

LAPORAN PRAKTIKUM

PEMROGRAMAN KOMPUTER (PYTHON)



Disusun Oleh:
Rafli Pratama
H1101241008

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS TANJUNGPURA PONTIANAK
2024

Praktikum 5

LIST PADA PYTHON

1.1. Dasar Teori

Konsep Dasar List :

Bahasa python memiliki paket built in disebut dengan list. List ditulis dalam tanda kurung siku []. List adalah struktur data pada python yang mampu menyimpan lebih dari satu data, seperti array. Perbedaan mendasar array dan list adalah pada array harus mengimpor library array dan array hanya menampung satu jenis tipe data, serta array lebih tepat untuk mengolah data yang besar. Ada beberapa hal yang dapat dilakukan dengan semua jenis list. Operasi ini meliputi pengindeksan, pengiris, penambahan, perbanyak, dan pengecekan keanggotaan. List (daftar) adalah suatu koleksi berurut beranggotakan nol atau lebih rujukan ke obyek data Python. List ditulis sebagai nilai-nilai terpisahkan koma di dalam kurung siku. List bersifat heterogen, artinya obyek-obyek data di dalamnya tidak harus berasal dari kelas yang sama dan koleksi tersebut dapat berikan ke suatu variabel. List kosong diwakili oleh [].

1.1.1 Cara Membuat List

List adalah tipe data yang paling serbaguna yang tersedia dalam bahasa Python, yang dapat ditulis sebagai daftar nilai yang dipisahkan koma (item) antara tanda kurung siku. Hal penting tentang list adalah item dalam list tidak boleh sama jenisnya. Membuat list sangat sederhana, tinggal memasukkan berbagai nilai yang dipisahkan koma di antara tanda kurung siku. Dibawah ini adalah contoh sederhana pembuatan list dalam bahasa Python.

#Contoh sederhana pembuatan list pada pemograman python

```
list1 = ['Jeruk', 'Mangga', 34, 56]
list2 = [11,22,33,44,55]
list3= ['a3', 'a2', True, False]
```

```
print(list1)
print(list2)
print(list3)
```

✓ 0.0s

```
['Jeruk', 'Mangga', 34, 56]
[11, 22, 33, 44, 55]
['a3', 'a2', True, False]
```

1.1.2 Akses nilai dari List dan memotong list

Mengakses nilai dalam list python adalah menggunakan tanda kurung siku untuk mengiris beserta indeks atau indeks untuk mendapatkan nilai yang tersedia pada indeks tersebut. Berikut adalah contoh cara mengakses nilai di dalam list python :

```
#Cara mengakses nilai di dalam list Python
list1 = ["Jeruk", 'mangga', 34, 54]
list2 = [11,22,33,44,55]
list3= ['a3', 'a2', True, False]

print(f'list1[0] : {list1[0]}')
print(f'list2[1:3] : {list2[1:3]}')
```

```
list1[0] : Jeruk
list2[1:3] : [22, 33]
```

1.1.3 Cara Menghapus isi List

Menghapus nilai di dalam list python, dapat digunakan salah satu pernyataan del jika kita tahu persis elemen yang dihapus. Adapun metode yang lain menggunakan pop dan remove.

```
#Contoh cara menghapus nilai pada list python
#menggunakan del
list = ['Jeruk', 'Mangga', 43, 54]

print(f'Kondisi Awal : {list}')
del list[2]
print(f'Kondisi Akhir : {list}')
```

```
Kondisi Awal : ['Jeruk', 'Mangga', 43, 54]
Kondisi Akhir : ['Jeruk', 'Mangga', 54]
```

```
#Contoh cara menghapus nilai pada list python
#menggunakan pop()-> menghapus index yg sesuai
list = ['Jeruk', 'Mangga', 43, 54]
```

```
print(f'Kondisi Awal : {list}')
list.pop(1)
print(f'Kondisi Akhir : {list}')
```

```
Kondisi Awal : ['Jeruk', 'Mangga', 43, 54]
Kondisi Akhir : ['Jeruk', 43, 54]
```

```
#Contoh cara menghapus nilai pada list python
#menggunakan remove() -> menghapus nilai yg sesuai
list = ['Jeruk', 'Mangga', 43, 54]
```

```
print(f'Kondisi Awal : {list}')
list.remove('Jeruk')
print(f'Kondisi Akhir : {list}')
```

```
Kondisi Awal : ['Jeruk', 'Mangga', 43, 54]
Kondisi Akhir : ['Mangga', 43, 54]
```

```

#perbedaan pop dan remove
#pop

data = []
a = int(input("Masukkan jumlah data : "))

for i in range(a):
    inp = int(input('Masukkan Data : '))
    data.append(inp)

print(data)

print('Berikut adalah proses penghapusan data')
for i in range (a):
    data.pop(0)
    print(f'Data yang hilang : {data}')
    if data == []:
        print('Tidak ada data yang bisa di hapus lagi karena kosong')

```

[3, 6, 5, 1]

Berikut adalah proses penghapusan data

Data yang hilang : [6, 5, 1]

Data yang hilang : [5, 1]

Data yang hilang : [1]

Data yang hilang : []

Tidak ada data yang bisa di hapus lagi karena kosong

```

#perbedaan pop dan remove
#del

data = []
a = int(input("Masukkan jumlah data : "))

for i in range(a):
    inp = int(input('Masukkan Data : '))
    data.append(inp)

print(data)

print('Berikut adalah proses penghapusan data')
for i in range (a):
    del data[0]
    print(f'Data yang hilang : {data}')
    if data == []:
        print('Tidak ada data yang bisa di hapus lagi karena kosong')

```

```

[5, 7, 9, 3]
Berikut adalah proses penghapusan data
Data yang hilang : [7, 9, 3]
Data yang hilang : [9, 3]
Data yang hilang : [3]
Data yang hilang : []
Tidak ada data yang bisa di hapus lagi karena kosong

```

1.1.4 List multi dimensi

List dapat juga memiliki lebih dari satu dimensi atau disebut dengan multi dimensi. List multi dimensi biasanya digunakan untuk menyimpan struktur data yang kompleks seperti tabel, matriks, graph, tree, dsb.

```

# List makanan dengan 2 dimensi
list_makan = [
    ["Siomay", "Bakso", "Mie Tiaw"],
    ["Pizza", "Spaghetti", "Dimsum"],
    ["Tempe", "Tahu", "Korket"]
]
# Cara mengakses list multidimensi
# misalkan kita ingin mengambil "tahu"
print (list_makan[2][1])
print(list_makan[1])

```

✓ 0.0s

```

Tahu
['Pizza', 'Spaghetti', 'Dimsum']

```

Angka dua 2 pada kode di atas, menunjukan indeks list baris yang akan kita akses.

Sedangkan angka 1 adalah list kolom yang akan diakses.

Hasil outputnya: “ Tahu “

Untuk menampilkan semua isi dalam list multi dimensi dapat menggunakan perulangan bersarang.

```
# List makanan dengan 2 dimensi
list_makan = [
    ["Siomay", "Bakso", "Mie Tiaw"],
    ["Pizza", "Spaghetti", "Dimsum"],
    ["Tempe", "Tahu", "Korket"]
]
# Cara menampilkan seluruh list multidimensi

for i in list_makan:
    for j in i:
        print(j)
```

✓ 0.0s

Siomay
Bakso
Mie Tiaw
Pizza
Spaghetti
Dimsum
Tempe
Tahu
Korket

1.1.5 Operasi pada List

Metode-metode yang disediakan oleh List di dalam Python dapat dilihat seperti berikut:

Nama Metode	Cara Penggunaan	Penjelasan
append	alist.append(item)	Menambahkan suatu item baru ke akhir list
insert	alist.insert(i,item)	Menyisipkan suatu item ke dalam list pada posisi ke-i
pop	alist.pop()	Menghapus & mengembalikan item terakhir dari dalam list
pop	alist.pop(i)	Menghapus dan mengembalikan item ke-i dari dalam list
sort	alist.sort()	Mengubah suatu list agar terurut
reverse	alist.reverse()	Mengubah suatu list dalam urutan terbalik
del	del alist[i]	Menghapus item pada posisi ke-i
index	alist.index(item)	Mengembalikan index dari kemunculan pertama dari item
count	alist.count(item)	Mengembalikan jumlah kehadiran dari item
remove	alist.remove(item)	Menghapus kemunculan pertama dari item

a) Append

Berikut adalah operasi insert untuk menambahkan data pada item terakhir.

list.append(item)

```
#append
list = [1, 2, 3, 4, 5]
print(list)
list.append(6)
print(list)
```

✓ 0.0s

[1, 2, 3, 4, 5]

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

b) Insert

Berikut adalah operasi insert untuk menambahkan data pada posisi item yang ditentukan.

list.insert(index, item)

```
#insert

list = [1,2,3,4,6]
print(list)
list.insert(4,5)
print(list)
```

✓ 0.0s

[1, 2, 3, 4, 6]

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

c) Pop

Berikut adalah operasi pop untuk menghilangkan item pada posisi yang ditentukan.

`list.pop(index)`

`list.pop()` -> menghapus pada index terakhir

```
# Operasi pop

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

✓ 0.0s

[1, 2, 3, 4, 5]

[1, 2, 4]

d) Sort

Berikut adalah operasi sort untuk mengurutkan nilai.

`list.sort()`

```
# Operasi sort

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

✓ 0.0s

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

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

e) Reverse

Berikut adalah operasi reverse untuk membalikkan urutan item data.

`list.reverse()`

```
# Operasi reverse

list = [1, 2, 3, 4, 5, True]
print(list)
list.reverse()
print(list)
```

✓ 0.0s

[1, 2, 3, 4, 5, True]

[True, 5, 4, 3, 2, 1]

f) Del

Method Del digunakan untuk menghapus list tersebut

```
#del
list1 = [1,2,3,4]
print(list1)
del list
print(list1)
```

⊗ 0.0s

[1, 2, 3, 4]

NameError Trace
Cell In[25], line 4
2 list1 = [1,2,3,4]
3 print(list1)
----> 4 del list
5 print(list1)

NameError: name 'list' is not defined

g) Index

Index digunakan untuk memanggil

h) Count

Untuk mengembalikan jumlah kehadiran item didalam list

```
#Count

list1 = [1, 2, 3, 4, 5]
print(list1)
list1.count(2)
print(list1)
```

✓ 0.0s

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

i) Remove

Berikut adalah operasi remove untuk menghapus data yang telah ditentukan.

list.remove(item)

```
# Operasi remove

list = [1, 2, 3, 4, 5]
print(list)
list.remove(5)
print(list)
```

✓ 0.0s

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

1.2. Tugas Praktikum

1.2.1. Soal 1

Buatlah algoritma mencari nilai kuadrat bilangan antar 1-10, dengan menggunakan array

```
data = []
for i in range (1,11):
    data.append(i)

for j in data:
    print(f'{j} x {j} = {j*j}')
```

Output :

```
1 x 1 = 1
2 x 2 = 4
3 x 3 = 9
4 x 4 = 16
5 x 5 = 25
6 x 6 = 36
7 x 7 = 49
8 x 8 = 64
9 x 9 = 81
10 x 10 = 100
```

Penjelasan :

Program akan menginisialisasi variabel data dengan list kosong lalu akan dilakukan perulangan for dengan index di mulai dari 1 dan diakhiri oleh 10, lalu di setiap perulangan tersebut index akan di append ke data dan mengisi list dari variabel data, lalu di lakukan perulangan kembali yang memanggil setiap nilai di list data dan setiap perulangan tersebut nilai akan dikalikan dengan nilai.

1.2.2 Soal 2

Terdapat sebuah array satu dimensi yang dibuat dengan int A[200], sudah ada isinya berupa nilai ujian mahasiswa. Susun algoritma untuk :

- Mencetak nilai yang terbesar, dan mencetak jumlah mahasiswa yang mendapat nilai terbesar tersebut.
- Mencetak nilai terbesar, dan mencetak berada di lokasi mana saja nilai yang terbesar tersebut berada dalam array

```

import random

data = []
for i in range (200):
    angka = random.randint(1,100)
    data.append(angka)
print(data)

data_maks = data[0]
jml_dataMaks = 0
for j in data:
    if j > data_maks:
        data_maks = j
        jml_dataMaks = 1
    elif j == data_maks:
        jml_dataMaks +=1
print(f'Nilai terbesar adalah {data_maks}')
print(f'Jumlah mahasiswa yang mendapatkan nilai {data_maks} adalah {jml_dataMaks} Orang')

data_map = []
for i in range(len(data)):
    if data[i] == data_maks:
        data_map.append(i)
print(f'yaitu pada posisi index ke- {data_map}')

```

Output :

```

[62, 100, 76, 70, 9, 27, 59, 19, 17, 93, 6, 59, 31, 9, 19, 23, 43,
Nilai terbesar adalah 100
Jumlah mahasiswa yang mendapatkan nilai 100 adalah 5 Orang
yaitu pada posisi index ke- [1, 25, 63, 81, 197]

```

Penjelasan :

Diawal program menggunakan sebuah library python yaitu random untuk melakukan fungsi random() setelah itu kita membuat variabel data yang berisi list kosong lalu kita lakukan perulangan sebanyak 200 kali yg dimana setiap perulangan tersebut data akan mengappend nilai angka yang merupakan fungsi random 1-100, setelah itu kita akan mengprint data, lalu dibuatlah variabel data_maks yaitu data indeks ke 0 dan jumlah data maks yaitu 0, setelah itu dilakukan perulangan yang memanggil semua nilai pada list data dan dilakukan pengecekan apakah ada nilai yang lebih besar daripada data indeks ke 0, jika ada maka tukar nilai data_maks, menjadi nilai list saat itu dan jumlahdatamaks = 1, dan jika nilai data saat itu sama dengan data maks maka jumlahdatamaks + 1, lalu kita print data maks dan jumlahdatamaks, setelah itu di buat lagi variabel data map, yg berisi list kosong lalu kita lakukan perulangan sebanyak jumlah nilai pada list data, jika didalam perulangan tersebut ada data index ke-i yg nilainya sama dengan data maks, maka data indeks ke-i di append ke data_map, lalu setelah perulangan selesai, print data_map

1.2.3 Soal 3

Seorang dosen ingin data-data nilai mahasiswanya dihitung dan dibuatkan programnya. Gunakan list untuk membuat program yang mempunyai ketentuan sebagai

berikut :

- a. Nama mahasiswa, nilai tugas, nilai UTS dan nilai UAS di input.
- b. Proses yang dilakukan untuk mendapatkan nilai murni dari masing-masing nilai adalah :
 - Nilai murni tugas = nilai tugas x 30%
 - Nilai murni UTS = nilai UTS x 30%
 - Nilai UAS = nilai UAS x 40%
 - Nilai akhir = Nilai murni tugas + Nilai murni UTS + Nilai murni UAS
- c. Ketentuan untuk grade nilai :
 - Nilai akhir ≥ 80 mendapat grade A
 - Nilai akhir ≥ 70 mendapat grade B
 - Nilai akhir ≥ 59 mendapat grade C
 - Nilai akhir ≥ 50 mendapat grade D
 - Nilai akhir < 50 mendapat grade E

```
data_mahasiswa = []
lagi = 'y'
jumlah = 0
while lagi == 'y' :
    for i in range(len(data_mahasiswa)):
        data_mahasiswa.pop(0)

    nama_mahasiswa = input('Masukkan nama mahasiswa : ')
    data_mahasiswa.append(nama_mahasiswa)

    mata_kuliah = input('Masukkan nama matakuliah : ')
    data_mahasiswa.append(mata_kuliah)

    nilai_tugas = int(input('Masukkan nilai tugas mahasiswa : '))
    data_mahasiswa.append(nilai_tugas)

    nilai_uts = int(input('Masukkan nilai UTS Mahasiswa : '))
    data_mahasiswa.append(nilai_uts)

    nilai_uas = int(input('Masukkan nilai UAS Mahasiswa : '))
    data_mahasiswa.append(nilai_uas)

    nilai_akhir = (nilai_tugas * 0.3) + (nilai_uts * 0.3) + (nilai_uas * 0.4)
    data_mahasiswa.append(nilai_akhir)

    if nilai_akhir >= 80 :
        grade = 'A'
    elif nilai_akhir >= 70:
```

```

        grade = 'B'
    elif nilai_akhir >= 60:
        grade = 'C'
    elif nilai_akhir >= 50:
        grade = 'D'
    else :
        grade = 'E'
    data_mahasiswa.append(grade)

    print('-----')
    print(f'data mahasiswa ke- {jumlah+1} ')
    print(data_mahasiswa)
    print(f>Nama mahasiswa \t: {data_mahasiswa[0]})
    print(f>Nama mata kuliah \t: {data_mahasiswa[1]})
    print(f>Nilai Tugas mata kuliah {mata_kuliah} \t: {data_mahasiswa[2]})
    print(f>Nilai UTS mata kuliah {mata_kuliah} \t: {data_mahasiswa[3]})
    print(f>Nilai UAS mata kuliah {mata_kuliah} \t: {data_mahasiswa[4]})
    print(f>Nilai Akhir mata kuliah {mata_kuliah} \t: {data_mahasiswa[5]})
    print(f>Grade \t: {data_mahasiswa[6]})
    jumlah += 1
    lagi = input('Apakah masih ingin menginput data mahasiswa? y/n')
    if lagi != 'y' :
        break

    print('-----')
    print('Terima Kasih')

```

Output :

```

-----
data mahasiswa ke- 1
['Rafli Pratama', 'PTSI', 90, 89, 95, 91.7, 'A']
Nama mahasiswa : Rafli Pratama
Nama mata kuliah : PTSI
Nilai Tugas mata kuliah PTSI : 90
Nilai UTS mata kuliah PTSI : 89
Nilai UAS mata kuliah PTSI : 95
Nilai Akhir mata kuliah PTSI : 91.7
Grade : A
-----
data mahasiswa ke- 2
['Atha', 'PTSI', 87, 90, 89, 88.69999999999999, 'A']
Nama mahasiswa : Atha
Nama mata kuliah : PTSI
Nilai Tugas mata kuliah PTSI : 87
Nilai UTS mata kuliah PTSI : 90
Nilai UAS mata kuliah PTSI : 89
Nilai Akhir mata kuliah PTSI : 88.69999999999999
Grade : A
-----
Terima Kasih

```

Penjelasan :

Di awal program kita membuat variabel `data_mahasiswa` dengan list kosong, lalu kita buat variabel lagi dengan nilai 'y' dan jumlah dengan nilai 0, lalu kita lakukan perulangan selama lagi == 'y' jalankan perulangan for sebanyak data mahasiswa dan setiap perulangan `data_mahasiswa` index ke 0 dihapus (ini digunakan untuk memastikan data kosong), lalu kita masukkan nama mahasiswa dan nilainya di append ke `data_mahasiswa`, lalu begitu juga dengan mata kuliah, tugas, uts, dan uas, lalu setelah itu kita menghitung nilai akhir dengan rumus $\text{tugas} \times 30\% + \text{uts} \times 30\% + \text{uas} \times 40\%$ lalu nilai akhir di append ke `data_mahasiswa`, setelah itu kita lakukan pengecekan jika nilai akhir ≥ 80 grade A, jika ≥ 70 grade = B, jika ≥ 60 grade C, Jika ≥ 50 grade D dan selain itu grade E, lali kita append grade ke data Mahasiswa, lalu kita print `data_mahasiswa` per index dan melakukan input untuk variabel lagi untuk mengakhiri perulangan jika nilai yang di inputkan bukan 'y'

1.3. Kesimpulan dan Saran

1.3.1. Kesimpulan

(contoh) Dari praktikum ini dapat disimpulkan :

1. List merupakan salah satu tipe data bawaan dari python
2. List berbeda dari Array, karena list dapat digunakan untuk tipe data yang berbeda beda sedangkan array tidak
3. Kita dapat membuat list kosong
4. Banyak method yang dapat digunakan pada list untuk mempermudah pengolahan nilai di dalamnya
5. Index pada list digunakan untuk memanggil nilai pada list tersebut
6. Kita dapat membuat list di dalam list

1.3.2. Saran

(contoh) Saran terhadap praktikum ini yaitu:

1. Teliti dalam memanggil index pada list karena perlu di ingat index di mulai dari 0
2. kurung [] untuk memanggil nilai list dengan index dan () untuk menggunakan index pada fungsi/method
3. perhatikan penggunaan tanda koma (,) untuk memisahkan nilai pada list

DAFTAR PUSTAKA

Modul *Perkuliahan dan Praktikum Algoritma dan Pemrograman* Oleh Ilhamsyah, S.Si., M.Cs.