

Nama : Malik

Muhammad

NIM : L200180072

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:

```
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:

```
>>> 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', 'ucapkan', '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', 'ucapkan', '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']
```

```

File Edit Shell Debug Options Window Help
>>> 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', 'god', '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']
>>> from sys import *
>>> dir()
['ModulePythonPertamaku', '__annotations__', '__builtins__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', 'api_version', 'argv', 'base_exec_prefix', 'base_prefix', 'breakpoint', 'buah', 'builtin_module_names', 'byteorder', 'call_tracing', 'callstats', 'copyright', 'daftarBaju', 'displayhook', 'dlhandle', 'dont_write_bytecode', 'exec_info', 'exec_prefix', 'executable', 'exit', 'flags', 'float_info', 'float_repr_style', 'get_asyncgen_hooks', 'get_coroutine_origin_tracking_depth', 'get_coroutine_wrapper', 'getcheckinterval', 'getdefaultencoding', 'getfilesystemencoding', 'getfilesystemerrors', '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', 'platform', 'prefix', 'set_asyncgen_hooks', 'set_coroutine_origin_tracking_depth', 'set_coroutine_wrapper', 'setcheckinterval', 'setprofile', 'setrecursionlimit', 'setswitchinterval', 'settrace', 'stderr', 'stdin', 'stdout', 'thread_info', 'ucap', 'ucapkanSalam', 'version', 'version_info', 'warnoptions', 'winver']
>>>

```

❖ 2.2 Class dan Object

Latihan 2.2 Sebuah kelas sederhana: Pesan. LatOOP2.py

Berikut adalah screenshot program dan hasil dari LatOOP2.py:

```

.py =
>>> pesanA = Pesan('Aku suka kuliah ini')
>>> pesanB = Pesan('Surakarta: the Spirit of Java')
>>> pesanA.cetakIni()
Aku suka kuliah ini
>>> pesanA.cetakJumlahKarakterku()
Kalimatku mempunyai 19 karakter.
>>> pesanB.cetakIni()
Surakarta: the Spirit of Java
>>> 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
>>>

```

```

""" Untuk memahami konsep Class dan 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

```

Latihan 2.3 Sebuah kelas sederhana lainnya. LatOOP3.py

```

gKelas.py
>>> obQ = sembarangKelas()
>>> obQ.metodeSatu()
>>> obQ.metodeSemblilan('Aku suka mie ayam')
>>>

```

Berikut adalah screenshot hasil dan program dari LatOOP3.py:

```

45.py
>>> p2 = Manusia('Budi')
>>> p2.ucapkanSalam()
Salam, nanaku Budi
>>> ak = Manusia('Abdul Karim')
>>> ak.ucapkanSalam()
Salam, nanaku 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 0x000023DA0C14808>
>>> 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 0x000023DA0C14808>
>>> ak.mengalikanDenganDua(8)
16
>>> |

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

```

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)

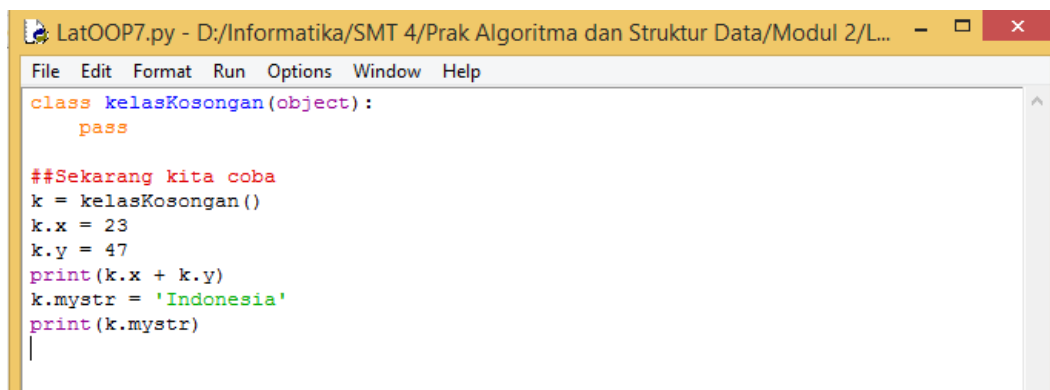
Jamil, NIM 234. Tinggal di Surakarta. Uang saku Rp 250000 tiap bulannya.
Andi, NIM 365. Tinggal di Magelang. Uang saku Rp 275000 tiap bulannya.
Sri, NIM 676. Tinggal di Yogyakarta. Uang saku Rp 240000 tiap bulannya.
Badu, NIM 334. Tinggal di Sragen. Uang saku Rp 230000 tiap bulannya.
Ilham, NIM 555.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



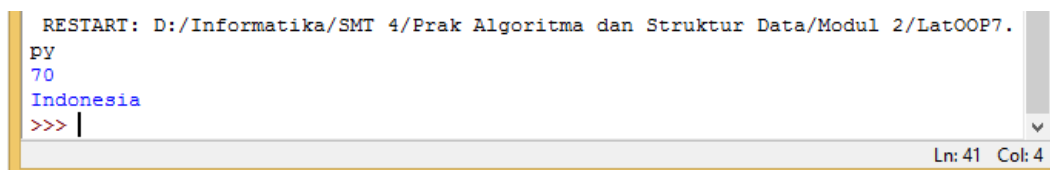
```

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 Pesannya.
 - b. Metode untuk menghitung jumlah huruf konsonan.

c. Metode untuk menghitung jumlah hurufvokal.

Jawab:

Berikut adalah screenshot dari program yang saya buat:

Berikut adalah program yang saya buat:

```
#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

```

class

Pesan(object):

"""

Sebuah class bernama Pesan.

Untuk memahami konsep Class dan

Object. """

definit(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 Pesanitu.
- b. Metode untuk menghitung jumlah hurufkonsonan.

```

>>> p9 = Pesan('Indonesia adalah negeri yang indah')
>>> p9.apakahTerkandung('ege')
True
>>> p9.apakahTerkandung('eka')
False
>>> p9.apakahTerkandung('Indo')
True
>>> p9.apakahTerkandung('satu')
False
>>> |

```

```
>>> 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 hurufvokal.

```
>>> 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 bawahini

a. Metode untuk mengambil kota tempat tinggal simahasiswa

b. Metode untuk memperbarui kota tinggal

c. Metode untuk menambah uangsaku

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 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 uangsaku

```
>>> 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
```

```
defstr(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. DariclassManusia,buatlahsebuahclassSiswaSMAYangmemuatmetode-metodebaru (kamu bebasmenentukan)

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'

definit(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):

definit(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 __mul__(self, n):
        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('bulutangkis')
#Berasal dari classManusia M1.ambilNama()
#Berasal dari class Mahasiswa M1.ambilNIM()
#Berasal dari class Mahasiswa M1.katakanPy()
#Berasal dari classMhsTIF

```

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)	