

Nama : Anang Prasetyo

NIM : L200180063

Kelas : C

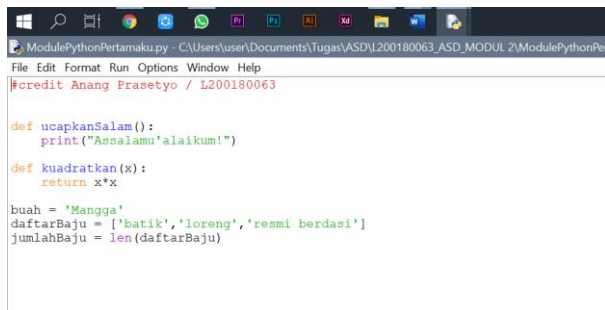
MODUL 2

MENGENAL OOP PADA PYTHON

❖ 2.1 Module

Latihan 2.1 Sebuah module sederhana. ModulePythonPertamaku.py

Berikut adalah screenshot program dari ModulePythonPertamaku.py:



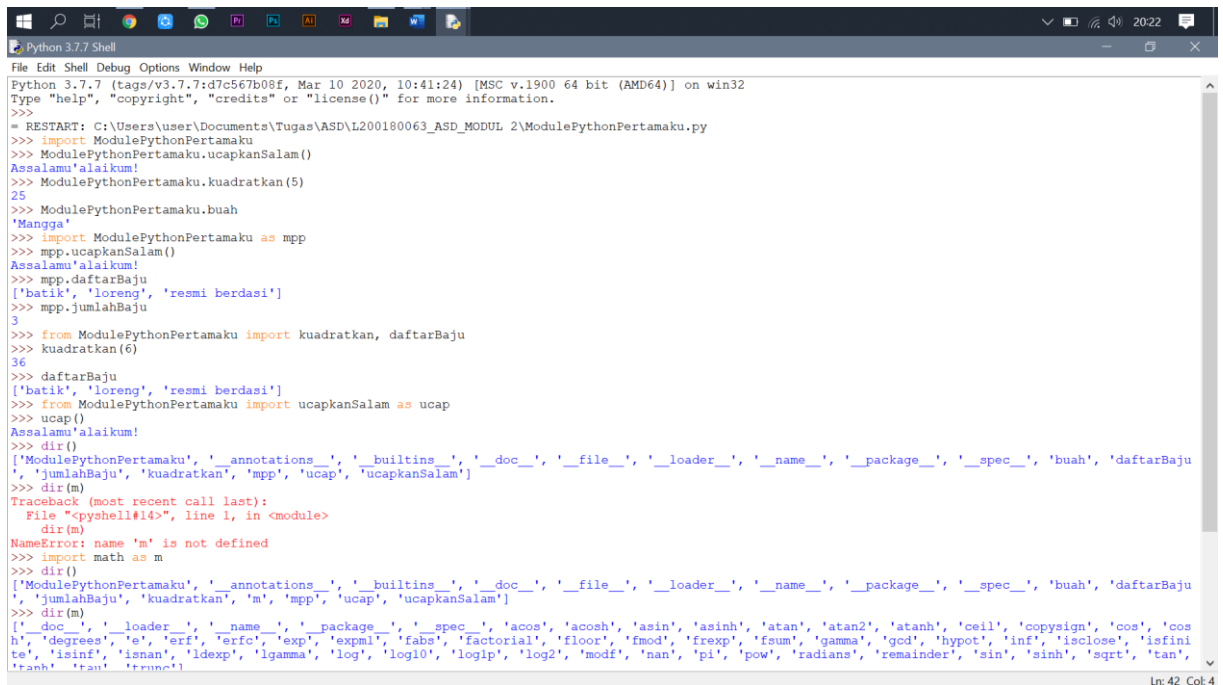
```
ModulePythonPertamaku.py - C:\Users\user\Documents\Tugas\ASD\L200180063_ASD_MODUL 2\ModulePythonPer
File Edit Format Run Options Window Help
#credit Anang Prasetyo / L200180063

def ucapkanSalam():
    print("Assalamu'alaikum!")

def kuadratkan(x):
    return x*x

buah = 'Mangga'
daftarBaju = ['batik', 'loreng', 'resmi berdasi']
jumlahBaju = len(daftarBaju)
```

Berikut adalah screenshot hasil setelah program ModulePythonPertamaku.py dijalankan:



```
Python 3.7.7 Shell
File Edit Shell Debug Options Window Help
Python 3.7.7 (tags/v3.7.7:d7c567b08f, Mar 10 2020, 10:41:24) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\user\Documents\Tugas\ASD\L200180063_ASD_MODUL 2\ModulePythonPertamaku.py
>>> import ModulePythonPertamaku
>>> ModulePythonPertamaku.ucapkanSalam()
Assalamu'alaikum!
>>> ModulePythonPertamaku.kuadratkan(5)
25
>>> ModulePythonPertamaku.buah
'Mangga'
>>> import ModulePythonPertamaku as mpp
>>> mpp.ucapkanSalam()
Assalamu'alaikum!
>>> mpp.daftarBaju
['batik', 'loreng', 'resmi berdasi']
>>> mpp.jumlahBaju
3
>>> from ModulePythonPertamaku import kuadratkan, daftarBaju
>>> kuadratkan(6)
36
>>> daftarBaju
['batik', 'loreng', 'resmi berdasi']
>>> from ModulePythonPertamaku import ucapkanSalam as ucap
>>> ucap()
Assalamu'alaikum!
>>> dir()
['ModulePythonPertamaku', '__annotations__', '__builtins__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', 'buah', 'daftarBaju', 'jumlahBaju', 'kuadratkan', 'mpp', 'ucap', 'ucapkanSalam']
>>> dir(m)
Traceback (most recent call last):
  File "<pyshell#14>", line 1, in <module>
    dir(m)
NameError: name 'm' is not defined
>>> import math as m
>>> dir()
['ModulePythonPertamaku', '__annotations__', '__builtins__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', 'buah', 'daftarBaju', 'jumlahBaju', 'kuadratkan', 'm', 'mpp', 'ucap', 'ucapkanSalam']
>>> 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', 'logip', 'log2', 'modf', 'nan', 'pi', 'pow', 'radians', 'remainder', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc']
Ln: 42 Col: 4
```



```
Python 3.7.7 Shell
File Edit Shell Debug Options Window Help
Python 3.7.7 (tags/v3.7.7:d7c567b08f, Mar 10 2020, 10:41:24) [MSC v.1900 64
bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\user\Documents\Tugas\ASD\L200180063_ASD_MODUL 2\LatOOP3
45.py
>>> p2 = Manusia('Budi')
>>> p2.ucapkanSalam()
Salaam, namaku Budi
>>> ak = Manusia('Abdul Karim')
>>> ak.ucapkanSalam()
Salaam, namaku Abdul Karim
>>> ak.keadaan
'lapar'
>>> ak.makan('nasi goreng')
Saya baru saja makan nasi goreng
>>> ak.keadaan
'kenyang'
>>> ak.olahraga
<bound method Manusia.olahraga of <__main__.Manusia object at 0x0000023DA0C
14808>
>>> ak.olahraga('renang')
Saya baru saja latihan renang
>>> ak.keadaan
'lapar'
>>> ak.makan('bakso')
Saya baru saja makan bakso
>>> ak.keadaan
'kenyang'
>>> ak.mengalikanDenganDua
<bound method Manusia.mengalikanDenganDua of <__main__.Manusia object at 0x
0000023DA0C14808>
>>> ak.mengalikanDenganDua(8)
16
>>> |

*LatOOP345.py - C:\Users\user\Documents\Tugas\ASD\L200180063_ASD_MODUL 2\...
File Edit Format Run Options Window Help
#credit Anang Prasetyo / L200180063

##LatOOP3

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 mengalikanDenganDua(self,n):
        return n*2
```

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)

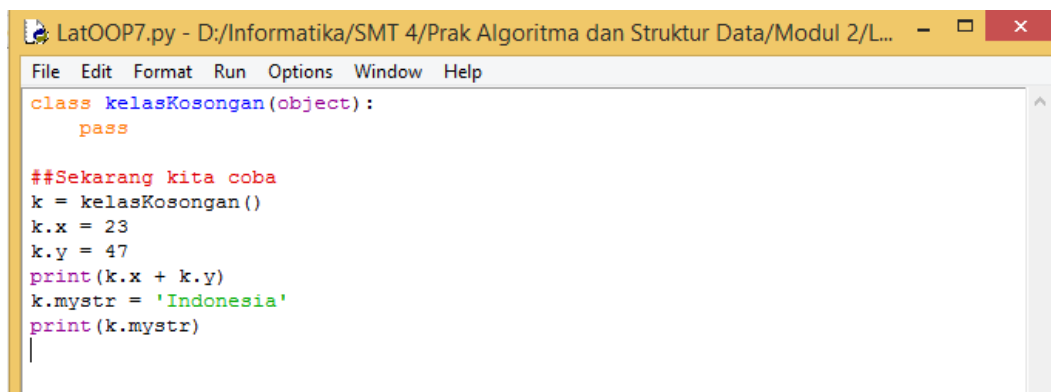
Jamil, NIM 234. Tinggal di Surakarta. Uang saku Rp 250000 tiap bulannya.
Andi, NIM 365. Tinggal di Magelang. Uang saku Rp 275000 tiap bulannya.
Sri, NIM 676. Tinggal di Yogyakarta. Uang saku Rp 240000 tiap bulannya.
Badu, NIM 334. Tinggal di Sragen. Uang saku Rp 230000 tiap bulannya.
Ilham, NIM 555. 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:



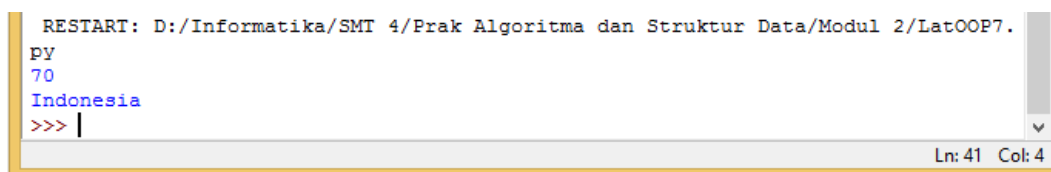
```

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:

```
"L200180063_ASD_Modul 2.py - C:\Users\user\Documents\Tugas\ASD\L200180063_ASD_MODUL 2\L200180063_ASD_Modul 2.py (3.7.7)"
File Edit Format Run Options Window Help
#Praktikum Algoritma dan Struktur Data
#Modul 2. Mengenal OOP pada Python
#2.6 Soal-soal untuk mahasiswa
#credit Anang Prasetyo / L200180063

#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
"""
```

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.

```

Python 3.7.7 (tags/v3.7.7:d7c567b08f, Mar 10 2020, 10:41:24) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\user\Documents\Tugas\ASD\L200180063_ASD_MODUL 2\L200180063_ASD_Modul 2.py
>>> p9 = Pesan('Indonesia adalah negeri yang indah')
>>> p9.apakahTerkandung('ege')
True
>>> p9.apakahTerkandung('eka')
False
>>> p9.apakahTerkandung('Indo')
True
>>> p9.apakahTerkandung('satu')
False
>>> |

```

b. Metode untuk menghitung jumlah huruf konsonan.

```
>>> p10 = Pesan('Surakarta')
>>> p10.hitungKonsonan()
5
>>> p11 = Pesan('Universitas')
>>> p11.hitungKonsonan()
6
>>> p12 = Pesan('Muhammadiyah')
>>> p12.hitungKonsonan()
7
>>> |
```

Ln: 104 Col: 4

c. Metode untuk menghitung jumlah huruf vokal.

```
>>> p10 = Pesan('Surakarta')
>>> p10.hitungVokal()
4
>>> p11 = Pesan('Universitas')
>>> p11.hitungVokal()
5
>>> p12 = Pesan('Muhammadiyah')
>>> p12.hitungVokal()
5
>>>
```

Ln: 149 Col: 4

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

- a. Metode untuk mengambil kota tempat tinggal si mahasiswa
- b. Metode untuk memperbarui kota tinggal
- c. Metode untuk menambah uang saku

Jawab:

Berikut adalah screenshot dari program yang saya buat:

```
#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 = []
```

```

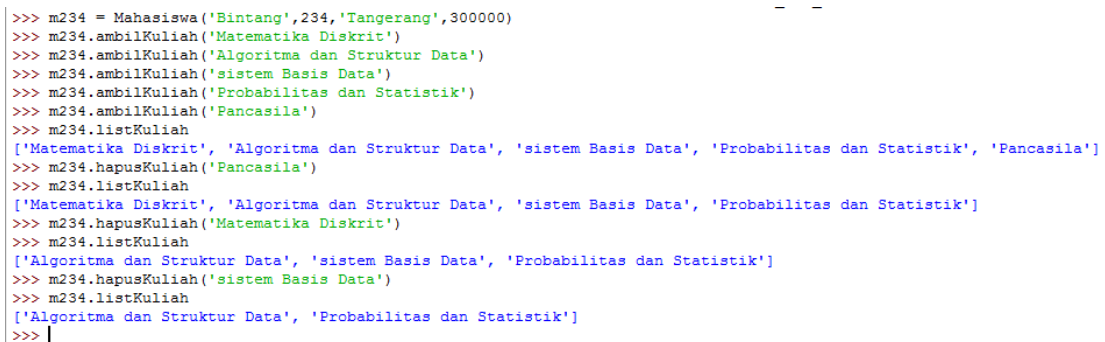
def listKuliah(self):
    return self.listKuliah

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 mengalikanDenganDua(self,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)	