nama urutkan yang dijalanyang didalamnya Dimana pertama tama saya mengurutkan list_angka memakai code list_angka.sort()

Kemudian saya membuat list kosong dengan nama ilang

Untuk setiap angka yang ke double maka akan dihilangkan dengan cara memasukan tiap elemen di list_angka ke ilang ketika belum ada elemen tersebut, jadi tidak akan ada double angka di dalam list ilang setelah itu list dibalik dengan cara [::-1]

Setelah selesai di proses dan di reverse maka kita hanya perlu menampilkan 3 dari bawah dengan [0:3] yang merupakan elemen ke 0,1,2 dari list.

SOAL 2

Meminta input user hingga menulis done kemudian menampilkan nilai minimal dan maksimalnya

Input :

```
def input_user(a):  
    list_angka=[]  
    while True:  
        angka=(input("masukan angka (ketik done ketika selesai): "))  
        if angka.lower() == "done":  
            break  
        else:  
            try:  
                angka=int(angka)  
                list_angka.append(angka)  
            except:  
                print ("masukan angka")  
  
        if len(list_angka)<2:  
            return ("jumlah inputan tidak sesuai ketentuan(masukan lebih dari 2 angka)")  
        return (f"{max(list_angka)} adalah nilai maksimal\n{min(list_angka)} adalah nilai minimal")  
print(input_user(input))
```

Output :

masukan angka (ketik done ketika selesai): 1
masukan angka (ketik done ketika selesai): 5
masukan angka (ketik done ketika selesai): 6
masukan angka (ketik done ketika selesai): 7
masukan angka (ketik done ketika selesai): 8
masukan angka (ketik done ketika selesai): 9
masukan angka (ketik done ketika selesai): apa
masukan angka
masukan angka (ketik done ketika selesai): done
9 adalah nilai maksimal
1 adalah nilai minimal

Jika salah inputan :

masukan angka (ketik done ketika selesai): 1

masukan angka (ketik done ketika selesai): done

jumlah inputan tidak sesuai ketentuan(masukan lebih dari 2 angka)

penjelasan :

pertama tama code akan menjalankan fungsi `input_user(input)` yang kemudian fungsi akan langsung masuk ke perulangan Dimana ketika masih belum memenuhi kondisi maka akan terus menerus meminta inputan hingga `input user == "done"`

setelah itu inputan diberi try and except untuk menangkal kesalahan input karena input haruslah angka, setelah memperoleh angka maka akan langsung di append/dimasukan kedalam `list_angka`

apabila banyak elemen di list angka < 2 maka akan menampilkan output `jumlah inputan tidak sesuai ketentuan(masukan lebih dari 2 angka)`

jika sudah sesuai ketentuan barulah menampilkan nilai maximum dan minimum nya memakai code

```
return (f"{max(list_angka)} adalah nilai maksimal\n{min(list_angka)} adalah nilai minimal")
```

SOAL 2

Membaca file txt dan menampilkan kata uniknya !

Input :

```
filename = input("nama file: ")
handle = open(filename,"r")
baca=handle.read()
pisah=baca.split(" ")
unik=[]
for i in pisah:
    if i != "\n":
        unik.append(i)
print("===== Isi Berita =====")
print (baca)
print()
print()
print ("===== Kata Unik =====")
print (unik)
```

Output :

nama file: berita.txt

===== Isi Berita =====

Contoh 6.4 Anda diminta untuk membuat suatu deret dengan tampilan pada gambar 6.10.

Pembahasan: Untuk dapat membuat kode program seperti di atas, maka logikanya adalah sebagai berikut:

Karena n = misalnya 10, maka n akan menjadi outer loop (baris). $i=n$ turun smp dengan 1.

Untuk inner loop kita harus menuliskan terlebih dahulu spasi kosong sebanyak $n-i-1$. Setelah itu digabungkan dengan menampilkan bintang (*) sebanyak sisanya ($1 \leq n-i-1$)

Untuk setiap loop pada inner, setelah habis maka lakukan enter / ganti baris dari outer loop.

Kode program dapat dibuat sebagai berikut:

===== Kata Unik =====

```
['Contoh', '6.4', 'Anda', 'diminta', 'untuk', 'membuat', 'suatu', 'deret', 'dengan', 'tampilan', 'pada',  
'gambar', '6.10.\nPembahasan:', 'Untuk', 'dapat', 'membuat', 'kode', 'program', 'seperti', 'di', 'atas,',  
'maka', 'logikanya', 'adalah\nsebagai', 'berikut:\nKarena', 'n', '=', 'misalnya', '10,', 'maka', 'n', 'akan',  
'menjadi', 'outer', 'loop', '(baris).', 'i=n', 'turun', 'smp', 'dengan', '1.\nUntuk', 'inner', 'loop', 'kita', 'harus',  
'menuliskan', 'terlebih', 'dahulu', 'spasi', 'kosong', 'sebanyak', 'n-i-1.', 'Setelah\nitu', 'digabungkan',  
'dengan', 'menampilkan', 'bintang', '(*)', 'sebanyak', 'sisanya', '(1', 's/d', 'n-i-1)\nUntuk', 'setiap', 'loop',  
'pada', 'inner,', 'setelah', 'habis', 'maka', 'lakukan', 'enter', '/', 'ganti', 'baris', 'dari', 'outer', 'loop.\nKode',  
'program', 'dapat', 'dibuat', 'sebagai', 'berikut:']
```

Penjelasan :

Pertama tama code akan meminta inputan user mengenai nama file yang ingin dibuka

Setelah itu barulah di open"r" untuk membaca isi file

```
baca=handle.read()  
pisah=baca.split(" ")  
unik=[]
```

3 code diatas akan membaca file dan membaginya per space(" ") dan membuat list kosong dengan nama unik

Untuk tiap elemen i pada pisah maka akan di masukan ke list unik

Setelah itu akan di tampilkan sebagai output memakai code :

```
print("===== Isi Berita =====")
print (baca)
print()
print()
print ("===== Kata Unik =====")
print (unik)
```