MODUL 3

DASAR PYTHON 3

3.1. Tujuan Praktikum:

Adapun Tujuan pada pratikum modul 3 yaitu:

1. Mampu memahami konsep dari python arrays, classes and object, inheritance, iterators, scope, modules, dates, math, JSON, RegEx.
2. Dapat mempraktikkan penerapan dari python arrays, classes and object, inheritance, iterators, scope, modules, dates, math, JSON, RegEx.

3.2. Alat dan Bahan Praktikum

1. Laptop

2. Mouse

3. Software *Visual Studio Code*

4. *Python*

3.3. Dasar Teori

3.3.1. Python Arrays

Array Python adalah kumpulan item yang digunakan untuk menyimpan beberapa nilai dari jenis yang sama bersama-sama. Python memiliki serangkaian metode bawaan yang dapat Anda gunakan pada daftar/array.

Arrays Method

Methods	Deskripsi
append()	Menambahkan elemen di akhir daftar
clear()	Menghapus semua elemen dari daftar
copy()	Mengembalikan salinan daftar
count()	Mengembalikan jumlah elemen dengan nilai yang ditentukan
extend()	Menambahkan elemen daftar (atau yang dapat diulang), ke akhir daftar saat ini

index()	Mengembalikan indeks elemen pertama dengan nilai yang ditentukan
insert()	Menambahkan elemen pada posisi yang ditentukan

pop()	Menghapus elemen pada posisi yang ditentukan
remove()	Menghapus item pertama dengan nilai yang ditentukan
reverse()	Membalikkan urutan daftar
sort()	Mengurutkan daftar

Contoh penggunaan Array

```
cars = ["Ford", "Volvo", "BMW"]

x = cars[0]

print(cars)
```

Hasil: `Ford`

3.3.2. Python Classes and Objects

Class adalah struktur data yang ditentukan pengguna yang mengikat anggota dan metode data ke dalam satu unit. Class adalah *blueprint* atau template kode untuk pembuatan objek. Dengan menggunakan class, Anda dapat membuat objek sebanyak yang Anda inginkan. Objek adalah contoh dari kelas.

Contoh pembuatan Class dan Object:

```
class MyClass:
  x = 5

p1 = MyClass()

print(p1.x)
```

Hasil: `5`

3.3.3. Python Inheritance

Python Inheritance memungkinkan kita untuk mendefinisikan kelas yang mewarisi semua metode dan properti dari kelas lain. Python Inheritance adalah fitur yang kuat dalam pemrograman berorientasi objek. Ini mengacu pada mendefinisikan kelas baru dengan sedikit atau tanpa modifikasi pada kelas yang ada.

Contoh Python Inheritance:

```
class Person:
  def init (self, fname, lname):
    self.firstname = fname
    self.lastname = lname
  def printname(self):
    print(self.firstname, self.lastname)


#Use the Person class to create an object, and then
execute the printname method:
x = Person("John", "Doe")
x.printname()
```

Hasil: `John Doe`

3.3.4. Python Iterators

Iterator dalam Python hanyalah objek yang dapat diulang. Objek yang akan mengembalikan data, satu elemen pada satu waktu. Secara teknis, objek iterator Python harus menerapkan dua metode khusus, __iter () dan __next (), yang secara kolektif disebut protokol iterator. Sebuah objek disebut iterable jika kita bisa mendapatkan iterator darinya.

Contoh Python Iterators

```
class MyNumbers:
  def iter (self):
    self.a = 1
    return self
  def next (self):
    x = self.a
```

```
        self.a += 1
        return x
myclass = MyNumbers()
myiter = iter(myclass)
print(next(myiter))
print(next(myiter))
print(next(myiter))
print(next(myiter))
print(next(myiter))
```

Hasil:

```
1
2
3
4
5
```

3.3.5. Python Scope

Ketika kita mendefinisikan variabel, fungsi atau nama kelas dalam suatu program, Ini hanya dapat diakses di wilayah tertentu dari program. Wilayah tertentu ini di mana nama, setelah didefinisikan, dapat digunakan untuk mengidentifikasi objek, variabel atau fungsi yang disebut ruang lingkup.

Contoh python scope:

```
x = 300
def myfunc():
  print(x)
myfunc()
print(x)
```

Hasil: `300 300`

3.3.6. Python Modules

Modul adalah file kode sumber Python dengan ekstensi .py. Nama modul adalah nama file Python tanpa ekstensi. Untuk menggunakan objek dari modul,

Anda mengimpornya menggunakan pernyataan impor.

Contoh penggunaan Python Modul:

```
def greeting(name):
  print("Hello, " + name)
```

simpan code diatas dengan nama mymodule.py

```
import mymodule
mymodule.greeting("Jonathan")
```

Hasil: `Hello, Jonathan`

3.3.7. Python Datetime

Datetime Python adalah kombinasi antara tanggal dan waktu. Atribut kelas ini mirip dengan kelas tanggal dan terpisah. Atribut ini termasuk hari, bulan, tahun, menit, kedua, mikrodetik, jam, dan tzinfo.

Contoh penggunaan Python Datetime:

```
import datetime
x = datetime.datetime.now()
print(x)
```

Hasil: `2022-10-04 20:13:48.341609`

3.3.8. Python Math

Python juga memiliki modul bawaan yang disebut matematika, yang memperluas daftar fungsi matematika. Untuk menggunakannya, Anda harus mengimpor modul matematika: impor matematika. Ketika Anda telah mengimpor modul matematika, Anda dapat mulai menggunakan metode dan konstanta modul.

Contoh penggunaan Python Match:

```
x = min(5, 10, 25)
y = max(5, 10, 25)
print(x)
print(y)
```

Hasil: `5 25`

3.3.9. Python JSON

JSON (JavaScript Object Notation) adalah file yang terutama digunakan untuk menyimpan dan mentransfer data sebagian besar antara server dan aplikasi web.

Python menyediakan modul yang disebut json yang dilengkapi dengan utilitas bawaan standar Python. Saat Anda mengonversi dari Python ke JSON, objek Python dikonversi menjadi JSON (JavaScript) yang setara yiatu:

Python	JSON
dict	Object
list	Array
tuple	Array
str	String

int	Number
float	Number
true	True
false	False
none	Null

Contoh Python JSON:

```
import json

# some JSON:

x = '{ "name":"John", "age":30, "city":"New
York"}' # parse x:
y = json.loads(x)

# the result is a Python dictionary:

print(y["age"])
```

Hasil: `30`

3.3.10. Python RegEX

Ekspresi reguler (atau regex) adalah urutan karakter yang menentukan pola pencarian. Dalam praktiknya, Anda akan menemukan ekspresi reguler di banyak aplikasi seperti mesin pencari, mencari dan mengganti dialog editor teks. Modul *re* menawarkan serangkaian fungsi yang memungkinkan kita mencari string untuk kecocokan yaitu:

Function	Deskripsi
findall	Mengembalikan daftar yang berisi semua kecocokan
search	Mengembalikan objek cocokkan jika ada kecocokan di mana saja dalam string
split	Mengembalikan daftar di mana string telah dibagi di setiap pertandingan
sub	Mengganti satu atau banyak kecocokan dengan string

Contoh Python RegEX:

```
import re

#Check if the string starts with "The" and ends with
"Spain": txt = "The rain in Spain"
x = re.search("^The.*Spain$",
txt) if x:
  print("YES! We have a
match!") else:
  print("No match")
```

Hasil: `YES! We have a match!`