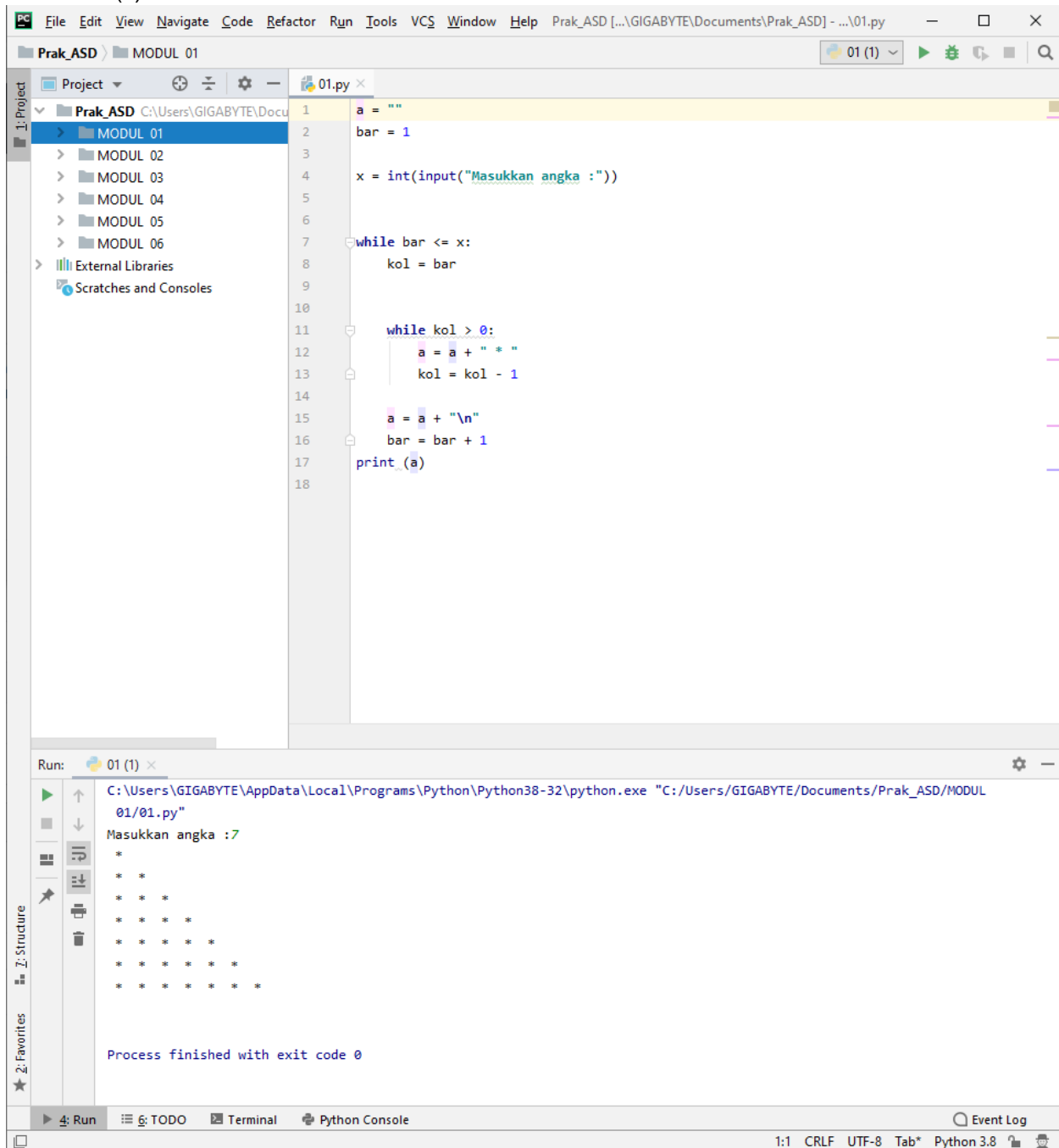


Nama : W. Faisal Hari Dewanto
NIM : L200180046
Kelas : B

Modul 1

1. Fungsi cetakSiku(x)



The screenshot displays a Python IDE with a project named 'Prak_ASD'. The left sidebar shows a file explorer with 'MODUL 01' selected. The main editor window shows the code for '01.py'.

```
1 a = ""
2 bar = 1
3
4 x = int(input("Masukkan angka :"))
5
6
7 while bar <= x:
8     kol = bar
9
10
11     while kol > 0:
12         a = a + " * "
13         kol = kol - 1
14
15     a = a + "\n"
16     bar = bar + 1
17 print(a)
18
```

The bottom panel shows the 'Run' output for '01 (1)'. The command executed is 'C:\Users\GIGABYTE\AppData\Local\Programs\Python\Python38-32\python.exe "C:/Users/GIGABYTE/Documents/Prak_ASD/MODUL 01/01.py"'. The output shows the input 'Masukkan angka :7' followed by a 7x7 grid of asterisks.

```
C:\Users\GIGABYTE\AppData\Local\Programs\Python\Python38-32\python.exe "C:/Users/GIGABYTE/Documents/Prak_ASD/MODUL 01/01.py"
Masukkan angka :7
*
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * * *

Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is UTF-8, the tab is 'Tab*', and the Python version is 3.8.

2. Fungsi yang menerima dua integer

The image shows two windows from a Python 3.8.1 environment. The left window is the Python Shell, and the right window is a text editor showing a Python file named 'modul 1 2.py'.

Python Shell (Left):

```
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\GIGABYTE\Documents\modul 1 2.py =====
>>> gambarlahPersegiEmpat(5,7)
0000000
0      0
0      0
0      0
0      0
0000000
>>>
```

Python File Editor (Right):

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

3. Fungsi menerima sebuah string dan hitung huruf konsonan

The image shows two windows from a Python 3.8.1 environment. The left window is the Python Shell, and the right window is a text editor showing a Python file named '03.py'.

Python Shell (Left):

```
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\03.py =====
>>> jumlahHurufVokal('Faisal')
(6, 3)
>>> jumlahHurufKonsonan('Faisal')
(6, 3)
>>> jumlahHurufKonsonan('FaisalHari')
(10, 5)
>>>
```

Python File Editor (Right):

```
def jumlahHurufVokal(x):
    vokal = "AIUEOaieuo"
    a = len(x)
    b = ""
    for k in x:
        if k in vokal:
            b+=k
    c = len(b)
    return (a,c)

def jumlahHurufKonsonan(x):
    konsonan = "BCDFGHJKLMNPQRSTVWXYZbcd fghjklmnpqrstvwxyz"
    a = len(x)
    b = ""
    for k in x:
        if k in konsonan:
            b+=k
    c = len(b)
    return (a,c)
```

4. Fungsi rerata

The image shows two windows from a Python 3.8.1 environment. The left window is the Python Shell, and the right window is a text editor showing a Python file named '04.py'.

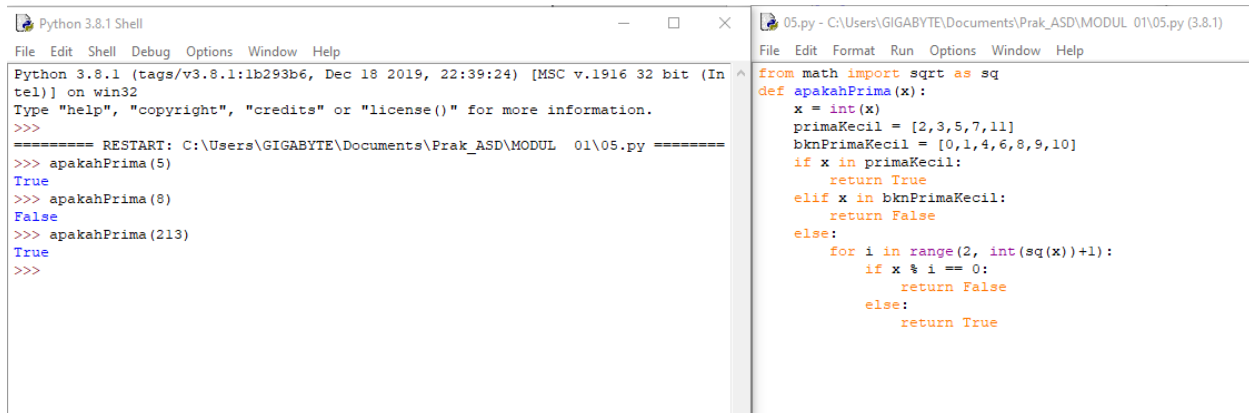
Python Shell (Left):

```
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\04.py =====
>>> rerata([1,2,3,4,5,6])
3.5
>>> g = [1,2,3,9,8,5,4,7]
>>> rerata(g)
4.875
>>>
```

Python File Editor (Right):

```
def rerata(b):
    k = 0
    for i in (b):
        k+=i
    hasil = k / len(b)
    return hasil
```

5. Fungsi bilangan prima

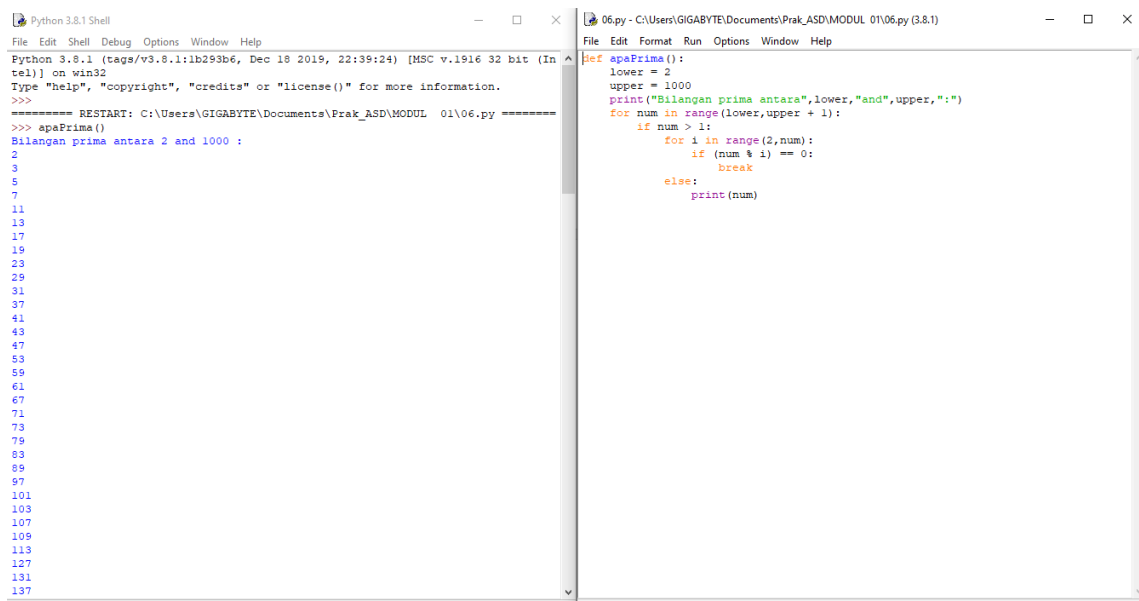


The screenshot shows a Python 3.8.1 Shell window on the left and a file editor window for 05.py on the right. The shell window displays the execution of the `apakahPrima` function for inputs 5, 8, and 213, returning `True`, `False`, and `True` respectively. The file editor shows the source code for `05.py`, which imports `sqrt` from the `math` module and defines the `apakahPrima` function. The function checks if a number `x` is in a list of small primes (`primaKecil = [2, 3, 5, 7, 11]`) or in a list of non-primes (`bknPrimaKecil = [0, 1, 4, 6, 8, 9, 10]`). If `x` is in either list, it returns `True` or `False` accordingly. Otherwise, it checks for divisibility from 2 to `int(sqrt(x))+1`; if any divisor is found, it returns `False`, otherwise `True`.

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\05.py =====
>>> apakahPrima(5)
True
>>> apakahPrima(8)
False
>>> apakahPrima(213)
True
>>>

05.py - C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\05.py (3.8.1)
File Edit Format Run Options Window Help
from math import sqrt as sq
def apakahPrima(x):
    x = int(x)
    primaKecil = [2,3,5,7,11]
    bknPrimaKecil = [0,1,4,6,8,9,10]
    if x in primaKecil:
        return True
    elif x in bknPrimaKecil:
        return False
    else:
        for i in range(2, int(sq(x))+1):
            if x % i == 0:
                return False
            else:
                return True
```

6. Fungsi bilangan prima

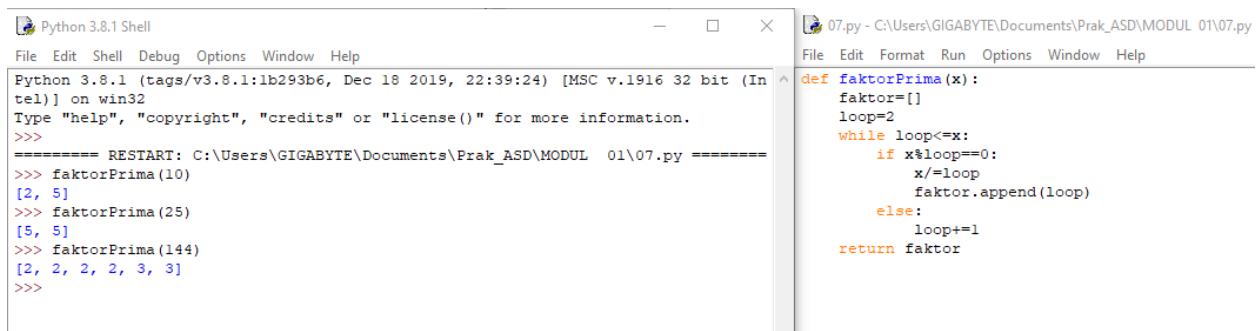


The screenshot shows a Python 3.8.1 Shell window on the left and a file editor window for 06.py on the right. The shell window displays the execution of the `apaPrima` function, which prints all prime numbers between 2 and 1000. The file editor shows the source code for `06.py`, which defines the `apaPrima` function. The function takes `lower` and `upper` as arguments and prints all prime numbers in the range `[lower, upper]`. It uses a nested loop: an outer loop for `num` from `lower` to `upper`, and an inner loop for `i` from 2 to `num`. If `num` is divisible by `i`, it breaks the inner loop. If the inner loop completes without breaking, `num` is a prime and is printed.

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\06.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

06.py - C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\06.py (3.8.1)
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)
```

7. Bilangan positif

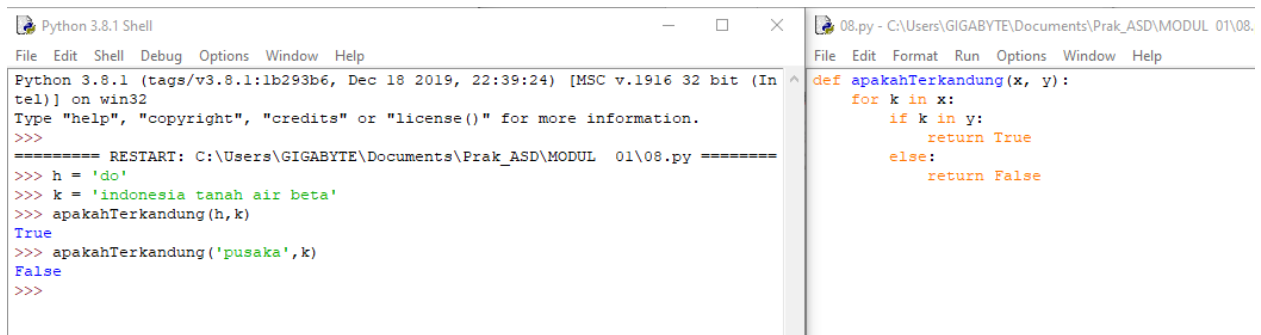


The screenshot shows a Python 3.8.1 Shell window on the left and a file editor window for 07.py on the right. The shell window displays the execution of the `faktorPrima` function for inputs 10, 25, and 144, returning their prime factorizations as lists. The file editor shows the source code for `07.py`, which defines the `faktorPrima` function. The function takes a number `x` and returns a list of its prime factors. It uses a while loop that divides `x` by the smallest prime factor `loop` until `x` becomes 1. The prime factors are stored in the `faktor` list.

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\07.py =====
>>> faktorPrima(10)
[2, 5]
>>> faktorPrima(25)
[5, 5]
>>> faktorPrima(144)
[2, 2, 2, 2, 3, 3]
>>>

07.py - C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\07.py
File Edit Format Run Options Window Help
def faktorPrima(x):
    faktor=[]
    loop=2
    while loop<=x:
        if x%loop==0:
            x/=loop
            faktor.append(loop)
        else:
            loop+=1
    return faktor
```

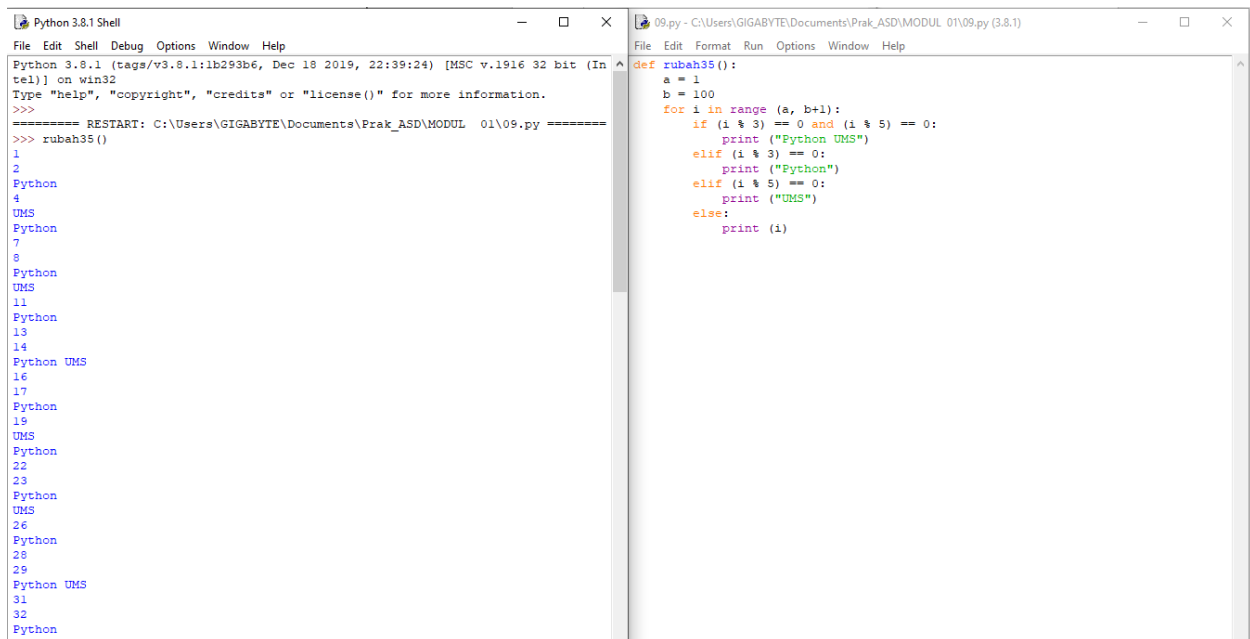
8. Fungsi



```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\08.py =====
>>> h = 'do'
>>> k = 'indonesia tanah air beta'
>>> apakahTerkandung(h,k)
True
>>> apakahTerkandung('pusaka',k)
False
>>>
```

```
08.py - C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\08.py
File Edit Format Run Options Window Help
def apakahTerkandung(x, y):
    for k in x:
        if k in y:
            return True
        else:
            return False
```

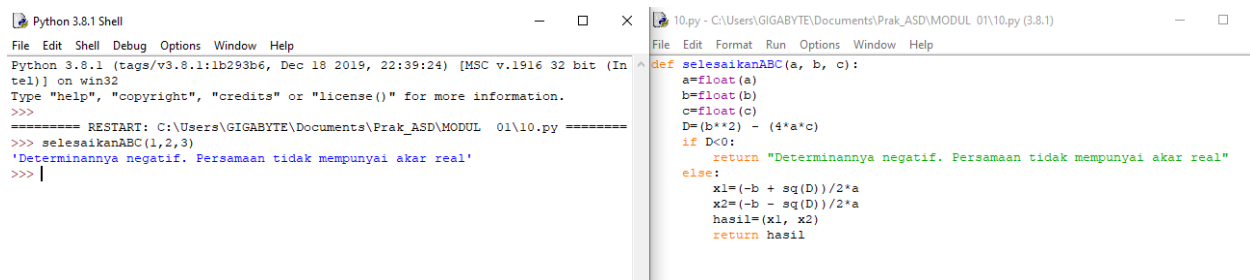
9. Program



```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\09.py =====
>>> rubah35()
1
2 Python
4 UMS
5 Python
7
8 Python
9 UMS
11 Python
13
14 Python UMS
16
17 Python
19 UMS
21 Python
23 UMS
26 Python
28
29 Python UMS
31
32 Python
>>>
```

```
09.py - C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\09.py (3.8.1)
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)
```

10. Modifikasi



```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\10.py =====
>>> selesaikanABC(1,2,3)
'Determinannya negatif. Persamaan tidak mempunyai akar real'
>>>
```

```
10.py - C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\10.py (3.8.1)
File Edit Format Run Options Window Help
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
```

11. Fungsi

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\11.py =====
>>> apakahKabisat(2000)
True
>>> apakahKabisat(2020)
True
>>> |

11.py - C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\11.py (3.8.1)
File Edit Format Run Options Window Help
def apakahKabisat(x):
    if (x % 4) == 0 and (x % 100) != 0 and (x % 400) != 0:
        return False
    elif (x % 4) == 0:
        return True
    else:
        return False
```

12. Fungsi

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\12.py =====
Permainan tebak angka.
Saya menyimpan sebuah angka bulat antara 1 sampai 100. Coba Tebak!
Masukkan tebakan ke-1:>40
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-2:>80
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-3:>50
Itu terlalu besar. Kesempatan habis. Nilainya adalah 22
>>>
>>>

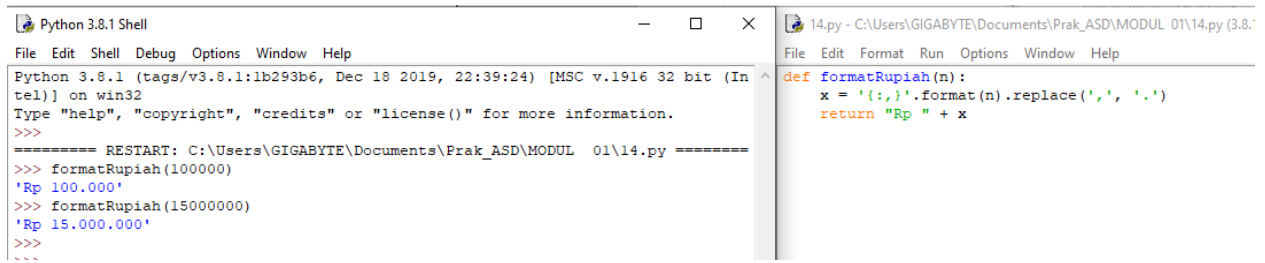
12.py - C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\12.py (3.8.1)
File Edit Format Run Options Window Help
from random import randint
print("""Permainan tebak angka.
Saya menyimpan sebuah angka bulat antara 1 sampai 100. Coba Tebak!""")
a = randint(1, 100)
for i in range(3):
    b = int(input("Masukkan tebakan ke-{}:>".format(i+1)))
    if b == a:
        print("Ya. Anda benar.")
    elif b > a:
        print("Itu terlalu besar. Kesempatan habis. Nilainya adalah",a)
    else:
        print("Itu terlalu besar. Coba lagi")
    elif b < a:
        print("Itu terlalu kecil. Kesempatan habis. Nilainya adalah",a)
    else:
        print("Itu terlalu kecil. Coba lagi")
```

13. Fungsi

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\13.py =====
>>> katakan(464693923919391939)
'empat ratus enam puluh empat enam ratus sembilan puluh tiga sembilan ratus dua puluh tiga sembilan ratus sembilan belas juta tiga ratus sembilan puluh satu ribu sembilan ratus tiga puluh enam'
>>> |

13.py - C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\13.py (3.8.1)
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"]
    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])
```

14. Fungsi



The image shows two windows from a Python IDE. The left window is a 'Python 3.8.1 Shell' with a menu bar (File, Edit, Shell, Debug, Options, Window, Help). It displays the Python version and architecture, followed by a restart message and the execution of a function. The right window is a script editor for '14.py' with a menu bar (File, Edit, Format, Run, Options, Window, Help). It contains the definition of the 'formatRupiah' function.

```
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\GIGABYTE\Documents\Prak_ASD\MODUL 01\14.py =====
>>> formatRupiah(100000)
'Rp 100.000'
>>> formatRupiah(15000000)
'Rp 15.000.000'
>>>
<<<
```

```
def formatRupiah(n):
    x = '{:,}'.format(n).replace(',','.')
    return "Rp " + x
```