

LAPORAN PRAKTIKUM

Modul 1

VARIABEL, TIPE DATA DAN OPERATOR



Disusun Oleh :

Alya Aulia Hanafi

NIM : 19104004

Dosen Pengampu :

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

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

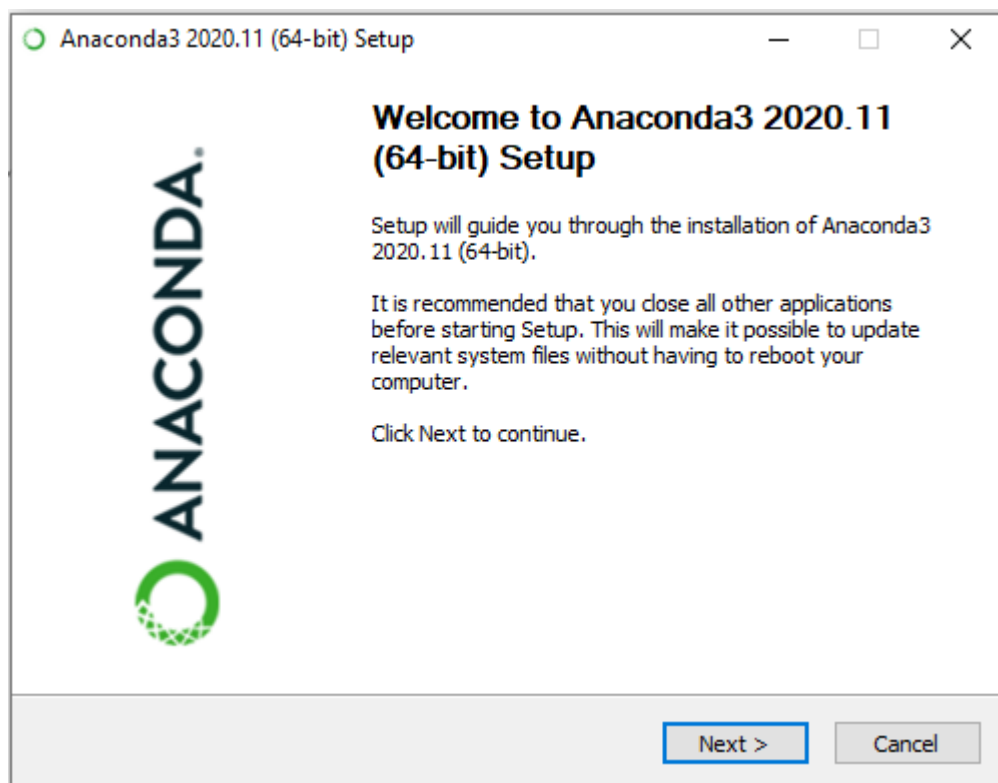
INSTITUT TEKNOLOGI TELKOM

PURWOKERTO

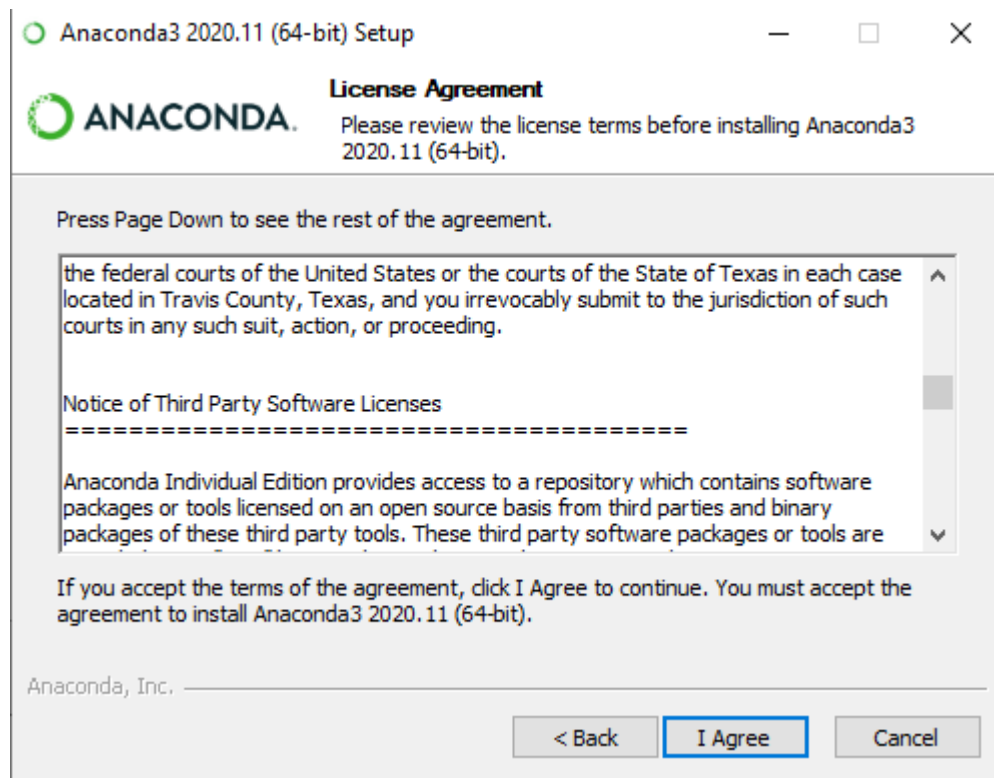
2020

A. Instalasi Software Anaconda

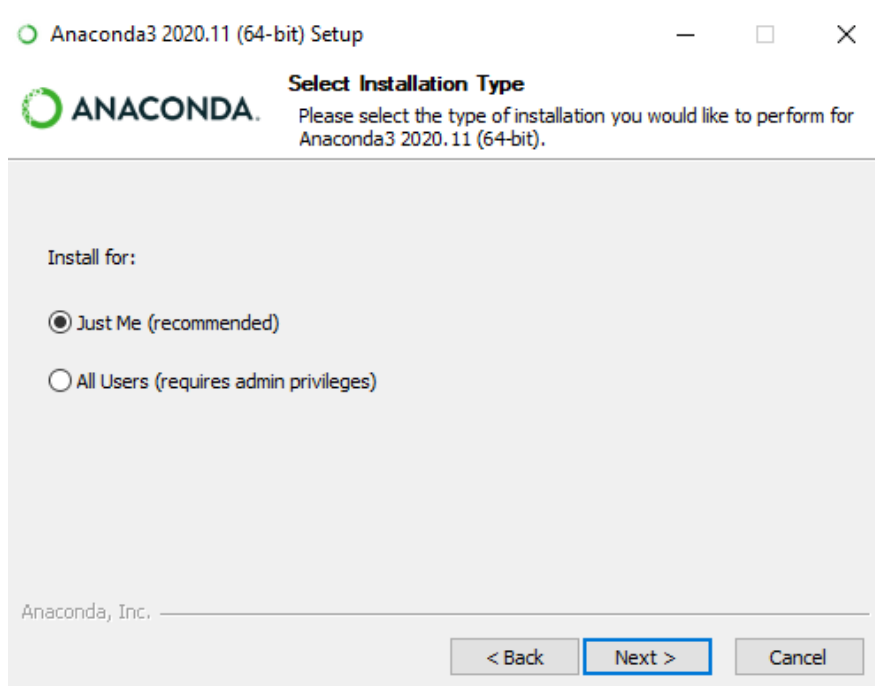
1. Langkah pertama adalah mendownload Anaconda3-2020.11-Windows-x86_64. Sesuaikan versi anaconda dengan perangkat. Ukuran file unduhan Anaconda3 457 mb. Setelah terdonwload, klik dua kali file anaconda lalu akan muncul tampilan seperti dibawah ini. Klik next untuk melanjutkan proses instalasi



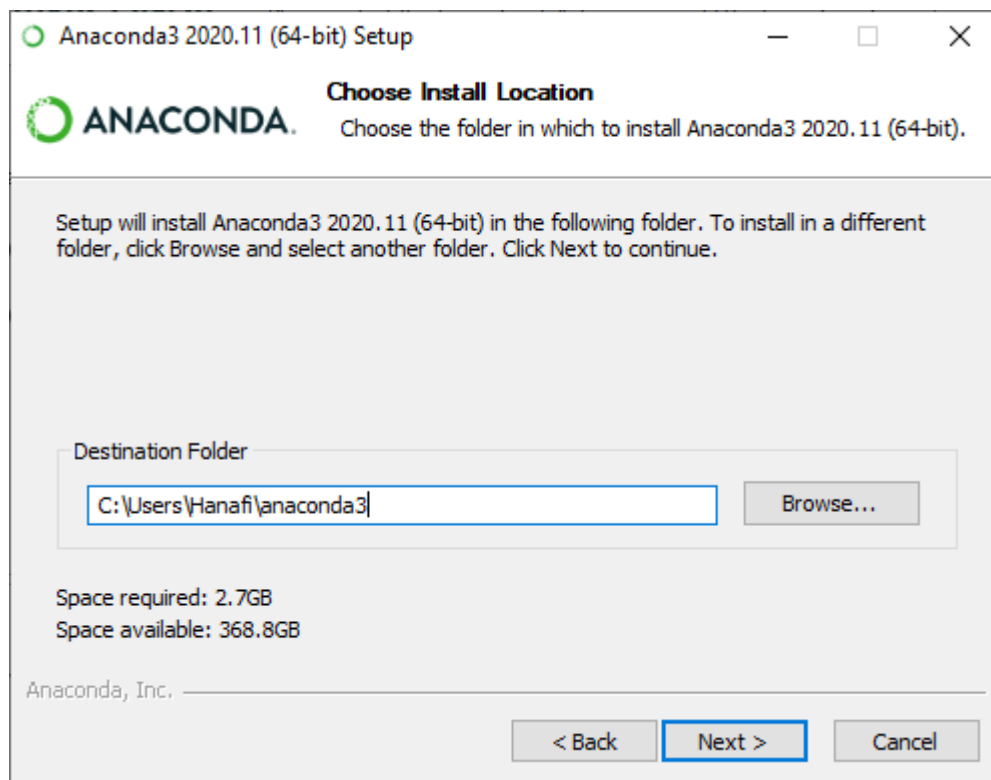
2. Pada jendela selanjutnya terdapat proses persetujuan user yang berkaitan license. Klik I agree untuk menyetujui license tersebut dan melanjutkan ke tahap berikutnya



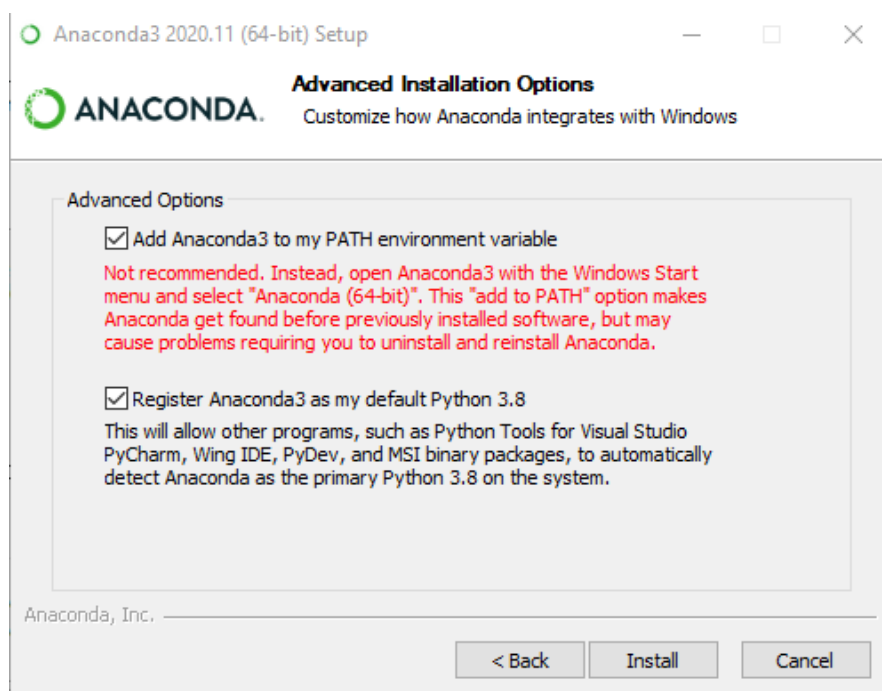
3. Pada tahap berikutnya terdapat pemilihan tipe instalasi. Pada bagian ini pilih sesuai kebutuhan. Anaconda memberikan 2 pilihan yaitu instalasi hanya untuk diri sendiri atau semua user dapat mengakses anaconda yang akan diinstal. Setelah memilih, klik next untuk melanjutkan proses instalasi



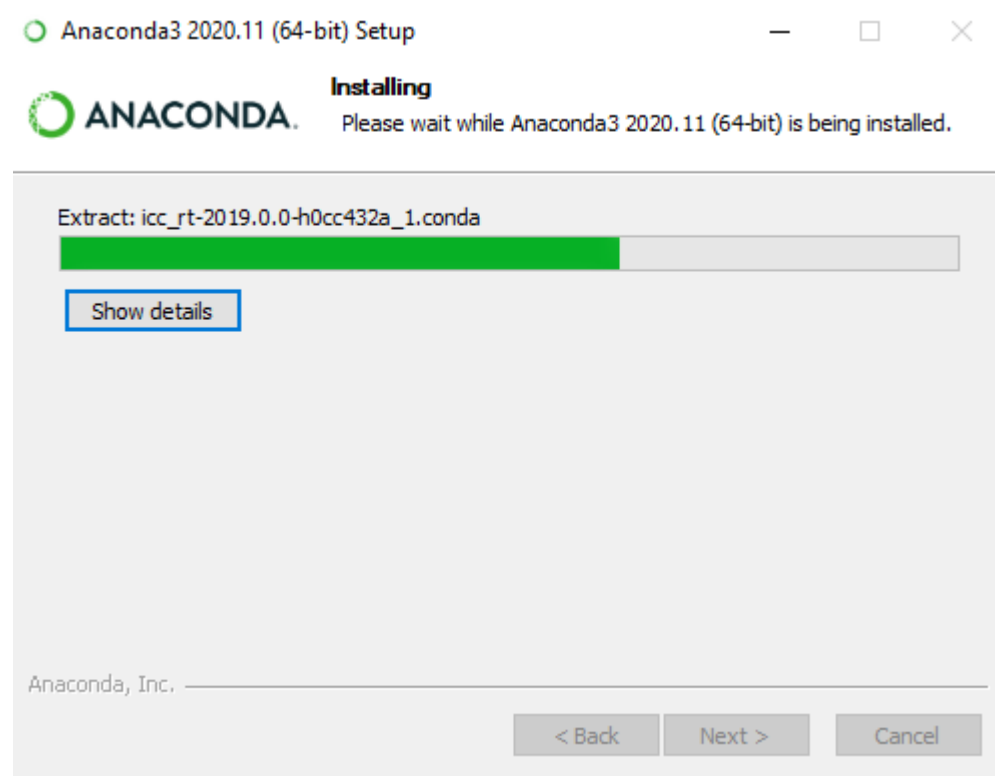
4. Pada tahap selanjutnya terdapat pemilihan tempat penyimpanan data anaconda. Setelah memilih tempat penyimpanan, klik next untuk melanjutkan



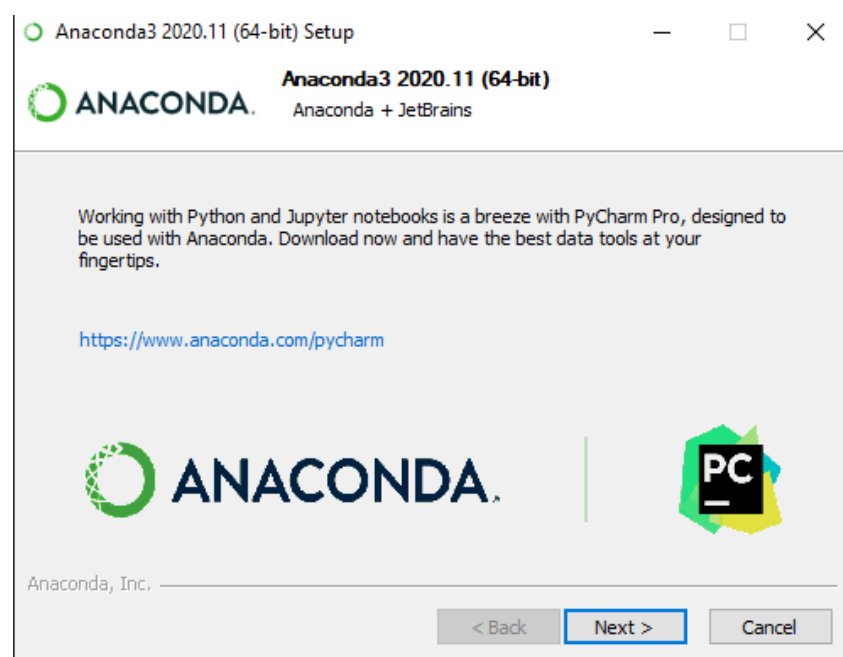
5. Selanjutnya pada tahap ini ceklis kedua pilihan dibawah untuk menambahkan Anaconda pada PATH kita dan mendaftarkan Anaconda sebagai default Python 3.8 kita. Lalu klik install untuk mulai menginstal Anaconda



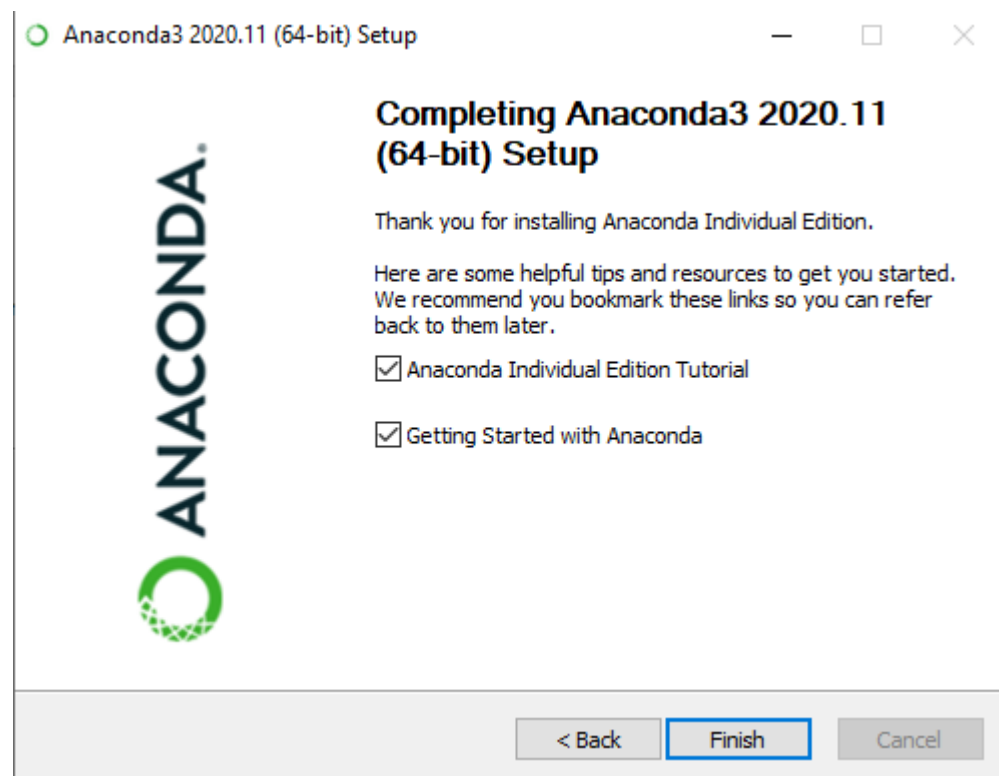
6. Tunggu proses instalasi Anaconda selesai



7. Saat selesai , Anaconda menyediakan link tools PyCharm. Kunjungi link bila membutuhkan. Jika tidak klik next



8. Klik finish dan proses Instalasi Anaconda selesai



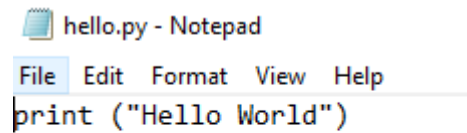
B. Praktikum

1. Contoh Instruksi pada Python

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

Penjelasan : variable dengan nama (nama) diisi dengan data ' ucok ', saat variable nama ditulis setelah pengisian data maka cmd akan menampilkan nama ucok. Pada python tipe data variable akan otomatis menyesuaikan dengan data yang dimasukan. Contoh lain variable umur diisi dengan data bernilai ' 20 ' lalu diberi instruksi print (nama , 'berumur', ' umur, 'tahun') maka akan menghasilkan output berupa kalimat ucok berumur 20 tahun, dimana ucok dan 20 merupakan data yang sudah diisi sebelumnya.

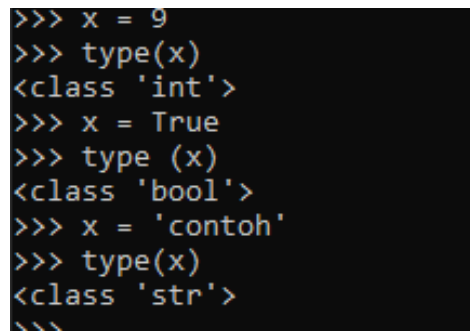
2.



```
hello.py - Notepad
File Edit Format View Help
print ("Hello World")
```

3. Variable dan Objek

A.

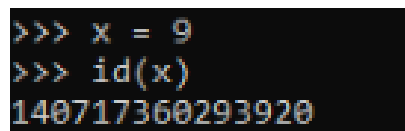


```
>>> x = 9
>>> type(x)
<class 'int'>
>>> x = True
>>> type(x)
<class 'bool'>
>>> x = 'contoh'
>>> type(x)
<class 'str'>
>>>
```

Penjelasan :

Bahasa Python bersifat dynamic typing yang artinya variable tidak dibatasi dengan tipe datanya. Sebagai contoh diatas variable x dapat diisi oleh beberapa tipe data mulai dari int , bool , dan juga str (String). Kode program (type (x)) digunakan untuk mengecek tipe data dari variable x setelah diisi oleh data. Pada saat variable x diisi dengan data (9) maka x bertipe data int, saat x diisi dengan True, maka tipe data x menjadi Boolean dan juga saat variable x diisi dengan kalimat ('contoh') , maka tipe data x menjadi str (String). Ini menandakan bahwa satu variable dapat dirubah tipe datanya sesuai kebutuhan.

B.



```
>>> x = 9
>>> id(x)
140717360293920
```

Penjelasan :

Kode diatas merupakan kode yang digunakan untuk mengetahui id dari variable dengan nilai 9. Id ini merupakan identitas unik yang dimiliki oleh variable.

C.

```
>>> x = 9
>>> id(x)
140717360293920
>>> y = 9
>>> id(y)
140717360293920
```

Penjelasan :

Variable berbeda (x , y) yang memiliki nilai sama yaitu (9), maka id dari kedua variable yang berbeda itu akan sama karena python menunjuk nilai yang sama

D.

```
>>> x = 9
>>> id(x)
140717360293920
>>> y = 9
>>> id(y)
140717360293920
>>> y = 7
>>> id(y)
140717360293856
```

Penjelasan :

Sebagai contoh lain, disini variable (y) diisi dengan nilai yang berbeda dengan sebelumnya yaitu (7) , saat id variable dicek disitu terlihat bahwa id variable (y) berbeda dengan variable(x dan y) yang memiliki nilai (9).

E.


```

>>> x = 9
>>> id(x)
140716789934112
>>> y = 9
>>> id(y)
140716789934112
>>> del y
>>> y
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'y' is not defined
>>> Y
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'Y' is not defined
>>> x
9
>>> id(x)
140716789934112
>>> x = True
>>> x
True

```

Penjelasan :

Pada kode diatas , variable y dihapus dengan perintah del. Disini yang akan dihapus hanya referensinya saja bukan nilai (9) yang ditunjuk oleh variable x dan y. Maka saat kita memanggil variable (y) maka peringatan eror akan keluar yang mengatakan bahwa variable y tidak teridentifikasi. Dan saat menuliskan variable (x) maka nilai (9) masih keluar. Jika x diisi dengan data (True) maka data (9) akan dianggap sampah karena tidak ditunjuk oleh variable apapun

4. 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
>>>

```

Penjelasan :

Python bersifat Case Sensitive sebagai contoh variable posisi berbeda dengan variable Posisi. Perbedaan besar kecilnya huruf pada penulisan sangat berpengaruh. Seperti kode diatas saat pembuatan variable posisi memiliki huruf p yang kecil, pada saat dipanggil dengan (posisi) maka nilai akan keluar. Sebaliknya jika posisi dipanggil dengan (Posisi) maka tidak ditemukan variable tersebut.

5. Perintah Program (Statement)

A.

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

Penjelasan :

Penggunaan titik koma pada python pada saat ada beberapa perintah dalam satu baris yang sama. Perintah print digunakan untuk mencetak nilai yang sudah diisikan pada variable.

B.

```
>>> 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
```

Penjelasan :

Penulisan perintah yang Panjang dapat dengan memecah menjadi beberapa baris dengan symbol(\). Antara baris satu dengan lainnya dihubungkan oleh tanda backslash.

C.

```
>>> print("Pemrograman GUI" +
... "dengan Python dan PyQt")
Pemrograman GUI dengan 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'}
>>>
```

Penjelasan :

Pada kode diatas , penulisan statement array jika ingin berbeda baris maka tidak perlu menggunakan backslash. Penulisan statement yang dipecah juga bisa dengan tanda plus. Penulisan array yang mengandung (..), {...}, [...] jika berbeda baris hanya perlu di beri titik koma pada akhir baris tidak perlu menggunakan backslash.

6. Tipe Numerik

A. Bilangan Bulat

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

Penjelasan :

Kode diatas merupakan contoh dari bilangan integral dengan basis decimal yaitu biner, octal dan juga heksadecimal. Jika kita menuliskan bilangan tersebut akan menghasilkan output berupa angka sesuai aturan bilangan decimal.

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

Penjelasan :

Berikutnya kode diatas merupakan contoh tipe bilangan Boolean yang bernilai True. Boolean hanya memiliki dua nilai yaitu True dan False.

```

>>> a = 15
>>> id(a)
140716789934304
>>> a +=5
>>> a
20
>>> id(a)
140716789934464
>>>

```

Penjelasan :

Kode diatas merupakan operasi aritmatika pada Python dimana, pertama variable a diisi dengan nilai 15 dan memiliki id variable seperti diatas. Lalu variable a ditambah dengan data bernilai 5. Maka sekarang variable a memiliki data yang bernilai 20 dan id pun ikut berubah.

B. Bilangan Riil

```

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

```

Penjelasan :

Selanjutnya ada bilangan riil yang terdiri dari float dan decimal. Folat menggunakan titik untuk tanda decimal. Seperti kode diatas, a diisi dengan data float. Saat dikalikan dengan 2 , maka data akan dikalikan 2 dan tetap bernilai float. Untuk tipe bilangan decimal sama dengan float , namun berbeda pada proses perhitungan nilai koma yang lebih akurat.

7. Tipe String

A

```

>>> s1 = 'pemrograman python'
>>> s2 = "pemrograman pyhton 2"
>>> s2 = '''pemrograman
... python 3'''
>>> s1[0], s1[1], s1[2]
('p', 'e', 'm')
>>>

```

Penjelasan :

Penulisan String memiliki 3 cara yaitu ditulis dengan petik tunggal ('), petik ganda (") dan petik tunggal atau ganda yang diulang sebanyak 3 kali (``) atau (""").

Pengaksesan karakter di dalam string dapat dilakukan dengan tanda [] diikuti nomer array.

B.

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

Penjelasan :

Pada penulisan karakter dalam string kita dapat menambahkan \n untuk enter, \t untuk petik tunggal, \t untuk tab. Seperti contoh diatas menggunakan \t untuk tab beberapa karakter.

C.

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

Penjelasan :

Kode diatas menggunakan tanda \t untuk tab dan \n untuk enter dan juga tanda plus untuk menampilkan data yang sudah dibuat sebelumnya.

D.

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

Penjelasan :

Kode diatas membangun string dengan tanda (==) artinya persamaan, tanda(!=) berarti tidak sama dan tanda < berarti lebih dari. Variable s1 dan s2 yang berisi data

string dibandingkan dengan menggunakan tanda tersebut. Jika perbandingan benar maka akan muncul output True , sebaliknya jika salah False.

E.

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

Penjelasan :

Pada kode diatas, substring diekstrak dengan menggunakan operator slice (:) dengan menuliskan indeks awal dan akhir. Berikut contoh penulisan :

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

F. Membuat String dengan format tertentu

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

Penjelasan :

Pada kode diatas menggabungkan tipe data int dan string dengan menggunakan %d , %f.

8. Tipe Koleksi

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

Penjelasan :

Tipe koleksi pada Python antara lain list, dictionary, tuple, set. Kode diatas adalah contoh tipe koleksi list. Objek list dibuat dengan tanda [], di dalamnya dipisahkan dengan koma antar item. Kode diatas list digabung dengan fungsi perulangan for.

C. Kesimpulan

Pada penulisan kode Python memiliki berbagai macam aturan. Dilihat dari cara penulisan, Bahasa Python lebih efisien dan lebih mudah dimengerti. Sebagai contoh penulisan variable yang fleksible dibanding dengan Bahasa lain.