

# **LAPORAN PRAKTIKUM**

## **Modul 1**

### **VARIABEL, TIPE DATA DAN OPERATOR**



**Disusun Oleh :**

**Fawwaz Muhammad Zulfikar**

NIM : 19104058

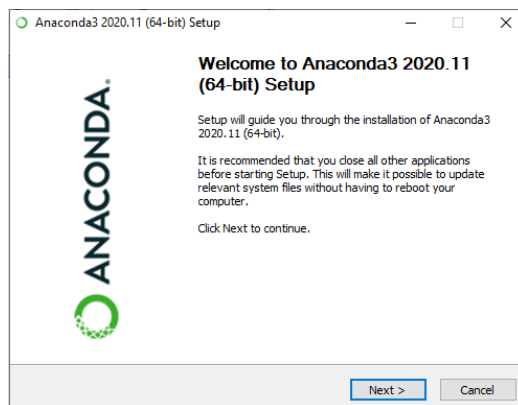
**Dosen Pengampu :**

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

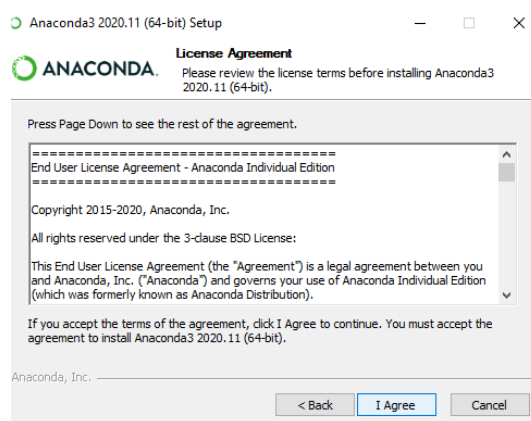
**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK  
FAKULTAS INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM  
PURWOKERTO  
2020**

## A. Instalasi Software Anaconda

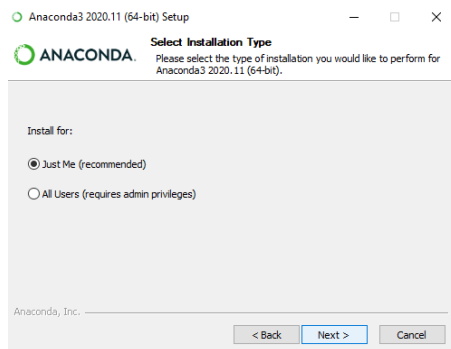
1. Langkah pertama adalah mendownload Anaconda3-2020.11-Windows-x86\_64. Sesuaikan versi anaconda dengan perangkat. Ukuran file unduhan Anaconda3 457 mb. Setelah terdonwload, klik dua kali file anaconda lalu akan muncul tampilan seperti dibawah ini. Klik next untuk melanjutkan proses instalasi



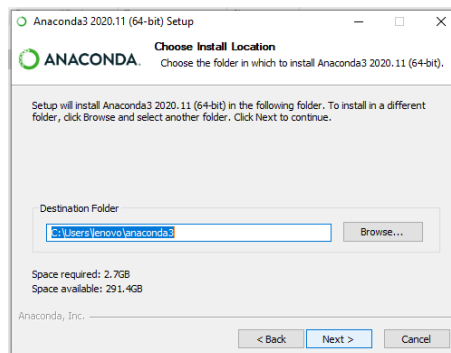
2. Pada jendela selanjutnya terdapat proses persejutuan user yang berkaitan license. Klik I agree untuk menyetujui license tersebut dan melanjutkan ke tahap berikutnya



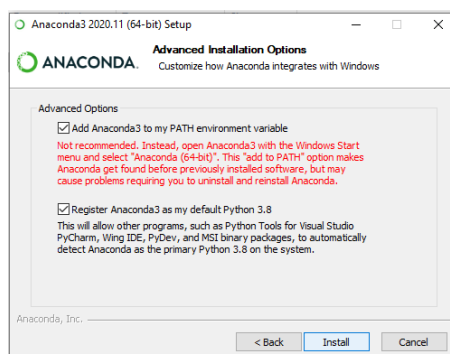
3. Pada tahap berikutnya terdapat pemilihan tipe instalasi. Pada bagian ini pilih sesuai kebutuhan. Anaconda memberikan 2 pilihan yaitu instalasi hanya untuk diri sendiri atau semua user dapat mengakses anaconda yang akan diinstal. Setelah memilih, klik next untuk melanjutkan proses instalasi



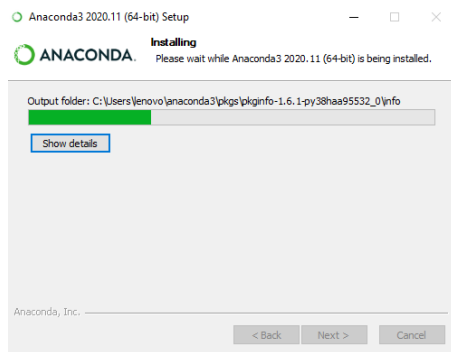
4. Pada tahap selanjutnya terdapat pemilihan tempat penyimpanan data anaconda. Setelah memilih tempat penyimpanan , klik next untuk melanjutkan



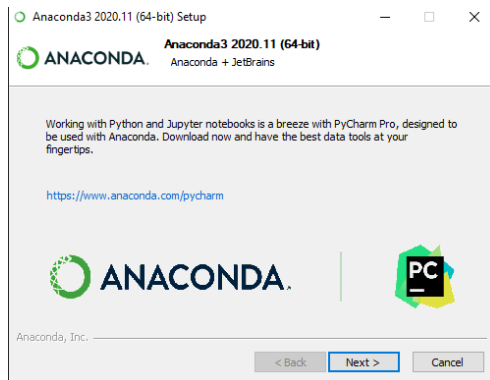
5. Selanjutnya pada tahap ini ceklis kedua pilihan dibawah untuk menambahkan Anaconda pada PATH kita dan mendaftarkan Anaconda sebagai default Python 3.8 kita. Lalu klik install untuk mulai menginstal Anaconda



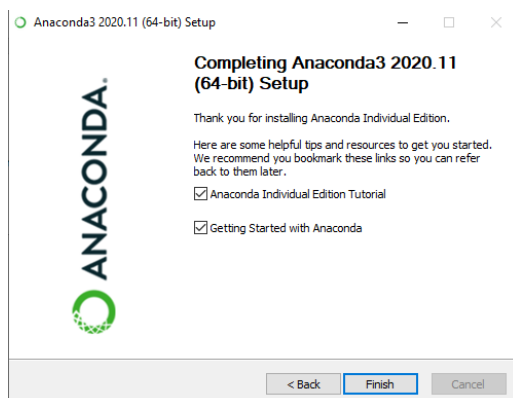
6. Tunggu proses instalasi Anaconda selesai



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 intruksi pada python

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

#### Penjelasan

Variable diatas diberi nama = 'Ucok' kemudian dipanggil variabelnya akan muncul 'Ucok', setelah itu diprint dengan memanggil nama variable kemudian muncul Ucok.

Setelah itu membuat variable baru diberi nama umur = 20, kemudian di print dengan memanggil variable nama dan memanggil variable umur.

```
>>> print('hello world')
hello world
>>>
```

2.

### 3. Variable dan Objek

A

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

#### Penjelasan

Python merupakan Bahasa dengan dynamic typing yaitu variabelnya tidak dibatasi oleh tipe datanya. Sebagai contoh, variabel yang sudah diisi dengan tipe bilangan bulat bisa diisi dengan bilangan riil, string, ataupun tipe data yang lain

B

```
>>> x = 9
>>> id(x)
140723941287968
```

#### 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)
140723941287968
>>> y = 9
>>> id(y)
140723941287968
>>>
```

#### 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)
140723941287968
>>> y = 9
>>> id(y)
140723941287968
>>> y = 7
>>> id(y)
140723941287904
>>>
```

#### 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)
140723941287968
>>> y = 9
>>> id(y)
140723941287968
>>> 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)
140723941287968
>>> 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
(300, 300)
>>> Posisi
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'Posisi' is not defined
>>>

```

### Penjelasan

Penulisan kode program pada python bersifat case sensitive. Dengan demikian misal variable **Posisi** akan berbeda dengan variable **posisi**.

## 5. Perintah Program (Statement)

A

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

### Penjelasan

Pada python setiap kode program yang dituliskan tidak harus diakhiri dengan sebuah statement (**biasanya tanda titik koma**) seperti pada Java dan C. Titik koma pada python hanya diberikan pada saat ada dua atau lebih statement pada satu baris yang sama.

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 hexadecimal
>>> c = 0x2f
>>> a
9
>>> b
19
>>> c
47
>>>
```

### Penjelasan

Dalam python terapat dua tipe bilangan bulat yaitu int dan bool. Selain tipe integral primitive python juga dapat menggunakan bilangan integral dengan basis decimal (10), biner (2), octal (8) maupun heksadesimal (16).

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

### Penjelasan

Tipe bilangan bulat yang kedua adalah tipe Boolean, dimana seperti yang telah kita ketahui tipe data boolean bernilai true atau false saja

```
>>> a = 15
>>> id(a)
140723941288160
>>> a += 5
>>> a
20
>>> id(a)
140723941288320
>>>
```

### 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

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

## 7. Tipe String

### A

```
>>> s1 = 'pemrograman python'
>>> s2 = "pemrograman python 2"
>>> s2 = '''pemrogramman
... python 3'''
>>> s1[0], s1[1], s2[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 = 'Pemrogaman Python dan PyQt'
>>> s1 = s[0:11]
>>> s1
'Pemrogaman '
>>> len(s1)
11
>>>
```

### Penjelasan

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

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

### Penjelasan

## 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.

## **Kesemipulan**

Bahasa Python memiliki keunikannya sendiri dibanding Bahasa-bahasa pemrograman lain, selain lebih simple Bahasa python lebih mudah dipahami ketimbang bahasa pemrograman lain. Bahasa python sendiri memiliki aturan yang berbeda juga dari bahasa yang lain.

