

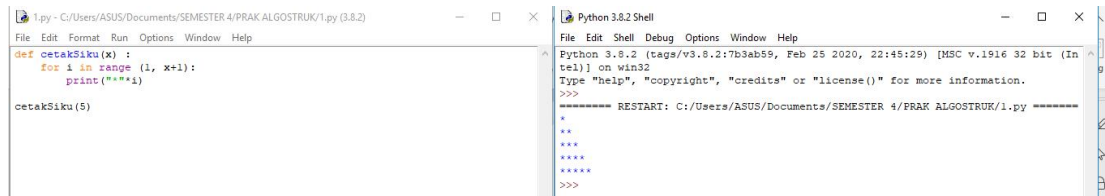
**NAMA : ELSA PUTRI ALIYYA**

**NIM : L200180108**

**KELAS : D**

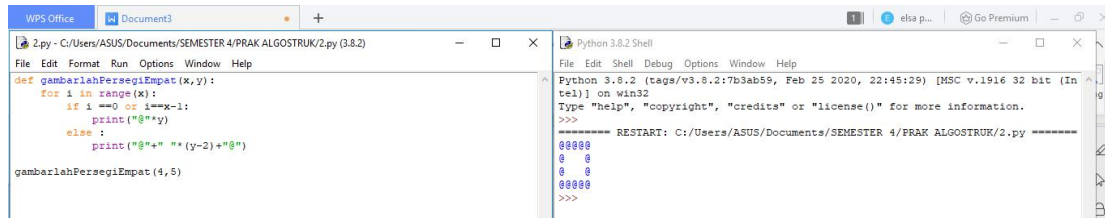
## **MODUL 1**

**1.**



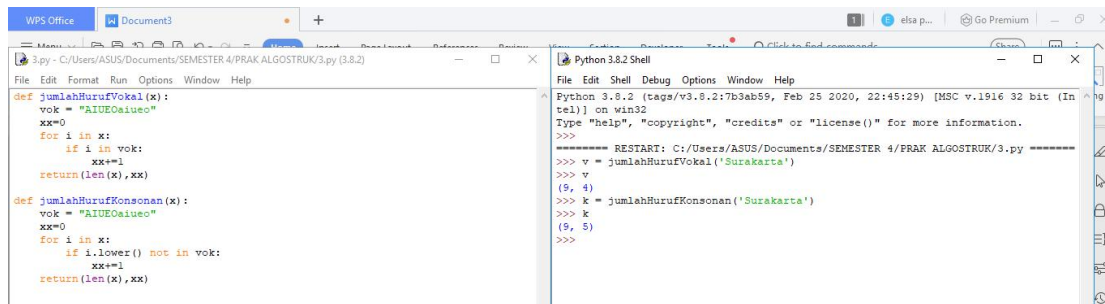
```
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/1.py =====
>>>
>>>
>>>
>>>
>>>
```

**2.**



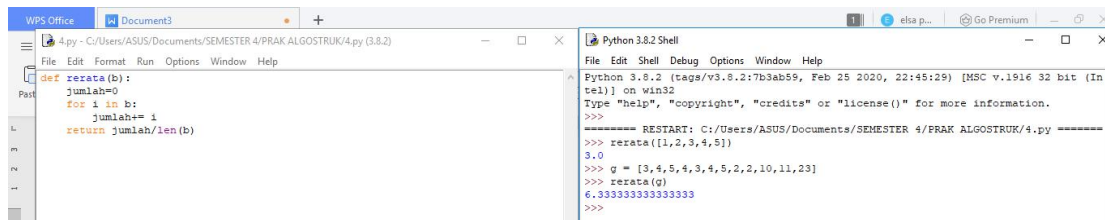
```
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/2.py =====
>>>
>>>
>>>
>>>
>>>
```

**3.**



```
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/3.py =====
>>> v = jumlahHurufVokal('Surakarta')
>>> v
(5, 4)
>>> k = jumlahHurufKonsonan('Surakarta')
>>> k
(9, 5)
>>>
```

**4.**



```
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/4.py =====
>>> rerata([1,2,3,4,5])
3.0
>>> g = [3,4,5,4,3,4,5,2,2,10,11,23]
>>> rerata(g)
6.333333333333333
>>>
```

5.

```

File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/5.py =====
17 adalah bilangan prima
97 adalah bilangan prima
123 bukan bilangan prima
>>>

```

6.

```

File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/6.py =====
2 True
3 True
4 False
5 True
6 False
7 True
8 False
9 False
10 False
11 True
12 False
13 True
14 False
15 True
16 False
17 True
18 False
19 True
20 False
21 True
22 False
23 True
24 False
25 True
26 False
27 True
28 False
29 True
30 False
31 True
32 False
33 True
34 False
35 True
36 False

```

The screenshot shows a WPS Office window with a Python script named `6.py` and a Python 3.8.2 Shell window. The script defines a function `apakahPrima(n)` to check if a number is prime. It uses a list `primaKecil` for small primes and a loop for larger numbers. The main loop prints the results for numbers from 2 to 1001.

```
from math import sqrt as sq
def apakahPrima(n):
    n = int(n)
    assert n >= 0
    primaKecil = [2,3,5,7,11]
    bukanPrKecil = [0,1,4,6,8,9,10]
    if n in primaKecil:
        return True
    elif n in bukanPrKecil:
        return False
    else:
        for i in range(2, int(sq(n))+1):
            if n%i == 0:
                return False
            break
        else:
            return True
for i in range(2,1001):
    print (str(i)+" "+str(apakahPrima(i)))
```

The shell window shows the output of the program, displaying a list of numbers from 2 to 1000, each followed by its primality status (True or False).

7.

The screenshot shows a WPS Office window with a Python script named `7.py` and a Python 3.8.2 Shell window. The script defines a function `faktorPrima(n)` to find the prime factors of a number. It uses a while loop to divide the number by its smallest prime factor until it becomes 1. The main loop calls the function for 10, 120, and 19.

```
def faktorPrima(n):
    p = []
    a = 2
    while a <= n:
        if n%a == 0:
            n /= a
            p.append(a)
        else:
            a += 1
    print(p)
faktorPrima(10)
faktorPrima(120)
faktorPrima(19)
```

The shell window shows the output of the program, displaying the prime factors for each input number: `[2, 5]` for 10, `[2, 2, 2, 3, 5]` for 120, and `[19]` for 19.

8.

The screenshot shows a WPS Office window with a Python script named `8.py` and a Python 3.8.2 Shell window. The script defines a function `apakahTerKandung(a,b)` to check if a word is in a list. It uses a list `h` containing the words "do", "Indonesia", and "tanah air beta". The main loop calls the function for "do" and "pusaka".

```
def apakahTerKandung(a,b):
    return a in b
h = "do"
k = "Indonesia tanah air beta"
print (apakahTerKandung(h,k))
print (apakahTerKandung("pusaka", k))
```

The shell window shows the output of the program, displaying `True` for "do" and `False` for "pusaka".

9.

9.py - C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/9.py (3.8.2)

```
for i in range(1,101):
    if i%3==0 and i%5==0:
        print("Python UMS")
    elif i%3==0:
        print("Python")
    elif i%5==0:
        print("UMS")
    else:
        print(i)
```

Python 3.8.2 Shell

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/9.py =====
1
2
Python
4
UMS
Python
7
8
Python
UMS
11
Python
13
14
Python UMS
16
17
Python
19
UMS
Python
22
23
Python
UMS
26
Python
28
29
Python UMS
31
32
Python
34
```

10.py - C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/10.py (3.8.2)

```
for i in range(1,101):
    if i%3==0 and i%5==0:
        print("Python UMS")
    elif i%3==0:
        print("Python")
    elif i%5==0:
        print("UMS")
    else:
        print(i)
```

Python 3.8.2 Shell

```
62
Python
64
UMS
67
Python
68
Python
UMS
71
Python
73
74
Python UMS
76
77
Python
79
UMS
Python
82
83
Python
UMS
86
Python
88
89
Python UMS
91
92
Python
94
UMS
97
98
Python
UMS
```

10.

10.py - C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/10.py (3.8.2)

```
from math import sqrt as akar
def selesaikanABC(a,b,c):
    a = float(a)
    b = float(b)
    c = float(c)
    D = b**2 - 4*a*c
    if D < 0:
        return "Determinannya negatif. Persamaan tidak mempunyai akar real."
    else:
        x1 = (-b + akar(D))/(2*a)
        x2 = (-b - akar(D))/(2*a)
        hasil = (x1,x2)
        return hasil
print(selesaikanABC(1,2,3))
```

Python 3.8.2 Shell

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/10.py =====
Determinannya negatif. Persamaan tidak mempunyai akar real.
>>>
```

11.

The image shows a WPS Office window with a document titled 'Document3'. The document contains a Python script named '11.py' located at 'C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/11.py (3.8.2)'. The script defines a function 'apekahKabisat(n)' that checks if a year is a leap year based on the following logic: if the year is divisible by 400, it is a leap year; if it is divisible by 100 but not 400, it is not a leap year; if it is divisible by 4 but not 100, it is a leap year; otherwise, it is not a leap year. The script then prints the results for the years 1896, 1897, 1900, 2000, 2004, 2100, and 2400.

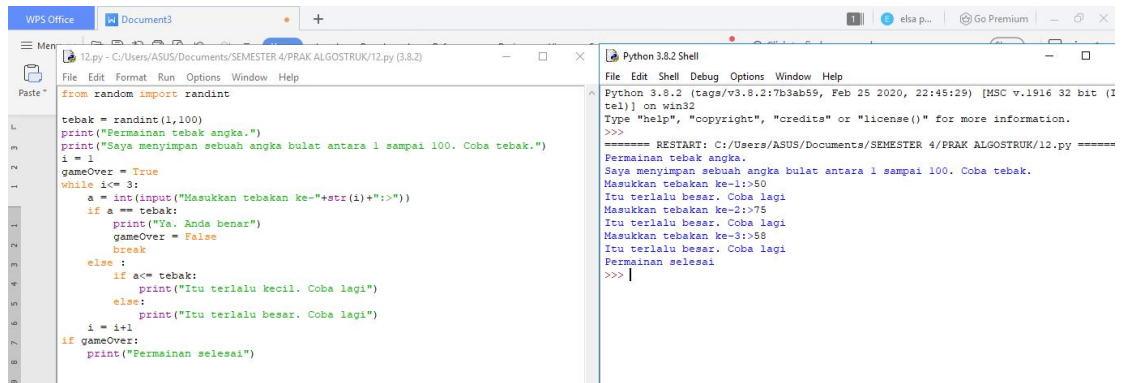
```
def apekahKabisat(n):
    if n%4==0:
        if n%100==0 and n%400==0:
            return True
        elif n%100==0 and n%400!=0:
            return False
        return True
    return False

print(apekahKabisat(1896))
print(apekahKabisat(1897))
print(apekahKabisat(1900))
print(apekahKabisat(2000))
print(apekahKabisat(2004))
print(apekahKabisat(2100))
print(apekahKabisat(2400))
```

The Python 3.8.2 Shell window shows the execution of the script. The output is as follows:

```
Python 3.8.2 Shell
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/11.py =====
True
False
False
True
True
False
True
>>> |
```

12.



The screenshot shows a WPS Office window with two panes. The left pane displays a Python script named 12.py, and the right pane shows the output of running this script in a Python 3.8.2 Shell.

**Python Script (12.py):**

```
from random import randint

tebak = randint(1,100)
print("Permainan tebak angka.")
print("Saya menyimpan sebuah angka bulat antara 1 sampai 100. Coba tebak.")
i = 1
gameOver = True
while i <= 3:
    a = int(input("Masukkan tebakan ke-"+str(i)+">"))
    if a == tebak:
        print("Ya. Anda benar")
        gameOver = False
        break
    else:
        if a < tebak:
            print("Itu terlalu kecil. Coba lagi")
        else:
            print("Itu terlalu besar. Coba lagi")
    i = i+1
if gameOver:
    print("Permainan selesai")
```

**Python 3.8.2 Shell Output:**

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/12.py =====
Permainan tebak angka.
Saya menyimpan sebuah angka bulat antara 1 sampai 100. Coba tebak.
Masukkan tebakan ke-1:>50
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-2:>75
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-3:>58
Itu terlalu besar. Coba lagi
Permainan selesai
>>> |
```

13.

```

def katakan(angka):
    satuan = ["satu", "dua", "tiga", "empat", "lima", "enam", "tujuh", "delapan",
              "sembilan", "sepuluh", "sebelas", "dua belas", "tiga belas",
              "empat belas", "lima belas", "enam belas", "tujuh belas",
              "delapan belas", "sembilan belas"]
    angka = '{0,.0f}'.format(int(angka))
    angka = angka.split(",")
    katakan = []
    idx = 1
    for x in angka[::-1]:
        seribu = False
        if idx == 2 and x[-1] != "0":
            if int(x) < 2:
                katakan.append("seribu")
            seribu = True
        else:
            katakan.append("ribu")
        if idx == 3 and x[-1] != "0":
            katakan.append("juta")
        if seribu == False:
            if int(x[-2:]) < 20 and int(x[-2:]) > 0:
                katakan.append(satuan[int(x[-2:])-1])
            elif int(x[-2:]) > 0:
                if int(x[-1]) != 0:
                    katakan.append(satuan[int(x[-1])-1])
                if int(x[-2]) != 0:
                    katakan.append(satuan[int(x[-2])-1] + " puluh")
            if int(x[0]) > 2 and len(x) == 3:
                katakan.append(satuan[int(x[0])-1] + " ratus")
            elif len(x) == 3 and int(x[0]) != 0:
                katakan.append("seratus")
            idx += 1
    return " ".join(katakan[::-1])

print(katakan(3125750))
  
```

```

Python 3.8.2 Shell
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/13.py =====
>>> tiga juta seratus dua puluh lima ribu tujuh ratus lima puluh
>>>
  
```

14.

```

def formatRupiah(num):
    a = "Rp {:,}".format(num)
    return ".".join(a.split(","))

print(formatRupiah(1500))
print(formatRupiah(2560000))
  
```

```

Python 3.8.2 Shell
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/14.py =====
>>> Rp 1.500
>>> Rp 2.560.000
>>>
  
```