

PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

MODUL 2

MENGENAL OOP PADA PYTHON



Oleh : Fariz Taufiqul Hafidz

Nim : L200210192

Kelas : F

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA**

2023

1. Pada Contoh 2.2, kita telah membuat class Pesan yang berisi beberapa metode. Bahkan metode-metode di bawah ini ke dalam class itu.
 - a) Metode untuk memeriksa apakah suatu string terkandung di object Pesan itu. Seperti ini hasilnya :

Modul 2.py - E:\Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul 2.py (3.10.10)
File Edit Format Run Options Window Help

```
### NOMOR 1 ###  
  
class Pesan(object):  
  
    def __init__(self, sebuahString):  
        self.teks = sebuahString  
    def cetakIni(self):  
        print(self.teks)  
    def cetakPakaiHurufKapital(self):  
        print(str.upper(self.teks))  
    def cetakPakaiHurufKecil(self):  
        print(str.lower(self.teks))  
    def jumKar(self):  
        return len(self.teks)  
    def cetakJumlahKarakterku(self):  
        print('kalimatku mempunyai', len(self.teks), 'karakter')  
    def perbarui(self, stringBaru):  
        self.teks = stringBaru  
    def apakahTerkandung(self, kata):  
        if kata in self.teks:  
            return True  
        else:  
            return False
```

Output :

IDLE Shell 3.10.10
File Edit Shell Debug Options Window Help
Python 3.10.10 (tags/v3.10.10:aad5f6a, Feb 7 2023, 17:20:36) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> = RESTART: E:\Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul 2 .py
>>> p9 = Pesan('Indonesia adalah negeri yang indah')
>>> p9.apakahTerkandung('ege')
True
>>> p9.apakahTerkandung('eka')
False

b) Metode untuk menghitung jumlah konsonan.

Modul 2.py - E:\Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul 2.py (3.10.10)

File Edit Format Run Options Window Help

```
### NOMOR 1 ###

class Pesan(object):

    def __init__(self, sebuahString):
        self.teks = sebuahString

    def cetakIni(self):
        print(self.teks)

    def cetakPakaiHurufKapital(self):
        print(str.upper(self.teks))

    def cetakPakaiHurufKecil(self):
        print(str.lower(self.teks))

    def jumKar(self):
        return len(self.teks)

    def cetakJumlahKarakterku(self):
        print('kalimatku mempunyai', len(self.teks), 'karakter')

    def perbarui(self, stringBaru):
        self.teks = stringBaru

    def apakahTerkandung(self, kata):
        if kata in self.teks:
            return True
        else:
            return False

    def hitungKonsonan(self):
        vokal = ['a', 'i', 'u', 'e', 'o', 'A', 'I', 'U', 'E', 'O']
        jml_Konsonan = 0
        for i in self.teks:
            if i not in vokal and i != ' ':
                jml_Konsonan += 1
        return jml_Konsonan
```

Output :

```
>>> p10 = Pesan('Surakarta')
>>> p10.hitungKonsonan()
5
```

c) Metode untuk menghitung jumlah huruf vokal.

Modul 2.py - E:\Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul 2.py (3.10.10)

File Edit Format Run Options Window Help

NOMOR 1

```
class Pesan(object):

    def __init__(self, sebuahString):
        self.teks = sebuahString
    def cetakIni(self):
        print(self.teks)
    def cetakPakaiHurufKapital(self):
        print(str.upper(self.teks))
    def cetakPakaiHurufKecil(self):
        print(str.lower(self.teks))
    def jumKar(self):
        return len(self.teks)
    def cetakJumlahKarakterku(self):
        print('kalimatku mempunyai', len(self.teks), 'karakter')
    def perbarui(self, stringBaru):
        self.teks = stringBaru
    def apakahTerkandung(self, kata):
        if kata in self.teks:
            return True
        else:
            return False

    def hitungKonsonan(self):
        vokal = ['a','i','u','e','o','A','I','U','E','O']
        jml_Konsonan = 0
        for i in self.teks:
            if i not in vokal and i != ' ':
                jml_Konsonan += 1
        return jml_Konsonan


    def hitungVokal(self):
        vokal = ['a','i','u','e','o','A','I','U','E','O']
        jml_Vokal = 0
        for i in self.teks:
            if i in vokal and i != ' ':
                jml_Vokal += 1
        return jml_Vokal
```

Output :

```
>>> p10.hitungVokal()
4
>>>
```

2. Lihat kembali contoh 2.4. Tambahkan beberapa metode seperti dijelaskan di bawah ini.

- a) Metode untuk mengambil kota tempat tinggal si mahasiswa. Seperti ini hasilnya :

 *Modul 2.py - E:\Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul 2.py (3.10.10)*

File Edit Format Run Options Window Help

```
### NOMOR 2,3,4,5 ###

class Manusia(object):
    keadaan = 'lapar'
    def __init__(self,nama):
        self.nama = nama
    def ucapkanSalam(self):
        print("Salam Namaku", self.nama)
    def makan(self,s):
        print("Saya baru saja makan",s)
        self.keadaan = 'kenyang'
    def olahraga(self,k):
        print("Saya baru saja latihan",k)
        self.keadaan = 'lapar'
    def mengalikanDenganDua(self,n):
        return n*2

class Mahasiswa(Manusia):
    def __init__(self,nama,NIM,kota,us):
        self.nama = nama
        self.NIM = NIM
        self.kotaTinggal = kota
        self.uangSaku = us
    def __str__(self):
        s = self.nama + ',NIM' + str(self.NIM) \
            + '.Tinggal di' + self.kotaTinggal \
            + '.Uang saku' + str(self.uangSaku) \
            + ' tiap bulannya.'
        return s
    def ambilNama(self):
        return self.nama
    def ambilNIM(self):
        return self.NIM
    def ambilUang(self):
        return self.uangSaku
    def makan(self,s):
        print("Saya baru saja makan",s,"sambil belajar")
        self.keadaan = 'kenyang'
    def ambilKota(self):
        return self.kotaTinggal

m1 = Mahasiswa('Fariz',100,'Pati',10000)
m2 = Mahasiswa('Taufiqul',101,'Semarang',20000)
m3 = Mahasiswa('Hafidz',102,'Kudus',30000)
```

Output :

```
>>> m3.ambilKota()  
'Kudus'  
>>> |
```

b) Metode untuk memperbarui kota tinggal. Seperti ini hasilnya :

```
class Mahasiswa (Manusia):  
    def __init__(self,nama,NIM,kota,us):  
        self.nama = nama  
        self.NIM = NIM  
        self.kotaTinggal = kota  
        self.uangSaku = us  
    def __str__(self):  
        s = self.nama + ',NIM' + str(self.NIM) \  
            + '.Tinggal di' + self.kotaTinggal \  
            + '.Uang saku' + str(self.uangSaku) \  
            + ' tiap bulannya.'  
        return s  
    def ambilNama(self):  
        return self.nama  
    def ambilNIM(self):  
        return self.NIM  
    def ambilUang(self):  
        return self.uangSaku  
    def makan(self,s):  
        print("Saya baru saja makan",s,"sambil belajar")  
        self.keadaan = 'kenyang'  
    def ambilKota(self):  
        return self.kotaTinggal  
    def perbaruiKota(self,k):  
        self.kotaTinggal = k  
        return self.kotaTinggal  
  
m1 = Mahasiswa('Fariz',100,'Pati',10000)  
m2 = Mahasiswa('Taufiqul',101,'Semarang',20000)  
m3 = Mahasiswa('Hafidz',102,'Kudus',30000)
```

Output :

```
>>> m3.perbaruiKota('Surakarta')  
'Surakarta'  
>>> m3.ambilKota()  
'Surakarta'  
>>> |
```

c) Metode untuk menambah uang saku. Seperti ini hasilnya :

```
class Mahasiswa (Manusia):
    def __init__(self,nama,NIM,kota,us):
        self.nama = nama
        self.NIM = NIM
        self.kotaTinggal = kota
        self.uangSaku = us
    def __str__(self):
        s = self.nama + ',NIM' + str(self.NIM) \
            + '.Tinggal di' + self.kotaTinggal \
            + '.Uang saku' + str(self.uangSaku) \
            + ' tiap bulannya.'
        return s
    def ambilNama(self):
        return self.nama
    def ambilNIM(self):
        return self.NIM
    def ambilUang(self):
        return self.uangSaku
    def makan(self,s):
        print("Saya baru saja makan",s,"sambil belajar")
        self.keadaan = 'kenyang'
    def ambilKota(self):
        return self.kotaTinggal
    def perbaruiKota(self,k):
        self.kotaTinggal = k
        return self.kotaTinggal
    def tambahUang(self,u):
        u = int(u)
        jumlah = self.uangSaku + u
        self.uangSaku = jumlah
        return self.uangSaku

m1 = Mahasiswa('Fariz',100,'Pati',10000)
m2 = Mahasiswa('Taufiqul',101,'Semarang',20000)
m3 = Mahasiswa('Hafidz',102,'Kudus',30000)
```

Output :

```
>>> m3.ambilUang()
30000
>>> m3.tambahUang(5000)
35000
>>> m3.ambilUang()
35000
>>> |
```

3. Masih di contoh 2.4. Buatlah suatu program untuk memasukkan data mahasiswa baru lewat Python Shell secara interaktif. Seperti sudah kamu duga, gunakanlah `input()`.

```
>>> a = input('Masukkan nama : ')
Masukkan nama : Giska
>>> b = input('Masukkan nim : ')
Masukkan nim : 8912
>>> c = input('Masukkan kota : ')
Masukkan kota : Surakarta
>>> d = input('Masukkan uang saku : ')
Masukkan uang saku : 50000
>>> m4 = Mahasiswa(a,b,c,d)
>>> m4.ambilNama()
'Giska'
>>> m4.ambilNIM()
'8912'
>>> m4.ambilKota()
'Surakarta'
>>> m4.ambilUang()
'50000'
>>>
```

4. Buatlah state baru di class Mahasiswa bernama `listKuliah` yang berupa list berisi daftar matakuliah yang diambil. Buat pula metode `ambilKuliah()` yang akan menambah daftar matakuliah ini. Contoh pemanggilan :

```
class Mahasiswa(Manusia):
    def __init__(self,nama,NIM,kota,us):
        self.nama = nama
        self.NIM = NIM
        self.kotaTinggal = kota
        self.uangSaku = us
    def __str__(self):
        s = self.nama + ',NIM' + str(self.NIM) \
            + '.Tinggal di' + self.kotaTinggal \
            + '.Uang saku' + str(self.uangSaku) \
            + ' tiap bulannya.'
        return s
    def ambilNama(self):
        return self.nama
    def ambilNIM(self):
        return self.NIM
    def ambilUang(self):
        return self.uangSaku
    def makan(self,s):
        print("Saya baru saja makan",s,"sambil belajar")
        self.keadaan = 'kenyang'
    def ambilKota(self):
        return self.kotaTinggal
    def perbaruiKota(self,k):
        self.kotaTinggal = k
        return self.kotaTinggal
    def tambahUang(self,u):
        u = int(u)
        jumlah = self.uangSaku + u
        return jumlah

    listKuliah = []
    def ambilkuliah(self,m):
        self.listKuliah.append(m)

m1 = Mahasiswa('Fariz',100,'Pati',10000)
m2 = Mahasiswa('Taufiqul',101,'Semarang',20000)
m3 = Mahasiswa('Hafidz',102,'Kudus',30000)
```


Output :

```
>>> m3.listKuliah
[]
>>> m3.ambilkuliah('Prak ASD')
>>> m3.listKuliah
['Prak ASD']
>>> m3.ambilkuliah('ASD')
>>> m3.listKuliah
['Prak ASD', 'ASD']
>>>
```

5. Berkaitan dengan nomer sebelumnya, buatlah metode untuk menghapus sebuah matakuliah dari listKuliah.

```
class Mahasiswa(Manusia):
    def __init__(self,nama,NIM,kota,us):
        self.nama = nama
        self.NIM = NIM
        self.kotaTinggal = kota
        self.uangSaku = us
    def __str__(self):
        s = self.nama + ',NIM' + str(self.NIM) \
            + '.Tinggal di' + self.kotaTinggal \
            + '.Uang saku' + str(self.uangSaku) \
            + ' tiap bulannya.'
        return s
    def ambilNama(self):
        return self.nama
    def ambilNIM(self):
        return self.NIM
    def ambilUang(self):
        return self.uangSaku
    def makan(self,s):
        print("Saya baru saja makan",s,"sambil belajar")
        self.keadaan = 'kenyang'
    def ambilKota(self):
        return self.kotaTinggal
    def perbaruiKota(self,k):
        self.kotaTinggal = k
        return self.kotaTinggal
    def tambahUang(self,u):
        u = int(u)
        jumlah = self.uangSaku + u
        return jumlah

    listKuliah = []
    def ambilkuliah(self,m):
        self.listKuliah.append(m)
    def hapusKuliah(self,x):
        self.listKuliah.remove(x)

m1 = Mahasiswa('Fariz',100,'Pati',10000)
m2 = Mahasiswa('Taufiqul',101,'Semarang',20000)
m3 = Mahasiswa('Hafidz',102,'Kudus',30000)
```

Output :

```
>>> m3.listKuliah
['Prak ASD', 'ASD']
>>> m3.hapusKuliah('Prak ASD')
>>> m3.listKuliah
['ASD']
>>> |
```

Ln: 85 Col: 0

6. Dari class Manusia, buatlah sebuah class SiswaSMA yang memuat metode-metode baru(kamu bebas menentukan).

```
###  NOMOR 6  ###

class siswaSMA(Manusia):

    def __init__(self,nama,no_induk,kelas,alamat):
        self.nama = nama
        self.nomor = no_induk
        self.kelas = kelas
        self.alamat = alamat

    def __str__(self):
        a = "Nama : "+self.nama \
            +"No Induk : "+str(self.no) \
            +"Tinggal di : "+self.alamat \
            +"Kelas : "+str(self.kelas)
        return a

    def ambilNama(self):
        return self.nama

    def ambilNoInduk(self):
        return self.nomor

    def ambilKelas(self):
        return self.kelas

    def ambilAlamat(self):
        return self.alamat

s1 = siswaSMA('Hafidz',205,'XII','Kudus')
```

Output :

```
>>> s1.ambilNama()
'Hafidz'
>>> s1.ambilNoInduk()
205
>>> s1.ambilKelas()
'XII'
>>> s1.ambilAlamat()
'Kudus'
>>> |
```

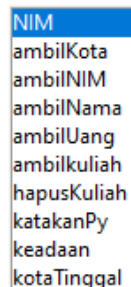
Ln: 94 Col: 0

7. Dengan membuat suatu instance dari class MhsTIF (halaman 25), beri keterangan pada setiap metode dan state yang tampak di object itu (lihat gambar di bawah): apakah metode/state itu berasal dari class Manusia, Mahasiswa, atau MhsTIF ?

```
###  NOMOR 7  ###
```

```
class MhsTIF(Mahasiswa):  
    def katakanPy(self):  
        print('Python is cool.')
```

```
= RESTART: E:\Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul 2  
.PY  
>>> m5 = MhsTIF('Zelly',105,'Sragen',45000)  
>>> m5.
```



NIM
ambilKota
ambilNIM
ambilNama
ambilUang
ambilkuliah
hapusKuliah
katakanPy
keadaan
kotaTinggal

- Dari class Manusia:
 - nama
 - keadaan
 - ucapkanSalam
 - makan
 - olahraga
 - mengalikanDenganDua
- Dari class Mahasiswa:
 - NIM
 - kotaTinggal
 - uangsaku
 - ambilNama
 - ambilNIM

- ambilUang
- ambilKota
- perbaruiKota
- tambahUang
- listKuliah
- ambilKuliah
- hapusKuliah
- Dari class MhsTIF:
 - katakanPy