

LAPORAN PRAKTIKUM

MODUL I

VARIABEL, TIPE DATA DAN OPERATOR



Disusun Oleh :

Novian Dwi Romadon

NIM : 19104011

Dosen

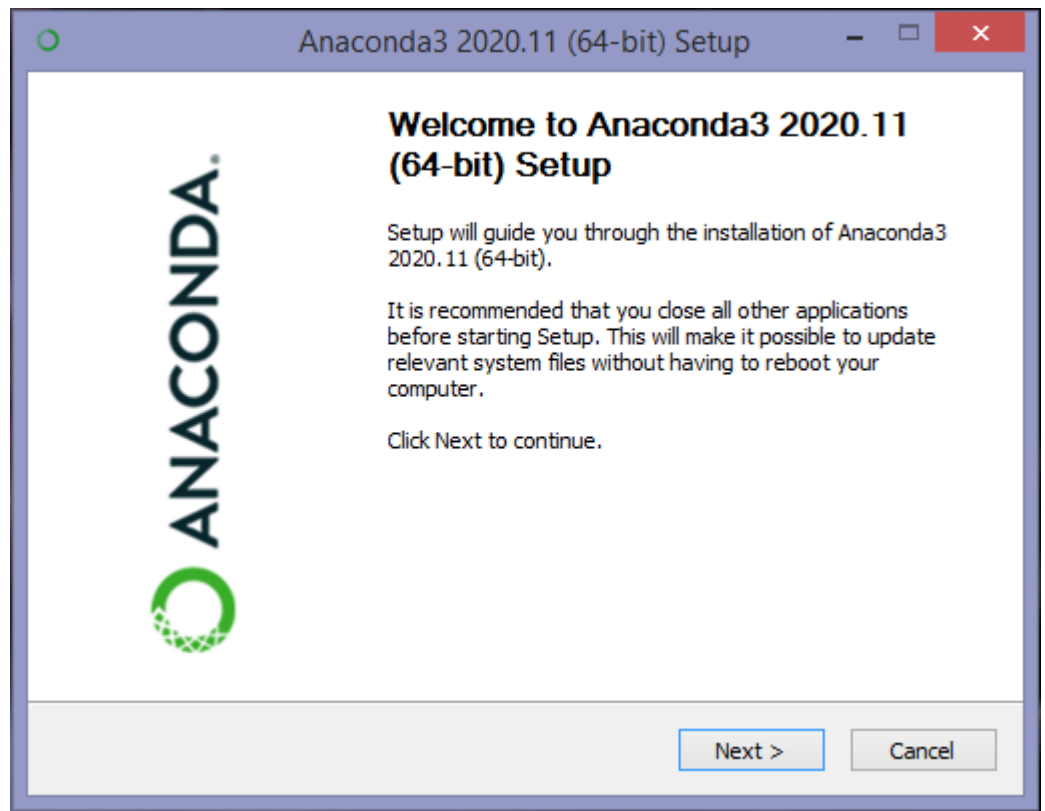
Ariq Cahya Wardhana, S.Kom., M.Kom.

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM
PURWOKERTO
2021**

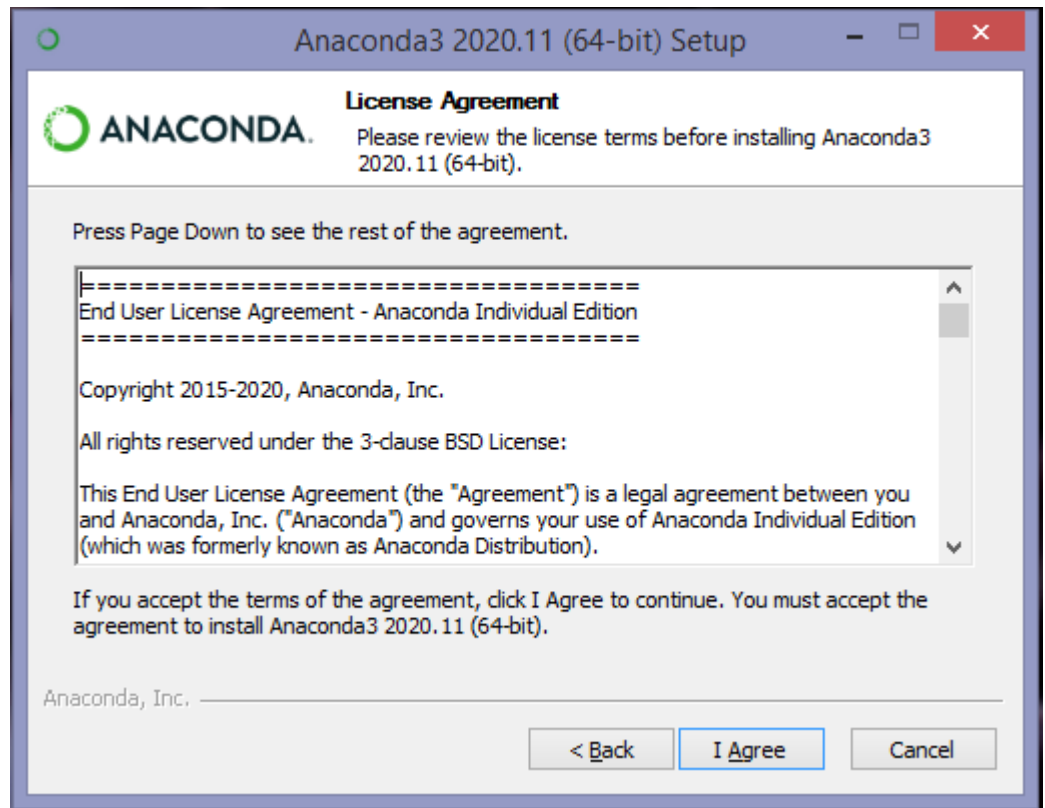
1. INSTALASI SOFTWARE PYTHON (ANACONDA)

Berikut merupakan cara instalasi Software Anaconda pada sistem operasi windows 64bit.

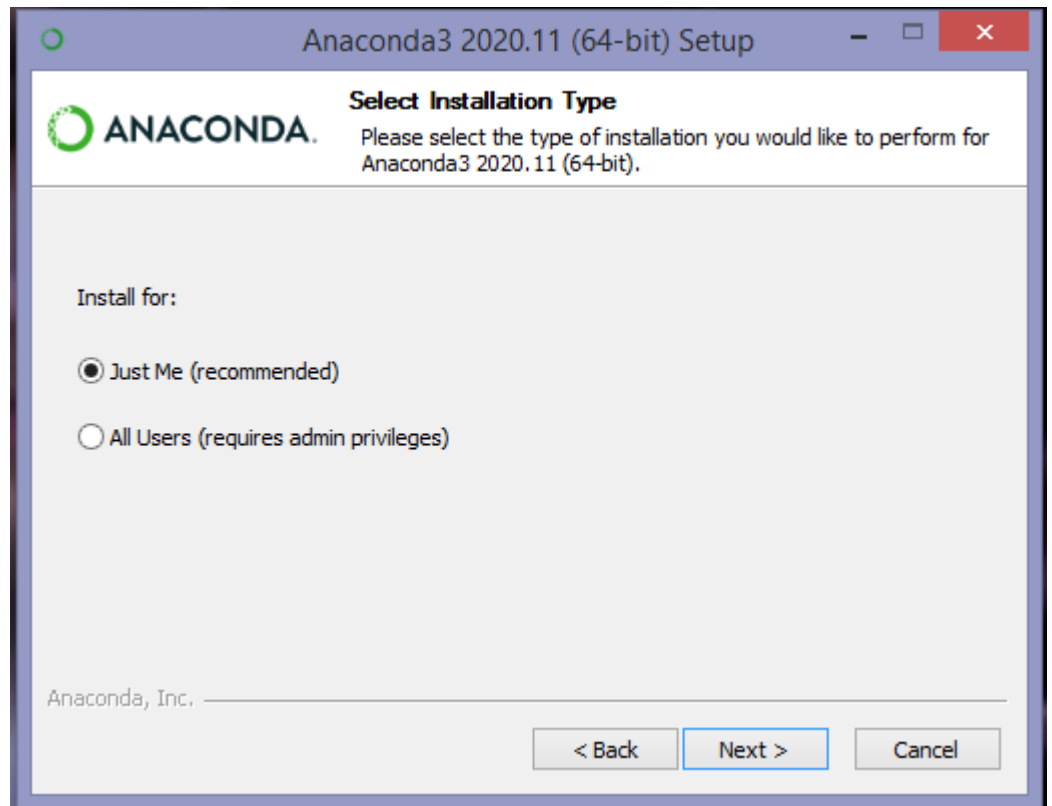
- a. Siapkan file installer Anaconda, lalu jalankan. Pada tahap awal install klik next.



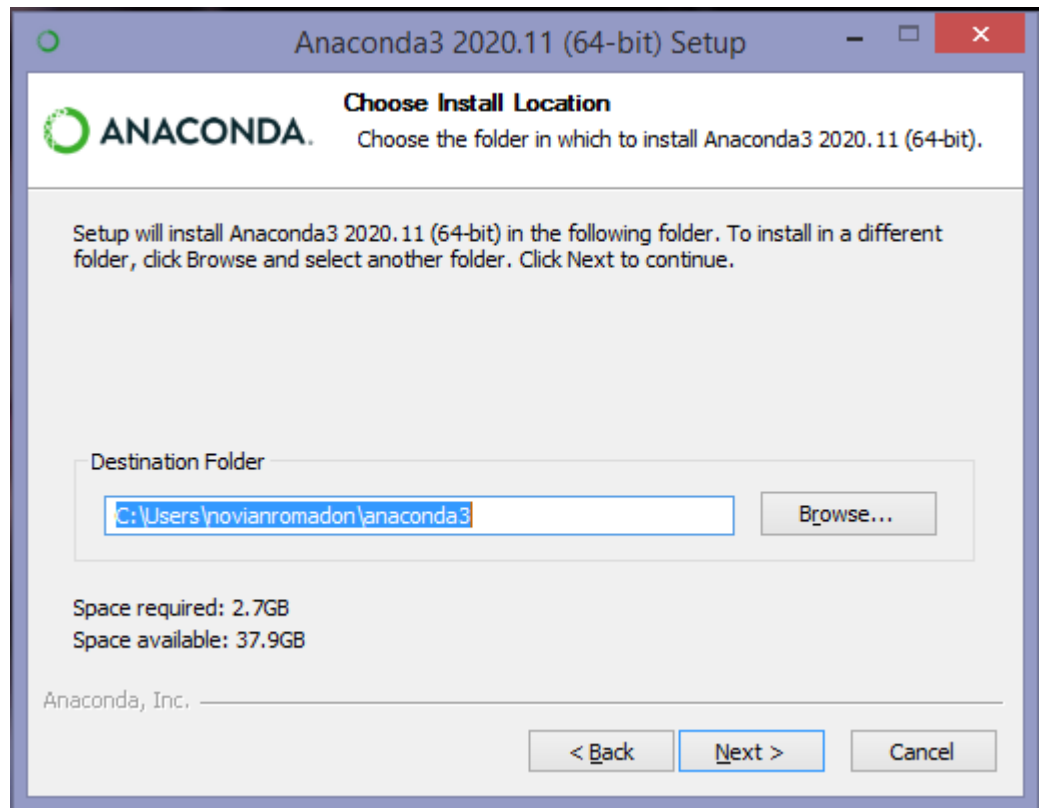
- b. Kemudian muncul persetujuan lisensi, klik I Agree.



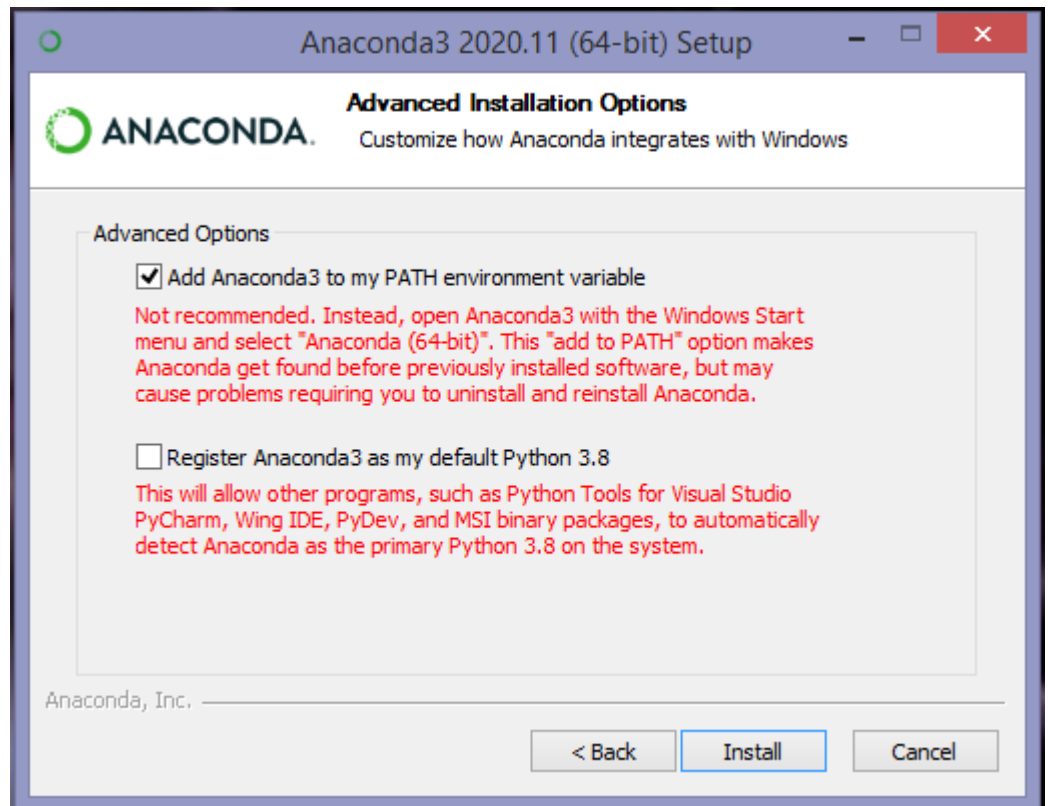
- c. Kemudian pilih Just Me yang di rekomendisi kan => Klik next.



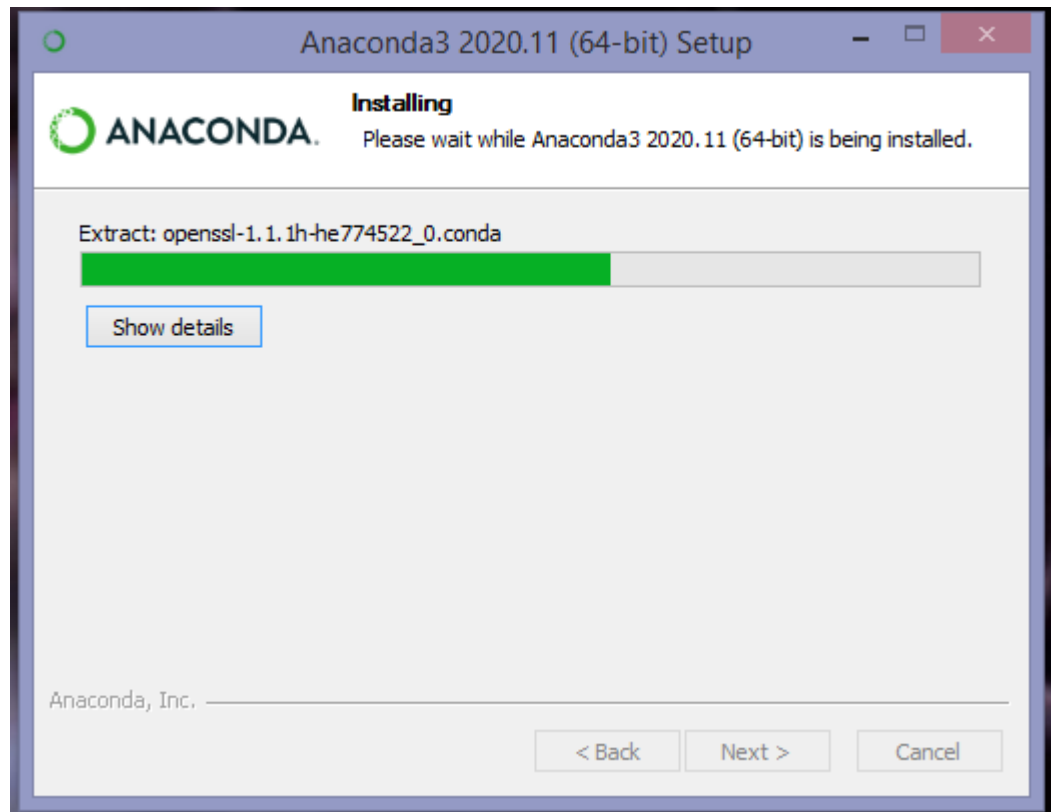
- d. Kemudian pilih lokasi penyimpanan aplikasi. Klik browser kemudian arahkan sesuai keinginan kita, klik next.



- e. Kemudian pilih Add Anaconda3 to my PATH environment variable. Untuk pilihan Register Anaconda3 as my default Python 3.8 bersifat Opsional, klik Install.



- f. Kemudian tunggu proses instal sampai selesai.



g. Kemudian klik Finish.



h. Menggunakan IDLE (Python Shell)

Berikut adalah contoh baris intruksi yang langsung dituliskan pada IDLE.

```
>>> nama = 'ucok'
>>> nama
'ucok'
>>> print(nama)
ucok
>>> umur = 20
>>> print(nama, "berumur", umur, "tahun")
ucok berumur 20 tahun
>>>
```

2. MEMBUAT DAN EKSEKUSI KODE PROGRAM PADA PYTHON

A. Membuat kode program

Membuat program bernama hello.py yang menampilkan “Hello World!”.

```
E:\KULIAH\Semester 4\Praktikum GUI\Modul 1\Laporan>python hello.py
Hello World!
E:\KULIAH\Semester 4\Praktikum GUI\Modul 1\Laporan>
```

3. VARIABEL DAN OBJEK

Berikut adalah contoh, variabel yang sudah diisi dengan tipe bilangan bulat bisa diisi dengan bilangan riil, string, ataupun tipe data yang lain karena Python merupakan Bahasa dengan dynamic typing.

```
>>>
>>> x = 9
>>> type(x)
<class 'int'>
>>> x = True
>>> type(x)
<class 'bool'>
>>> x = 'contoh'
>>> type(x)
<class 'str'>
>>>
```

```
>>>
>>> x = 9
>>> id(x)
140704916711456
>>>
```

id pada potongan kode di atas merupakan sebuah identitas unik yang dimiliki oleh setiap variable. Cara mendapatkan id adalah dengan menggunakan perintah `id('nama_variabel')`.

```
>>>
>>> x = 9
>>> id(x)
140704916711456
>>> y = 9
>>> id(y)
140704916711456
>>>
```

Dari potongan kode di atas, jika kita memanggil id untuk variable x maupun y maka akan muncul id yang sama karena variable x maupun y memiliki id referensi yang sama.

```
>>>
>>> x = 9
>>> id(x)
140704916711456
>>> y = 9
>>> id(y)
140704916711456
>>> del y
>>> y
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'y' is not defined
>>> x
9
>>> id(x)
140704916711456
>>> x = True
>>>
```

Referensi objek variabel x akan dipindahkan dari objek '9' ke objek 'True'. Dengan demikian objek lama (9) akan diklaim sebagai sampah karena objek tersebut tidak ditunjuk oleh variable apapun.

4. PYTHON BERSIFAT CASE-SENSITIVE

Variable Posisi akan berbeda dengan variable posisi. Karena penulisan kode program pada python bersifat case sensitive.

```
>>>
>>> posisi = (300,300)
>>> posisi
(300, 300)
>>>
>>> Posisi
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'Posisi' is not defined
>>>
```

5. PERINTAH PROGRAM (STATEMENT)

Titik koma pada python hanya diberikan pada saat ada dua atau lebih statement pada satu baris yang sama.

```
>>> a = 1; b = 2; c = 3
>>> print(a); print(b); print(c)
1
2
3
>>>
```

Perintah program ditulis dalam satu baris kode, tetapi jika perintah yang dituliskan panjang maka setiap baris harus dihubungkan dengan tanda backslash (\).

```
>>> if isinstance(x,int) and \
... x > 0 and \
... x % 2 == 1:
...     print("%d adalah bilangan bulat ganjil positif" %x)
... 9 adalah bilangan bulat ganjil positif
```

Tanda backslash tidak diperlukan jika menulis perintah kode dalam bentuk array atau kode yang terdapat diantara tanda (...), [...] atau {...}.

```

>>> print("Pemrograman GUI" +
... 'dengan Python dan PyQt')
Pemrograman GUIDengan Python dan PyQt
>>> data = [
... 100,
... 200,
... 300
... ]
>>> kamus = {
... 'one': 'satu',
... 'two': 'dua',
... 'three': 'tiga'
... }
>>> data
[100, 200, 300]
>>> kamus
{'one': 'satu', 'two': 'dua', 'three': 'tiga'}
>>>

```

6. TIPE NUMERIK

A. Bilangan Bulat

Python dapat menggunakan bilangan integral dengan basis decimal (10), biner (2), octal (8) maupun heksadesimal (16).

```

>>> # bilangan biner
>>> a = 0b1001
>>> # bilangan oktal
>>> b = 0o23
>>> # bilangan heksadesimal
>>> c = 0x2f
>>> a
9
>>> b
19
>>> c
47
>>>

```

Tipe Boolean

```

>>> a = True
>>> type(a)
<class 'bool'>
>>> int(a)
1
>>>

```

Proses perhitungan dan penambahan bilangan pada python akan menghasilkan objek baru, hal ini terlihat dari id nya.

```

>>>
>>> a = 15
>>> id(a)
140704213641440
>>> a += 5
>>> a
20
>>> id(a)
140704213641600
>>>

```

B. Bilangan Riil

Python menyediakan tipe float, decimal.Decimal dan complex. Tipe bilangan float menggunakan titik untuk tanda desimalnya.

```

>>> a = 123.456
>>> a
123.456
>>> a * 2
246.912
>>>

```

7. TIPE STRING

Objek string dapat dibuat dengan tiga cara yaitu:

- Menggunakan tanda petik tunggal
- Menggunakan tanda petik ganda
- Menggunakan tanda petik tunggal ataupun ganda yang direpetisi sebanyak tiga kali

```

>>>
>>> s1 = 'pemrograman python'
>>> s2 = "pemrograman python 2"
>>> s3 = '''pemrograman
python 3'''
>>>

```

Mengakses karakter di dalam string dengan tanda [] diikuti nomor array-nya.

```

>>>
>>> s1[0], s1[1], s1[2]
('p', 'e', 'm')
>>>

```

Dalam string dapat memberikan karakter khusus antara lain \n untuk memberikan enter, \' untuk memberikan petik tunggal, \t untuk memberikan tab.

```

>>>
>>> data = 'p001\tspidol\t\t9000\np002\tpensil\t\t6000'
>>> print(data)
p001      spidol          9000
p002      pensil         6000
>>>

```

Juga dapat menggabungkan dua objek string menjadi satu dengan operator +.

```

>>>
>>> data = '\tharga\n' + data
>>> print(data)
        harga
p001      spidol          9000
p002      pensil         6000
>>>

```

A. Membandingkan String

Membandingkan tipe data string dengan menggunakan operator >, <=, >=

```

>>>
>>> s1 = 'python'
>>> s2 = 'PYTHON'
>>> s1 == s2
False
>>> s1 != s2
True
>>> s1 < s2
False
>>>

```

B. Mengekstrak Substring

Di dalam string dapat diekstrak dengan menggunakan operator slice (:) dengan menyertakan indeks awal dan akhir sebagai penanda. Dan len berfungsi untuk menghitung jumlah string.

```

>>> s = 'Pemrograman Python dan PyQt'
>>> s1 = s[0:11]
>>> s1
'Pemrograman'
>>> len(s1)
11

```

Kode tersebut mengambil substring dari variable s mulai dari indeks ke 0 sampai indeks ke 11.

```

>>> s = s[:11]
>>> s = s[:8]
>>> s = s[8:]
>>> s = s[0:11:2]
>>> s = s[0:11:1]
>>> s = s[0:11:3]

```

C. Membuat String dengan format tertentu

Python dapat menggabungkan tipe data atau format lain ke dalam string menggunakan \$s, %f, %s dll.

```
>>>
>>> s = 'balonku ada %d, kempes %d tinggal %f' % (5,1,4.5)
>>> s
'balonku ada 5, kempes 1 tinggal 4.500000'
>>>
```

8. TIPE KOLEKSI

Beberapa tipe koleksi antara lain list, dictionary, tuple dan set. Cara akses list dapat digabungkan dengan fungsi perulangan dasar seperti for, while dan lain-lain.

```
>>> list = ['balon', 'budi', 'ada', 5]
>>> for item in list:
...     print (item)
...
balon
budi
ada
5
>>>
```

Menghapus elemen pada list..

```
>>> del list[1]
>>> list
['balon', 'ada', 5]
```

Merubah value pada list.

```
>>>
>>> list[1] = 'papaya'
>>> list
['balon', 'papaya', 5]
>>>
```

Menambah elemen pada list.

```
>>>
>>> list.extend([1])
>>> list
['balon', 'papaya', 5, 1]
>>>
```

Merubah list yang di buat dengan perintah perintah tersebut!

```
>>>
>>> hp = ['samsung', 'oppo', 'iphone', 2, 5]
>>> hp
['samsung', 'oppo', 'iphone', 2, 5]
>>> del hp[2]
>>> hp
['samsung', 'oppo', 2, 5]
>>> hp[2] = 3
>>> hp
['samsung', 'oppo', 3, 5]
>>> hp.extend(['LG'])
>>> hp
['samsung', 'oppo', 3, 5, 'LG']
>>>
```