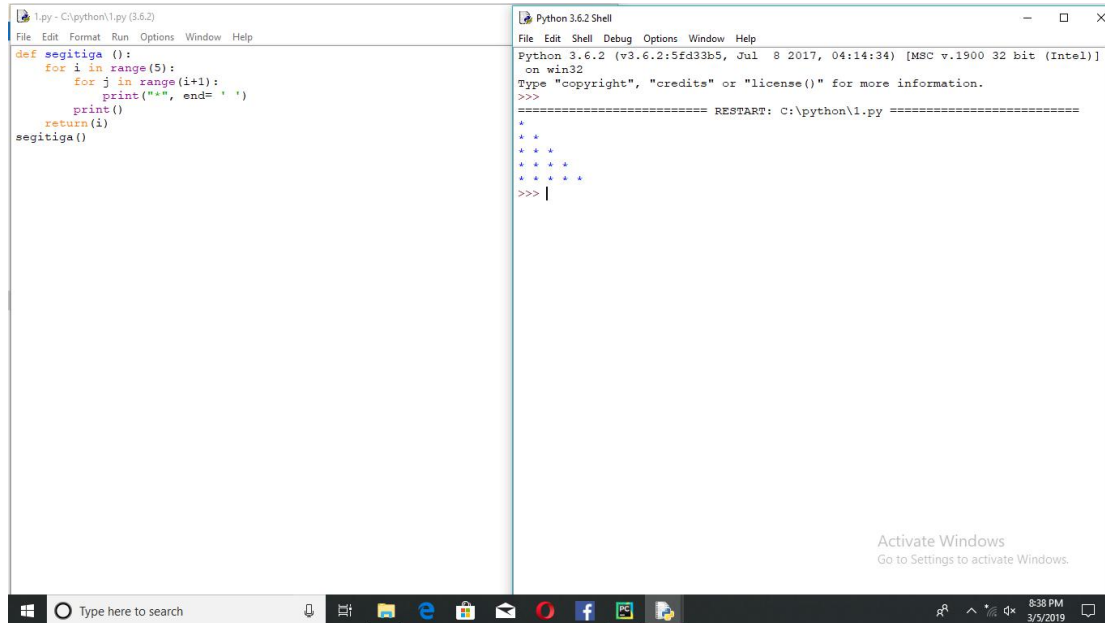


Nama : Rifqi Alwan P
NIM : L200180008
Kelas : A

MODUL 1

1.



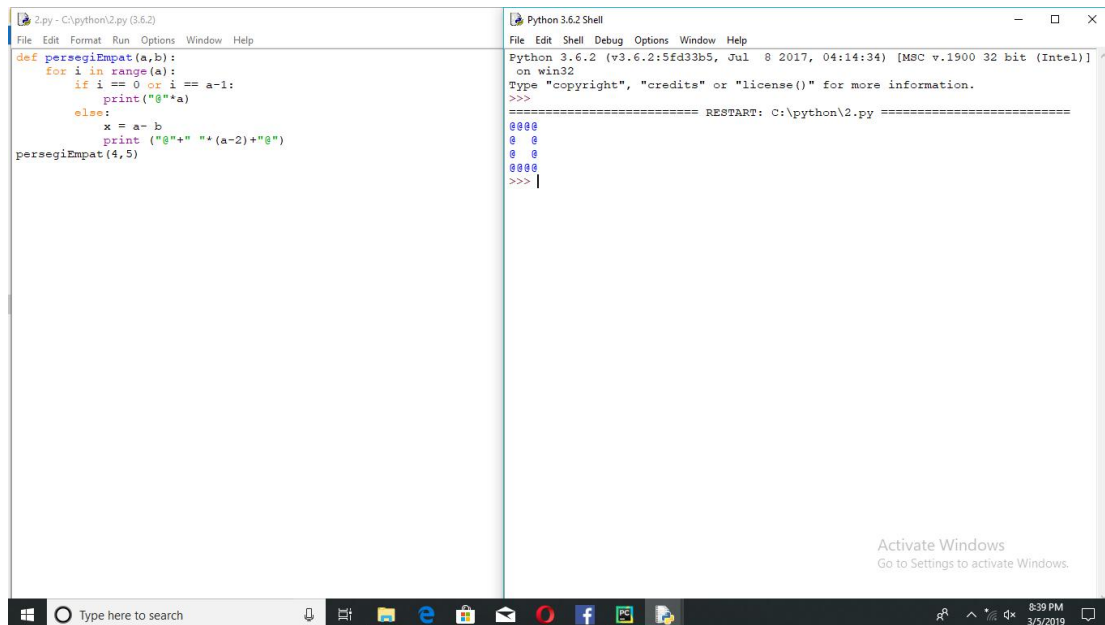
The screenshot shows a Python 3.6.2 IDE with two windows. The left window, titled '1.py - C:\python\1.py (3.6.2)', contains the following code:

```
def segitiga ():  
    for i in range(5):  
        for j in range(i+1):  
            print("+", end= ' ' )  
            print()  
        return(i)  
segitiga()
```

The right window, titled 'Python 3.6.2 Shell', shows the output of the script. It displays the Python version and architecture, followed by a restart command and a series of asterisks forming a triangle:

```
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]  
on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: C:\python\1.py =====  
*  
* *  
* * *  
* * * *  
* * * * *  
>>> |
```

2.



The screenshot shows a Python 3.6.2 IDE with two windows. The left window, titled '2.py - C:\python\2.py (3.6.2)', contains the following code:

```
def persegiEmpat(a,b):  
    for i in range(a):  
        if i == 0 or i == a-1:  
            print("#"*a)  
        else:  
            x = a- b  
            print ("#"+" " *(a-2)+"#")  
persegiEmpat(4,5)
```

The right window, titled 'Python 3.6.2 Shell', shows the output of the script. It displays the Python version and architecture, followed by a restart command and a series of hash symbols forming a rectangle:

```
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]  
on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: C:\python\2.py =====  
####  
# #  
# #  
####  
>>> |
```

3.

The screenshot shows a Python 3.6.2 IDE with a script that defines two functions: `jumlahHurufVokal` and `jumlahHurufKonsonan`. The `jumlahHurufVokal` function counts the number of vowels in a string, and the `jumlahHurufKonsonan` function counts the number of consonants. The script then calls these functions with the input "Surakarta" and prints the results.

```
def jumlahHurufVokal(input):  
    total = 0  
    voc = ["a", "i", "u", "e", "o"]  
    for i in input:  
        if i in voc:  
            total+=1  
    return [len(input), total]  
  
def jumlahHurufKonsonan(input):  
    total = 0  
    voc = ["a", "i", "u", "e", "o"]  
    for i in input:  
        if i in voc:  
            total+=1  
    return [len(input), len(input)-total]  
  
v = jumlahHurufVokal("Surakarta")  
k = jumlahHurufKonsonan("Surakarta")  
  
print(v)  
print(k)
```

The Python 3.6.2 Shell window shows the output of the script:

```
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]  
on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: C:\python\3ab.py =====  
>>> [9, 4]  
[9, 5]  
>>>
```

4.

The screenshot shows a Python 3.6.2 IDE with a script that defines a function `rerata` to calculate the average of a list of numbers. The script then calls this function with the input [1, 2, 3, 4, 5] and prints the result.

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

The Python 3.6.2 Shell window shows the output of the script:

```
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]  
on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: C:\python\4.py =====  
>>> rerata([1,2,3,4,5])  
3.0  
>>>
```

5.

The screenshot shows a Python IDE window titled '5.py - C:\python\5.py (3.6.2)' and a 'Python 3.6.2 Shell' window. The IDE contains a function `apakahPrima(n)` that checks if a number is prime. It uses a list `primaKecil` for small primes and a loop for larger numbers. The shell shows the function being called for 17, 97, and 123, all returning `True`.

```

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

print(apakahPrima(17))
print(apakahPrima(97))
print(apakahPrima(123))

```

```

Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\python\5.py =====
False
False
False
>>>

```

6.

The screenshot shows a Python IDE window titled '6.py - C:\python\6.py (3.6.2)' and a 'Python 3.6.2 Shell' window. The IDE contains a function `apakahPrima(n)` similar to the one in the previous screenshot. The shell shows the function being called for numbers from 962 to 1000, returning `False` for most and `True` for some.

```

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)))

```

```

Python 3.6.2 Shell
962 False
963 False
964 False
965 False
966 False
967 True
968 False
969 False
970 False
971 True
972 False
973 False
974 False
975 False
976 False
977 True
978 False
979 False
980 False
981 False
982 False
983 True
984 False
985 False
986 False
987 False
988 False
989 False
990 False
991 True
992 False
993 False
994 False
995 False
996 False
997 True
998 False
999 False
1000 False

```

7.

The screenshot shows a Windows 10 desktop environment. On the left is a Python IDE window titled '7.py - C:\python\7.py (3.6.2)'. It contains the following code:

```
def faktorPrima(x) :
    a = []
    b = []
    hasil = 0
    bil = x
    prima = True
    for i in range(2,x):
        prima = True
        for u in range(2, i) :
            if i % u == 0 :
                prima = False
        if prima :
            a.append(i)
    idx = 0
    while bil > 1 :
        try:
            if (bil%a[idx]) == 0 :
                hasil = bil/a[idx]
                bil = hasil
                b.append(a[idx])
            else :
                idx = idx + 1
        except IndexError :
            break
    print (b)
```

On the right is a 'Python 3.6.2 Shell' window. It displays the output of running the script:

```
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\python\7.py =====
>>> faktorPrima(10)
[2, 5]
>>> faktorPrima(120)
[2, 2, 2, 3, 5]
>>> faktorPrima(19)
[1]
>>> |
```

The taskbar at the bottom shows the search bar and various application icons. A watermark 'Activate Windows' is visible in the bottom right corner.

8.

The screenshot shows a Windows 10 desktop environment. On the left is a Python IDE window titled '8.py - C:/python/8.py (3.6.2)'. It contains the following code:

```
def apakahTerkandung(a,b):
    return a in b

h = "do"
k = "Indonesia tanah air beta"
print (apakahTerkandung(h, k))
print (apakahTerkandung("pusaka", k))
```

On the right is a 'Python 3.6.2 Shell' window. It displays the output of running the script:

```
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/python/8.py =====
True
False
>>> |
```

The taskbar at the bottom shows the search bar and various application icons. A watermark 'Activate Windows' is visible in the bottom right corner.

9.

The screenshot shows a Python IDE with a file named `9.py` and a Python 3.6.2 Shell. The code in `9.py` is as follows:

```
for i in range(1,100):
    if (i % 3) == 0 and (i % 5) == 0 :
        i = "Python UMS"
    elif (i % 3) == 0:
        i = "Python"
    elif (i % 5) == 0:
        i = "UMS"
    print(i)
```

The Python 3.6.2 Shell shows the output of the program, which is a sequence of numbers from 1 to 99, with the words "Python" and "UMS" replacing numbers that are divisible by 3 or 5. The output is as follows:

```
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
```

10.

The screenshot shows a Python IDE with a file named `10.py` and a Python 3.6.2 Shell. The code in `10.py` is as follows:

```
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):
        print("Determinan negatif. Persamaan tidak mempunyai akar")
    else:
        x1 = (-b + akar(D))/(2*a)
        x2 = (-b - akar(D))/(2*a)
        hasil = (x1,x2)
        return hasil
```

The Python 3.6.2 Shell shows the output of the program, which is the result of calling the `selesaikanABC` function with arguments `(1,2,3)`. The output is as follows:

```
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
On win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\python\10.py =====
>>> selesaikanABC(1,2,3)
Determinan negatif. Persamaan tidak mempunyai akar real.
>>> |
```

11.

The screenshot shows a Python 3.6.2 IDE with two windows. The left window, titled '11.py - C:\python\11.py (3.6.2)', contains the following code:

```
def apakahKabisat(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(apakahKabisat(1896))
print(apakahKabisat(1897))
print(apakahKabisat(1900))
print(apakahKabisat(2000))
print(apakahKabisat(2004))
print(apakahKabisat(2100))
print(apakahKabisat(2400))
```

The right window, titled 'Python 3.6.2 Shell', shows the output of the script:

```
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\python\11.py =====
True
False
False
True
True
False
True
>>> |
```

12.

The screenshot shows a Python 3.6.2 IDE with two windows. The left window, titled '12.py - C:\python\12.py (3.6.2)', contains the following code:

```
import random

r = random.randint(1,100)
a = """Permainan tebak angka.
Saya menyimpan sebuah angka bulat antara 1 sampai 100. Coba Tebak.

print(a)

b = "Masukkan tebakan ke-"
f = "> "
c = 1
d = str(c)

for i in range(1,100):
    e = (b+d+f)
    a = int(input(e))
    c+=1
    d = str(c)
    if(a < r):
        print("Itu terlalu kecil. Coba lagi.")
    elif(a > r):
        print("Itu terlalu besar. Coba lagi.")
    elif(a == r):
        print("Ya. Anda benar")
        break
```

The right window, titled 'Python 3.6.2 Shell', shows the output of the script:

```
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\python\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.
Masukkan tebakan ke-4:> 10
Itu terlalu kecil. Coba lagi.
Masukkan tebakan ke-5:> 20
Itu terlalu kecil. Coba lagi.
Masukkan tebakan ke-6:> 25
Itu terlalu kecil. Coba lagi.
Masukkan tebakan ke-7:> 35
Itu terlalu besar. Coba lagi.
Masukkan tebakan ke-8:> 30
Itu terlalu kecil. Coba lagi.
Masukkan tebakan ke-9:> 31
Itu terlalu kecil. Coba lagi.
Masukkan tebakan ke-10:> 32
Itu terlalu kecil. Coba lagi.
Masukkan tebakan ke-11:> 33
Itu terlalu kecil. Coba lagi.
Masukkan tebakan ke-12:> |
```

13.

The screenshot shows a Windows desktop with two windows. The left window is a text editor titled '13.py - C:\python\13.py (3.6.2)' containing a Python script. The script defines a function `katakan` that takes a number `n` and returns its Indonesian word representation. It uses a list `angka` for digits 1-10 and a series of `if` statements for larger numbers. The main part of the script sets `a = 1` and enters a `while` loop that prompts the user to input a number, then prints the result from the `katakan` function. The right window is a 'Python 3.6.2 Shell' showing the execution of the script. It displays the prompt `>>>`, the function definition, and the execution of `RESTART: C:\python\13.py`. The output shows the program asking for input and displaying the word representation for 20000, 1550000, and 1.

```
def katakan(bil):
    angka = ["", "Satu ", "Dua ", "Tiga ", "Empat ", "Lima ", "Enam ",
             "Tujuh ", "Delapan ", "Sembilan ", "Sepuluh ", "Sebelas "]
    n = int(bil)
    if n >= 0 and n <= 11:
        hasil = angka[n]
    elif n < 20:
        hasil = katakan(n-10) + " Belas "
    elif n < 100:
        hasil = katakan(n/10) + " Puluh " + katakan(n%10)
    elif n < 200:
        hasil = ""
    elif n < 1000:
        hasil = " Seratus " + katakan(n-100)
    elif n < 10000:
        hasil = katakan(n/100) + " Ratus " + katakan(n%100)
    elif n < 2000:
        hasil = " Seribu " + katakan(n-1000)
    elif n < 1000000:
        hasil = katakan(n/1000) + " Ribu " + katakan(n%1000)
    elif n < 1000000000:
        hasil = katakan(n/1000000) + " Juta " + katakan(n%1000000)
    elif n > 1000000000:
        hasil = "Maaf, program tidak membaca angka lebih dari Satu
    return hasil

a = 1
while a != 0:
    a = input(' Masukkan angka dari 1 sd 1.000.000.000: ')
    huruf = katakan(a)
    print(huruf + ' Rupiah')

Masukkan angka dari 1 sd 1.000.000.000: 20000
Dua Puluh Ribu Rupiah
Masukkan angka dari 1 sd 1.000.000.000: 1550000
Satu Juta Lima Ratus Lima Puluh Ribu Rupiah
Masukkan angka dari 1 sd 1.000.000.000: 1
```

14.

The screenshot shows a Windows desktop with two windows. The left window is a text editor titled '14.py - C:\python\14.py (3.6.2)' containing a Python script. The script defines a function `formatRupiah` that takes a number `n` and returns its Indonesian word representation. It uses a list `angka` for digits 1-10 and a series of `if` statements for larger numbers. The main part of the script sets `a = 1` and enters a `while` loop that prompts the user to input a number, then prints the result from the `formatRupiah` function. The right window is a 'Python 3.6.2 Shell' showing the execution of the script. It displays the prompt `>>>`, the function definition, and the execution of `RESTART: C:\python\14.py`. The output shows the program asking for input and displaying the word representation for 1500, 2560000, and 2560000.

```
def formatRupiah(n):
    y = str(n)
    if len(y) <= 3:
        return 'Rp ' + y
    else:
        p = y[-3:]
        q = y[:-3]
        return (formatRupiah(q) + ',' + p)
    print ('Rp' + (formatRupiah(q)) + '.' + p)

a = 1
while a != 0:
    a = input(' Masukkan angka dari 1 sd 1.000.000.000: ')
    huruf = formatRupiah(a)
    print(huruf + ' Rupiah')

formatRupiah(1500)
'Rp 1.500'
formatRupiah(2560000)
'Rp 2.560.000'
formatRupiah(2560000)
'Rp 2.560.000'
```