

Nama : MuchFatan Rahmadan
Nim : L200180061
Kelas : C

MODUL 2

MENGENAL OOP PADA PYTHON

2.1 Module

Latihan 2.1 Sebuah module sederhana. ModulePythonPertamaku.py

Berikut adalah screenshot program dari ModulePythonPertamaku.py:

The screenshot displays two windows from a Windows 10 desktop environment. The top-left window is a 'Python 3.7.6 Shell' showing the execution of a script. The top-right window is a text editor showing the source code of the script. The bottom-left window is another 'Python 3.7.6 Shell' showing the execution of the script after a restart. The bottom-right window is the same text editor showing the source code.

Python 3.7.6 Shell (Top Left):

```
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/LABSI-14/AppData/Local/Programs/Python/Python37/a.py ====
>>> ucapkansalam()
Assalamu'alaikum!
>>> kuadratkan(8)
64
>>> buah
'mangga'
>>> daftarbaju
['batik', 'loreng', 'resmi berdasi']
>>> jumlahbaju
3
>>> import a as mpp
>>> mpp.ucapkankansalam()
Assalamu'alaikum!
>>> mpp.daftarbaju
['batik', 'loreng', 'resmi berdasi']
>>> mpp.jumlahbaju
3
>>> import kuadratkan, jumlahbaju
Traceback (most recent call last):
  File "<pyshell#9>", line 1, in <module>
    import kuadratkan, jumlahbaju
ModuleNotFoundError: No module named 'kuadratkan'
>>> from baju import kuadratkan, daftarbaju
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    from baju import kuadratkan, daftarbaju
ModuleNotFoundError: No module named 'baju'
>>> from a import kuadratkan, daftarbaju
>>> kuadratkan(7)
49
>>> daftarbaju
['batik', 'loreng', 'resmi berdasi']
>>> from a import ucapkansalam as ucap
>>> ucap()
Assalamu'alaikum!
```

Python File Editor (Top Right):

```
def ucapkansalam():
    print("Assalamu'alaikum!")

def kuadratkan(x):
    return x*x

buah = 'mangga'
daftarbaju = ['batik', 'loreng', 'resmi berdasi']
jumlahbaju = len(daftarbaju)

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
```

Python 3.7.6 Shell (Bottom Left):

```
>>> from a import ucapkansalam as ucap
>>> ucap()
Assalamu'alaikum!
>>> dir()
['_annotations_', '_builtins_', '_doc_', '_file_', '_loader_', '_name_', '_package_', '_spec_', 'buah', 'daftarbaju', 'jumlahbaju', 'kuadratka', 'n', 'mpp', 'pesan', 'ucap', 'ucapkankansalam']
>>> import math as m
>>> dir()
['_annotations_', '_builtins_', '_doc_', '_file_', '_loader_', '_name_', '_package_', '_spec_', 'buah', 'daftarbaju', 'jumlahbaju', 'kuadratka', 'm', 'mpp', 'pesan', 'ucap', 'ucapkankansalam']
>>> dir(m)
['_doc_', '_loader_', '_name_', '_package_', '_spec_', 'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'nan', 'pi', 'pow', 'radians', 'remainder', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc']
>>> from sys import *
>>> dir()
['_annotations_', '_builtins_', '_doc_', '_file_', '_loader_', '_name_', '_package_', '_spec_', 'api_version', 'argv', 'base_exec_prefix', 'base_prefix', 'breakpointhook', 'buah', 'builtin_module_names', 'byteorder', 'call_tracing', 'callstats', 'copyright', 'daftarbaju', 'displayhook', 'dillhandle', 'dont_write_bytecode', 'exc_info', 'excepthook', 'exec_prefix', 'executable', 'exit', 'flags', 'float_info', 'float_repr_style', 'get_asynogen_hooks', 'get_coroutine_origin_tracking_depth', 'get_coroutine_wrapper', 'getallstateblocks', 'getcheckinterval', 'getdefaultencoding', 'getfilesystemencodingerrors', 'getfile_systemencoding', 'getprofile', 'getrecursionlimit', 'getrefcount', 'getsizeof', 'getswitchinterval', 'gettrace', 'getwindowsversion', 'hash_info', 'hexversion', 'implementation', 'int_info', 'intern', 'is_finalizing', 'jumlahbaju', 'kuadratkan', 'last_traceback', 'last_type', 'last_value', 'm', 'maxsize', 'maxunicode', 'meta_path', 'modules', 'mpp', 'path', 'path_hooks', 'path_importer_cache', 'pesan', 'platform', 'prefix', 'set_asynogen_hooks', 'set_coroutine_origin_tracking_depth', 'set_coroutine_wrapper', 'setcheckinterval', 'setprofile', 'setrecursionlimit', 'setswitchinterval', 'settrace', 'stderr', 'stdin', 'stdout', 'thread_read_info', 'ucap', 'ucapkankansalam', 'version', 'version_info', 'warnoptions', 'winver']
>>>
```

Python File Editor (Bottom Right):

```
def ucapkansalam():
    print("Assalamu'alaikum!")

def kuadratkan(x):
    return x*x

buah = 'mangga'
daftarbaju = ['batik', 'loreng', 'resmi berdasi']
jumlahbaju = len(daftarbaju)

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
```

✚ 2.2 Class dan Object

Latihan 2.2 Sebuah kelas sederhana: Pesan. LatOOP2.py

Berikut adalah screenshot program dan hasil dari LatOOP2.py:

The screenshot shows two windows from a Windows desktop. The left window is the 'Python 3.7.6 Shell' with the following code and output:

```
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/LABSI-14/AppData/Local/Programs/Python/Python37/a.py ===
>>> pesanA = Pesan('Aku suka kuliah ini')
>>> pesanB = Pesan('Surakarta: The Spirit of Java')
>>> pesanA.cetakini()
Aku suka kuliah ini
>>> pesanA.cetakjumlahkarakterku()
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    pesanA.cetakjumlahkarakterku()
NameError: name 'pesanA' is not defined
>>> pesanA.cetakjumlahkarakterku()
kalimatku mempunyai 19 karakter.
>>> pesanB.cetakjumlahkarakterku()
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    pesanB.cetakjumlahkarakterku()
NameError: name 'pesanB' is not defined
>>> pesanB.cetakjumlahkarakterku()
kalimatku mempunyai 29 karakter.
>>> pesanA.cetakpakaihurufkapital()
AKU SUKA KULIAH INI
>>> pesanA.cetakpakaihurufkecil()
aku suka kuliah ini
>>> pesanA.perbarui('aku senang struktur data')
>>> pesanA.cetakini()
aku senang struktur data
>>>
```

The right window is 'a.py - C:/Users/LABSI-14/AppData/Local/Programs/Python/Python37/a.py (3.7.6)' with the following code:

```
def ucapkansalam():
    print("Assalamu'alaikum!")

def kuadratkan(x):
    return x*x

buah = 'mangga'
daftarbaju = ['batik', 'loreng', 'resmi berdasi']
jumlahbaju = len(daftarbaju)

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
```

The screenshot shows two windows from a Windows desktop. The left window is the 'Python 3.7.6 Shell' with the following code and output:

```
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/LABSI-14/AppData/Local/Programs/Python/Python37/a.py ===
>>> obQ = sembarangkelas()
>>> obQ.metodesatu()
>>> obQ.metodesembilan()
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    obQ.metodesembilan()
TypeError: metodesembilan() missing 1 required positional argument: 'stringBaru'
>>> obQ.metodesembilan('aku sika mie ayam')
>>>
```

The right window is 'a.py - C:/Users/LABSI-14/AppData/Local/Programs/Python/Python37/a.py (3.7.6)' with the following code:

```
def ucapkansalam():
    print("Assalamu'alaikum!")

def kuadratkan(x):
    return x*x

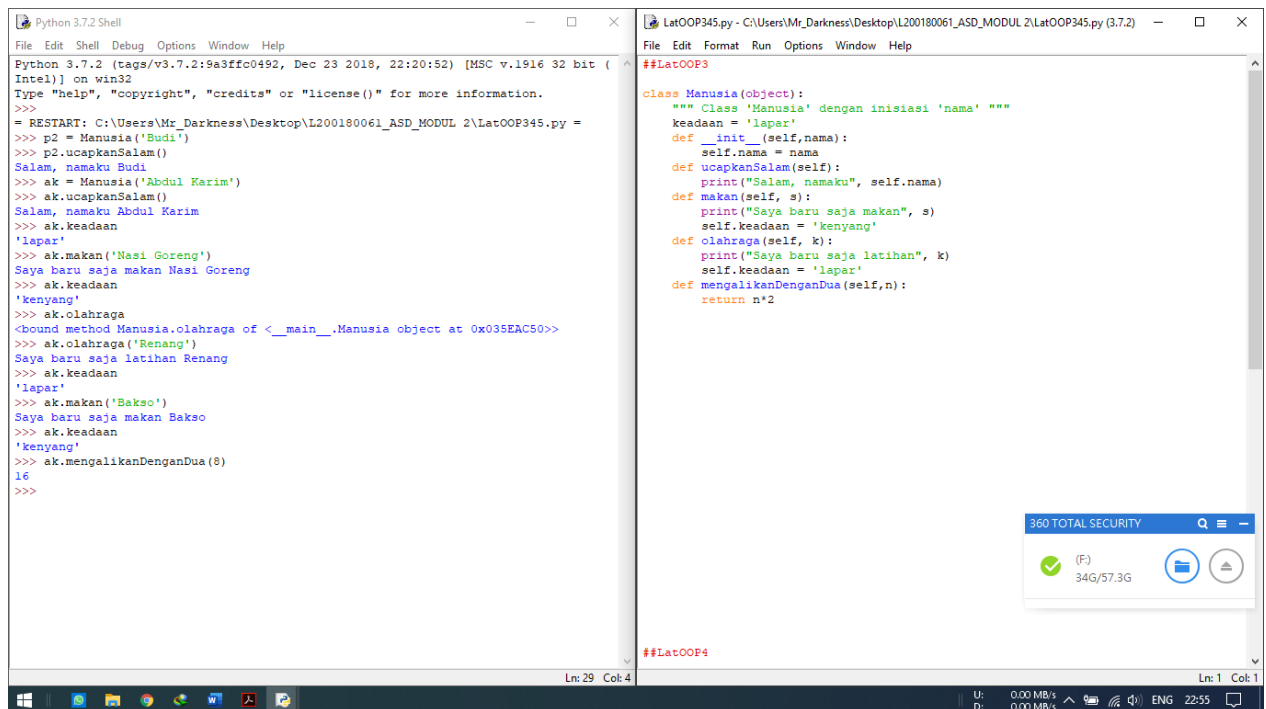
buah = 'mangga'
daftarbaju = ['batik', 'loreng', 'resmi berdasi']
jumlahbaju = len(daftarbaju)

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

class sembarangkelas(object):
    def metodesatu(self):
        pass
    def metodesembilan(self, stringBaru):
        pass
```

Latihan 2.3 Sebuah kelas sederhana lainnya. LatOOP3.py

Berikut adalah screenshot hasil dan program dari LatOOP3.py:



2.2.1 Pewarisan

Latihan 2.4 Membuat sebuah class yang bisa menampung data-data mahasiswa: nama, NIM, kotaTinggal, uangSaku. LatOOP4

Berikut adalah screenshot program dari LatOOP4:

```
##LatOOP4

class Mahasiswa(Manusia):
    """Class Mahasiswa yang dibangun dari class Manusia."""
    def __init__(self,nama,NIM,kota,us):
        """Metode inisiasi ini menutupi metode inisiasi di class Manusia."""
        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 Rp ' + str(self.uangSaku) \
            + ' tiap bulannya.'
        return s
    def ambilNama(self):
        return self.nama
    def ambilNIM(self):
        return self.NIM
    def ambilUangSaku(self):
        return self.uangSaku
    def makan(self,s):
        """Metode ini menutupi metode 'makan'-nya class Manusia.
        Mahasiswa kalau makan sambil belajar."""
        print("Saya baru saja makan",s,"sambil belajar.")
        self.keadaan = 'kenyang'

#Ada kelanjutannya (lihat di "Soal-soal untuk Mahasiswa")
```

Berikut adalah screenshot hasil setelah program LatOOP4 dijalankan:

```

>>>
RESTART: D:\Informatika\SMT 4\Prak Algoritma dan Struktur Data\Modul 2\LatOOP3.py
>>> m1 = Mahasiswa('Jamil',234,'Surakarta',250000)
>>> m2 = Mahasiswa('Andi',365,'Magelang',275000)
>>> m3 = Mahasiswa('Sri',676,'Yogyakarta',240000)
>>> m1.ambilNama()
'Jamil'
>>> m2.ambilNIM()
365
>>> m3.ucapkanSalam()
Salaam, namaku Sri
>>> m3.keadaan
'lapar'
>>> m3.makan('gado-gado')
Saya baru saja makan gado-gado sambil belajar.
>>> m3.keadaan
'kenyang'
>>> print(m3)
Sri, NIM 676. Tinggal di Yogyakarta. Uang saku Rp 240000 tiap bulannya.
>>>

```

Latihan 2.5 Membuat class MhsTIF yang didasarkan pada class Mahasiswa. LatOOP5

Berikut adalah screenshot program dari LatOOP5:

```

##LatOOP5

class MhsTIF(Mahasiswa):      #perhatikan class induknya: Mahasiswa
    """Class MhsTIF yang dibangun dari class Mahasiswa"""
    def katakanPy(self):
        print('Python is cool.')

```

Ln: 59 Col: 32

Berikut adalah screenshot hasil setelah program LatOOP5 dijalankan:

```

>>>
RESTART: D:\Informatika\SMT 4\Prak Algoritma dan Struktur Data\Modul 2\LatOOP3.py
>>> m4 = MhsTIF('Badu',334,'Sragen',230000)
>>> m4.katakanPy()
Python is cool.
>>> print(m4)
Badu, NIM 334. Tinggal di Sragen. Uang saku Rp 230000 tiap bulannya.
>>> m4.keadaan
'lapar'
>>> m4.makan('pecel')
Saya baru saja makan pecel sambil belajar.
>>> m4.keadaan
'kenyang'
>>> m4.ucapkanSalam()
Salaam, namaku Badu
>>>

```

✚ 2.3 Object dan List

Latihan 2.6 Daftar Mahasiswa

Berikut adalah screenshot dari daftar mahasiswa dengan menggunakan program pada latihan 2.4:

```

RESTART: D:\Informatika\SMT 4\Prak Algoritma dan Struktur Data\Modul 2\LatOOP3.
py
>>> m1 = Mahasiswa('Jamil',234,'Surakarta',250000)
>>> m2 = Mahasiswa('Andi',365,'Magelang',275000)
>>> m3 = Mahasiswa('Sri',676,'Yogyakarta',240000)
>>> m4 = MhsTIF('Badu',334,'Sragen',230000)
>>> m5 = Mahasiswa('Ilham',555,'Jakarta',300000)
>>> m6 = Mahasiswa('Rizky',100,'Bandung',350000)
>>> daftar = [m1,m2,m3,m4,m5,m6]          #tambahkan lainnya jika kamu punya
>>> for i in daftar: print(i.NIM)          #tekan <Enter> dua kali

234
365
676
334
555
100
>>> for i in daftar: print(i)

Jamil, NIM 234. Tinggal di Surakarta. Uang saku Rp 250000 tiap bulannya.
Andi, NIM 365. Tinggal di Magelang. Uang saku Rp 275000 tiap bulannya.
Sri, NIM 676. Tinggal di Yogyakarta. Uang saku Rp 240000 tiap bulannya.
Badu, NIM 334. Tinggal di Sragen. Uang saku Rp 230000 tiap bulannya.
Ilham, NIM 555. Tinggal di Jakarta. Uang saku Rp 300000 tiap bulannya.
Rizky, NIM 100. Tinggal di Bandung. Uang saku Rp 350000 tiap bulannya.
>>> daftar[2].ambilNama()
'Sri'

```

✚ 2.4 Class sebagai namespace

Latihan 2.7. LatOOP7.py

Berikut adalah screenshot program dari LatOOP7.py:

```

File Edit Format Run Options Window Help
class kelasKosongan(object):
    pass

##Sekarang kita coba
k = kelasKosongan()
k.x = 23
k.y = 47
print(k.x + k.y)
k.mystr = 'Indonesia'
print(k.mystr)

```

Berikut adalah screenshot hasil setelah program LatOOP7.py dijalankan:

```

RESTART: D:/Informatika/SMT 4/Prak Algoritma dan Struktur Data/Modul 2/LatOOP7.
py
70
Indonesia
>>> |
Ln: 41 Col: 4

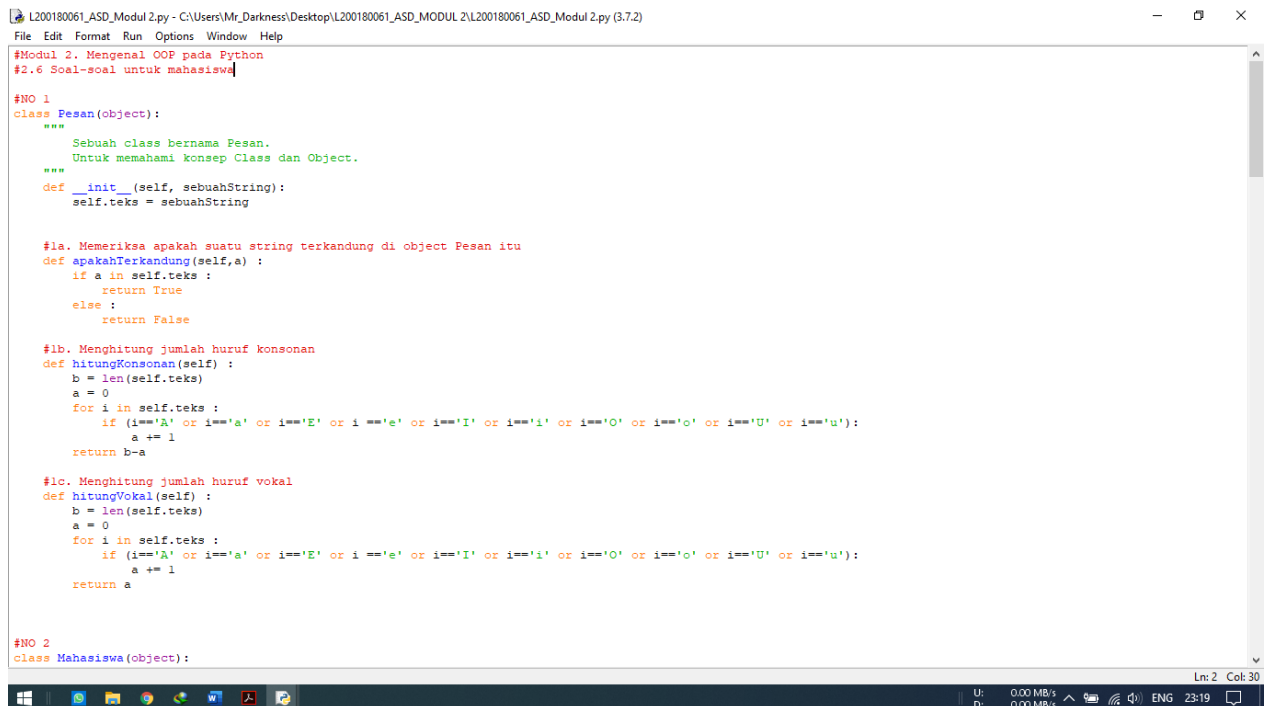
```

✚ 2.6 Soal – Soal Untuk Mahasiswa

1. Pada contoh 2.2 kita telah membuat class Pesan yang berisi beberapa metode. Tambahkan metode – metode di bawah ini ke dalam class itu.
 - a. Metode untuk memeriksa apakah suatu string terkandung di object Pesan itu.
 - b. Metode untuk menghitung jumlah huruf konsonan.
 - c. Metode untuk menghitung jumlah huruf vokal.

Jawab:

Berikut adalah screenshot dari program yang saya buat:



```
#Modul 2. Mengenal OOP pada Python
#2.6 Soal-soal untuk mahasiswa

#NO 1
class Pesan(object):
    """
    Sebuah class bernama Pesan.
    Untuk memahami konsep Class dan Object.
    """
    def __init__(self, sebuahString):
        self.teks = sebuahString

    #1a. Memeriksa apakah suatu string terkandung di object Pesan itu
    def apakahTerkandung(self,a) :
        if a in self.teks :
            return True
        else :
            return False

    #1b. Menghitung jumlah huruf konsonan
    def hitungKonsonan(self) :
        b = len(self.teks)
        a = 0
        for i in self.teks :
            if (i=='A' or i=='a' or i=='E' or i=='e' or i=='I' or i=='i' or i=='O' or i=='o' or i=='U' or i=='u'):
                a += 1
        return b-a

    #1c. Menghitung jumlah huruf vokal
    def hitungVokal(self) :
        b = len(self.teks)
        a = 0
        for i in self.teks :
            if (i=='A' or i=='a' or i=='E' or i=='e' or i=='I' or i=='i' or i=='O' or i=='o' or i=='U' or i=='u'):
                a += 1
        return a

#NO 2
class Mahasiswa(object):
```

Berikut adalah program yang saya buat: class

Pesan(object):

"""

Sebuah class bernama Pesan.

Untuk memahami konsep Class dan Object.

"""

def __init__(self, sebuahString):

self.teks = sebuahString

#1a. Memeriksa apakah suatu string terkandung di object Pesan itu

def apakahTerkandung(self,a) : if a in self.teks : return

True else :

return False

#1b. Menghitung jumlah huruf konsonan

def hitungKonsonan(self) :

b = len(self.teks) a = 0 for i in self.teks : if (i=='A' or i=='a' or i=='E' or

i=='e' or i=='I' or i=='i' or i=='O' or i=='o' or i=='U' or i=='u'):

```

        a += 1

    return b-a

```

```

#1c. Menghitung jumlah huruf vokal
def hitungVokal(self) :
    b = len(self.teks)
    a = 0
    for i in self.teks :
        if (i=='A' or i=='a' or i=='E' or i=='e' or i=='I' or i=='i' or
            i=='O' or i=='o' or i=='U' or i=='u'):
            a += 1

    return a

```

Berikut adalah screenshot dari hasil ketika program diatas dijalankan:

a. Metode untuk memeriksa apakah suatu string terkandung di object Pesan itu.

The screenshot shows two windows. The left window is a Python 3.7.2 Shell with the following output:

```

Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\Mr_Darkness\Desktop\L200180061_ASD_MODUL 2\L200180061_ASD_Modul 2.py
>>> p9 = Pesan('Indonesia adalah negeri yang indah')
>>> p9.apakahTerandung('ege')
True
>>> p9.apakahTerandung('eka')
False
>>> p9.apakahTerandung('Indo')
True
>>> p9.apakahTerandung('satu')
False
>>> |

```

The right window shows the Python script L200180061_ASD_Modul 2.py with the following code:

```

#Modul 2. Mengenal OOP pada Python
#2.6 Soal-soal untuk mahasiswa

#NO 1
class Pesan(object):
    """
    Sebuah class bernama Pesan.
    Untuk memahami konsep Class dan Object.
    """
    def __init__(self, sebuahString):
        self.teks = sebuahString

    #1a. Memeriksa apakah suatu string terkandung di object Pesan itu
    def apakahTerandung(self,a) :
        if a in self.teks :
            return True
        else :
            return False

    #1b. Menghitung jumlah huruf konsonan
    def hitungKonsonan(self) :
        b = len(self.teks)
        a = 0
        for i in self.teks :
            if (i=='A' or i=='a' or i=='E' or i=='e' or i=='I' or i=='i' or i=='O' or i=='o' or i=='U' or i=='u'):
                a += 1
        return b-a

    #1c. Menghitung jumlah huruf vokal
    def hitungVokal(self) :
        b = len(self.teks)
        a = 0
        for i in self.teks :
            if (i=='A' or i=='a' or i=='E' or i=='e' or i=='I' or i=='i' or i=='O' or i=='o' or i=='U' or i=='u'):
                a += 1
        return a

```

b. Metode untuk menghitung jumlah huruf konsonan.

The screenshot shows two windows. The left window is a Python 3.7.2 Shell with the following code:

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\Mr_Darkness\Desktop\L200180061_ASD_MODUL 2\L200180061_ASD_Modul 1 2.py
>>> p9 = Pesanan('Indonesia adalah negeri yang indah')
>>> p9.apakahTerandung('ege')
True
>>> p9.apakahTerandung('eka')
False
>>> p9.apakahTerandung('Indo')
True
>>> p9.apakahTerandung('satu')
False
>>> p10 = Pesanan('Surakarta')
>>> p10.hitungKonsonan()
5
>>> p11 = Pesanan('Universitas')
>>> p11.hitungKonsonan()
6
>>> |
```

The right window is a Python file editor showing the following code:

```
#Modul 2. Mengenal OOP pada Python
#2.6 Soal-soal untuk mahasiswa

#NO 1
class Pesanan(object):
    """
    Sebuah class bernama Pesanan.
    Untuk memahami konsep Class dan Object.
    """
    def __init__(self, sebuahString):
        self.teks = sebuahString

    #1a. Memeriksa apakah suatu string terkandung di object Pesanan itu
    def apakahTerandung(self,a) :
        if a in self.teks :
            return True
        else :
            return False

    #1b. Menghitung jumlah huruf konsonan
    def hitungKonsonan(self) :
        b = len(self.teks)
        a = 0
        for i in self.teks :
            if (i=='A' or i=='a' or i=='E' or i=='e' or i=='I' or i=='i' or i=='O' or i=='o' or i=='U' or i=='u' or i=='Y' or i=='y') :
                a += 1
        return b-a

    #1c. Menghitung jumlah huruf vokal
    def hitungVokal(self) :
        b = len(self.teks)
        a = 0
        for i in self.teks :
            if (i=='A' or i=='a' or i=='E' or i=='e' or i=='I' or i=='i' or i=='O' or i=='o' or i=='U' or i=='u' or i=='Y' or i=='y') :
                a += 1
        return a
```

c. Metode untuk menghitung jumlah huruf vokal.

The screenshot shows two windows. The left window is a Python 3.7.2 Shell with the following code:

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\Mr_Darkness\Desktop\L200180061_ASD_MODUL 2\L200180061_ASD_Modul 1 2.py
>>> p9 = Pesanan('Indonesia adalah negeri yang indah')
>>> p9.apakahTerandung('ege')
True
>>> p9.apakahTerandung('eka')
False
>>> p9.apakahTerandung('Indo')
True
>>> p9.apakahTerandung('satu')
False
>>> p10 = Pesanan('Surakarta')
>>> p10.hitungKonsonan()
5
>>> p11 = Pesanan('Universitas')
>>> p11.hitungKonsonan()
6
>>> p10.hitungVokal()
4
>>> p11.hitungVokal()
5
>>>
```

The right window is a Python file editor showing the following code:

```
#Modul 2. Mengenal OOP pada Python
#2.6 Soal-soal untuk mahasiswa

#NO 1
class Pesanan(object):
    """
    Sebuah class bernama Pesanan.
    Untuk memahami konsep Class dan Object.
    """
    def __init__(self, sebuahString):
        self.teks = sebuahString

    #1a. Memeriksa apakah suatu string terkandung di object Pesanan itu
    def apakahTerandung(self,a) :
        if a in self.teks :
            return True
        else :
            return False

    #1b. Menghitung jumlah huruf konsonan
    def hitungKonsonan(self) :
        b = len(self.teks)
        a = 0
        for i in self.teks :
            if (i=='A' or i=='a' or i=='E' or i=='e' or i=='I' or i=='i' or i=='O' or i=='o' or i=='U' or i=='u' or i=='Y' or i=='y') :
                a += 1
        return b-a

    #1c. Menghitung jumlah huruf vokal
    def hitungVokal(self) :
        b = len(self.teks)
        a = 0
        for i in self.teks :
            if (i=='A' or i=='a' or i=='E' or i=='e' or i=='I' or i=='i' or i=='O' or i=='o' or i=='U' or i=='u' or i=='Y' or i=='y') :
                a += 1
        return a
```

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

- Metode untuk mengambil kota tempat tinggal si mahasiswa
- Metode untuk memperbarui kota tinggal
- Metode untuk menambah uang saku **Jawab:**

Berikut adalah screenshot dari program yang saya buat:


```
L200180061_ASD_Modul 2.py - C:\Users\Mr.Darkness\Desktop\L200180061_ASD_MODUL 2\L200180061_ASD_Modul 2.py (3.7.2)
File Edit Format Run Options Window Help

##      return a
##
##
##
##NO 2
class Mahasiswa(object):
    """Sebuah class bernama Mahasiswa untuk menyimpan data-data mahasiswa"""
    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 Rp ' + str(self.uangSaku) \
            + ' tiap bulannya.'
        return s
    def ambilNama(self):
        return self.nama
    def ambilNIM(self):
        return self.NIM
    def ambilUangSaku(self):
        return self.uangSaku
    def makan(self,s):
        """Metode ini menutupi metode 'makan'-nya class Manusia.
        Mahasiswa kalau makan sambil belajar."""
        print("Saya baru saja makan",s,"sambil belajar.")
        self.keadaan = 'kenyang'

#2a. Mengambil kota tempat tinggal
def ambilKotaTinggal(self):
    return self.kotaTinggal

#2b. Memperbarui kota tinggal
def perbaruiKotaTinggal(self, kotaBaru):
    self.kotaTinggal = kotaBaru

#2c. Menambah uang saku
def tambahUangSaku(self, tambahUang):
    usBaru = self.uangSaku + tambahUang
    self.uangSaku = usBaru
```

Berikut adalah program yang saya buat: class

Mahasiswa(object):

"""Sebuah class bernama Mahasiswa untuk menyimpan data-data mahasiswa"""

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 Rp ' + str(self.uangSaku) \

+ ' tiap bulannya.'

return s def

ambilNama(self):

return self.nama def

ambilNIM(self): return

self.NIM def

ambilUangSaku(self):

```

return self.uangSaku    def
makan(self,s):
    """Metode ini menutupi metode 'makan'-nya class Manusia.
    Mahasiswa kalau makan sambil belajar."""    print("Saya baru
saja makan",s,"sambil belajar.")    self.keadaan = 'kenyang'

```

```

#2a. Mengambil kota tempat tinggal
def ambilKotaTinggal(self):    return
self.kotaTinggal

```

```

#2b. Memperbarui kota tinggal    def
perbaruiKotaTinggal(self, kotaBaru):
self.kotaTinggal = kotaBaru

```

```

#2c. Menambah uang saku    def
tambahUangSaku(self, tambahUang):
usBaru = self.uangSaku + tambahUang
self.uangSaku = usBaru

```

Berikut adalah screenshot dari hasil ketika program diatas dijalankan: a.

Metode untuk mengambil kota tempat tinggal si mahasiswa

```

>>> m9 = Mahasiswa('Ade',155,'Surabaya',250000)
>>> m9.ambilKotaTinggal()
'Surabaya'
>>> m9.perbaruiKotaTinggal('Sleman')
>>> m9.ambilKotaTinggal()
'Sleman'
>>> m10 = Mahasiswa('Indah',267,'Surakarta',200000)
>>> m10.ambilKotaTinggal()
'Surakarta'
>>> m10.perbaruiKotaTinggal('Semarang')
>>> m10.ambilKotaTinggal()
'Semarang'
>>> |

```

Ln: 58 Col: 4

b. Metode untuk memperbarui kota tinggal

```

>>> m9 = Mahasiswa('Ade',155,'Surabaya',250000)
>>> m9.ambilKotaTinggal()
'Surabaya'
>>> m9.perbaruiKotaTinggal('Sleman')
>>> m9.ambilKotaTinggal()
'Sleman'
>>> m10 = Mahasiswa('Indah',267,'Surakarta',200000)
>>> m10.ambilKotaTinggal()
'Surakarta'
>>> m10.perbaruiKotaTinggal('Semarang')
>>> m10.ambilKotaTinggal()
'Semarang'
>>> |

```

Ln: 58 Col: 4

c. Metode untuk menambah uang saku

```

>>> m7 = Mahasiswa('Eric',215,'Malang',270000)
>>> m7.ambilUangSaku()
270000
>>> m7.tambahUangSaku(50000)
>>> m7.ambilUangSaku()
320000
>>> m8 = Mahasiswa('Bagas',199,'Bogor',300000)
>>> m8.ambilUangSaku()
300000
>>> m8.tambahUangSaku(100000)
>>> m8.ambilUangSaku()
400000
>>> |

```

Ln: 86 Col: 4

3. Masih di contoh 2.4. Buatlah suatu program untuk memasukkan data mahasiswa baru lewat Python Shell secara interaktif. Gunakanlah input()

Jawab:

Berikut adalah screenshot dari program yang saya buat:

```

#NO 3
class Mahasiswa(object):
    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 Rp ' + str(self.uangSaku) \
            + ' tiap bulannya.'
        return s

x = input("Masukkan nama -> ")
z = input("Masukkan NIM -> ")
w = input("Masukkan kotaTinggal -> ")
v = input("Masukkan uangSaku -> ")
y = Mahasiswa(x, z, w, v)
print(y)

```

Berikut adalah program yang saya buat:

```

class Mahasiswa(object):
    def __init__(
(self, nama, NIM, kota, us):
        self.nama = nama
        self.NIM = NIM

```

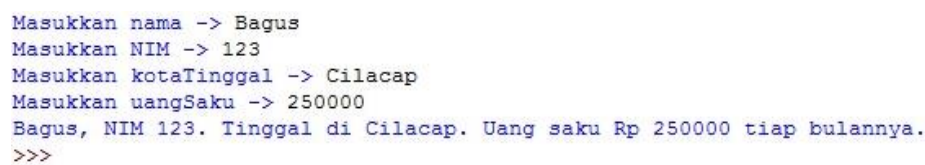
```
self.kotaTinggal = kota
```

```
self.uangSaku = us
```

```
def __str__(self):  
    s = self.nama + ', NIM ' + str(self.NIM) \  
+ '. Tinggal di ' + self.kotaTinggal \  
    + '. Uang saku Rp ' + str(self.uangSaku) \  
    + ' tiap bulannya.'  
    return s
```

```
x = input("Masukkan nama -> ") z =  
input("Masukkan NIM -> ") w =  
input("Masukkan kotaTinggal -> ") v =  
input("Masukkan uangSaku -> ") y =  
Mahasiswa(x, z, w, v) print(y)
```

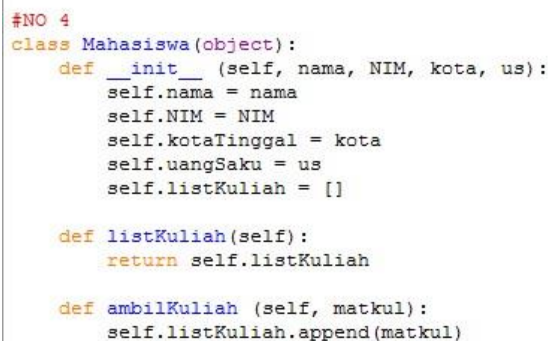
Berikut adalah screenshot dari hasil ketika program diatas dijalankan:



```
Masukkan nama -> Bagus  
Masukkan NIM -> 123  
Masukkan kotaTinggal -> Cilacap  
Masukkan uangSaku -> 250000  
Bagus, NIM 123. Tinggal di Cilacap. Uang saku Rp 250000 tiap bulannya.  
>>>
```

4. Buatlah state baru di class Mahasiswa bernama listKuliah yang berupa list berisi list mata kuliah yang diambil. Buat pula metode ambilKuliah() yang akan menambah daftar mataKuliah ini **Jawab:**

Berikut adalah screenshot dari program yang saya buat:



```
#NO 4  
class Mahasiswa(object):  
    def __init__(self, nama, NIM, kota, us):  
        self.nama = nama  
        self.NIM = NIM  
        self.kotaTinggal = kota  
        self.uangSaku = us  
        self.listKuliah = []  
  
    def listKuliah(self):  
        return self.listKuliah  
  
    def ambilKuliah (self, matkul):  
        self.listKuliah.append(matkul)
```

Berikut adalah program yang saya buat:

```
class Mahasiswa(object):    def __init__
```

```
(self, nama, NIM, kota, us):
```

```
    self.nama = nama
```

```
self.NIM = NIM
```

```
self.kotaTinggal = kota
```

```
self.uangSaku = us
```

```
self.listKuliah = []
```

```
def listKuliah(self):
```

```
    return self.listKuliah
```

```
def ambilKuliah (self, matkul):
```

self.listKuliah.append(matkul) Berikut adalah screenshot dari

hasil ketika program diatas dijalankan:

```
>>> m234 = Mahasiswa('Bintang',234,'Tangerang',300000)
>>> m234.listKuliah
[]
>>> m234.ambilKuliah('Matematika Diskrit')
>>> m234.listKuliah
['Matematika Diskrit']
>>> m234.ambilKuliah('Algoritma dan Struktur Data')
>>> m234.listKuliah
['Matematika Diskrit', 'Algoritma dan Struktur Data']
>>> m235 = Mahasiswa('Gita',235,'Yogyakarta',270000)
>>> m235.listKuliah
[]
>>> m235.ambilKuliah('Pemrograman Berorientasi Objek')
>>> m235.listKuliah
['Pemrograman Berorientasi Objek']
>>> m235.ambilKuliah('Pemrograman Visual')
>>> m235.listKuliah
['Pemrograman Berorientasi Objek', 'Pemrograman Visual']
>>> m235.ambilKuliah('Logika dan Himpunan')
>>> m235.listKuliah
['Pemrograman Berorientasi Objek', 'Pemrograman Visual', 'Logika dan Himpunan']
>>>
```

5. Berkaitan dengan nomor sebelumnya, buatlah metode untuk menghapus sebuah mataKuliah dari listKuliah **Jawab:**

Berikut adalah screenshot dari program yang saya buat:

```
#NO 5
class Mahasiswa(object):
    def __init__(self, nama, NIM, kota, us):
        self.nama = nama
        self.NIM = NIM
        self.kotaTinggal = kota
        self.uangSaku = us
        self.listKuliah = []

    def listKuliah(self):
        return self.listKuliah

    def hapusKuliah (self, matkul):
        self.listKuliah.remove(matkul)

    def ambilKuliah (self, matkul):
        self.listKuliah.append(matkul)
```

Berikut adalah program yang saya buat:

```
class Mahasiswa(object):
    def __init__(self, nama, NIM, kota, us):
        self.nama = nama
        self.NIM = NIM
        self.kotaTinggal = kota
        self.uangSaku = us
        self.listKuliah = []
```

```
    def listKuliah(self):
        return self.listKuliah
```

```
    def hapusKuliah (self, matkul):
        self.listKuliah.remove(matkul)
```

```
    def ambilKuliah (self, matkul):
        self.listKuliah.append(matkul)
```

Berikut adalah screenshot dari hasil ketika program diatas dijalankan:

```

>>> m234 = Mahasiswa('Bintang',234,'Tangerang',300000)
>>> m234.ambilKuliah('Matematika Diskrit')
>>> m234.ambilKuliah('Algoritma dan Struktur Data')
>>> m234.ambilKuliah('sistem Basis Data')
>>> m234.ambilKuliah('Probabilitas dan Statistik')
>>> m234.ambilKuliah('Pancasila')
>>> m234.listKuliah
['Matematika Diskrit', 'Algoritma dan Struktur Data', 'sistem Basis Data', 'Probabilitas dan Statistik', 'Pancasila']
>>> m234.hapusKuliah('Pancasila')
>>> m234.listKuliah
['Matematika Diskrit', 'Algoritma dan Struktur Data', 'sistem Basis Data', 'Probabilitas dan Statistik']
>>> m234.hapusKuliah('Matematika Diskrit')
>>> m234.listKuliah
['Algoritma dan Struktur Data', 'sistem Basis Data', 'Probabilitas dan Statistik']
>>> m234.hapusKuliah('sistem Basis Data')
>>> m234.listKuliah
['Algoritma dan Struktur Data', 'Probabilitas dan Statistik']
>>>

```

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

Berikut adalah screenshot dari program yang saya buat:

```

#NO 6
class Manusia(object):
    keadaan = 'lapar'
    def __init__(self,nama):
        self.nama = nama
    def ucapkanSalam(self):
        print("Salaam, 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'

class siswaSMA(Manusia):
    def __init__(self, nama, noAbsen, kelas, jurusan):
        self.nama = nama
        self.noAbsen = noAbsen
        self.kelas = kelas
        self.jurusan = jurusan

    def ambilNama(self):
        return self.nama

    def ambilNoAbsen(self):
        return self.noAbsen

    def ambilKelas(self):
        return self.kelas

    def ambilJurusan(self):
        return self.jurusan

```

Berikut adalah program yang saya buat:

```

class Manusia(object):
    keadaan = 'lapar'
    def __init__(self,nama):
        self.nama = nama
    def ucapkanSalam(self):
        print("Salaam, 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'
```

```
class siswaSMA(Manusia):  def __init__(self,
nama, noAbsen, kelas, jurusan):    self.nama =
nama    self.noAbsen = noAbsen    self.kelas =
kelas    self.jurusan = jurusan
```

```
    def ambilNama(self):
return self.nama
```

```
    def ambilNoAbsen(self):
return self.noAbsen
```

```
    def ambilKelas(self):
return self.kelas
```

```
    def ambilJurusan(self):
return self.jurusan
```

Berikut adalah screenshot dari hasil ketika program diatas dijalankan:

```

>>> s1 = siswaSMA('Andi',1,10,'IPA')
>>> s2 = siswaSMA('Bagas',5,11,'IPS')
>>> s1.ucapkanSalam()
Salaam, namaku Andi
>>> s1.ambilKelas()
10
>>> s2.ucapkanSalam()
Salaam, namaku Bagas
>>> s2.ambilNama()
'Bagas'
>>> s1.ambilJurusan()
'IPA'
>>> s2.ambilNoAbsen()
5
>>> s1.keadaan
'lapar'
>>> s1.makan('mie ayam')
Saya baru saja makan mie ayam
>>> s1.keadaan
'kenyang'
>>> s2.olahraga('basket')
Saya baru saja latihan basket
>>> s2.keadaan
'lapar'
>>> s2.makan('nasi goreng')
Saya baru saja makan nasi goreng
>>> s2.keadaan
'kenyang'
>>> s1.ambilNama()
'Andi'
>>>

```

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

Jawab:

Berikut adalah screenshot dari program yang saya buat:

```
#NO 7
class Manusia(object):
    """ Class 'Manusia' dengan inisiasi 'nama' """
    keadaan = 'lapar'
    def __init__(self,nama):
        self.nama = nama
    def ucapkanSalam(self):
        print("Salaam, 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 mengalikanDua(n):
        return n*2

class Mahasiswa(Manusia):
    """Class Mahasiswa yang dibangun dari class Manusia."""
    def __init__(self,nama,NIM,kota,us):
        """Metode inisiasi ini menutupi metode inisiasi di class Manusia."""
        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 Rp ' + str(self.uangSaku) \
            + ' tiap bulannya.'
        return s
    def ambilNama(self):
        return self.nama
    def ambilNIM(self):
        return self.NIM
    def ambilUangSaku(self):
        return self.uangSaku
    def makan(self,s):
        """Metode ini menutupi metode 'makan'-nya class Manusia.
        Mahasiswa kalau makan sambil belajar."""
        print("Saya baru saja makan",s,"sambil belajar.")
        self.keadaan = 'kenyang'

class MhsTIF(Mahasiswa):
    """Class MhsTIF yang dibangun dari class Mahasiswa"""
    def katakanPy(self):
        print('Python is cool.')

M1 = MhsTIF('Doni',2327,'Klaten',350000)
M1.ucapkanSalam() #Berasal dari class Manusia
M1.olahraga('bulu tangkis') #Berasal dari class Manusia
M1.ambilNama() #Berasal dari class Mahasiswa
M1.ambilNIM() #Berasal dari class Mahasiswa
M1.katakanPy() #Berasal dari class MhsTIF
```

Berikut adalah program yang saya buat: class

Manusia(object):

""" Class 'Manusia' dengan inisiasi 'nama' """

keadaan = 'lapar' def __init__(self,nama):

self.nama = nama def ucapkanSalam(self):

```

        print("Salaam, 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 mengalikanDua(n):
    return n*2

```

```

class Mahasiswa(Manusia):
    """Class Mahasiswa yang dibangun dari class Manusia."""
    def __init__(self,nama,NIM,kota,us):
        """Metode inisiasi ini menutupi metode inisiasi di class Manusia."""
        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 Rp ' + str(self.uangSaku) \
        + ' tiap bulannya.'
        return s
    def ambilNama(self):
        return self.nama
    def ambilNIM(self):
        return self.NIM
    def ambilUangSaku(self):
        return self.uangSaku
    def makan(self,s):

```

```

        """Metode ini menutupi metode 'makan'-nya class Manusia.
Mahasiswa kalau makan sambil belajar."""    print("Saya baru
saja makan",s,"sambil belajar.")    self.keadaan = 'kenyang'

```

```

class MhsTIF(Mahasiswa):
    """Class MhsTIF yang dibangun dari class Mahasiswa"""
    def katakanPy(self):
        print('Python is cool.')

```

```

M1 = MhsTIF('Doni',2327,'Klaten',350000)
M1.ucapkanSalam()    #Berasal dari class Manusia
M1.olahraga('bulu tangkis')    #Berasal dari class Manusia
M1.ambilNama()    #Berasal dari class Mahasiswa
M1.ambilNIM()    #Berasal dari class Mahasiswa
M1.katakanPy()    #Berasal dari class MhsTIF

```

Berikut adalah screenshot dari hasil ketika program diatas dijalankan:

```

Salaam, namaku Doni
Saya baru saja latihan bulu tangkis
Python is cool.
>>> M2 = MhsTIF('Rina',555,'Semarang',300000)
>>> M2.makan('sate')
Saya baru saja makan sate sambil belajar.
>>>

```

Berikut adalah daftar metode atau state dan asal class-nya:

Class Manusia	Class Mahasiswa	Class MhsTIF
keadaan	nama	katakanPy(self)
	NIM	
ucapkanSalam(self)	kota	
	us	
olahraga(self, k)	ambilNama(self)	

	ambilNIM(self)	
mengalikanDenganDua(self,n)	ambilUangSaku(self)	
	makan(self,s)	