

LAPORAN PRAKTIKUM
MODUL 1
VARIABEL, TIPE DATA DAN OPERATOR



Disusun Oleh:

Hasna Rizki Violina

NIM:19104045

Dosen :

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

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

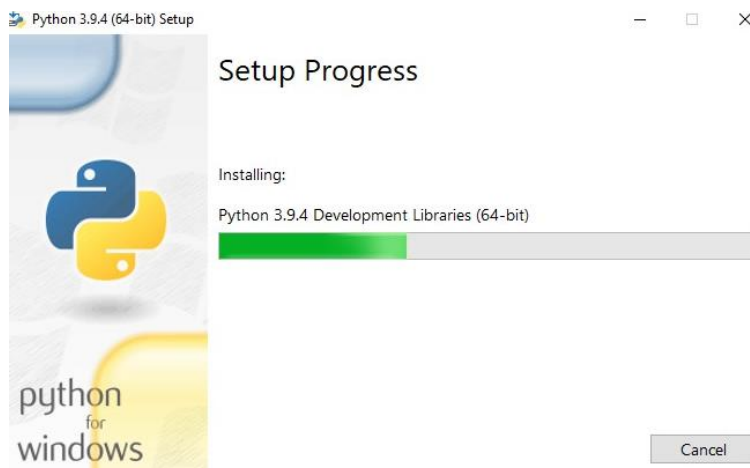
2021

1. INSTALASI SOFTWARE PYTHON

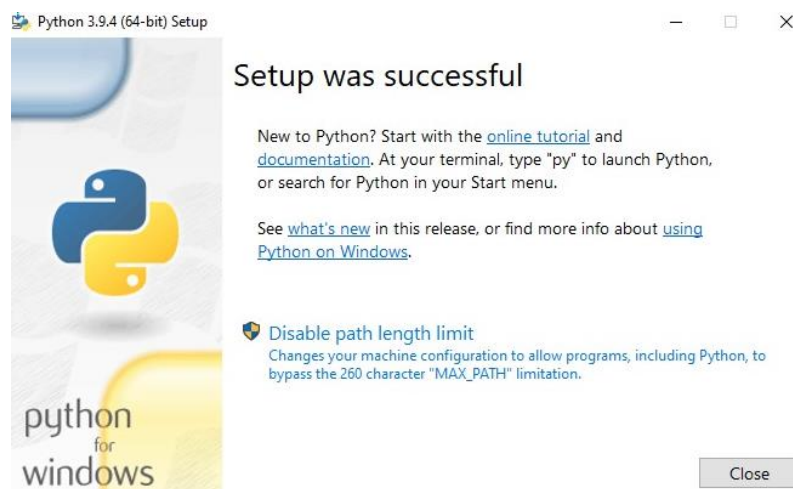
- a. Klik dua kali pada file instalasi. Lalu klik install now.



- b. Menunggu hingga proses instalasi selesai.



- c. Proses instalasi selesai.



2. MEMBUAT DAN EKSEKUSI KODE PROGRAM PADA PYTHON

Membuat kode program sederhana yang menampilkan output “Hello World!” dengan menggunakan kode *print()* untuk mencetaknya.

```
>>> print("Hello World!")
Hello World!
```

3. VARIABEL DAN OBJEK

Membuat variable dengan langsung mengisi *value*-nya yang mana *value* tersebut dapat berupa bilangan bulat, bilangan riil, string, ataupun tipe data lain. Variable x dapat berubah ubah tipe datanya sesuai dengan kebutuhan.

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

Menampilkan identitas unik dari sebuah variable dengan menggunakan `id('nama variabel')`.

```
>>> x = 9
>>> id(x)
2067613641264
```

Untuk setiap variable, jika memiliki value yang sama maka python akan menunjukan nilai/identitas unik yang sama untuk variable yang berbeda. Namun, jika value-nya berbeda, maka identitas uniknya berbeda juga.

```
>>> x = 9
>>> id(x)
2067613641264
>>> y = 9
>>> id(y)
2067613641264
>>>
>>> y = 7
>>> id(y)
2067613641200
```

Menggunakan perintah *del* untuk menghapus variable y, maka yang akan terhapus hanya referensinya saya, bukan objek '9' yang tadi ditunjuk oleh variable x dan y.

```
>>> x = 9
>>> id(x)
2067613641264
>>> y = 9
>>> id(y)
2067613641264
>>> del y
>>> y
Traceback (most recent call last):
  File "<pyshell#30>", line 1, in <module>
    y
NameError: name 'y' is not defined
>>> x
9
>>> id(x)
2067613641264
```

Dengan menambahkan 'x = True' maka referensi objek variable x berpindah dari '9' ke 'True'.

4. PYTHON BERSIFAT *CASE-SENSITIVE*

Bersifat *case-sensitive* yang artinya kasis huruf besar dan huruf kecil memiliki beda arti. Seperti pada variable di kode berikut ini,

```
>>> posisi = (300,300)
>>> posisi
(300, 300)
>>> Posisi
Traceback (most recent call last):
  File "<pyshell#38>", line 1, in <module>
    Posisi
NameError: name 'Posisi' is not defined
...
```

Variable 'posisi' dan 'Posisi' tidak sama.

5. PERINTAH PROGRAM (STATEMENT)

Menggunakan tanda titik koma untuk menuliskan lebih dari satu statement pada satu baris.

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

Memecah baris kode yang panjang menjadi beberapa baris kode dengan menghubungkannya menggunakan tanda *backslash* (\) di setiap akhir baris.

```
>>> x = 9
>>> 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
```

Namun *backslash* tidak diperlukan ketika menulis baris kode dalam bentuk array atau kode yang memiliki 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

Bilangan bulat pada python yaitu int dan bool. Yang pertama dapat menuliskan menggunakan bilangan integral dengan basis decimal, biner, octa, maupun heksadesimal.

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

Yang kedua yaitu tipe Boolean

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

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

```
>>> a = 15
>>> id(a)
2067613641456
>>> a += 5
>>> a
20
>>> id(a)
2067613641616
```

b. Bilangan Riil

Tipe bilangan riil, python menyediakan tipe float, decimal, dan complex. Pada tipe float menggunakan titik untuk tanda desimalnya. Hal tersebut sama dengan tipe decimal, tetapi tipe decimal digunakan untuk melakukan perhitungan dengan nilai koma yang lebih presisi.

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

7. TIPE STRING

Mendeklarasikan objek string menggunakan tiga cara berikut ini

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

Mengakses karakter didalam string dapat menggunakan tanda [] diikuti nomor arraynya.

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

Pada python dapat memberikan karakter dengan menggunakan tanda '\n' untuk enter, '\ ' untuk petik tunggal, dan '\t' untuk tab.

```
...
>>> data = 'p001\tpidol\t\t9000\np002\tpensil\t\t6000'
>>> print(data)
p001      pidol          9000
p002      pensil        6000
```

Menggabungkan dua objek string dapat menggunakan operator '+'.

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

a. Membandingkan string

Dalam membandingkan string dapat menggunakan operator ==, >, <, <=, >=.

Sedangkan untuk membandingkan id objek string dapat menggunakan *is*.

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

b. Mengekstrak substring

Mengekstrak string dengan menggunakan operator slice (:) disertai indeks awal dan akhir sebagai penanda. Pada kode dibawah ini, mengambil substring dari variable s mulai dari indeks ke 0 sampai indeks ke 11.

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

Jika tidak menyertakan indeks maka string yang akan diekstrak adalah sepanjang string tersebut / string sisanya.

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

c. Membuat string dengan format tertentu

Memnggabungkan tipe data atau format lain ke dalam string yang telah dibuat dengan menggunakan \$d, %f, %s dan lain sebagainya.

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

8. TIPE KOLEKSI

Membuat objek list dengan menggunakan tanda [], tiap objeknya dipisahkan dengan menggunakan koma dan dapat terdiri dari berbagai macam tipe data. List dapat diakses dengan fungsi perulangan seperti pada kode berikut yang menggunakan fungsi perulangan *for*.

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

balon
budi
ada
5
```

Menghapus elemen list dengan menggunakan perintah `del namaList['indeksList']`. Kode berikut akan menghapus indeks ke-2 pada list.

```
>>> del list[2]
>>> list
['balon', 'budi', 5]
...
```

Mengubah elemen list dengan menggunakan perintah `namaList['indeks'] = value baru`. Kode dibawah ini mengubah indeks ke-1 dengan value 'budi' menjadi 'ada'.

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

Menambahkan elemen dengan menggunakan perintah `namaList.extend([list])`. Kode dibawah ini menambahkan elemen 'Loh'.

```
>>> list.extend(['loh'])
>>> list
['balon', 'ada', 5, 'loh']
...
```

Latihan merubah list yang telah saya buat dengan beberapa perintah diatas.

```
>>> hogwarts = [1,2,'Harry','Hermione','Ron']
>>> hogwarts
[1, 2, 'Harry', 'Hermione', 'Ron']
>>> del hogwarts[0]
>>> hogwarts
[2, 'Harry', 'Hermione', 'Ron']
>>> hogwarts[0] = 'Dumbledore'
>>> hogwarts
['Dumbledore', 'Harry', 'Hermione', 'Ron']
>>> hogwarts.extend(['Lily'])
>>> hogwarts
['Dumbledore', 'Harry', 'Hermione', 'Ron', 'Lily']
```