Lab 1.3.3 – Python Programming Review

Tujuan :

Part 1 : Menjalankan DEVASC VM

Part 2 : Start Python dan VS Code

Part 3 : Review tipe data dan variable

Part 4 : Review List dan Kamus data

Part 5 : Review Function input

Part 6 : Review If, For, While

Part 7 : Review Method untuk akses file

Resource yang dibutuhkan :

1 PC /Laptop dengan operating sistem yang ada

Virtual Box / VMWare

DEVASC Virtual Machine

Petunjuk :

Part 1 : Menjalankan DEVASC VM

- Jalankan DEVASC VM yang sebelumnya sudah di import di virtual box/VMWare.

Part 2 : Memulai Python dan VS Code

Langkah 1 : Start Python

a. Untuk memulai python, kita lalukan pengecekan versi python yang ada di DEVASC VM. Buka terminal yang ada di DEVASC VM. Lalu ketikkan perintah **python3 –V**

devasc@labvm:~$ **python3 –V**

Python 3.8.2

Catatan :

Jika versi python masih versi 2, maka anda bisa melakukan update version dengan mengetikkan **sudo apt-get install python3**

b. Untuk mulai menggunakan python, jika kita sudah berada dalam bahasa pemrograman python ditandai dengan tanda “**>>>**”.

devasc@labvm~$ **python3**

Python 3.8.2 (default, Mar 13 2020, 10:14:16)

[GCC 9.3.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

>>>

Langkah 2 : Menggunakan interpreter sebagai calculator

a. Mencoba operasi matematika

Cobalah beberapa operasi matematika dibawah ini :

```
>>> 2+3
5
>>> 10-4
6
>>> 2*4
8
>>> 20/5
4.0
>>> 3**2
9
```

Langkah 3 : menggunakan interpreter untuk menampilkan string

String ditandai dengan **single quote (‘)** atau **double quote (“)**.

a. Ketikkan “Hello World” or ‘Hello World’

```
>>> "Hello World!"
'Hello World!'
>>> 'Hello World!'
'Hello World!'
```

b. Fungsi **print** dapat digunakan untuk menampilkan string.

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

c. Untuk keluar dari interactive interpreter, ketikkan **quit()**

```
>>> quit()
devasc@labvm:~$
```

Langkah 4 : Open VS Code dan buat script hello world

a. Open VS Code. (Cari di desktop DEVASC VM)

b. Setelah terbuka VS Codenya, klik **File > NewFile,** untuk membuka file baru.

c. Di dalam file baru tersebut, kemudian ketikkan print(“Hello World!”)

d. Simpan file baru /script tsb dengan nama **hello-world.py** di letakkan dalam directory : **labs/devnet-erc/python**

e. Jalankan script yang sudah anda buat dengan cara klik **Run > Run Without Debugging**.

f. Jika menjalan script melalui terminal, caranya :

- Masuk ke dalam directory iini : **labs/devnet-erc/python.** Dengan perintah cd.
- Setelah berada di directory tsb, maka ketikkan **python3 hello-world.py**

devasc@labvm:~/labs/devnet-src/python$ python3 hello-world.py

Hello World!

devasc@labvm:~/labs/devnet-src/python$

Part 3 : Review tipe data dan variable.

Langkah 1 : gunakan interactive interpreteur untuk mencoba beberapa type data dan variable.

Ketikkan seperti dibawah ini :

```
>>> type(98)
<class 'int'>
>>> type(98.6)
<class 'float'>
>>> type("Hi!")
<class 'str'>
>>> type(True)
<class 'bool'>
```

Langkah 2 : Review tipe data Boolean yang digunakan sebagai operator.

```
>>> 1<2
True>>> 1<1
False
>>> 1==1
True
>>> 1>=1
True
>>> 1<=1
True
```

Langkah 3 : menggunakan interpreteur untuk membuat variable

>>> **x=3**

>>> **x*5**

15

>>> **"Cisco"*x**

'CiscoCiscoCisco'

Langkah 4 : menggunakan interpreter untuk menggabungkan string

>>> **str1="Cisco"**

>>> **str2="Networking"**

>>> **str3="Academy"**

>>> **space=" "**

>>> **print(str1+space+str2+space+str3)**

Cisco Networking Academy

Jika tanda menggunakan variable space, dapat menggunakan “,”

>>> **print(str1,str2,str3)**

Cisco Networking Academy

Langkah 5 : review casting dan printing dengan tipe data yang berbeda.

a. Casting adalah converting / mengubah tipe data . contohnya dibawah ini silahkan di coba :

   >>> **x=3**

   >>> **print("The value of x is " + x)**

   Traceback (most recent call last):

   File "<stdin>", line 1, in <module>

   TypeError: can only concatenate str (not "int") to str

   >>>

b. Gunakan fungsi str(), untuk mengubah tipe data interger ke string

   >>> **print("The value of x is " + str(x))**

   The value of x is 3

   >>> **type(x)**

   <class 'int'>

c. Catatan : Tipe data untuk variable x masing integer, untuk mengconvert tipe data, deklarasikan kembali variable tsb ke tipe data yang baru.

   >>> **x=str(x)**

   >>> **type(x)**

   <class 'str'>

d. Tipe data float

```
>>> num = 22/7
>>> f"The value of num is  {num}"
'The value of num is 3.142857142857143'
>>> pi = "{:.2f}".format(num)
>>> f"The value of pi is {pi}."
'The value of pi is 3.14.'
>>>
```

Part 4 : Review List dan Kamus Data

Langkah 1 : Membuat dan mengubah list

a. List disebut juga dengan array.

Coba script dibawah ini :

```
>>> hostnames=["R1","R2","R3","S1","S2"]
>>> type(hostnames)
<class 'list'>
>>> len(hostnames)
5
>>> hostnames
['R1', 'R2', 'R3', 'S1', 'S2']
```

b. Menampilkan index dari array

```
>>> hostnames[0]
'R1'
>>> hostnames[-1]
'S2'
>>> hostnames[0]="RTR1"
>>> hostnames
['RTR1', 'R2', 'R3', 'S1', 'S2']
>>> del hostnames[3]
>>> hostnames
['RTR1', 'R2', 'R3', 'S2']
>>>
```

Langkah 2 : membuat dan mengubah dictionary

a. Dictionary adalah mengacak array dari sebuah objek. Tiap objek terdiri atas sebuah key/nilai berpasangan.

- Membuat dictionary dengan braces {}

  Coba script dibawah ini :

  >>> **ipAddress={"R1":"10.1.1.1","R2":"10.2.2.1","R3":"10.3.3.1"}**

  >>> **type(ipAddress)**

  <class 'dict'>

b. >>> **ipAddress**

  {'R1': '10.1.1.1', 'R2': '10.2.2.1', 'R3': '10.3.3.1'}

  >>> **ipAddress['R1']**

  '10.1.1.1'

  >>> **ipAddress["S1"]="10.1.1.10"**

  >>> **ipAddress**

  {'R1': '10.1.1.1', 'R2': '10.2.2.1', 'R3': '10.3.3.1', 'S1': '10.1.1.10'}

  >>>

c. >>> **ipAddress["R3"]=["10.3.3.1","10.3.3.2","10.3.3.3"]**

  >>> **ipAddress**

  {'S1': '10.1.1.10', 'R2': '10.2.2.1', 'R1': '10.1.1.1', 'R3': ['10.3.3.1', '10.3.3.2', '10.3.3.3']}

  >>>

Part 5 : Review Fungsi Input

Langkah 1 : Membuat variable untuk menyimpan inputan user dan kemudian menampilkan isinya .

Tuliskan script ni melalui interpreter/ melalui terminal dalam python.

>>> **firstName = input("What is your first name? ")**

What is your first name? **User_Name**

>>> **print("Hello " + firstName +"!")**

Hello User_Name!

>>>

Langkah 2 : Membuat script untuk mengumpulkan informasi personal

a. Buka VS Code

b. Open new file – simpan dengan nama : **personal-info.py** letakkan di dalam directory : **~/labs/devnet-src/python**.

c. Tuliskan script dibawah ini di dalam file **personal-info.py**

```
firstName = input("What is your first name? ")
lastName = input("What is your last name? ")
location = input("What is your location? ")
age = input("What is your age? ")
print("Hi " + firstName, lastName + "! Your location is " + location + " and
you are " + age + " years old.")
```

d. Run **personal-info.py** melalui terminal dengan cara :

devasc@labvm:~/labs/devnet-src$ **python3 person-info.py**

What is your first name? **Bob**

What is your last name? **Smith**

What is your location? **London**

What is your age? **36**

Hi Bob Smith! Your location is London and you are 36 years old.

devasc@labvm:~/labs/devnet-src$ ^C

Part 6 : Review fungsi If, For dan While

Langkah 1 : Buat Fungsi If/else

a. Open new file dan simpan dengan nama file : **if-vlan.py.**

```
nativeVLAN = 1
dataVLAN = 100
if nativeVLAN == dataVLAN:
 print("The native VLAN and the data VLAN are the same.")
else:
 print("The native VLAN and the data VLAN are different.")
```

b. Run script diatas.

Hasil run seperti dibawah ini :

```
The native VLAN and the data VLAN are different.
```

c. Modify **nativeVLAN** dan **dataVLAN,** lalu run lagi,

```
The native VLAN and the data VLAN are the same.
```

Langkah 2 : create function if/elif/else

a. Create new file dan simpan dengan nama : **if-acl.py**. tuliskan script dibawah ini :

```
aclNum = int(input("What is the IPv4 ACL number? "))
if aclNum >= 1 and aclNum <= 99:
      print("This is a standard IPv4 ACL.")
elif aclNum >=100 and aclNum <= 199:
```

```
        print("This is an extended IPv4 ACL.")
    else:
         print("This is not a standard or extended IPv4 ACL.")
```

b. Run **if-acl.py,**

devasc@labvm:~/labs/devnet-src/python$ **python3 if-acl.py**

What is the IPv4 ACL number? **10**

This is a standard IPv4 ACL.

devasc@labvm:~/labs/devnet-src/python$ **python3 if-acl.py**

What is the IPv4 ACL number? **110**

This is an extended IPv4 ACL.

devasc@labvm:~/labs/devnet-src/python$ **python3 if-acl.py**

What is the IPv4 ACL number? **200**

This is not a standard or extended IPv4 ACL.

devasc@labvm:~/labs/devnet-src/python$

Langkah 3 : Create for loop

Lakukan langkah dibawah ini :

a. devasc@labvm:~/labs/devnet-src/python$ **python3**

Python 3.8.2 (default, Mar 13 2020, 10:14:16)

[GCC 9.3.0] on linux

Type "help", "copyright", "credits" or "license" for more information.>>>

devices=["R1","R2","R3","S1","S2"]

>>> **for item in devices:**

... print(item)

...

R1

R2

R3

S1

S2

>>>

b. jalankan script lagi dan tambahkan if

devasc@labvm:~/labs/devnet-src/python$ python3

```
Python 3.8.2 (default, Mar 13 2020, 10:14:16)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.>>>
devices=["R1","R2","R3","S1","S2"]
>>> for item in devices:
...         if “R” in item:
...             print(item)
…
R1
R2
R3
>>>
```

c. tambahkan fungsi append()

```
>>> switches=[]
>>> for item in devices:
...         if "S" in item:
...                     switches.append(item)
...
>>> switches
['S1', 'S2']
>>>
```

Langkah 4 : While Loop

a. open new file dan simpan file dengan nama: **while-loop.py**

masukkan script ini :

```
x=input("Enter a number to count to: ")
x=int(x)
y=1
while y<=x:
 print(y)
 y=y+1
```

b. save dan run script diatas.

```
devasc@labvm:~/labs/devnet-src/python$ python3 while-loop.py
Enter a number to count to: 10
1
2
3
4
5
6
7
8
9
10
devasc@labvm:~/labs/devnet-src/python$
```

c. modifikasi script diatas dengan menambahkan **fungsi while y <=x**,

```
x=input("Enter a number to count to: ")
x=int(x)
y=1
while True:
        print(y)
        y=y+1
        if y>x:
                break
```

d. save dan run script.

Langkah 5 : menggunakan while – loop

a. modify file while-loop.py

```
while True:
        x=input("Enter a number to count to: ")
        if x == 'q' or x == 'quit':
                break
        x=int(x)
        y=1
```

```
while True:
        print(y)
        y=y+1
        if y>x:
                break
```

b. save dan run script

devasc@labvm:~/labs/devnet-src/python$ python3 while-loop.py

Enter a number to count to: **3**

1

2

3

Enter a number to count to: **5**

1

2

3

4

5

Enter a number to count to: **quit**

devasc@labvm:~/labs/devnet-src/python$

Bagian 7 : Review method untuk file access

Langkah 1 : buat program untuk membaca sebuah file external

Method yang digunakan adalah : open()

a. buat file baru dan beri nama : **file-access.py**
b. tambahkan script dibawah ini :

```
file=open("devices.txt","r")
for item in file:
    print(item)
file.close()
```

c. save and run, dan hasilnya harus seperti dibawah ini

devasc@labvm:~/labs/devnet-src/python$ **python3 file-access.py**

Cisco 819 Router

Cisco 881 Router

Cisco 888 Router

Cisco 1100 Router

Cisco 4321 Router

Cisco 4331 Router

Cisco 4351 Router

Cisco 2960 Catalyst Switch

Cisco 3850 Catalyst Switch

Cisco 7700 Nexus Switch

Cisco Meraki MS220-8 Cloud Managed Switch

Cisco Meraki MX64W Security Appliance

Cisco Meraki MX84 Security Appliance

Cisco Meraki MC74 VoIP Phone

Cisco 3860 Catalyst Switch

devasc@labvm:~/labs/devnet-src/python$

Langkah 2 : Remove line kosong dari hasil/output

a. edit file **file-access.py** dengan memasukkan method **strip()**

```
file=open("devices.txt","r")
for item in file:
        item=item.strip()
        print(item)
file.close()
```

b. Save and run,

devasc@labvm:~/labs/devnet-src/python$ **python3 file-access.py**

Cisco 819 Router
Cisco 881 Router
Cisco 888 Router
Cisco 1100 Router
Cisco 4321 Router
Cisco 4331 Router
Cisco 4351 Router
Cisco 2960 Catalyst Switch
Cisco 3850 Catalyst Switch
Cisco 7700 Nexus Switch

Cisco Meraki MS220-8 Cloud Managed Switch
Cisco Meraki MX64W Security Appliance
Cisco Meraki MX84 Security Appliance
Cisco Meraki MC74 VoIP Phone
Cisco 3860 Catalyst Switch
devasc@labvm:~/labs/devnet-src/python$

Langkah 3 : copy content file ke dalam list variable

a. Modify file-access.py :

```
devices=[]
file=open("devices.txt","r")
for item in file:
    item=item.strip()
    devices.append(item)
file.close()
print(devices)
```

b. save and run

devasc@labvm:~/labs/devnet-src/python$ **python3 file-access.py**

['Cisco 819 Router', 'Cisco 881 Router', 'Cisco 888 Router', 'Cisco 1100 Router', 'Cisco 4321 Router', 'Cisco 4331 Router', 'Cisco 4351 Router', 'Cisco 2960 Catalyst Switch', 'Cisco 3850 Catalyst Switch', 'Cisco 7700 Nexus Switch', 'Cisco Meraki MS220-8 Cloud Managed Switch', 'Cisco Meraki MX64W Security Appliance', 'Cisco Meraki MX84 Security Appliance', 'Cisco Meraki MC74 VoIP Phone', 'Cisco 3860 Catalyst Switch']
devasc@labvm:~/labs/devnet-src/python$

laangkah 4 : Challenge , dan buat sebuah script yang dapat mengijinkan user untuk menambahkan device.

1. Open new file dan simpan dengan nama : **file-access-input.py**
2.
```
file = open("devices.txt", "a")
while True:
    newItem = input("Enter device name: ")
    if newItem == "exit":
        print("All done!")
        break
    file.write(newItem  +  "\n")
file.close()
```
3. save and run

devasc@labvm:~/labs/devnet-src/python$ **python3 file-access-input.py**

Enter device name: **Cisco 1941 Router**

Enter device name: **Cisco 2950 Catalyst Switch**

Enter device name: **exit**

All done!

devasc@labvm:~/labs/devnet-src/python$