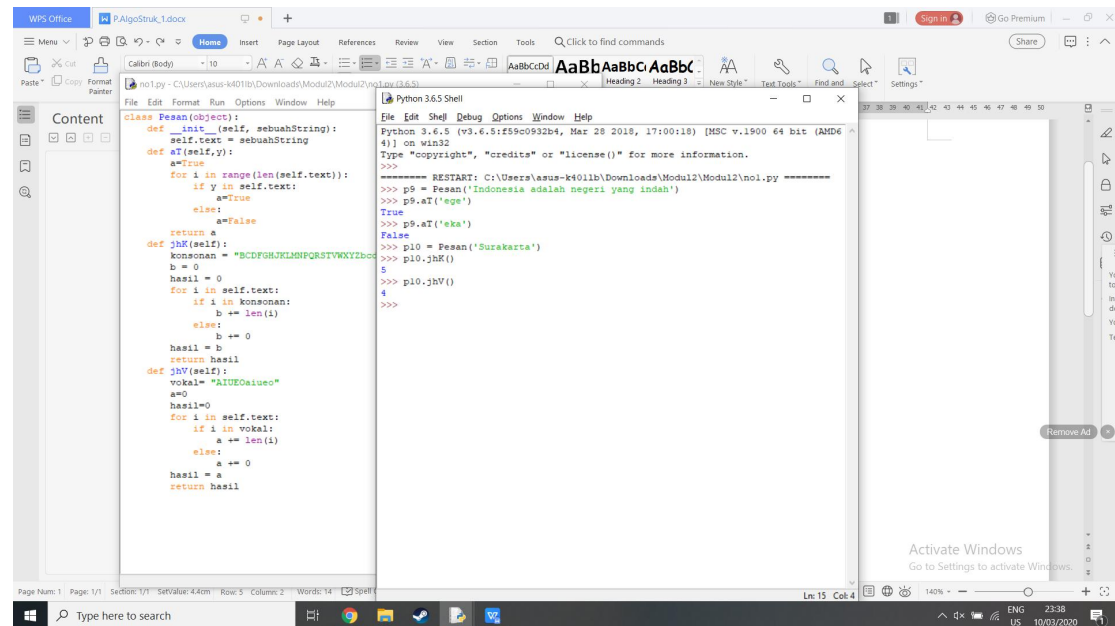


Nama : Amartya Maulana
Kelas : G
Praktikum Algoritma dan Struktur Data Bab II

1.

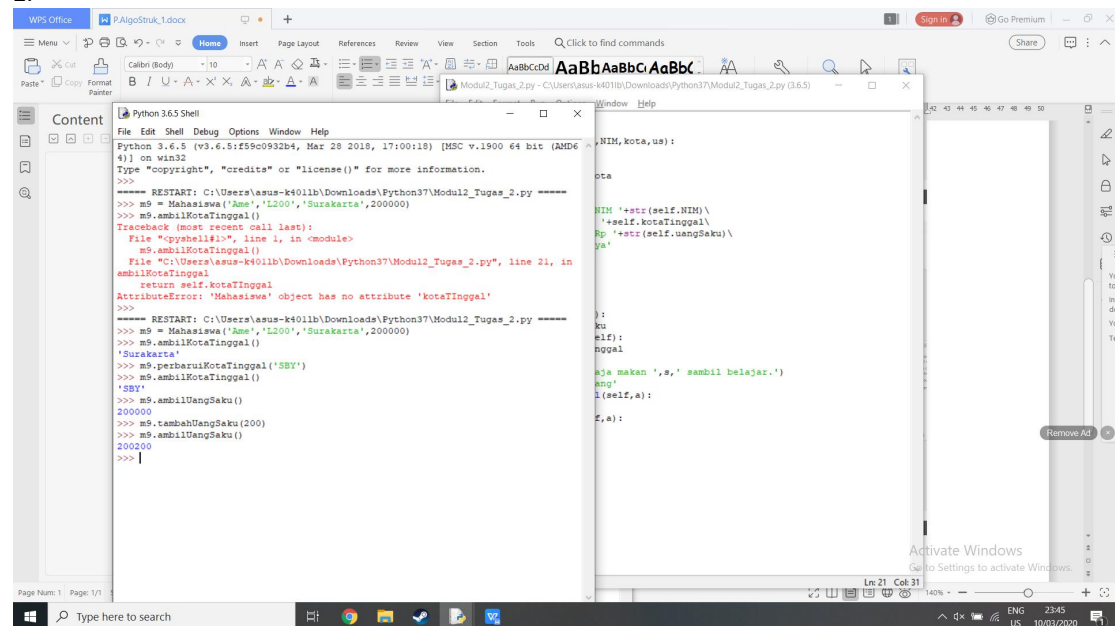


The screenshot shows a WPS Office window with a document titled 'P.AlgoStruk_1.docx'. The document contains a Python class named 'Pesan' with methods for initializing text, checking if a character is a vowel or consonant, and counting them. The Python 3.6.5 Shell is open, showing the execution of the script. The output indicates that the script is running successfully, with the text 'Indonesia adalah negeri yang indah' being processed.

```
class Pesan(object):
    def __init__(self, sebuahString):
        self.text = sebuahString
    def __str__(self):
        return self.text
    def isVowel(self, a):
        for i in range(len(self.text)):
            if i in self.text:
                a=True
            else:
                a=False
        return a
    def jhK(self):
        konsonan = "BCDFGHJKLMNPQRSTVWXYZbc"
        b = 0
        hasil = 0
        for i in self.text:
            if i in konsonan:
                b += len(i)
            else:
                b += 0
        hasil = b
        return hasil
    def jhV(self):
        vokal = "AIUEOaiueo"
        a=0
        hasil=0
        for i in self.text:
            if i in vokal:
                a += len(i)
            else:
                a += 0
        hasil = a
        return hasil
```

```
Python 3.6.5 Shell
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\asus-k4011b\Downloads\Modul2\Modul2\nol.py (3.6.5) =====
>>> p5 = Pesan('Indonesia adalah negeri yang indah')
>>> p5.at('ege')
True
>>> p5.at('eka')
False
>>> p10 = Pesan('Surakarta')
>>> p10.jhK()
5
>>> p10.jhV()
4
>>>
```

2.

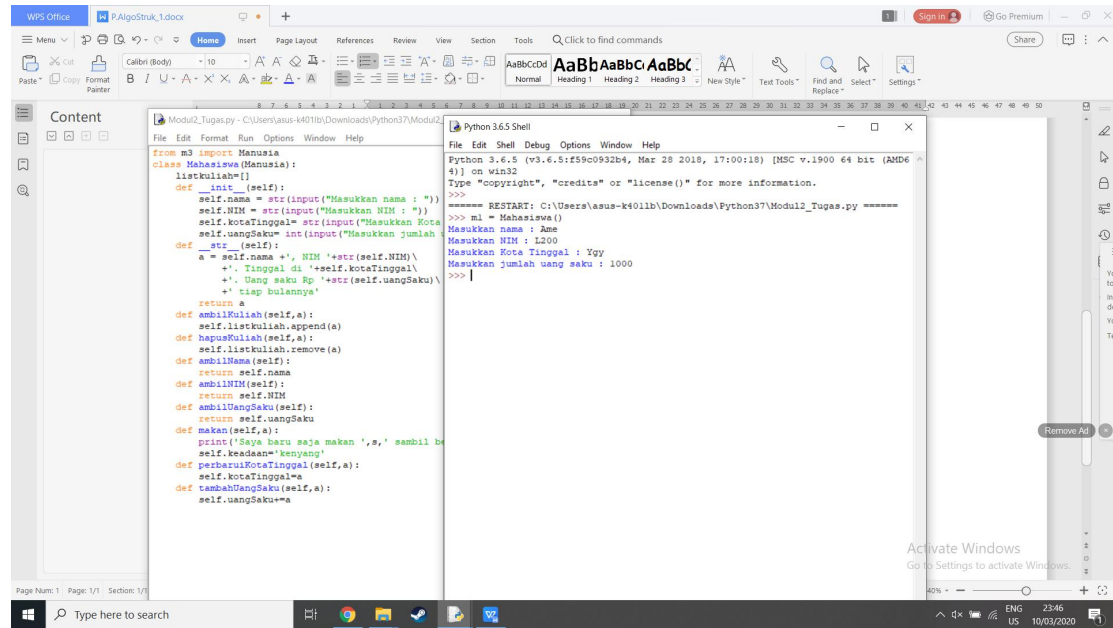


The screenshot shows a WPS Office window with a document titled 'P.AlgoStruk_1.docx'. The document contains a Python script for a class named 'Mahasiswa' with methods for initializing data, adding and removing items from a list, and displaying the list. The Python 3.6.5 Shell is open, showing the execution of the script. The output indicates that the script is running successfully, with the text 'Surakarta' being processed.

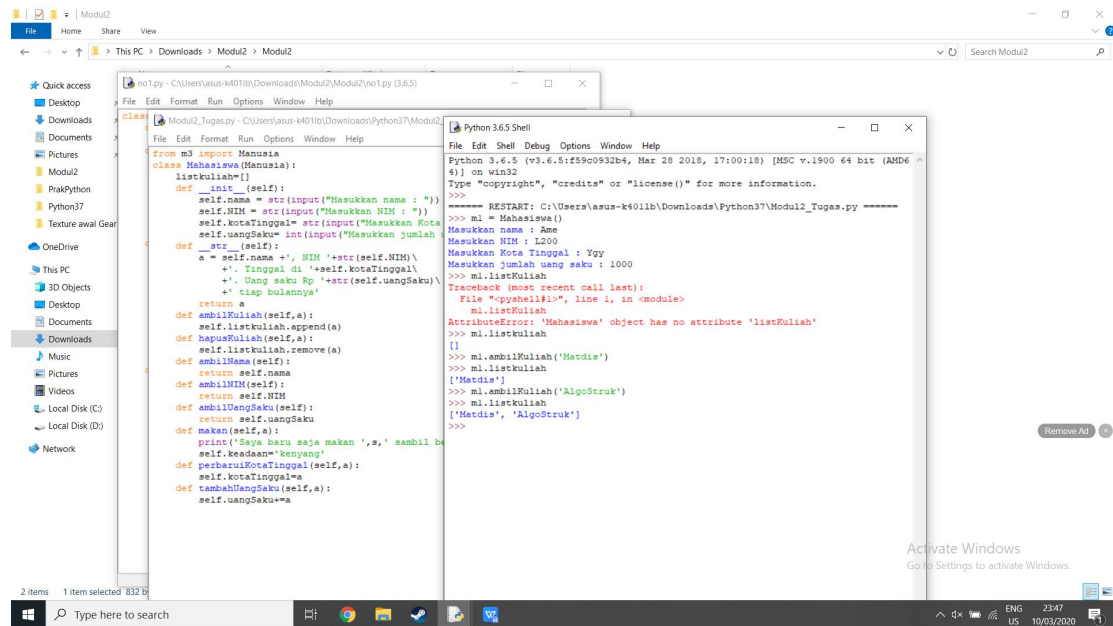
```
class Mahasiswa:
    def __init__(self, nama, l200, 'Surakarta', 200000):
        self.nama = nama
        self.l200 = l200
        self.Surakarta = 'Surakarta'
        self.200000 = 200000
    def tambahUangSaku(self, uang):
        self.200000 += uang
    def ambilUangSaku(self):
        return self.200000
    def ambilKotaTinggal(self):
        return self.Surakarta
    def perbaruiKotaTinggal(self, kota):
        self.Surakarta = kota
    def __str__(self):
        return self.nama + ' ' + self.l200 + ' ' + self.Surakarta + ' ' + str(self.200000)
```

```
Python 3.6.5 Shell
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\asus-k4011b\Downloads\Python37\Modul2_Tugas_2.py (3.6.5) =====
>>> m5 = Mahasiswa('Susi', 'L200', 'Surakarta', 200000)
>>> m5.ambilKotaTinggal()
Surakarta
>>> m5.perbaruiKotaTinggal('SBY')
>>> m5.ambilKotaTinggal()
SBY
>>> m5.ambilUangSaku()
200000
>>> m5.tambahUangSaku(200)
200200
>>> m5.ambilUangSaku()
200200
>>>
```

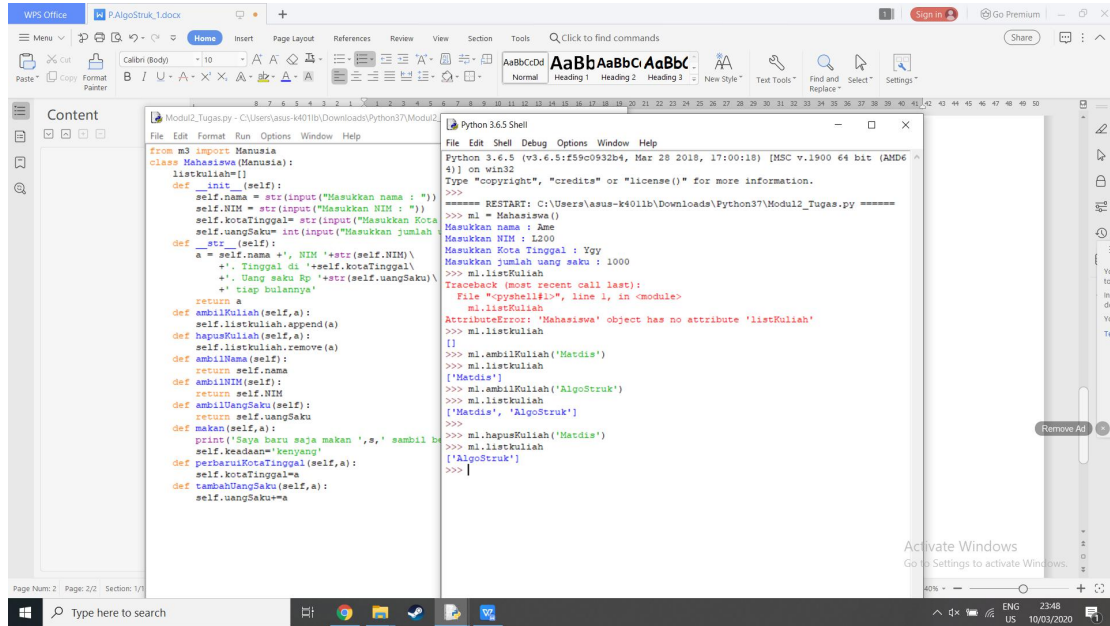
3.



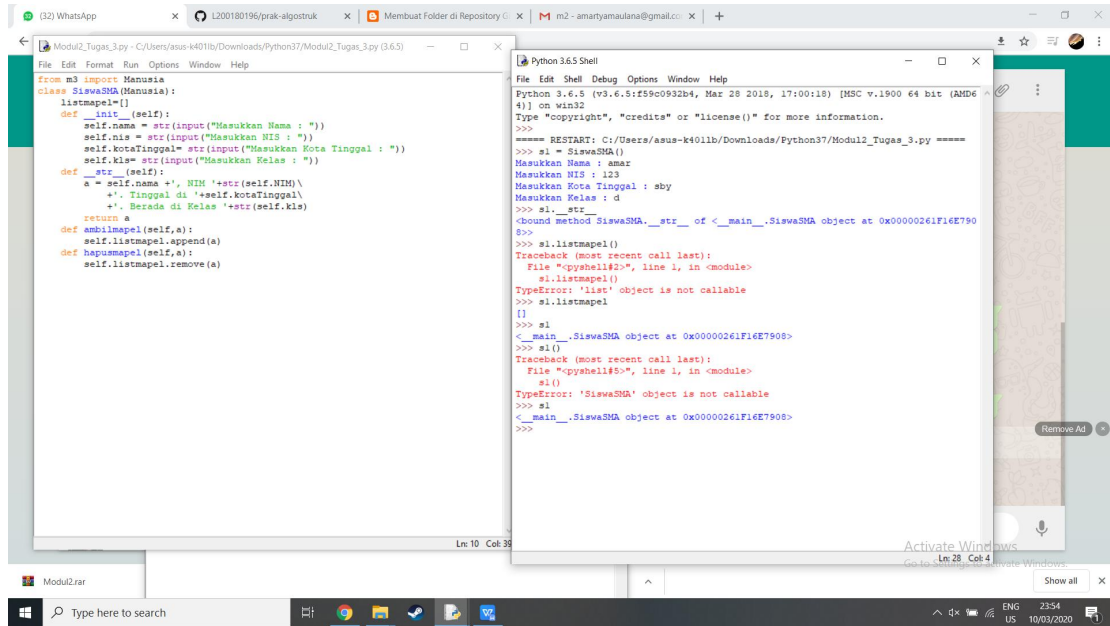
4.



5.



6.



7.

The screenshot shows a Python IDE with two windows. The left window displays a Python script with a class hierarchy where `MhsTIF` inherits from `Mahasiswa`. The right window is a Python 3.6.5 Shell showing the execution of the script. The output demonstrates that an instance of `MhsTIF` can access attributes and methods defined in the `Mahasiswa` class, such as `ambilKotaTinggal`, `ambilNIM`, `ambilNama`, `ambilJangSaku`, `katakanPy`, `kaduan`, and `kotaTinggal`. The shell also shows a `Traceback` error for a `TypeError: 'list' object is not callable` and a `TypeError: 'mDD' object is not callable`.

```

from m3 import Mahasiswa
class S1(Mahasiswa):
    def __init__(self, nama, nim, kota):
        self.nama = nama
        self.nim = nim
        self.kota = kota
    def katakanPy(self):
        print('Pythin is cool..')

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

# Execution in Python Shell
Python 3.6.5 (v3.6.5:559c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/asus-k401lb/Downloads/Python37/Modul2_Tugas_3.py =====
>>> s1 = SiswaSMA()
Masukkan Nama : amar
Masukkan NIS : 123
Masukkan Kota Tinggal : sbj
Masukkan Kelas : d
>>> s1._str_
<bound method SiswaSMA._str_ of <_main_.SiswaSMA object at 0x00000261F16E7950>
>>> s1.listmapel()
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    s1.listmapel()
TypeError: 'list' object is not callable
>>> s1.listmapel
[]
>>> s1
<_main_.SiswaSMA object at 0x00000261F16E7950>
>>> s1()
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    s1()
TypeError: 'SiswaSMA' object is not callable
>>> s1.ambilKotaTinggal
'amar'
>>> s1.ambilNIM
123
>>> s1.ambilNama
'amar'
>>> s1.ambilJangSaku
[]
>>> s1.katakanPy
'Pythin is cool..'
>>> s1.kaduan
[]
>>> s1.kotaTinggal
'sbj'
>>> s1.mDD
'Pythin is cool..'
>>> s1.makan
'Pythin is cool..'
>>> a = MhsTIF('Domi', '12300', 'JKT', 12312)
>>> a

```

Metode/state yang keluar dari object itu berasal dari class Mahasiswa, tapi melewati class MhsTIF karena pada MhsTIF mengalami pewarisan dari class Mahasiswa jadi kita bisa menggunakan seluruh metode/state yang berada pada class Mahasiswa.