

Nama : Yusrina Khairin Rusydina

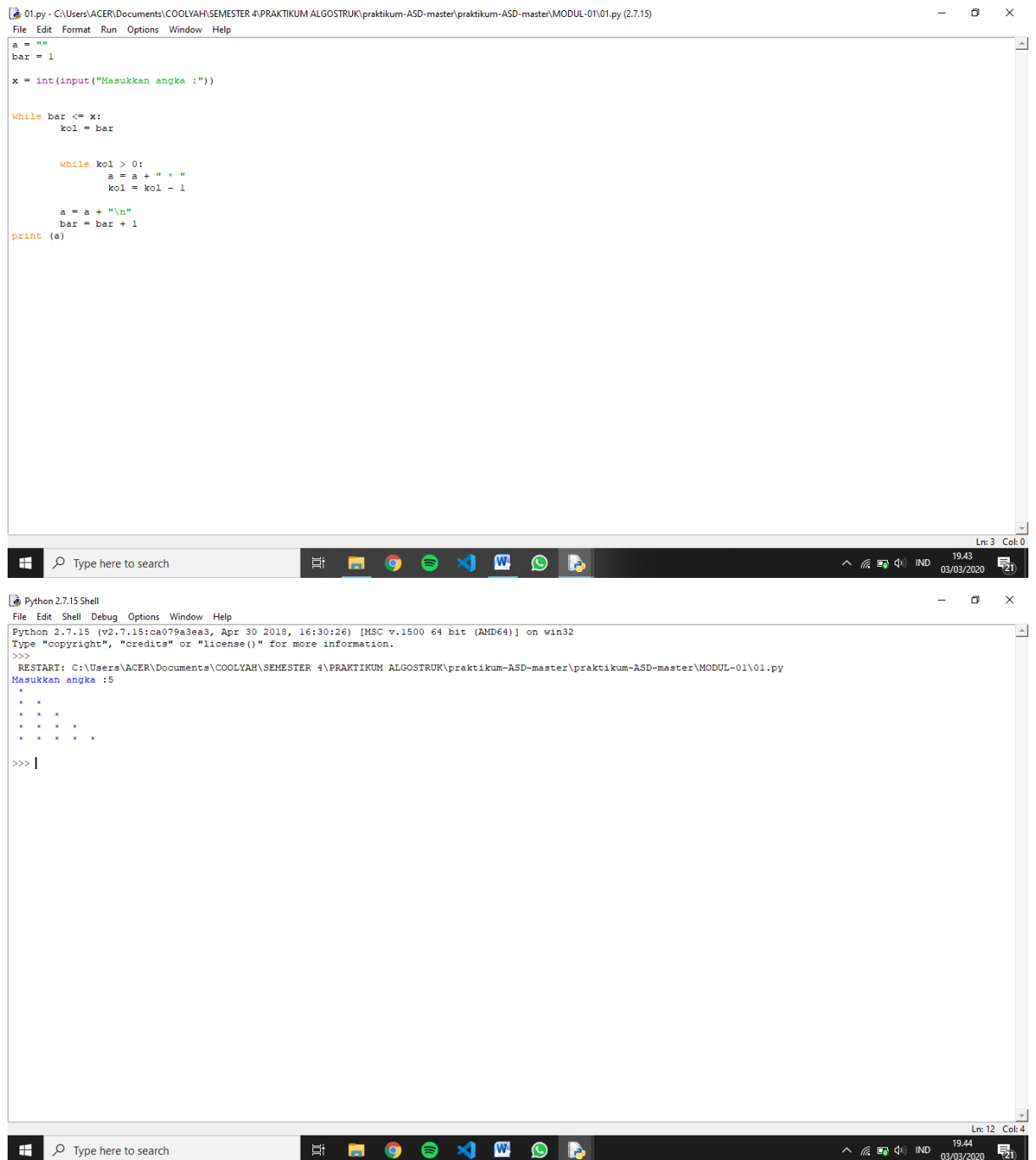
NIM : L200180025

Kelas : B

MODUL 1

1. Fungsi

cetakSiku(x)



```
01.py - C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\01.py (2.7.15)
File Edit Format Run Options Window Help

a = ""
bar = 1

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

while bar <= x:
    kol = bar

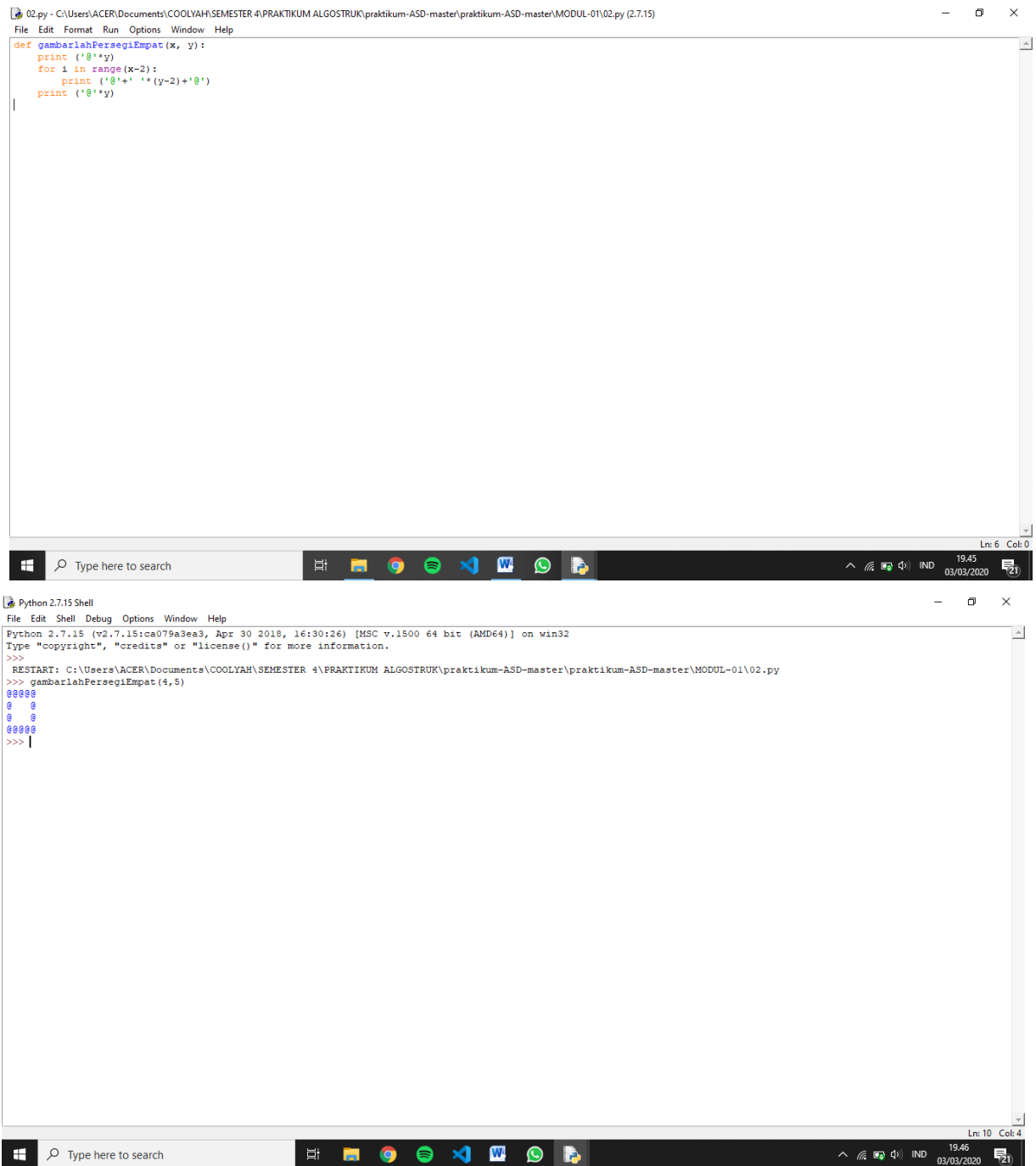
    while kol > 0:
        a = a + " * "
        kol = kol - 1

    a = a + "\n"
    bar = bar + 1
print(a)
```

```
Python 2.7.15 Shell
File Edit Shell Debug Options Window Help
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\01.py
Masukkan angka :5
*
* *
* * *
* * * *
* * * * *

>>> |
```

2. Fungsi yang menerima dua integer positif



The image shows two windows from a Windows operating system. The top window is a Python IDE (likely PyCharm) with the file path `C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\02.py (2.7.15)`. The code in the editor is:

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

The bottom window is a "Python 2.7.15 Shell" with the same file path. It shows the execution of the function `gambarlahPersegiEmpat(4,5)`, which produces the following output:

```
>>> gambarlahPersegiEmpat(4,5)  
@@@@@  
@  @  
@  @  
@@@@@
```

3. Fungsi menerima sebuah string dan hitung huruf konsonan

```
03.py - C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\03.py (2.7.15)
File Edit Format Run Options Window Help

##no 3a
def jumlahHurufVokal(x):
    vokal = "AIUEOaiueo"
    a = len(x)
    b = ""
    for k in x:
        if k in vokal:
            b+=k
    c = len(b)
    return (a,c)

##no 3b
def jumlahHurufKonsonan(x):
    konsonan = "BCDFGHJKLMNPQRSTVWXYZbodfghjklmnpqrstvwxyz"
    a = len(x)
    b = ""
    for k in x:
        if k in konsonan:
            b+=k
    c = len(b)
    return (a,c)
```

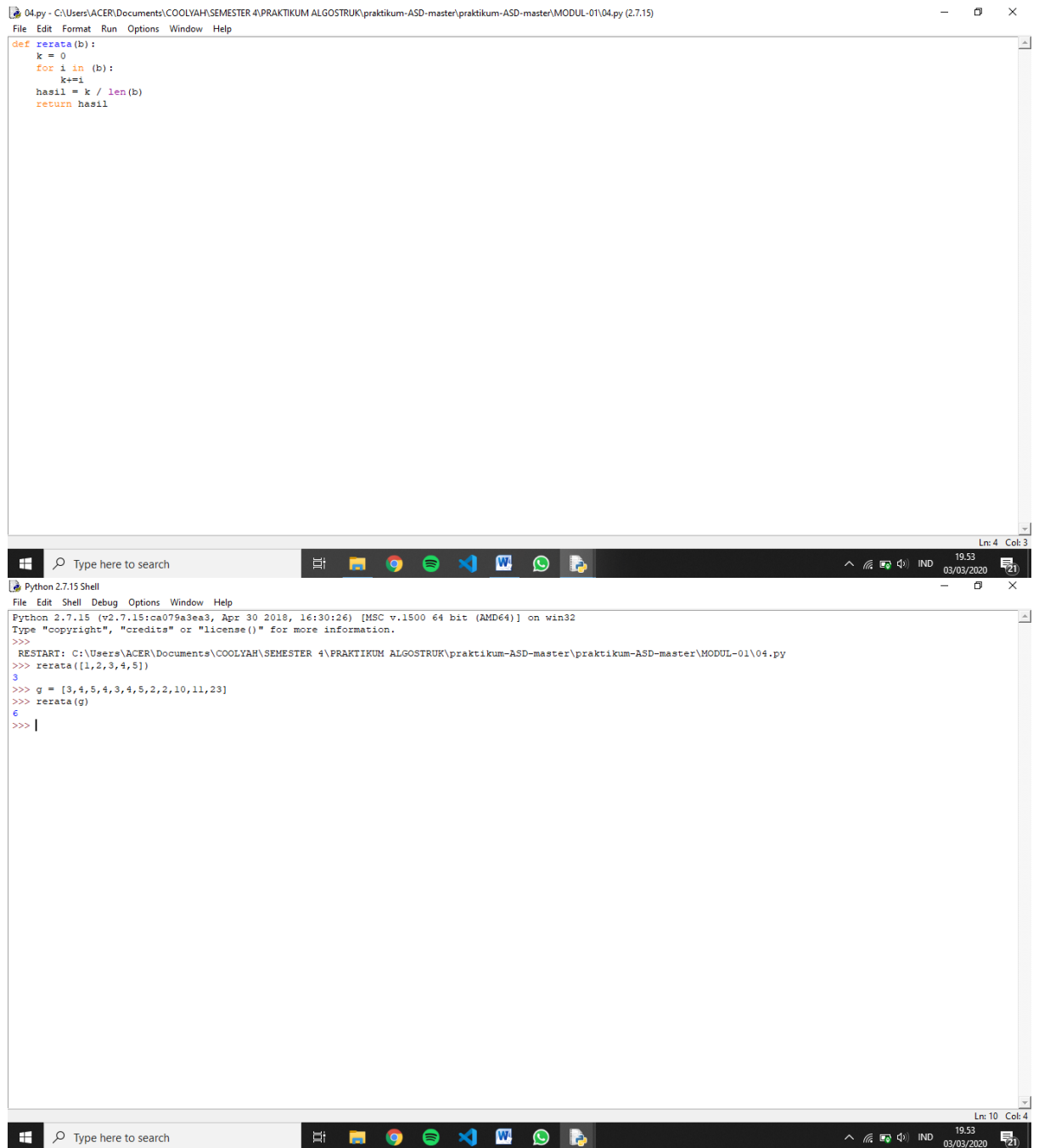


```
Python 2.7.15 Shell
File Edit Shell Debug Options Window Help

Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\03.py
>>> k = jumlahHurufVokal('Rina')
>>> jumlahHurufVokal('Rina')
(4, 2)
>>> jumlahHurufKonsonan('Dafforina')
(9, 5)
>>>
```



4. Fungsi rerata

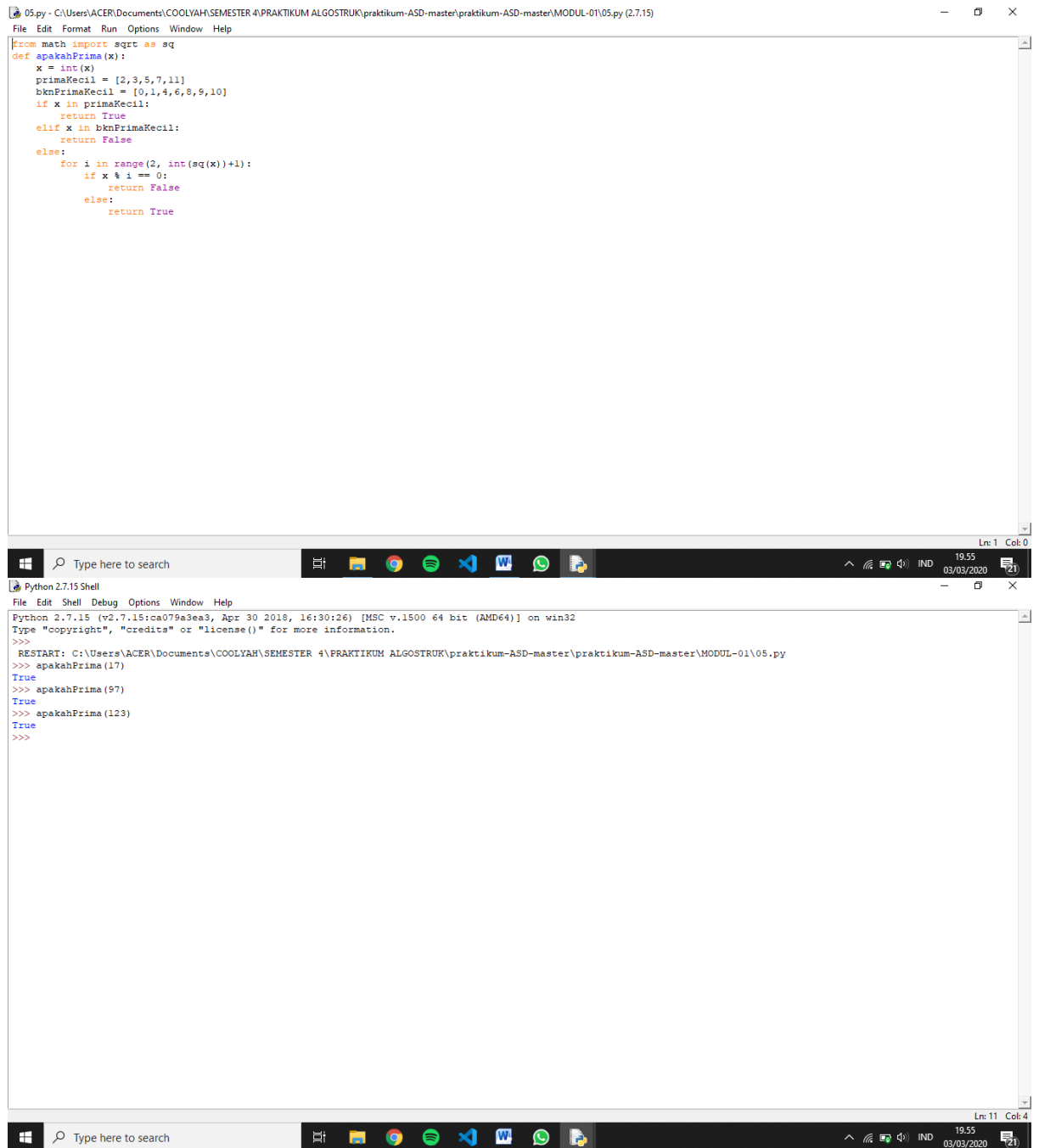


The image shows a screenshot of a Python IDE (likely IDLE) with two windows. The top window is a Python script file named '04.py' located at 'C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\04.py (2.7.15)'. It contains a function definition for 'rerata(b)'. The bottom window is a 'Python 2.7.15 Shell' showing the execution of the function. The shell output shows the function being called with a list of numbers, and the result is printed.

```
def rerata(b):  
    k = 0  
    for i in (b):  
        k+=1  
    hasil = k / len(b)  
    return hasil
```

```
Python 2.7.15 Shell  
File Edit Shell Debug Options Window Help  
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\04.py  
>>> rerata([1,2,3,4,5])  
3  
>>> g = [3,4,5,4,3,4,5,2,2,10,11,23]  
>>> rerata(g)  
6  
>>> |
```

5. Fungsi bil prima



The image shows a screenshot of a Python IDE (likely PyCharm) with a file named `05.py` open. The file contains a function `apakahPrima(x)` that checks if a number `x` is prime. The function uses a list of known primes (`primaKecil = [2, 3, 5, 7, 11]`) and a set of known primes (`bknPrimaKecil = {0, 1, 4, 6, 8, 9, 10}`) for quick checks. If `x` is not in these sets, it checks for divisibility by integers from 2 to $\sqrt{x} + 1$.

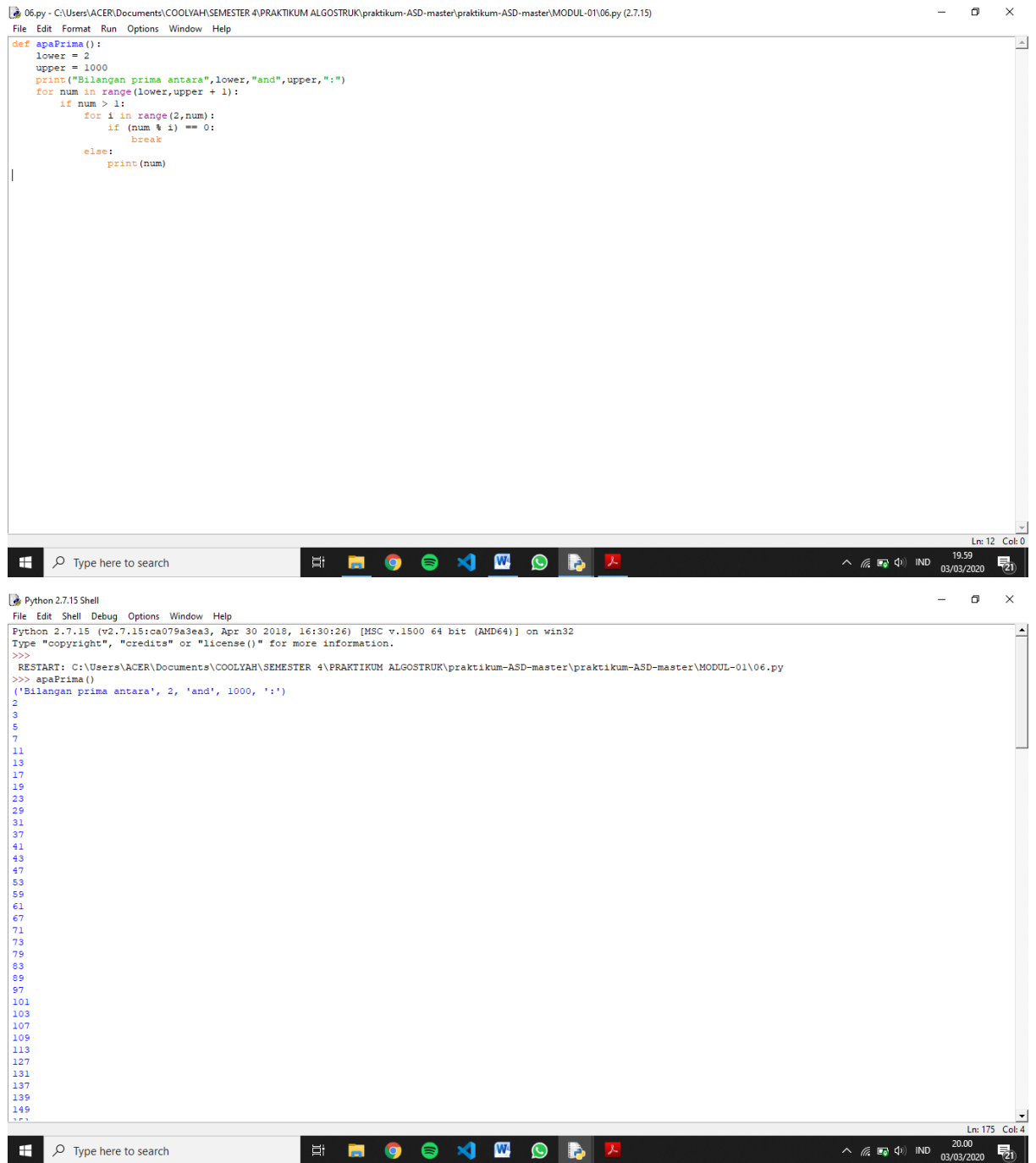
```
from math import sqrt as sq
def apakahPrima(x):
    x = int(x)
    primaKecil = [2, 3, 5, 7, 11]
    bknPrimaKecil = {0, 1, 4, 6, 8, 9, 10}
    if x in primaKecil:
        return True
    elif x in bknPrimaKecil:
        return False
    else:
        for i in range(2, int(sq(x))+1):
           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

Below the editor, a Python 2.7.15 Shell window shows the execution of the function. It restarts the shell and runs the following commands:

```
>>> apakahPrima(17)
True
>>> apakahPrima(97)
True
>>> apakahPrima(123)
True
>>>
```

The Windows taskbar at the bottom shows the date as 03/03/2020 and the time as 19:55.

6. Fungsi bil prima

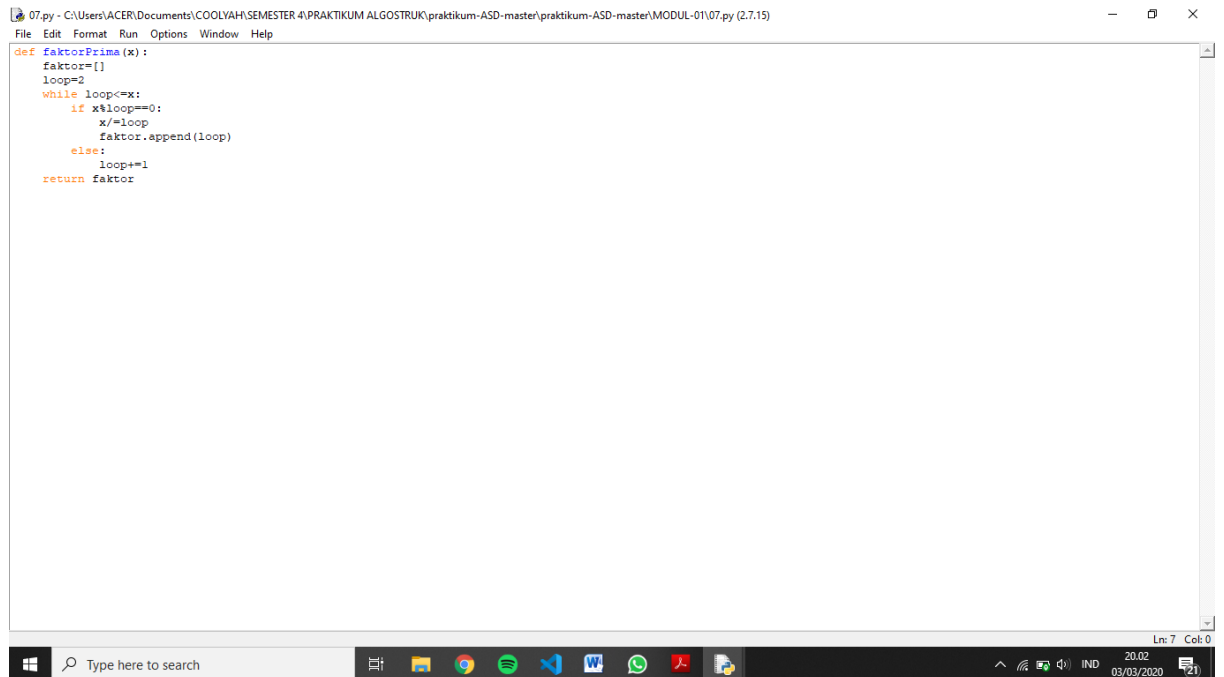


The image shows a Windows desktop with two application windows. The top window is a Python 2.7.15 IDE titled '06.py - C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\06.py (2.7.15)'. It contains a function definition for `apaPrima()` that prints prime numbers between 2 and 1000. The bottom window is a 'Python 2.7.15 Shell' titled 'Python 2.7.15 Shell', showing the execution of the `apaPrima()` function. The output displays the first 175 prime numbers, with the first few being 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, 139, 149, and so on. The taskbar at the bottom shows various icons including File Explorer, Chrome, and communication apps like WhatsApp and Telegram.

```
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)
```

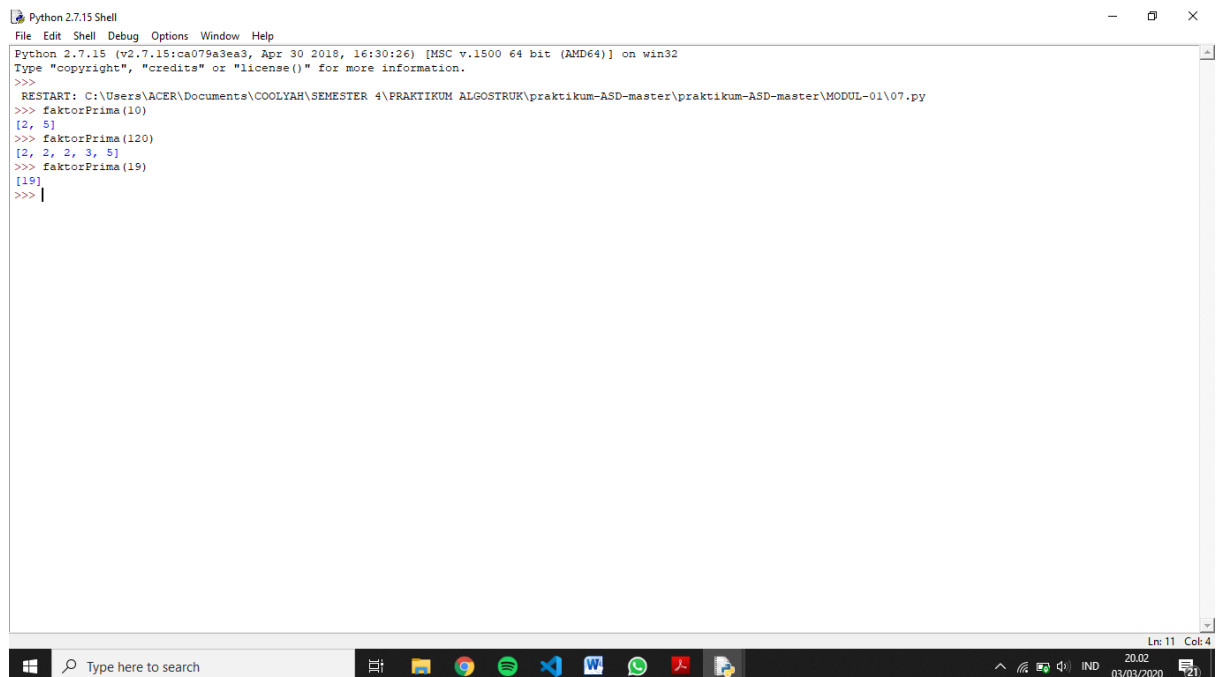
```
Python 2.7.15 Shell
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\06.py
>>> 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
139
149
...
```

7. Bil positif



The screenshot shows a Python IDE window titled "07.py - C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\07.py (2.7.15)". The code defines a function `faktorPrima(x)` that calculates the prime factors of a positive integer `x`. The function initializes an empty list `faktor` and a variable `loop` to 2. It enters a `while` loop that continues as long as `loop` is less than or equal to `x`. Inside the loop, it checks if `x` is divisible by `loop` (`x%loop==0`). If true, it divides `x` by `loop` (`x/=loop`) and appends `loop` to the `faktor` list. If false, it increments `loop` by 1. Finally, it returns the `faktor` list.

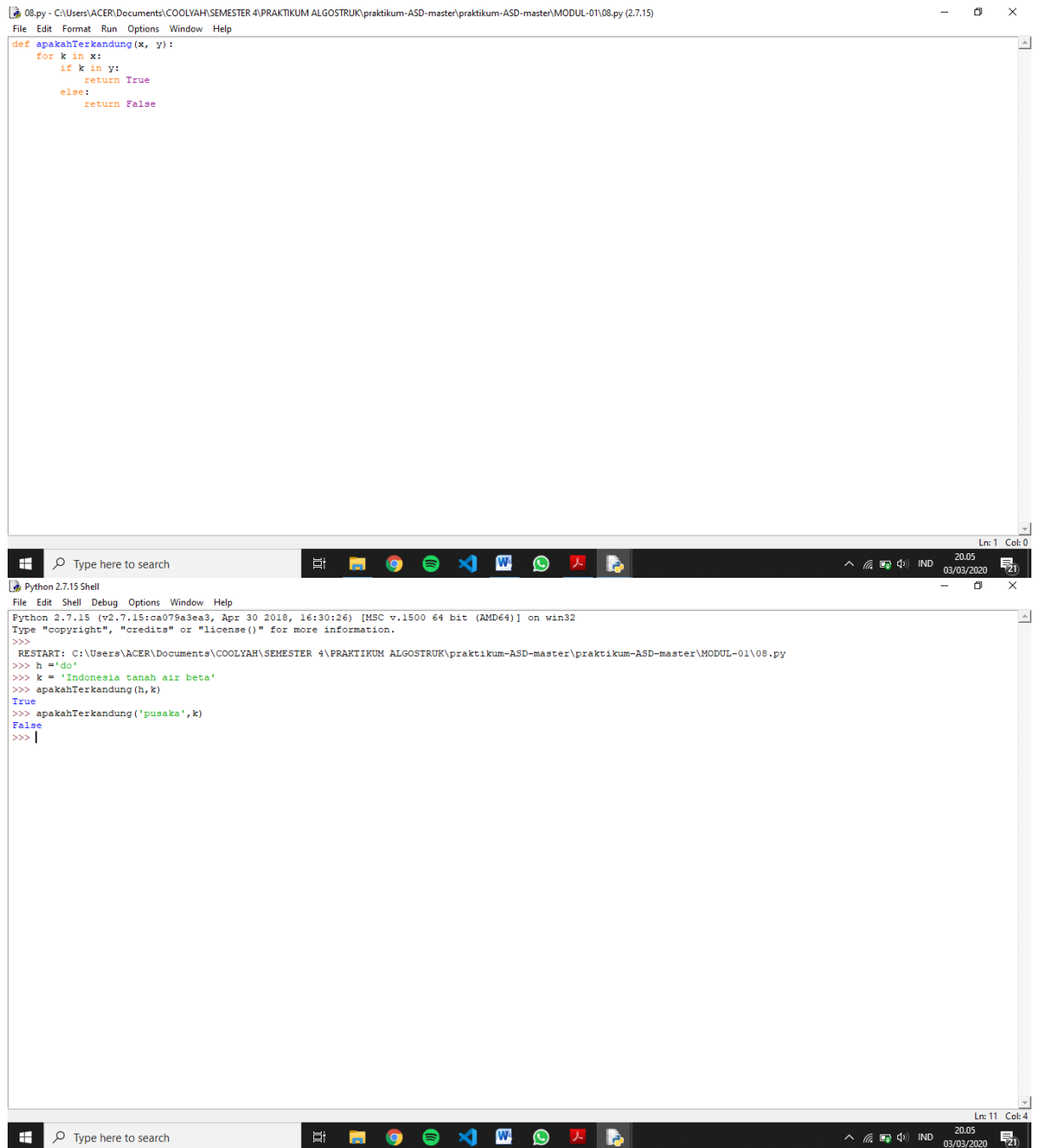
```
def faktorPrima(x):  
    faktor=[]  
    loop=2  
    while loop<=x:  
        if x%loop==0:  
            x/=loop  
            faktor.append(loop)  
        else:  
            loop+=1  
    return faktor
```



The screenshot shows a Python 2.7.15 Shell window. It displays the execution of the `faktorPrima` function for three different inputs: 10, 120, and 19. The output for each call is shown on a new line, indicating the list of prime factors.

```
Python 2.7.15 Shell  
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\07.py  
>>> faktorPrima(10)  
[2, 5]  
>>> faktorPrima(120)  
[2, 2, 2, 3, 5]  
>>> faktorPrima(19)  
[19]  
>>> |
```

8. Fungsi



The screenshot displays a Python IDE window titled "08.py - C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\08.py (2.7.15)". The code editor contains a function definition:

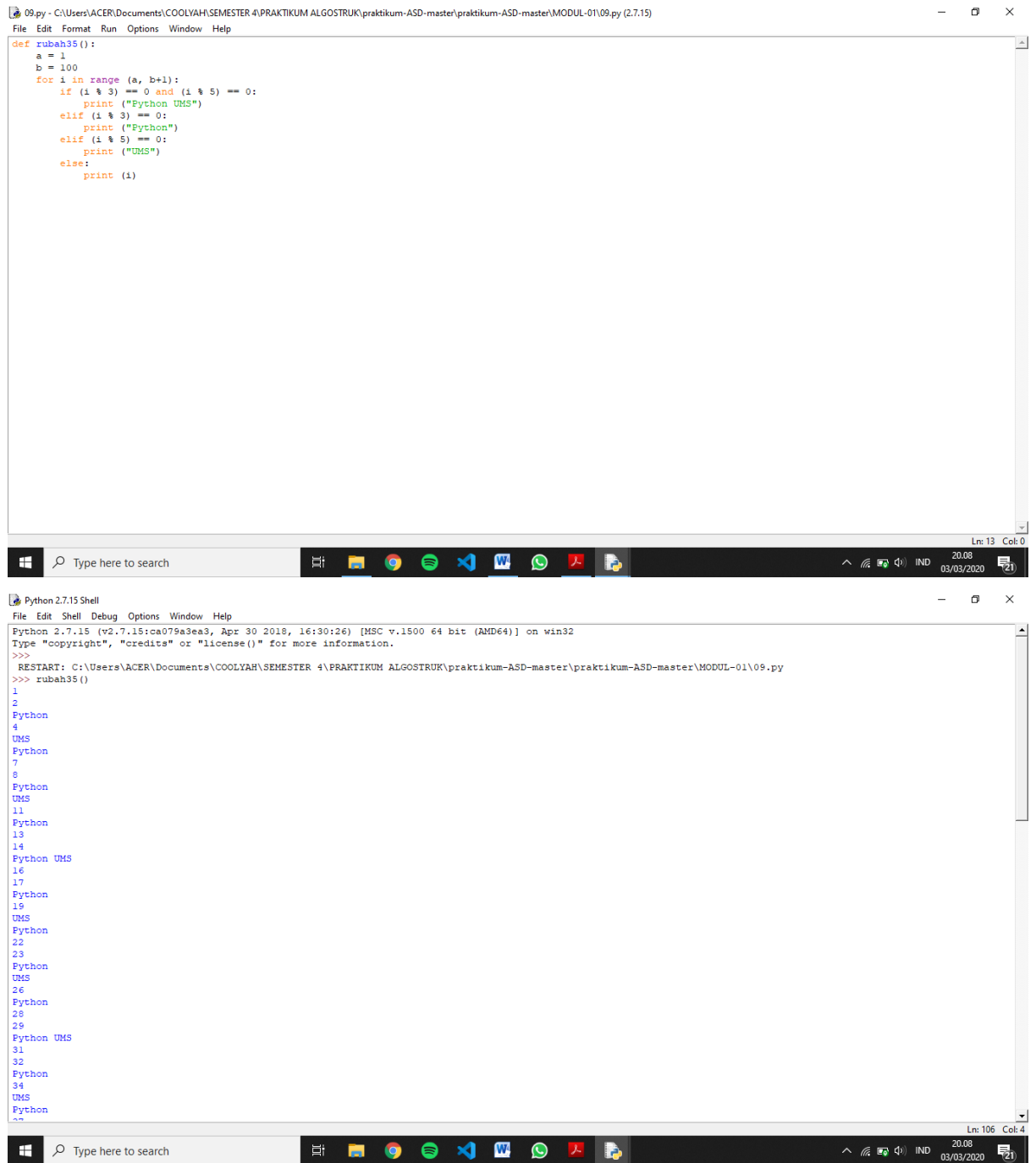
```
def apakahTerkandung(x, y):  
    for k in x:  
        if k in y:  
            return True  
        else:  
            return False
```

Below the code editor is a "Python 2.7.15 Shell" window. It shows the execution of the function with the following commands and output:

```
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\08.py  
>>> h = 'do'  
>>> k = 'Indonesia tanah air beta'  
>>> apakahTerkandung(h, k)  
True  
>>> apakahTerkandung('pusaka', k)  
False  
>>> |
```

The taskbar at the bottom shows the Windows Start button, a search bar, and several application icons including File Explorer, Google Chrome, Spotify, Visual Studio Code, Word, and WhatsApp. The system tray on the right indicates the date and time as 03/03/2020, 20:05.

9. Program

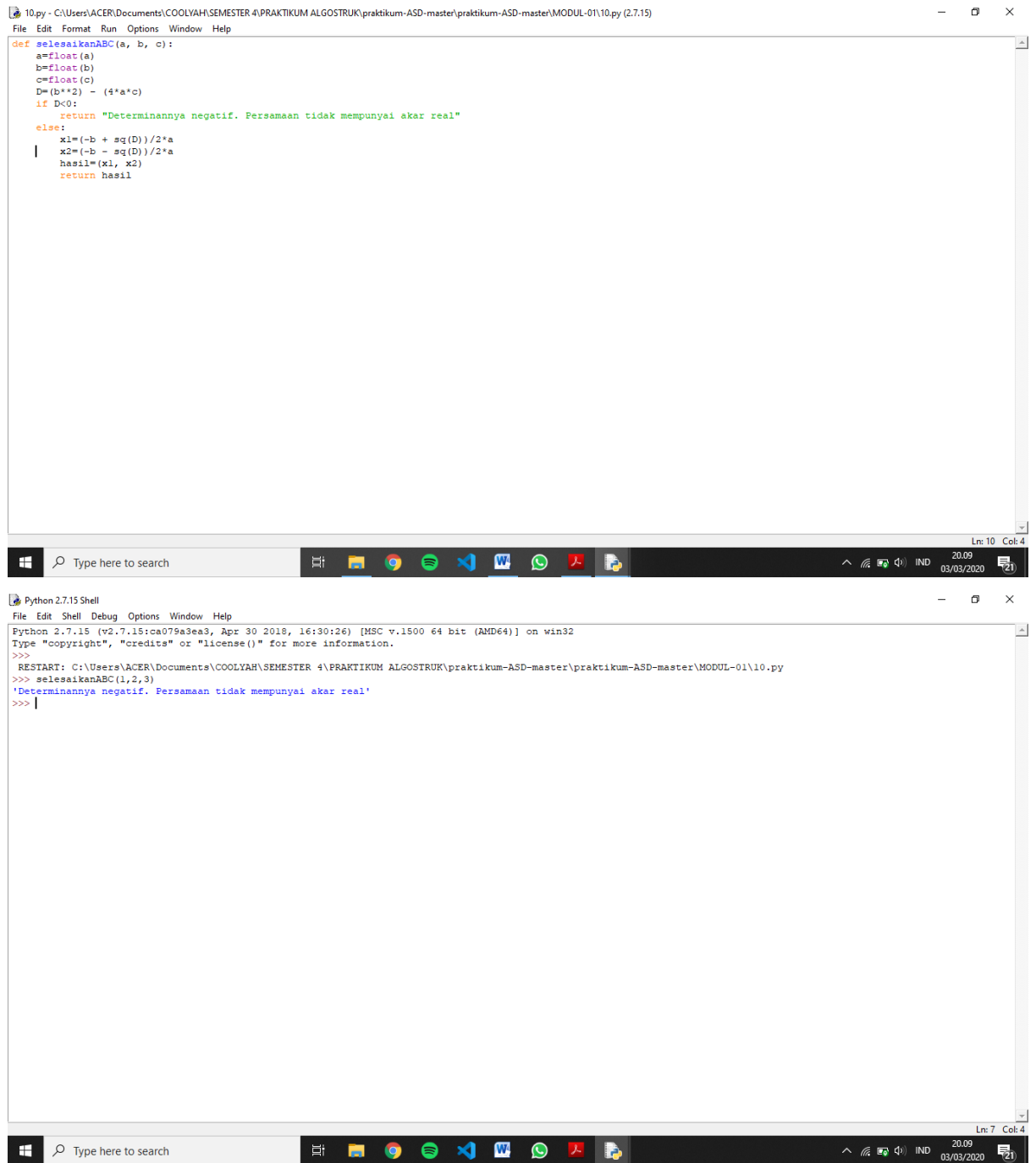


The image shows a screenshot of a Windows desktop with two windows open. The top window is a Python IDE titled '09.py - C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\09.py (2.7.15)'. It contains a Python function named 'rubah35()' that iterates from 1 to 100. For each number 'i', it checks if 'i' is divisible by 3 and 5. If both, it prints 'Python UMS'. If only one, it prints 'Python'. Otherwise, it prints 'UMS'. The bottom window is a 'Python 2.7.15 Shell' showing the execution of the script. It displays the output of the 'rubah35()' function, which is a sequence of 'Python' and 'UMS' strings corresponding to the numbers 1 through 100. The taskbar at the bottom shows various application icons and the system clock indicating 20:08 on 03/03/2020.

```
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 2.7.15 Shell
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\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
>>>
```

10. Modifikasi



```
10.py - C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\10.py (2.7.15)
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

Python 2.7.15 Shell
File Edit Shell Debug Options Window Help
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\10.py
>>> selesaikanABC(1,2,3)
'Determinannya negatif. Persamaan tidak mempunyai akar real'
>>>
```

11. Fungsi

```
11.py - C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum
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

Python 2.7.15 Shell
File Edit Shell Debug Options Window Help
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2016, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\11.py
>>> apakahKabisat(2000)
True
>>> apakahKabisat(2020)
True
>>> |
```

12. Fungsi

The image shows a Windows desktop environment. At the top, a Python script editor window is open, displaying a game logic script. The script generates a random number between 1 and 100 and allows a user to guess it three times. It provides feedback on whether the guess is too high, too low, or correct. The script is saved as 12.py in the directory C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\12.py (2.7.15).

```
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 tebakkan ke-{}:>".format(i+1)))
    if b == a:
        print("Ya. Anda benar.")
    elif b > a:
        if i >= 2:
            print("Itu terlalu besar. Kesempatan habis. Nilainya adalah",a)
        else:
            print("Itu terlalu besar. Coba lagi")
    else:
        if i >= 2:
            print("Itu terlalu kecil. Kesempatan habis. Nilainya adalah",a)
        else:
            print("Itu terlalu kecil. Coba lagi")
```

Below the script editor, a Windows taskbar is visible with various application icons. Below the taskbar, a Python 2.7.15 Shell window is open, showing the execution of the script. The shell output matches the script's logic, showing the user's input and the program's response.

```
Python 2.7.15 Shell
File Edit Shell Debug Options Window Help
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\12.py
Permainan tebak angka.
Saya menyimpan sebuah angka bulat antara 1 sampai 100. Coba Tebak!
Masukkan tebakkan ke-1:>57
Itu terlalu besar. Coba lagi
Masukkan tebakkan ke-2:>30
Itu terlalu kecil. Coba lagi
Masukkan tebakkan ke-3:>46
('Itu terlalu besar. Kesempatan habis. Nilainya adalah', 45)
>>> |
```

13. Fungsi

```
13.py - C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\13.py (2.7.15)
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])

Python 2.7.15 Shell
File Edit Shell Debug Options Window Help

Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\13.py
>>> katakan(99987654)
'sembilan puluh sembilan juta sembilan ratus delapan puluh tujuh ribu enam ratus lima puluh empat'
>>> |
```

14. Fungsi

```
14.py - C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\14.py (2.7.15)
File Edit Format Run Options Window Help

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

```
Python 2.7.15 Shell
File Edit Shell Debug Options Window Help

Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\ACER\Documents\COOLYAH\SEMESTER 4\PRAKTIKUM ALGOSTRUK\praktikum-ASD-master\praktikum-ASD-master\MODUL-01\14.py
>>> formatRupiah(1500)
'Rp 1.500'
>>> formatRupiah(2560000)
'Rp 2.560.000'
>>> |
```