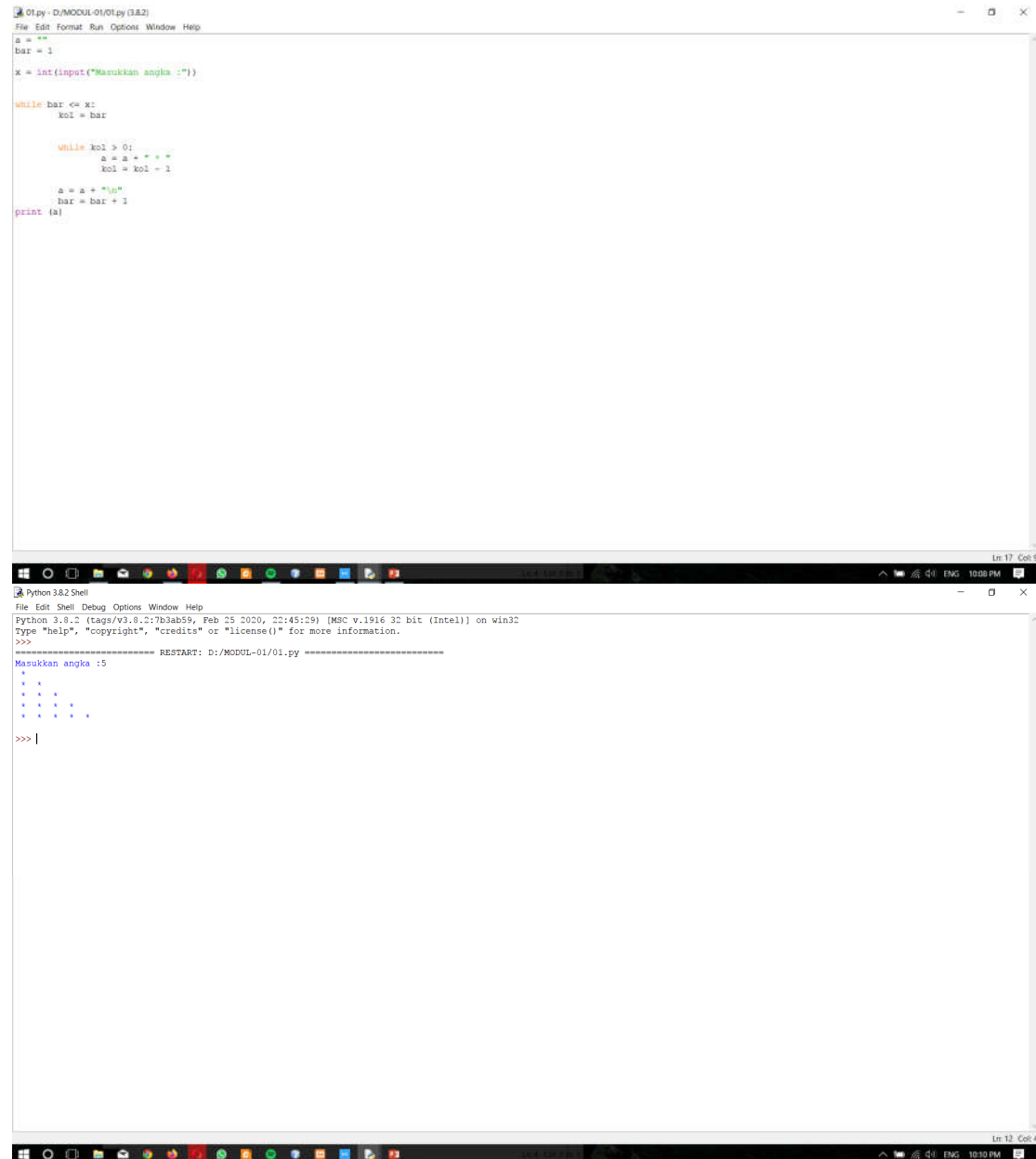


Nama : Luqman Hanung Asidiq  
NIM : L200180035  
Kelas : B

## Laporan Praktikum Algoritma dan Struktur Data Modul 1

1.



```
O1.py - D:/MODUL-01/O1.py (3.8.2)
File Edit Format Run Options Window Help
a = ""
bar = 1

x = int(input("Masukkan angka :"))

while bar <= x:
    kol = bar

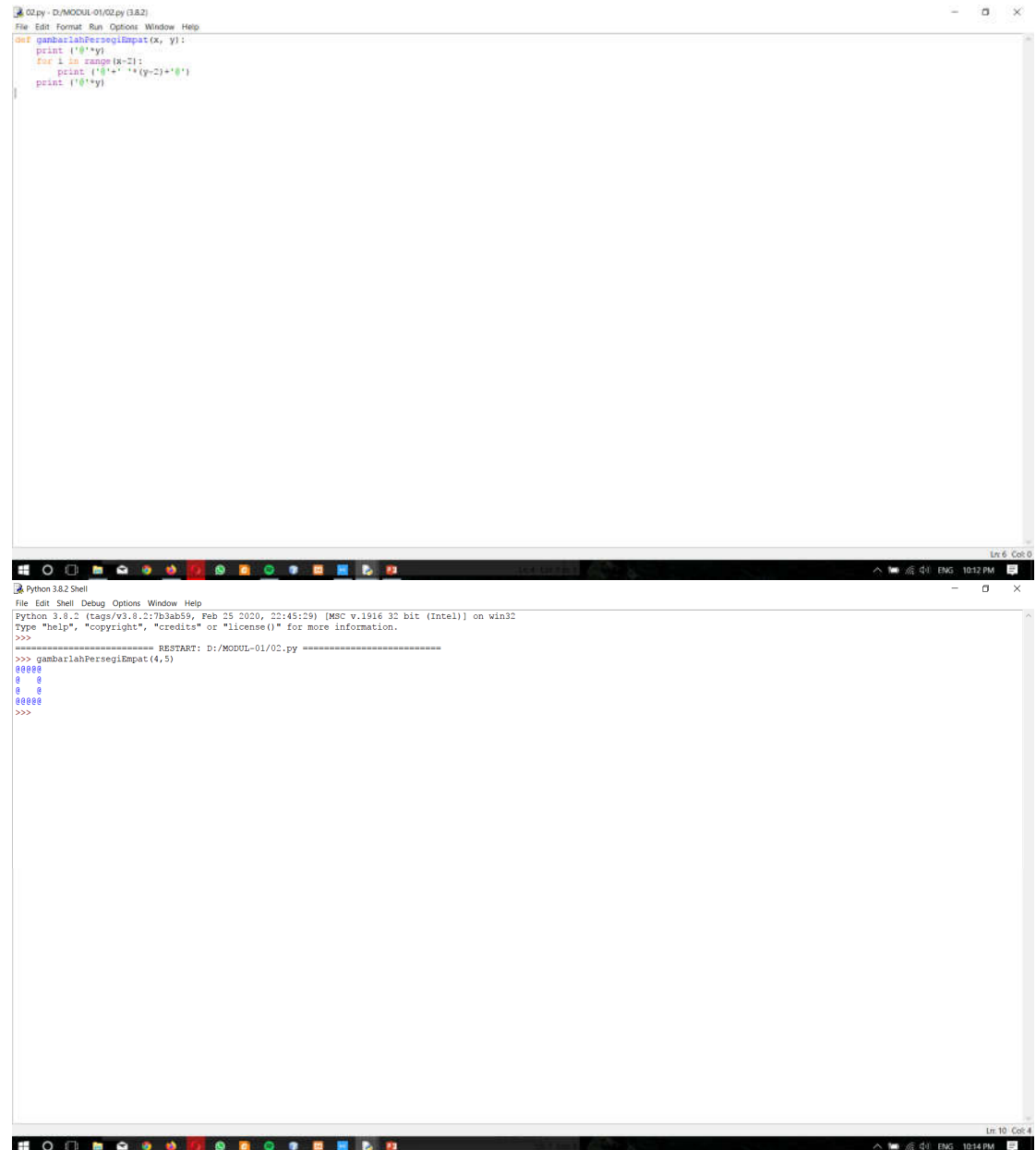
    while kol > 0:
        a = a + " * "
        kol = kol - 1

    a = a + "\n"
    bar = bar + 1

print(a)
```

```
Python 3.8.2 Shell
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: D:/MODUL-01/O1.py =====
Masukkan angka :5
*
* *
* * *
* * * *
* * * * *
>>> |
```

2.



The image shows a Windows desktop environment with two windows open. The top window is a Python IDE titled '02.py - D:\MODUL-01\02.py (3.8.2)'. It contains the following code:

```
def gambarlahPersegiRmpat(x, y):  
    print ('0'*y)  
    for i in range(x-2):  
        print ('0'*x+'*(y-2)+'0')  
    print ('0'*y)
```

The bottom window is a 'Python 3.8.2 Shell'. It shows the execution of the function with the command: `>>> gambarlahPersegiRmpat(4,5)`. The output is a 4x5 grid of characters: the first and last rows are '00000', and the middle two rows are '0000\*(3-2)0000'.

```
Python 3.8.2 Shell  
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: D:\MODUL-01\02.py =====  
>>> gambarlahPersegiRmpat(4,5)  
00000  
0000*(3-2)0000  
0000*(3-2)0000  
00000  
>>>
```

3.

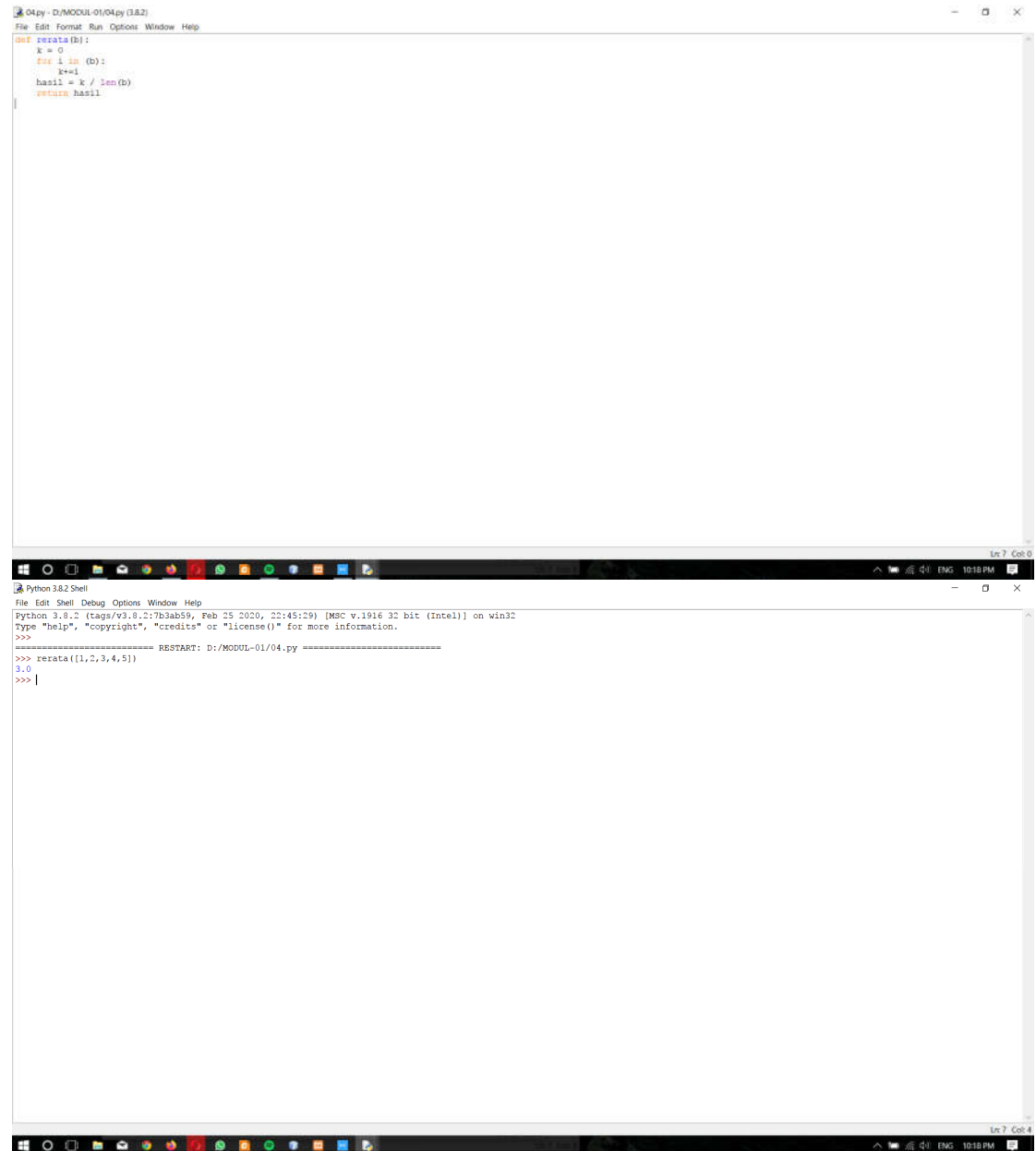
```
O1py - D:\MOCKOL-01\O1py (3.8.2)
File Edit Format Run Options Window Help

#func 3a
def jumlahHurufVokal(x):
    vokal = "AIUEOaiueo"
    a = len(x)
    b = 0
    for k in x:
        if k in vokal:
            b+=1
    c = len(b)
    return (a,c)

#func 3b
def jumlahHurufKonsonan(x):
    konsonan = "BCDFGHJKLMNPQRSTVWXYZbcdfghjklmnpqrstvwxyz"
    a = len(x)
    b = 0
    for k in x:
        if k in konsonan:
            b+=1
    c = len(b)
    return (a,c)

Python 3.8.2 Shell
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: D:\MOCKOL-01\O1py =====
>>> jumlahHurufVokal('budi')
(4, 2)
>>> jumlahHurufKonsonan("dudi")
(4, 2)
>>>
```

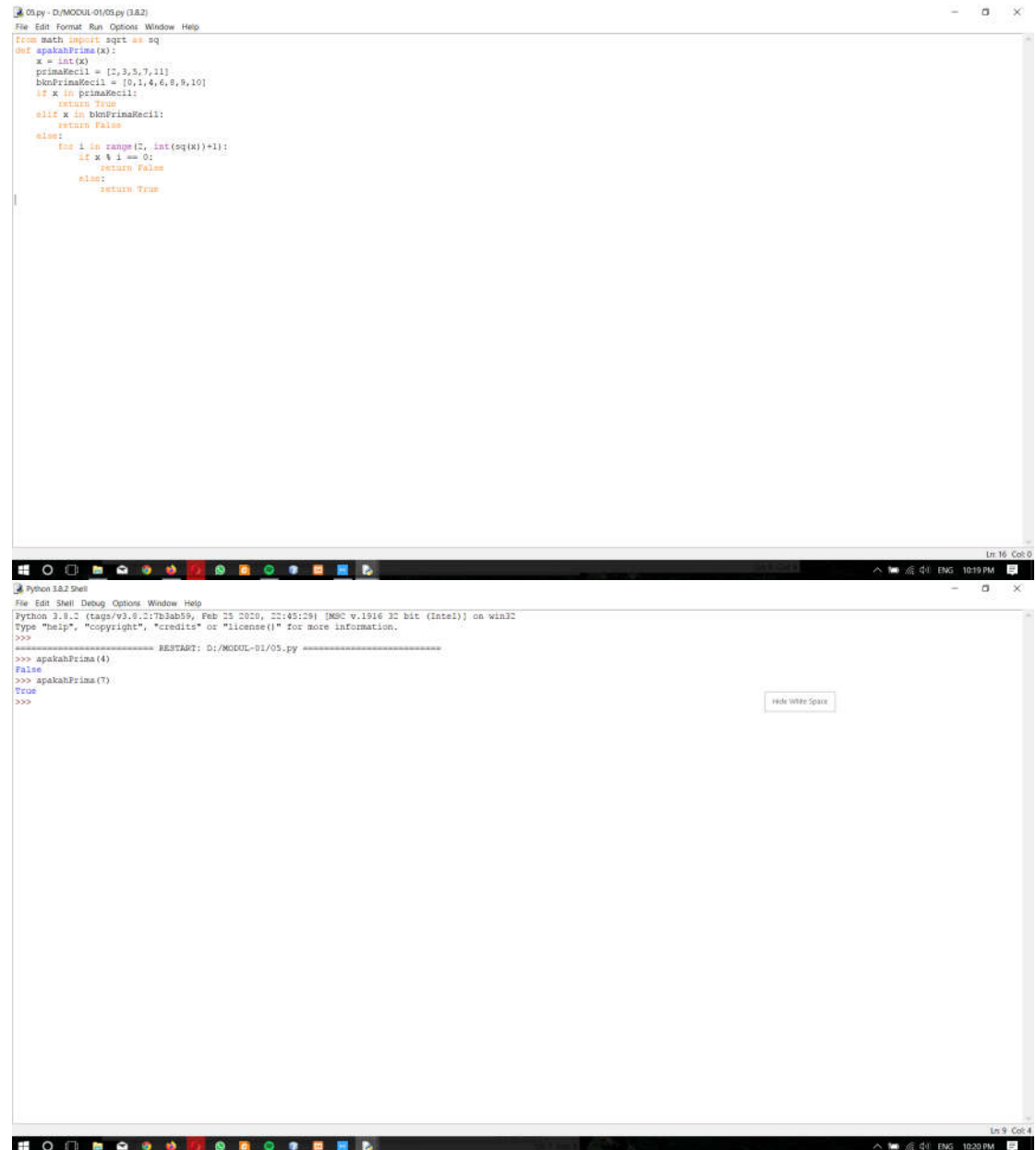
4.



```
O4.py - D:\MODUL-01\O4.py (3.8.2)
File Edit Format Run Options Window Help
def rerata(D):
    k = 0
    for i in D:
        k+=i
    hasil = k / len(D)
    return hasil

Python 3.8.2 Shell
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: D:\MODUL-01\O4.py =====
>>> rerata([1,2,3,4,5])
3.0
>>> |
```

5.



The image shows a Python IDE window titled '05.py - D:\MOCKOL-01\05.py (382)' and a Python Shell window titled 'Python 3.8.2 Shell'.

The IDE window contains the following code:

```
from math import sqrt as sq
def apakahPrima(x):
    n = int(x)
    primaRecil = [2,3,5,7,11]
    bknPrimaRecil = [0,1,4,6,8,9,10]
    if x in primaRecil:
        return True
    elif x in bknPrimaRecil:
        return False
    else:
        for i in range(2, int(sq(x))+1):
            if x % i == 0:
                return False
            else:
                return True
```

The Python Shell window shows the execution of the function:

```
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: D:\MOCKOL-01\05.py =====
>>> apakahPrima(4)
False
>>> apakahPrima(7)
True
>>>
```

The task is to modify the function to return True for prime numbers and False for non-prime numbers. The current code already does this, but it has a bug: it returns False for prime numbers greater than 11. The fix is to remove the 'else' block and return True at the end of the function.

```
def apakahPrima(x):
    n = int(x)
    primaRecil = [2,3,5,7,11]
    bknPrimaRecil = [0,1,4,6,8,9,10]
    if x in primaRecil:
        return True
    elif x in bknPrimaRecil:
        return False
    else:
        for i in range(2, int(sq(x))+1):
            if x % i == 0:
                return False
        return True
```

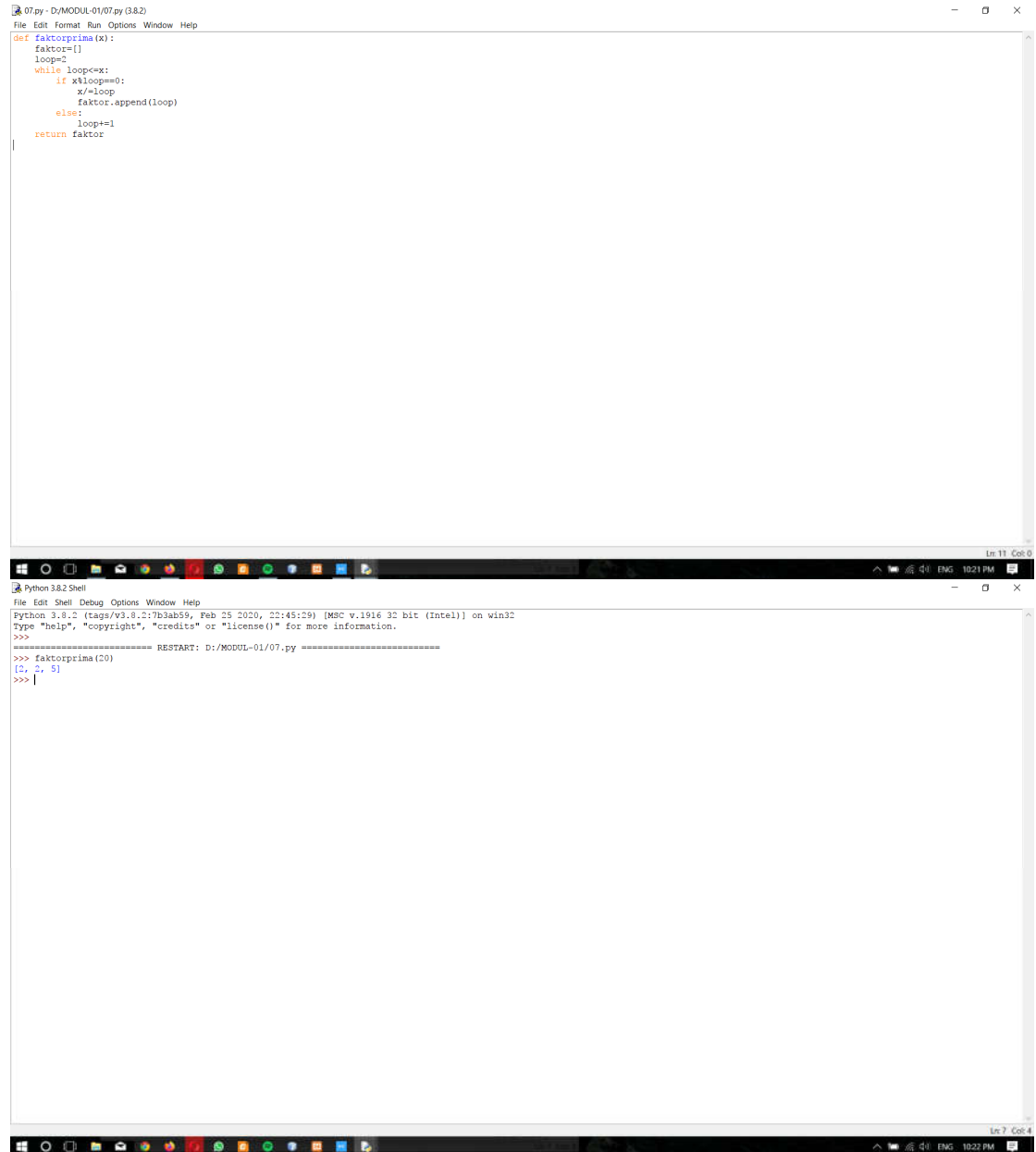
6.

```
O6.py - D:\MODUL-01\O6.py (3.8.2)
File Edit Format Run Options Window Help

def apaPrima():
    lower = 2
    upper = 1000
    print("Bilangan prima antara",lower,"and",upper,":")
    for num in range(lower,upper + 1):
        if num > 1:
            for i in range(2,num):
                if (num % i) == 0:
                    break
            else:
                print(num)

Python 3.8.2 Shell
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: D:\MODUL-01\O6.py =====
>>> apaPrima()
Bilangan prima antara 2 and 1000 :
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
101
103
107
109
113
127
131
137
```

7.



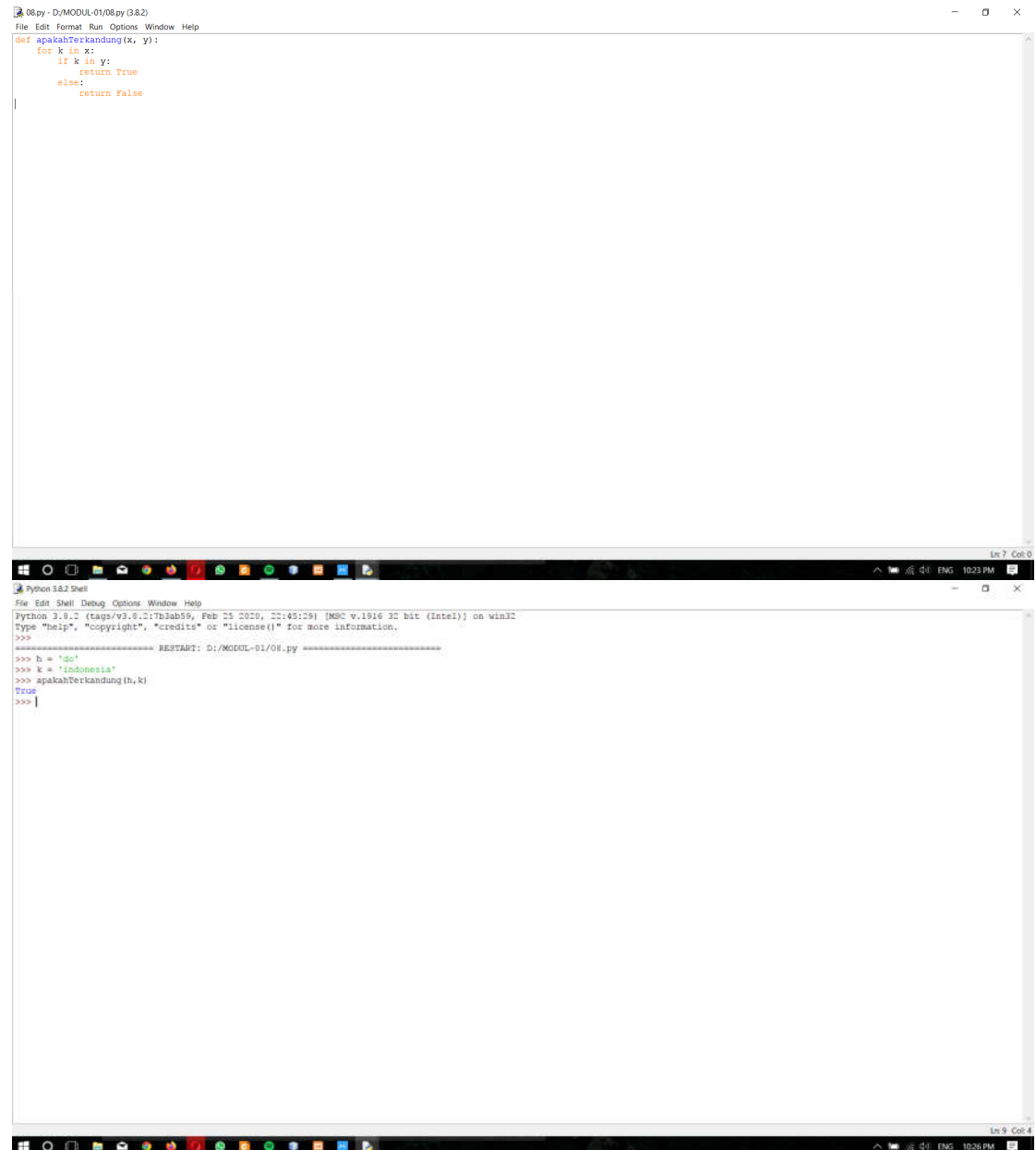
The image shows a screenshot of a Python IDE with two windows. The top window is a text editor titled '07.py - D:/MODUL-01/07.py (3.8.2)' containing a function definition for 'faktorprima(x)'. The function initializes an empty list 'faktor', sets 'loop=2', and enters a while loop 'while loop<=x:'. Inside the loop, it checks 'if x%loop==0:'. If true, it divides 'x' by 'loop' and appends 'loop' to 'faktor'. If false, it increments 'loop' by 1. The function returns 'faktor'.

```
def faktorprima(x):  
    faktor=[]  
    loop=2  
    while loop<=x:  
        if x%loop==0:  
            x/=loop  
            faktor.append(loop)  
        else:  
            loop+=1  
    return faktor
```

The bottom window is a 'Python 3.8.2 Shell' titled 'Python 3.8.2 Shell'. It shows the execution of 'faktorprima(20)', which returns the list '[2, 2, 5]'. The shell also displays the Python version and a restart message.

```
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: D:/MODUL-01/07.py =====  
>>> faktorprima(20)  
[2, 2, 5]  
>>>
```

8.



The image shows a screenshot of a computer screen with two windows. The top window is a Python IDE titled '08.py - D:/MODUL-01/08.py (3.8.2)'. It contains the following Python code:

```
def apakahTerkandung(x, y):  
    for k in x:  
        if k in y:  
            return True  
        else:  
            return False
```

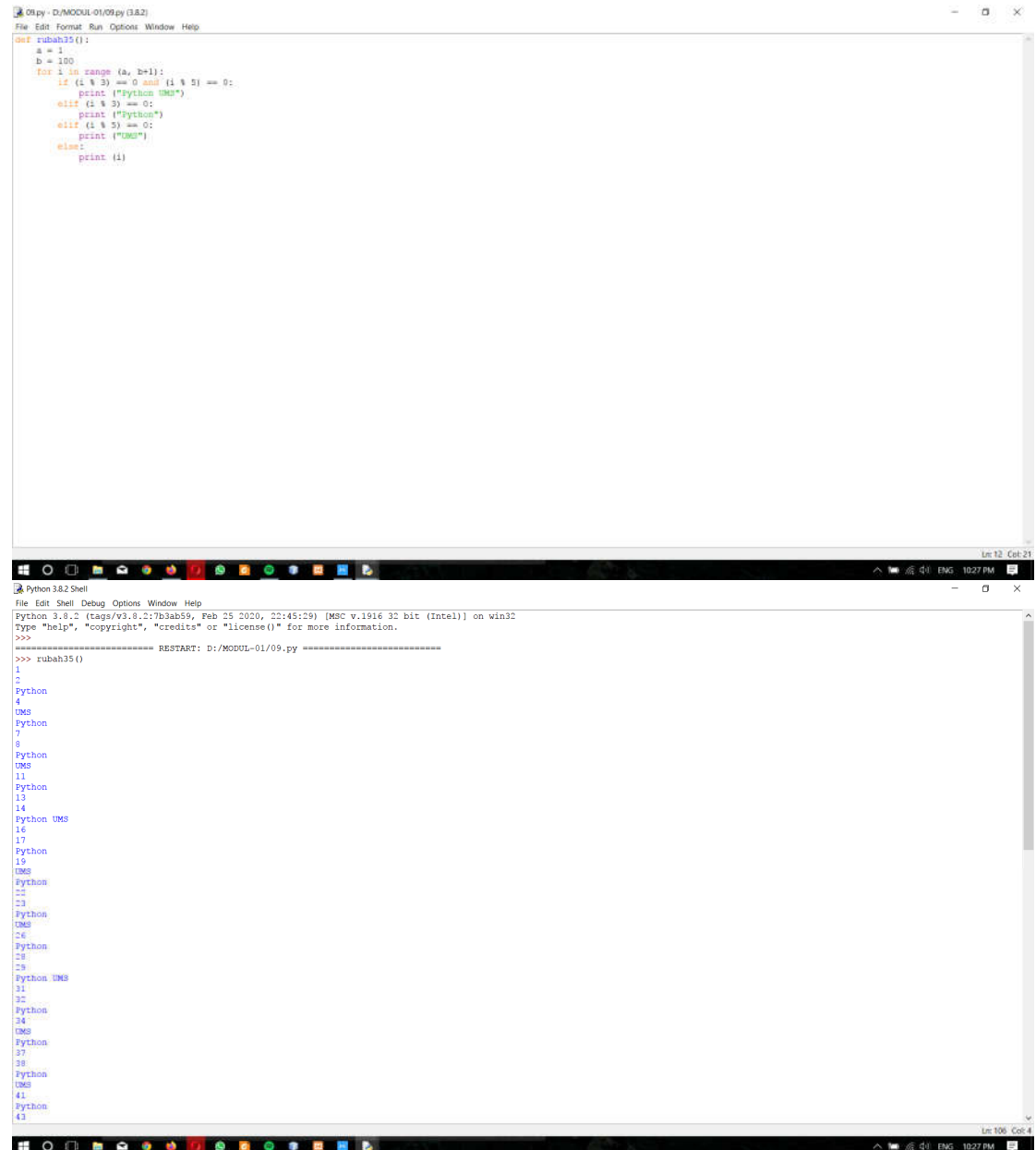
The bottom window is a 'Python 3.8.2 Shell' titled 'Python 3.8.2 Shell'. It shows the execution of the code from the IDE. The output is as follows:

```
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: D:/MODUL-01/08.py =====  
>>> h = 'do'  
>>> k = 'Indonesia'  
>>> apakahTerkandung(h, k)  
True  
>>> |
```

The task is to complete the function `apakahTerkandung(x, y)` by adding a loop that iterates over the characters in `x` and checks if any character is present in `y`. If a character is found, the function should return `True`; otherwise, it should return `False`.



9.



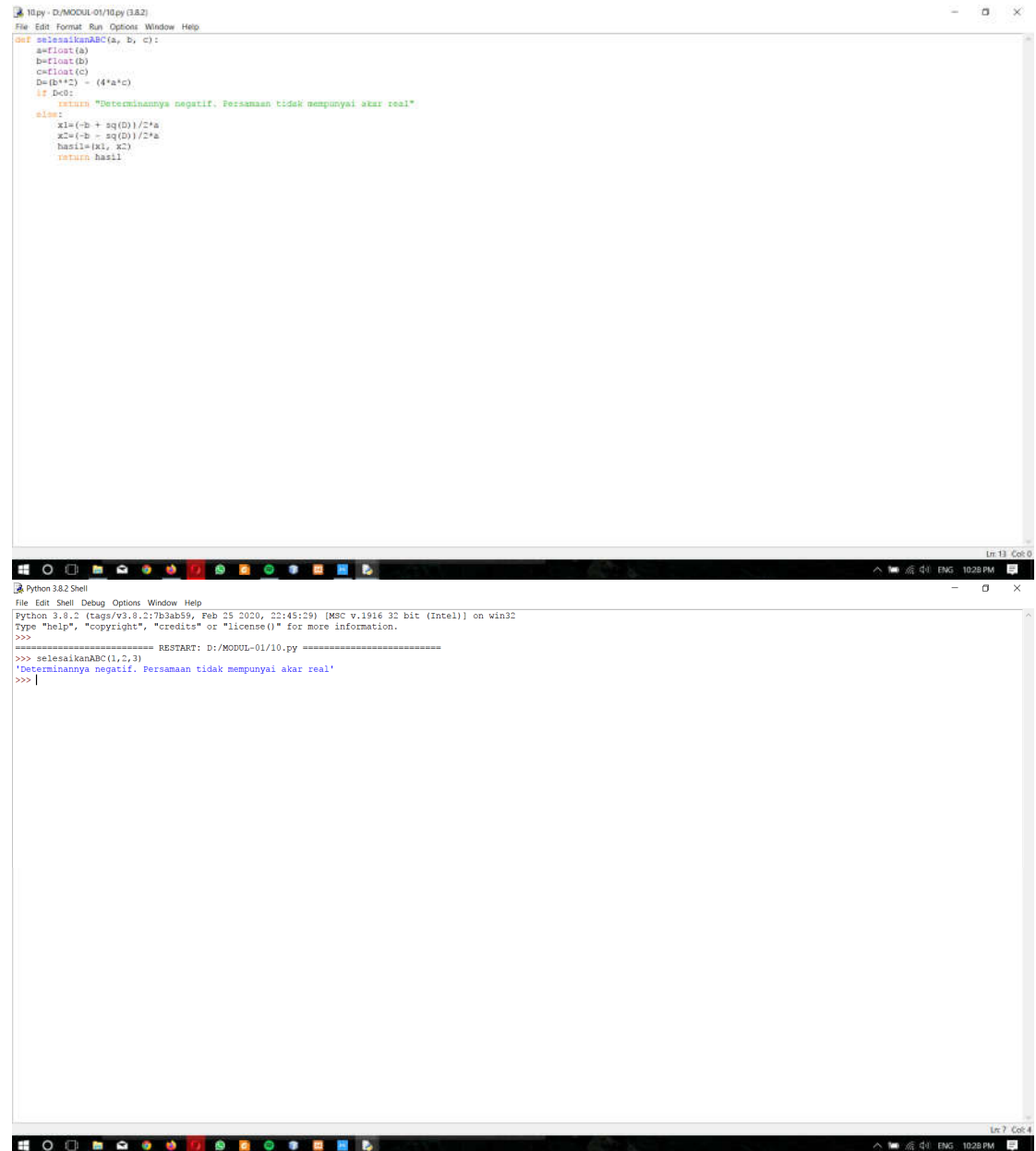
The image shows a screenshot of a Python IDE with two windows. The top window is a script editor titled '09.py - D:\MODUL-01\09.py (3.8.2)'. It contains a Python script that iterates from 1 to 100 and prints 'Python' or 'UMS' based on divisibility rules. The bottom window is a 'Python 3.8.2 Shell' showing the execution of the script, with the output 'Python' and 'UMS' alternating every four lines.

```
09.py - D:\MODUL-01\09.py (3.8.2)
File Edit Format Run Options Window Help

def rubah35():
    a = 1
    b = 100
    for i in range (a, b+1):
        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
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: D:/MODUL-01/09.py =====
>>> rubah35()
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
UMS
Python
37
38
Python
UMS
41
Python
43
```

10.



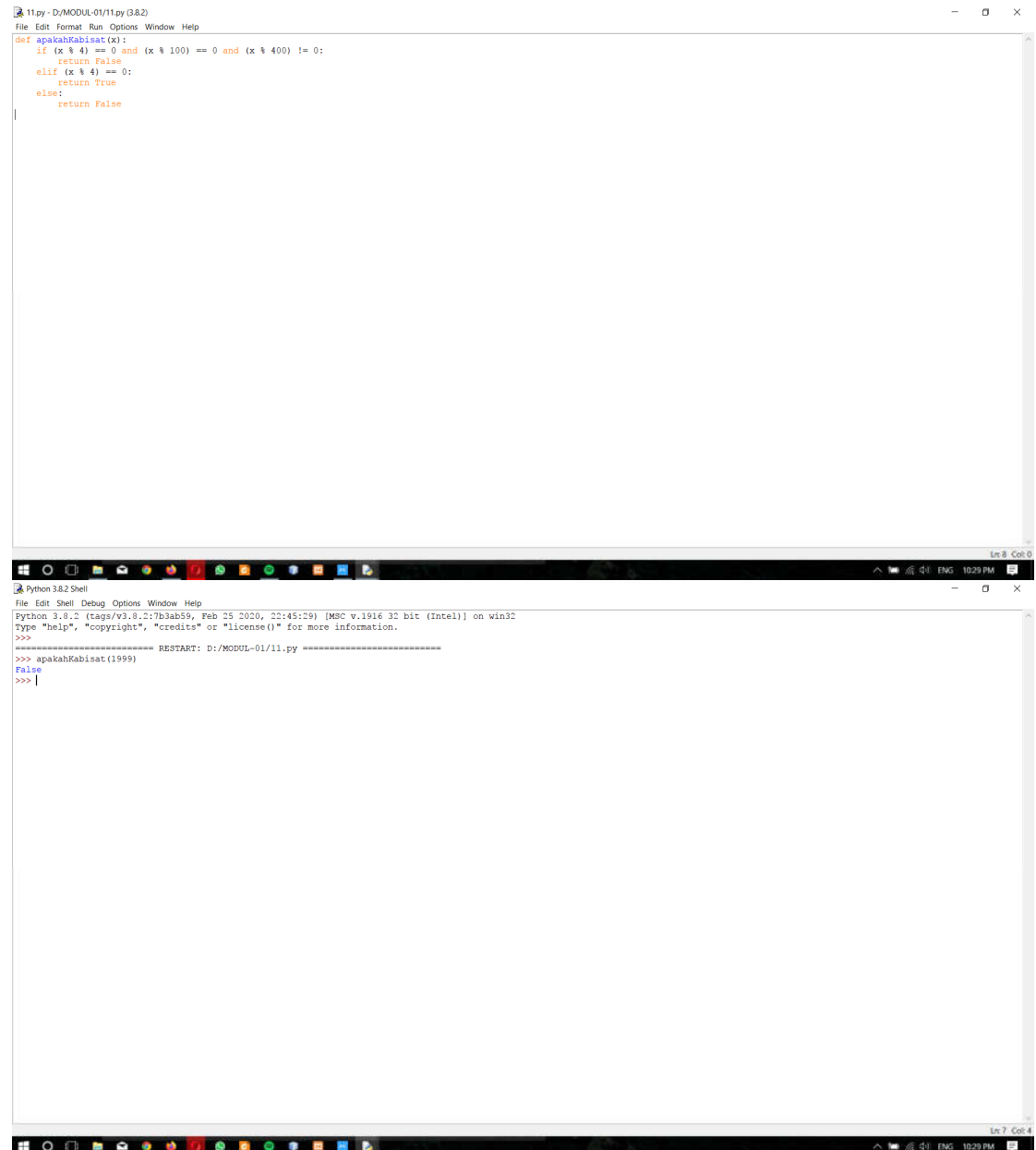
The image shows a Python IDE window titled '10.py - D:\MODUL-01\10.py (3.8.2)'. The script defines a function `selesaikanABC(a, b, c)` that calculates the roots of a quadratic equation  $ax^2 + bx + c = 0$ . It uses the discriminant  $D = b^2 - 4ac$  to determine if the roots are real. If  $D < 0$ , it returns a message in Indonesian: "Determinannya negatif. Persamaan tidak mempunyai akar real". Otherwise, it calculates the two roots  $x_1$  and  $x_2$  and returns them as a list.

```
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 + sq(D))/2*a
        x2=(-b - sq(D))/2*a
        hasil=[x1, x2]
        return hasil
```

Below the IDE, a 'Python 3.8.2 Shell' window shows the execution of the script. It displays the restart message and the output of the function call `selesaikanABC(1,2,3)`, which is the Indonesian message: "Determinannya negatif. Persamaan tidak mempunyai akar real".

```
Python 3.8.2 Shell
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: D:\MODUL-01\10.py =====
>>> selesaikanABC(1,2,3)
'Determinannya negatif. Persamaan tidak mempunyai akar real'
>>> |
```

11.

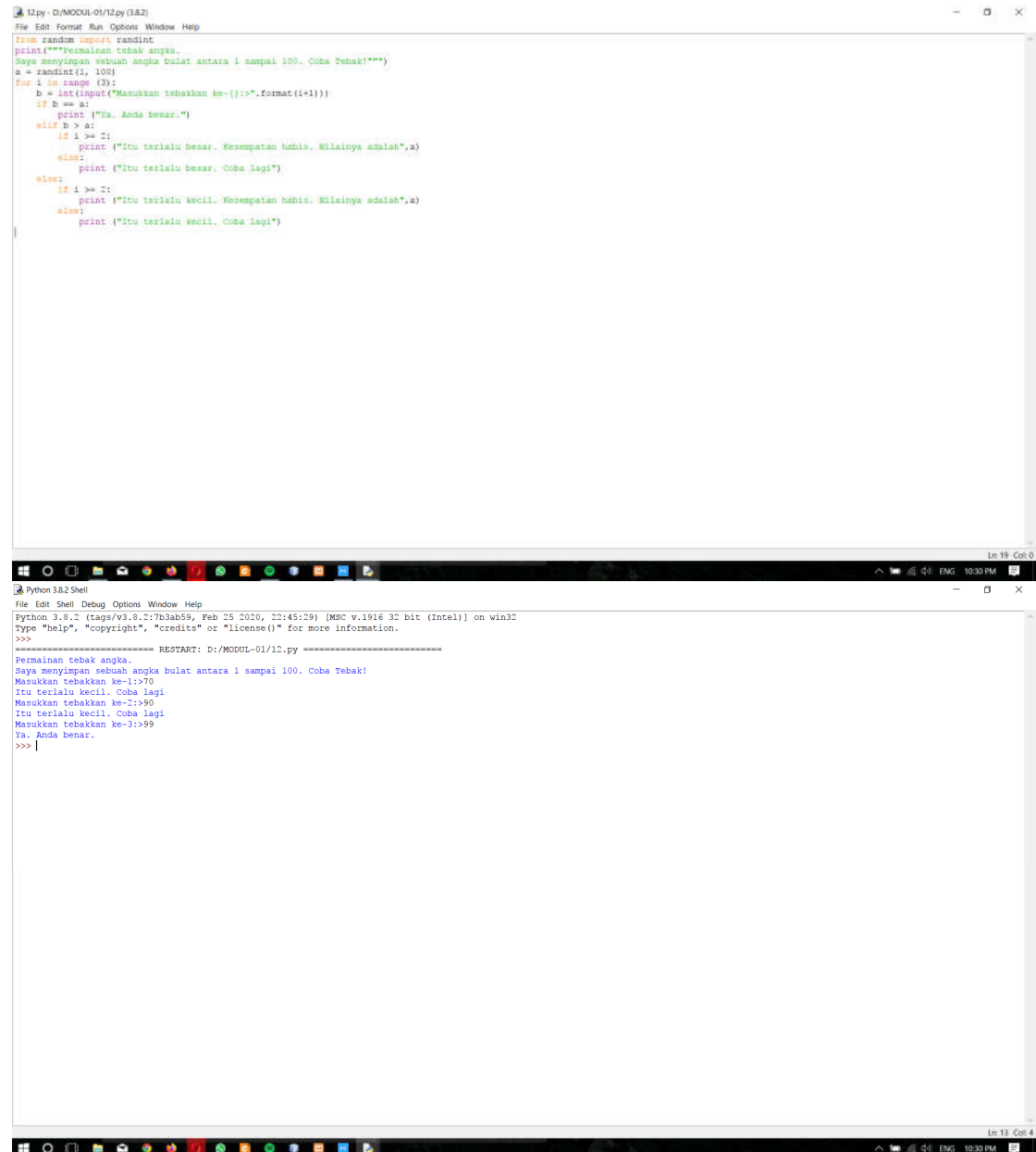


The image shows a screenshot of a Python IDE with two windows. The top window, titled '11.py - D:/MODUL-01/11.py (3.8.2)', contains a function definition for `apakahKabisat(x)`. The function checks if a year `x` is a leap year by testing if it is divisible by 4, 100, and 400. The bottom window, titled 'Python 3.8.2 Shell', shows the function being called with the argument 1999, resulting in the output `False`.

```
11.py - D:/MODUL-01/11.py (3.8.2)
File Edit Format Run Options Window Help
def apakahKabisat(x):
    if (x % 4) == 0 and (x % 100) != 0 and (x % 400) != 0:
        return True
    elif (x % 4) == 0:
        return True
    else:
        return False

Python 3.8.2 Shell
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: D:/MODUL-01/11.py =====
>>> apakahKabisat(1999)
False
>>> |
```

12.



The image shows a Python IDE window titled "12.py - D:\MODUL-01\12.py (382)". The script implements a number guessing game. It imports the random module and generates a random number 'a' between 1 and 100. It then enters a loop where the user is prompted to guess a number 'b'. The program checks if 'b' is equal to 'a', greater than 'a', or less than 'a', providing feedback and allowing the user to guess again. The script is as follows:

```
from random import randint
print("""Permainan tebak angka.
Saya menyimpan sebuah angka bulat antara 1 sampai 100. Coba Tebak!""")
a = randint(1, 100)
for i in range(3):
    b = int(input("Masukkan tebakan ke-{}>".format(i+1)))
    if b == a:
        print("Ya, Anda benar.")
    elif b > a:
        if i >= 2:
            print("Itu terlalu Besar. Kesempatan habis. Silainya adalah",a)
        else:
            print("Itu terlalu Besar. Coba lagi")
    else:
        if i >= 2:
            print("Itu terlalu kecil. Kesempatan habis. Silainya adalah",a)
        else:
            print("Itu terlalu kecil. Coba lagi")
```

Below the script, the Python Shell output is shown. It indicates a restart of the script and displays the game's execution, including the user's guesses and the program's feedback.

```
Python 3.8.2 Shell
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: D:\MODUL-01\12.py =====
Permainan tebak angka.
Saya menyimpan sebuah angka bulat antara 1 sampai 100. Coba Tebak!
Masukkan tebakan ke-1:>70
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-2:>90
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-3:>99
Ya, Anda benar.
>>> |
```

13.

```
llpy - D:\MODUL-01\llpy (3A2)
File Edit Format Run Options Window Help
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"]
    if
        angka = '%i,%i,%i' % (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]) < 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[-3]) != 0:
                        katakan.append(satuan[int(x[-3]) - 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])

Python 3.8.2 Shell
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: D:\MODUL-01\llpy =====
>>> katakan(230345)
'seratus tiga puluh tiga ratus empat puluh lima'
>>> |
```

14.



The image shows a screenshot of a Windows desktop with two windows open. The top window is a Python IDE titled "14.py - D:\MODUL-01\14.py (3.8.2)". It contains a single function definition: `def FormatRupiah(n):` followed by an indented line `x = '{:,}'.format(n).replace(',','')`, and another indented line `return "Rp " + x`. The bottom window is a "Python 3.8.2 Shell". It displays the Python version and build information, followed by a prompt `>>>`. The user has entered `>>> FormatRupiah(14000)`, and the shell has responded with `'Rp 14.000'`. The taskbar at the bottom shows various application icons and the system clock indicating 10:32 PM.

```
14.py - D:\MODUL-01\14.py (3.8.2)
File Edit Format Run Options Window Help
def FormatRupiah(n):
    x = '{:,}'.format(n).replace(',','')
    return "Rp " + x

Python 3.8.2 Shell
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: D:\MODUL-01\14.py =====
>>> FormatRupiah(14000)
'Rp 14.000'
>>> |
```