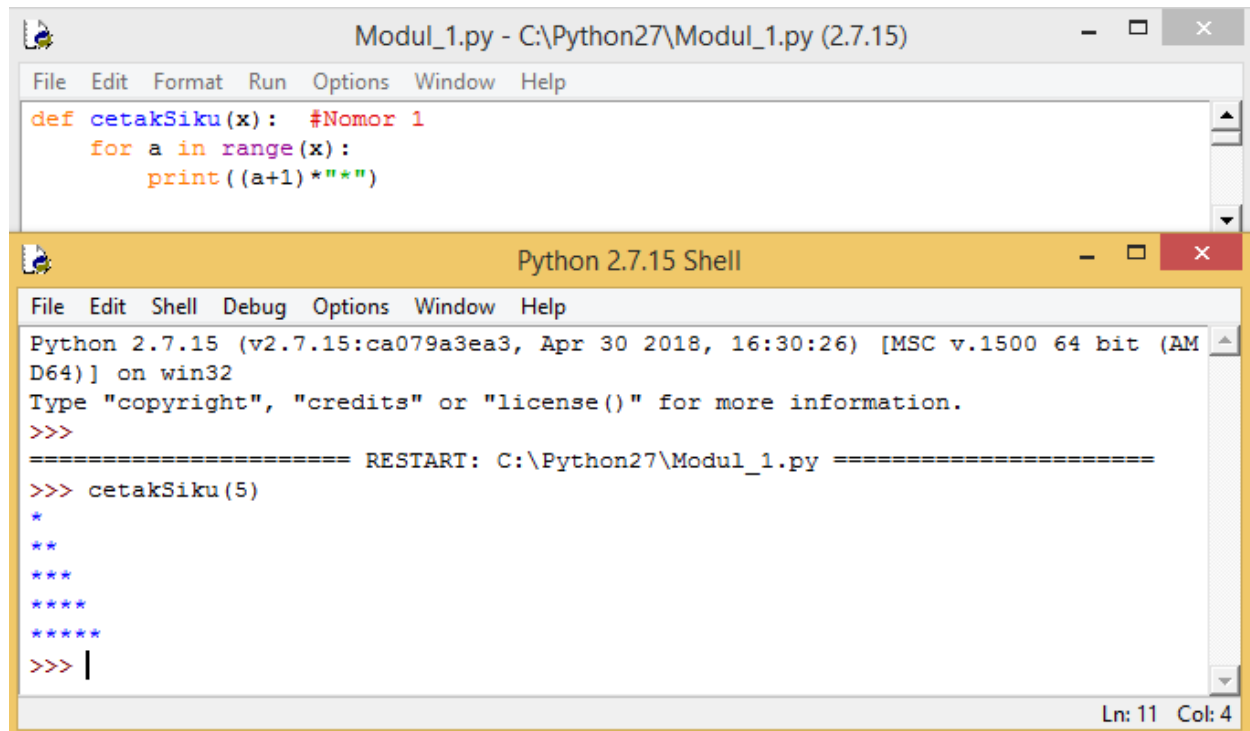


Nama : Anggit Astriani
NIM : L200180111

MODUL 1

Nomor 1



The image shows a screenshot of a Python IDE with two windows. The top window, titled 'Modul_1.py - C:\Python27\Modul_1.py (2.7.15)', contains the following Python code:

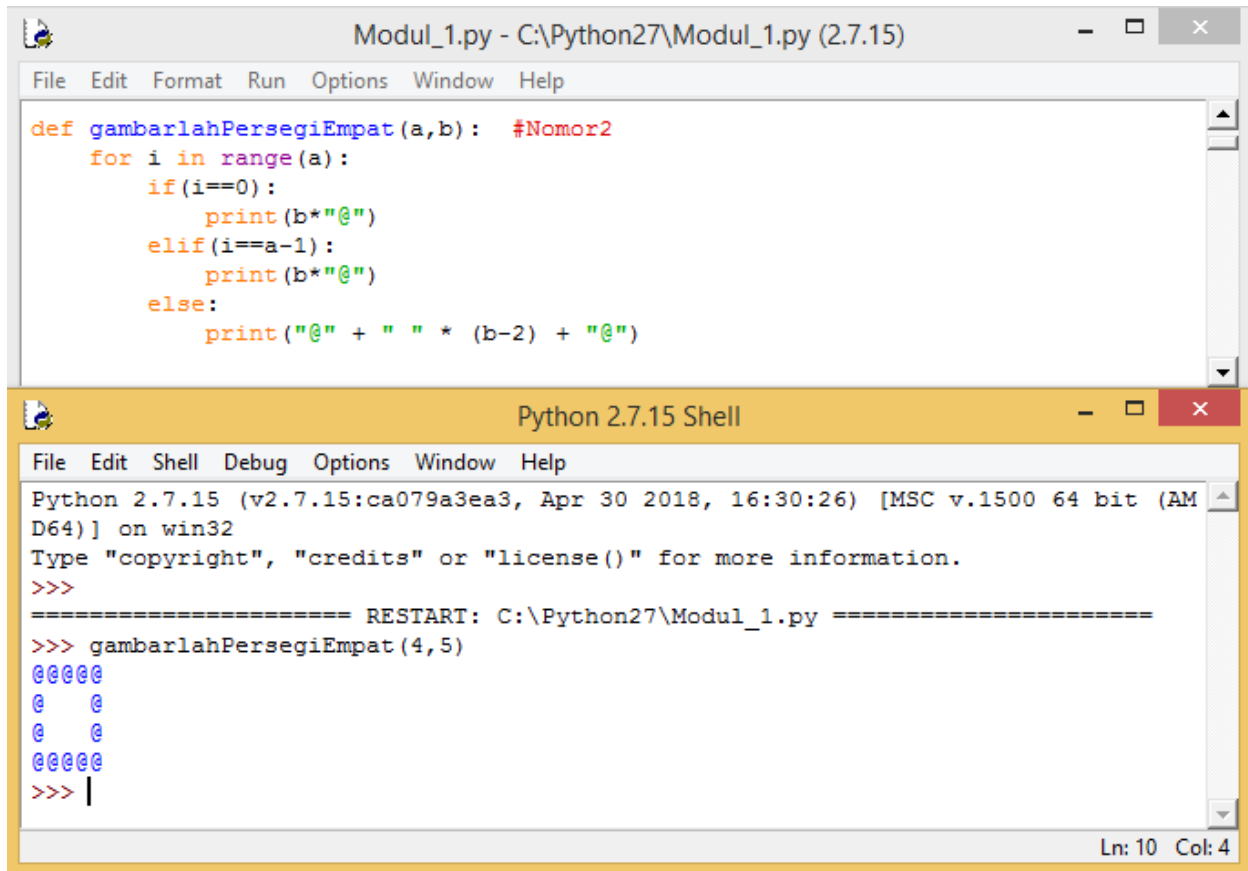
```
def cetakSiku(x): #Nomor 1
    for a in range(x):
        print((a+1)*"*")
```

The bottom window, titled 'Python 2.7.15 Shell', shows the execution of the script. It displays the Python version and architecture, followed by a restart message. The output of the script for `cetakSiku(5)` is shown as five lines of asterisks:

```
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:\Python27\Modul_1.py =====
>>> cetakSiku(5)
*
**
***
****
*****
>>> |
```

The status bar at the bottom right of the shell window indicates 'Ln: 11 Col: 4'.

Nomor 2



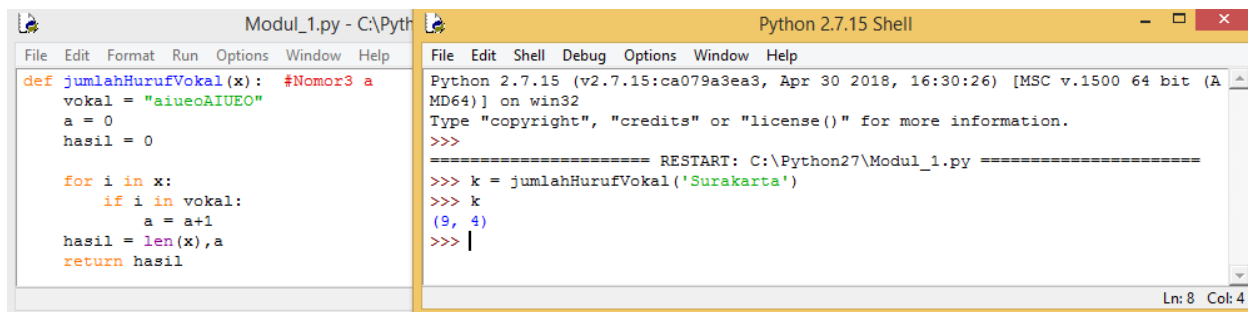
The screenshot shows two windows from a Python IDE. The top window, titled 'Modul_1.py - C:\Python27\Modul_1.py (2.7.15)', contains a Python function named `gambarlahPersegiEmpat(a,b)`. The function uses a `for` loop to iterate from 0 to `a-1`. Inside the loop, it checks if `i` is 0 or `a-1` to print the first and last columns of a square, and otherwise prints the middle columns. The bottom window, titled 'Python 2.7.15 Shell', shows the execution of the script. It displays the Python version and architecture, followed by a restart message. The function is then called with `gambarlahPersegiEmpat(4,5)`, which outputs a 4x5 grid of characters: the first and last columns are filled with '@' and the middle three columns are filled with spaces.

```
def gambarlahPersegiEmpat(a,b): #Nomor2
    for i in range(a):
        if(i==0):
            print(b*"@")
        elif(i==a-1):
            print(b*"@")
        else:
            print("@ " + " " * (b-2) + "@")
```

```
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:\Python27\Modul_1.py =====
>>> gambarlahPersegiEmpat(4,5)
@@@@@
@   @
@   @
@   @
@@@@@
>>> |
```

Ln: 10 Col: 4

Nomor 3a



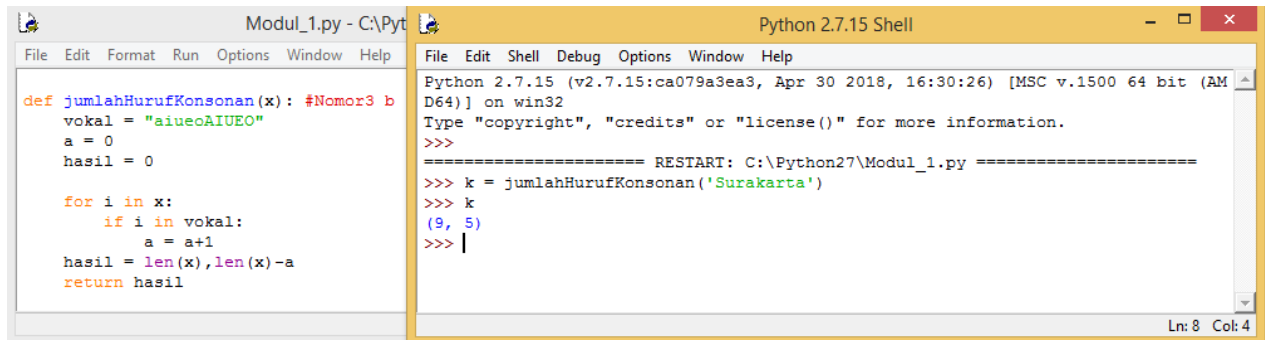
The screenshot shows two windows from a Python IDE. The top window, titled 'Modul_1.py - C:\Python27\Modul_1.py (2.7.15)', contains a Python function named `jumlahHurufVokal(x)`. The function initializes a string `vokal` with the vowels 'a', 'i', 'u', 'e', 'o', 'A', 'I', 'U', 'E', 'O', sets `a` to 0, and then iterates through each character in `x`. If a character is a vowel, it increments `a`. Finally, it returns a tuple containing the length of `x` and the count `a`. The bottom window, titled 'Python 2.7.15 Shell', shows the execution of the script. It displays the Python version and architecture, followed by a restart message. The function is then called with `jumlahHurufVokal('Surakarta')`, which outputs the tuple `(9, 4)`, representing the total length of the string and the number of vowels.

```
def jumlahHurufVokal(x): #Nomor3 a
    vokal = "aiueoAIUEO"
    a = 0
    hasil = 0
    for i in x:
        if i in vokal:
            a = a+1
    hasil = len(x),a
    return hasil
```

```
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:\Python27\Modul_1.py =====
>>> k = jumlahHurufVokal('Surakarta')
>>> k
(9, 4)
>>> |
```

Ln: 8 Col: 4

Nomor 3b



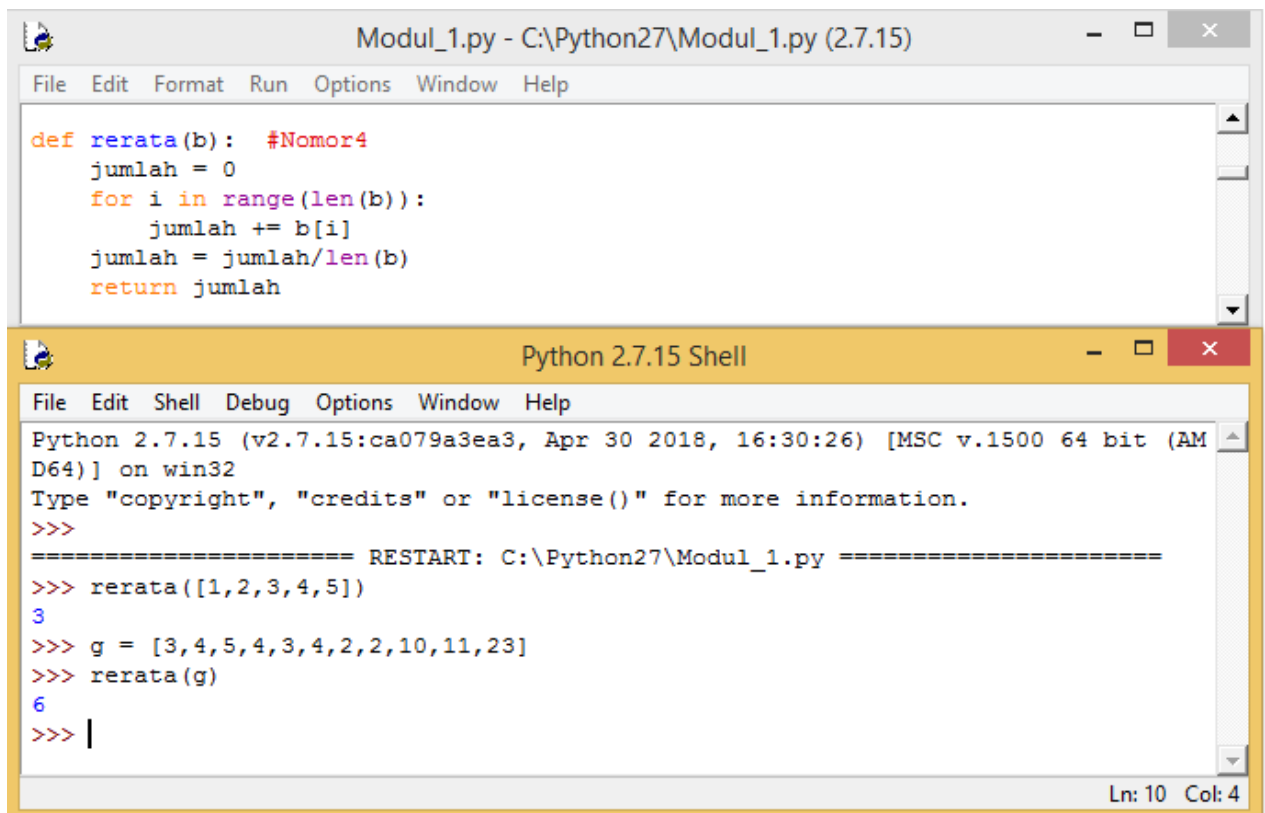
The image shows a Python IDE window titled 'Modul_1.py - C:\Python27\Modul_1.py' and a 'Python 2.7.15 Shell' window. The script in the IDE defines a function `jumlahHurufKonsonan(x)` that counts consonants in a string. The shell shows the function being called with the string 'Surakarta', resulting in the output (9, 5).

```
def jumlahHurufKonsonan(x): #Nomor3 b
    vokal = "aiueoAIUEO"
    a = 0
    hasil = 0

    for i in x:
        if i in vokal:
            a = a+1
    hasil = len(x),len(x)-a
    return hasil

>>> k = jumlahHurufKonsonan('Surakarta')
>>> k
(9, 5)
>>> |
```

Nomor 4

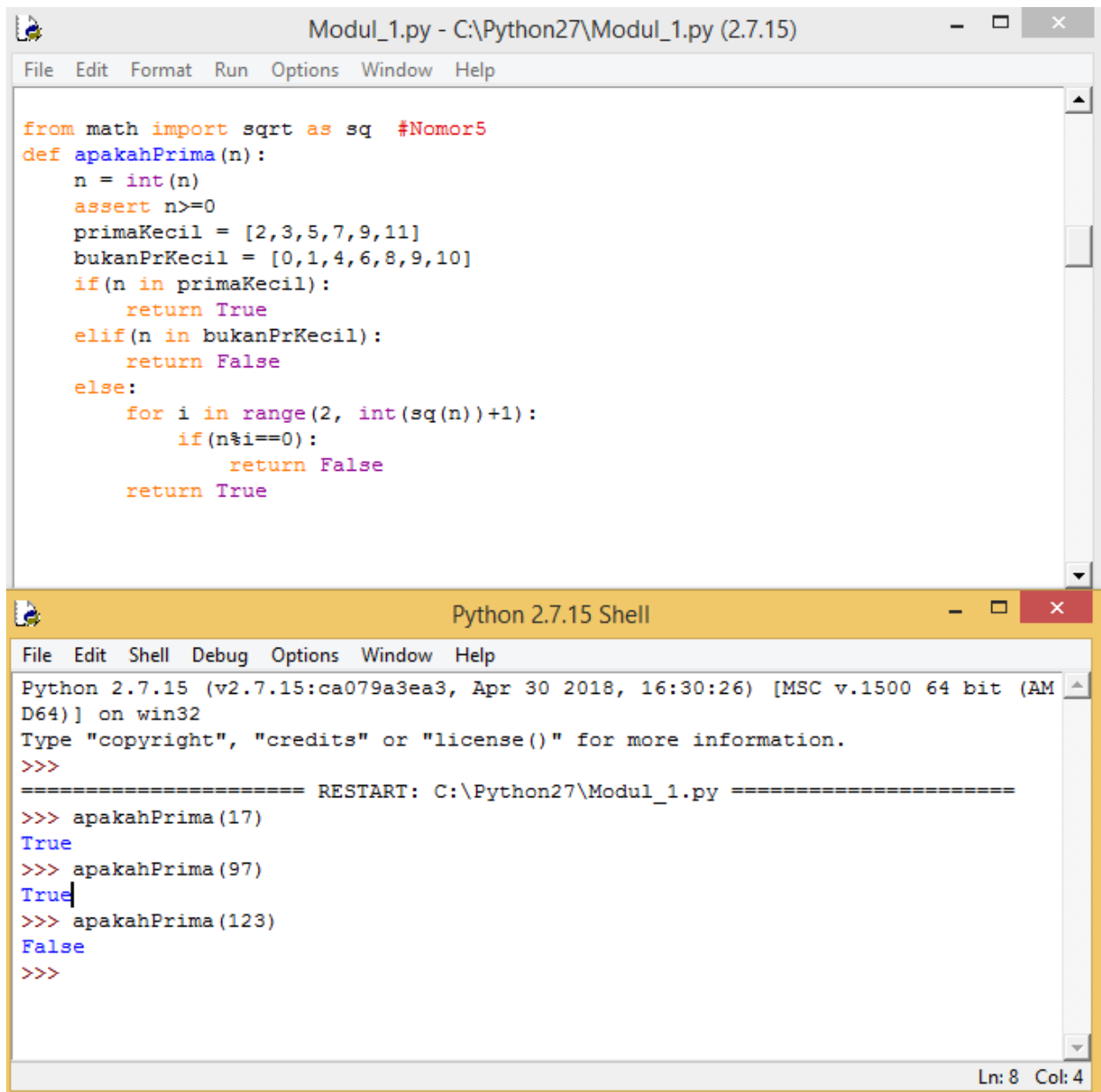


The image shows a Python IDE window titled 'Modul_1.py - C:\Python27\Modul_1.py (2.7.15)' and a 'Python 2.7.15 Shell' window. The script in the IDE defines a function `rerata(b)` that calculates the average of a list. The shell shows the function being called with the list [1, 2, 3, 4, 5], resulting in the output 3, and then with a larger list, resulting in the output 6.

```
def rerata(b): #Nomor4
    jumlah = 0
    for i in range(len(b)):
        jumlah += b[i]
    jumlah = jumlah/len(b)
    return jumlah

>>> rerata([1,2,3,4,5])
3
>>> g = [3,4,5,4,3,4,2,2,10,11,23]
>>> rerata(g)
6
>>> |
```

Nomor 5



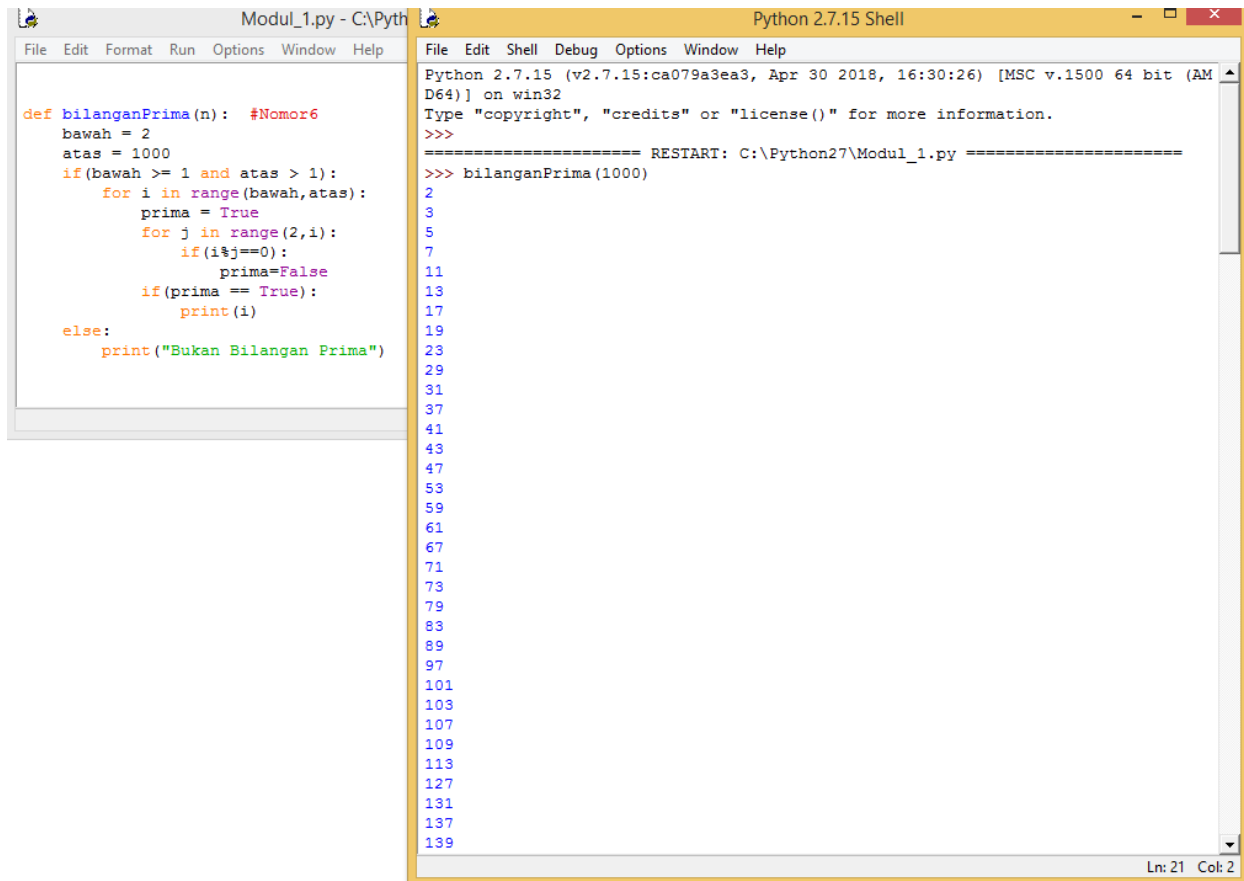
The image shows a screenshot of a Python IDE with two windows. The top window, titled 'Modul_1.py - C:\Python27\Modul_1.py (2.7.15)', contains a Python script. The script defines a function 'apakahPrima(n)' that checks if a number is prime. It uses a list of small primes and a loop to check divisibility. The bottom window, titled 'Python 2.7.15 Shell', shows the execution of the script. It displays the output of the function for inputs 17, 97, and 123, which are True, True, and False respectively.

```
from math import sqrt as sq #Nomor5
def apakahPrima(n):
    n = int(n)
    assert n>=0
    primaKecil = [2,3,5,7,9,11]
    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
        return True
```

```
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:\Python27\Modul_1.py =====
>>> apakahPrima(17)
True
>>> apakahPrima(97)
True
>>> apakahPrima(123)
False
>>>
```

Ln: 8 Col: 4

Nomor 6

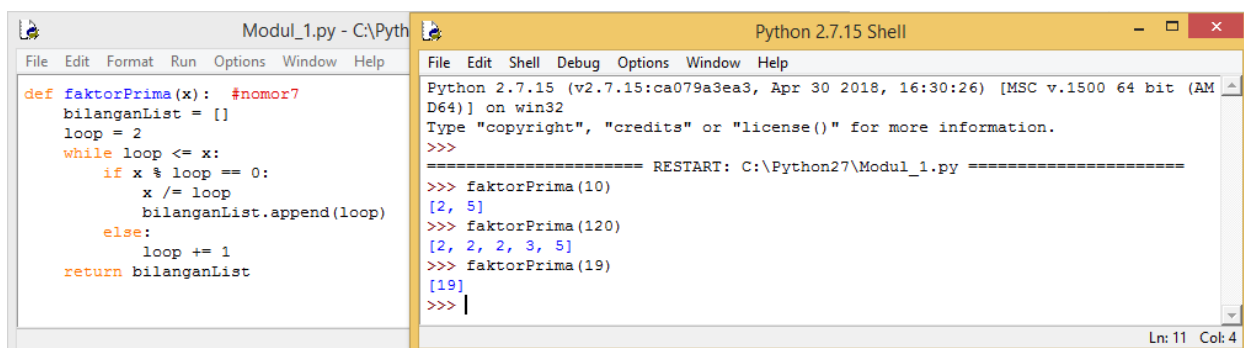


The image shows a Python IDE with two windows. The left window, titled 'Modul_1.py - C:\Pyth...', contains a function `bilanganPrima(n)` that finds prime numbers between `bawah` and `atas`. The right window, titled 'Python 2.7.15 Shell', shows the execution of `bilanganPrima(1000)`, which outputs a list of prime numbers from 2 to 997.

```
def bilanganPrima(n): #Nomor6
    bawah = 2
    atas = 1000
    if(bawah >= 1 and atas > 1):
        for i in range(bawah,atas):
            prima = True
            for j in range(2,i):
                if(i%j==0):
                    prima=False
            if(prima == True):
                print(i)
    else:
        print("Bukan Bilangan Prima")

>>> bilanganPrima(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
```

Nomor 7

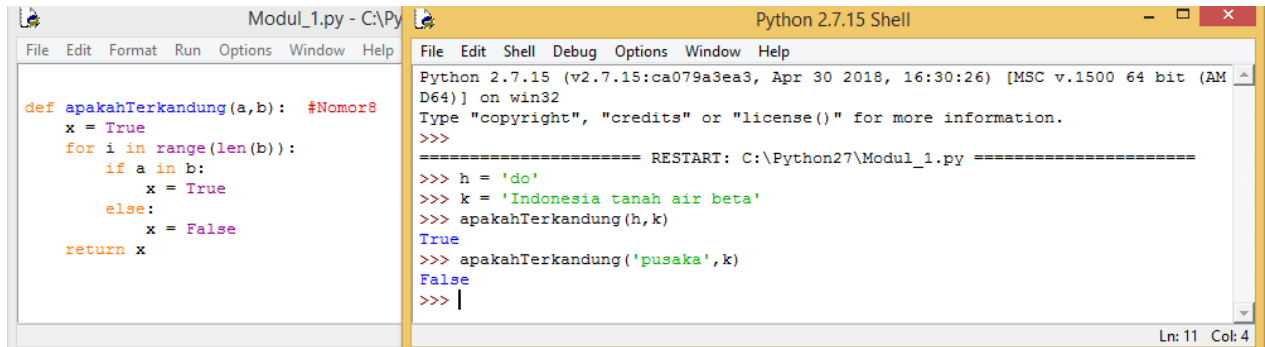


The image shows a Python IDE with two windows. The left window, titled 'Modul_1.py - C:\Pyth...', contains a function `faktorPrima(x)` that finds the prime factors of `x` and returns them as a list. The right window, titled 'Python 2.7.15 Shell', shows the execution of `faktorPrima(10)`, `faktorPrima(120)`, and `faktorPrima(19)`, which outputs the lists `[2, 5]`, `[2, 2, 2, 3, 5]`, and `[19]` respectively.

```
def faktorPrima(x): #nomor7
    bilanganList = []
    loop = 2
    while loop <= x:
        if x % loop == 0:
            x /= loop
            bilanganList.append(loop)
        else:
            loop += 1
    return bilanganList

>>> faktorPrima(10)
[2, 5]
>>> faktorPrima(120)
[2, 2, 2, 3, 5]
>>> faktorPrima(19)
[19]
>>> |
```

Nomor 8



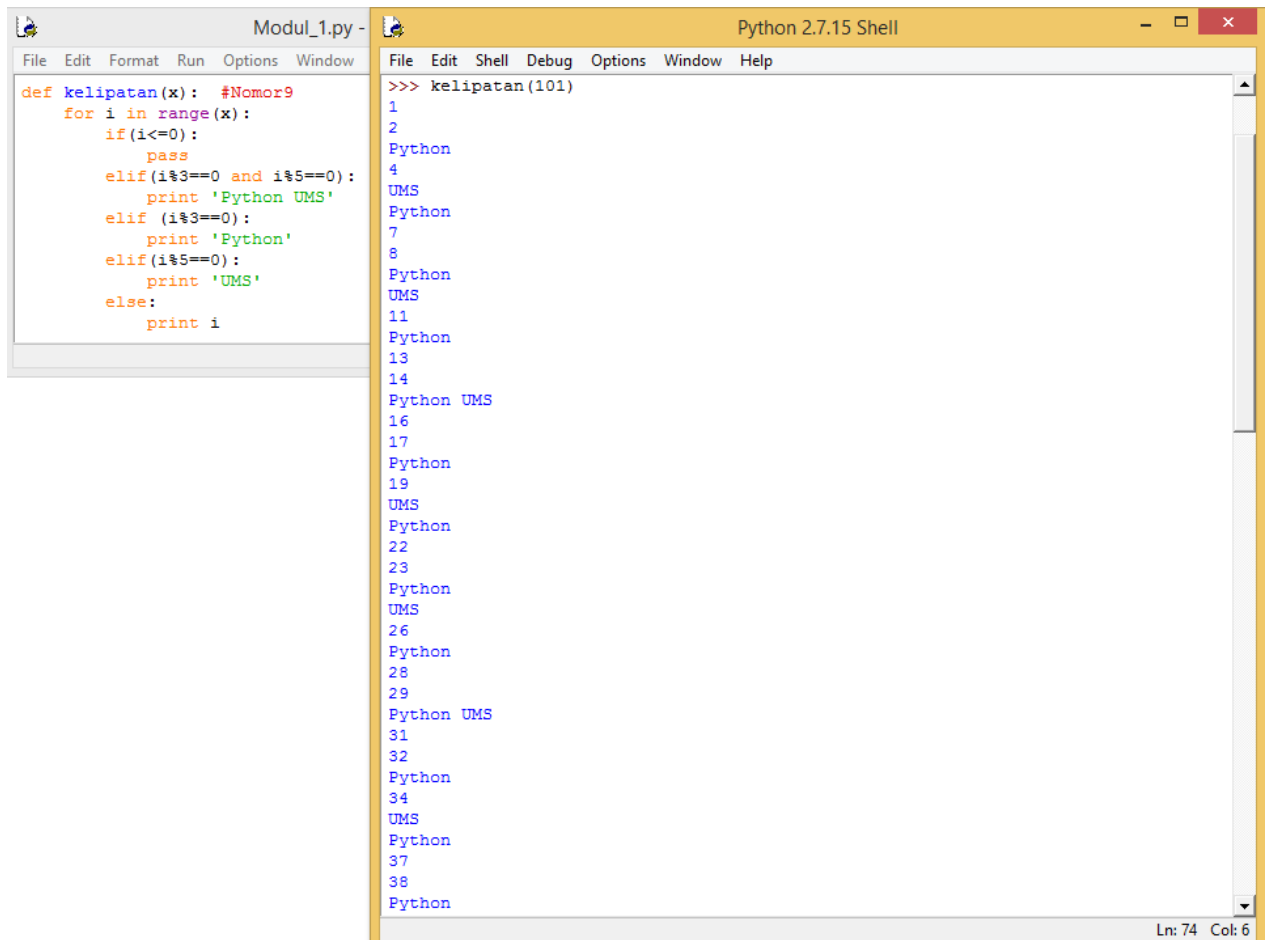
The screenshot shows a Python IDE with two windows. The left window, titled 'Modul_1.py - C:\Py', contains a function definition for 'apakahTerkandung'. The right window, titled 'Python 2.7.15 Shell', shows the execution of the function with test cases.

```
def apakahTerkandung(a,b): #Nomor8
    x = True
    for i in range(len(b)):
        if a in b:
            x = True
        else:
            x = False
    return x
```

```
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:\Python27\Modul_1.py =====
>>> h = 'do'
>>> k = 'Indonesia tanah air beta'
>>> apakahTerkandung(h,k)
True
>>> apakahTerkandung('pusaka',k)
False
>>> |
```

Ln: 11 Col: 4

Nomor 9



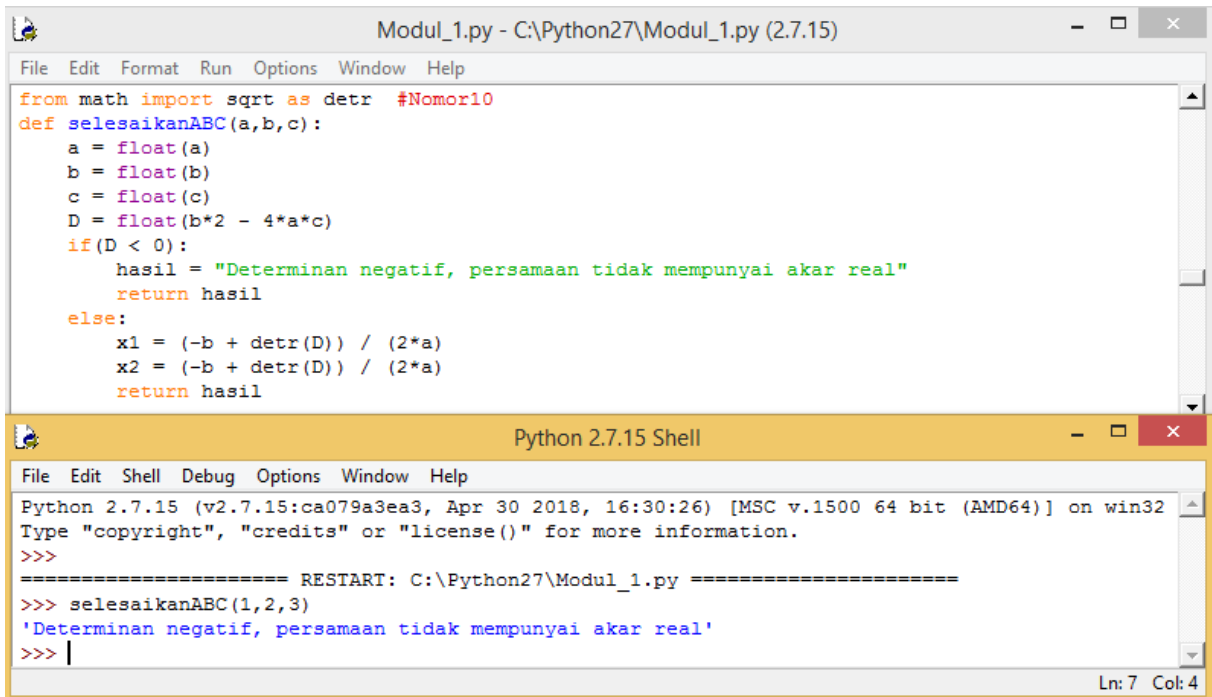
The screenshot shows a Python IDE with two windows. The left window, titled 'Modul_1.py - C:\Py', contains a function definition for 'kelipatan'. The right window, titled 'Python 2.7.15 Shell', shows the execution of the function with the argument 101.

```
def kelipatan(x): #Nomor9
    for i in range(x):
        if i<=0:
            pass
        elif(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
```

```
>>> kelipatan(101)
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
```

Ln: 74 Col: 6

Nomor 10



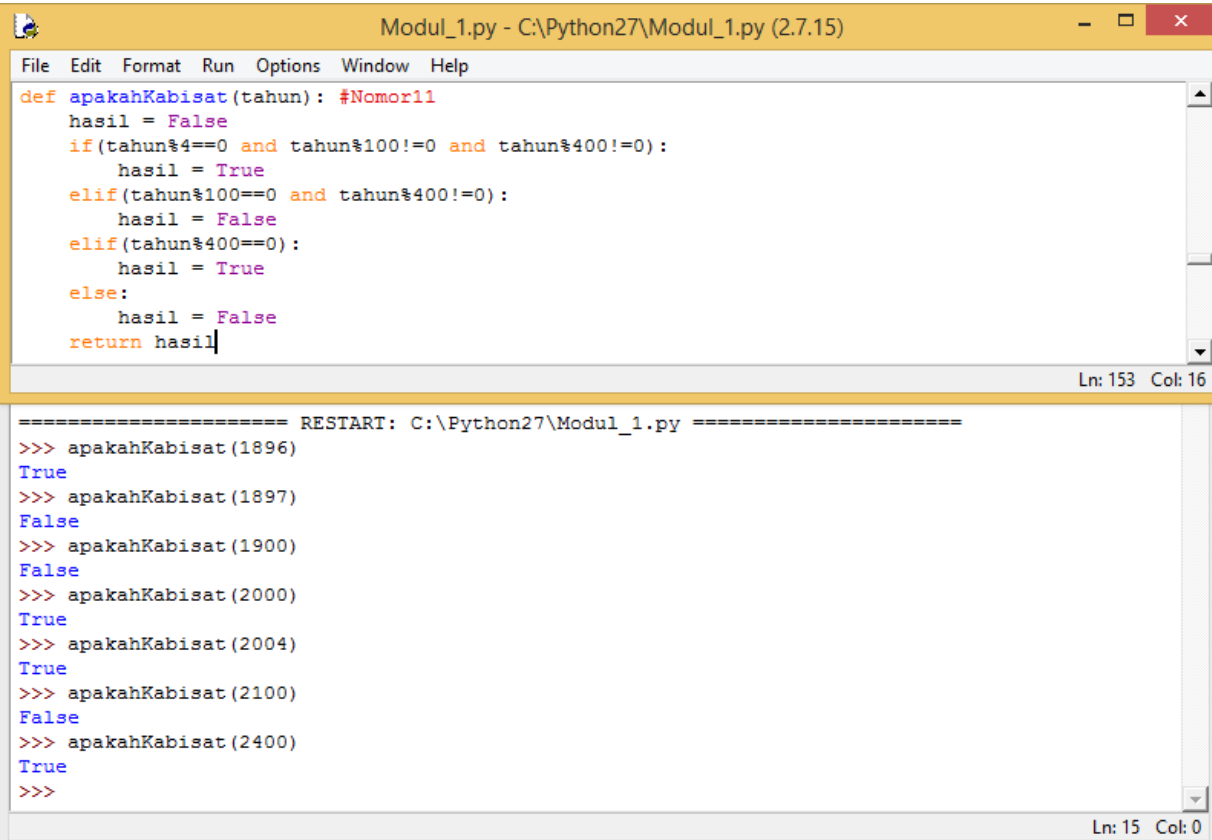
The screenshot shows a Python IDE window titled "Modul_1.py - C:\Python27\Modul_1.py (2.7.15)". The code defines a function `selesaikanABC(a,b,c)` that calculates the discriminant $D = b^2 - 4ac$. If $D < 0$, it returns the string "Determinan negatif, persamaan tidak mempunyai akar real". Otherwise, it calculates the two roots x_1 and x_2 using the quadratic formula and returns them. Below the code editor is a "Python 2.7.15 Shell" window showing the execution of the function with arguments (1, 2, 3), which results in the expected output string.

```
from math import sqrt as detr #Nomor10
def selesaikanABC(a,b,c):
    a = float(a)
    b = float(b)
    c = float(c)
    D = float(b*2 - 4*a*c)
    if(D < 0):
        hasil = "Determinan negatif, persamaan tidak mempunyai akar real"
        return hasil
    else:
        x1 = (-b + detr(D)) / (2*a)
        x2 = (-b + detr(D)) / (2*a)
        return hasil
```

```
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:\Python27\Modul_1.py =====
>>> selesaikanABC(1,2,3)
'Determinan negatif, persamaan tidak mempunyai akar real'
>>>
```

Ln: 7 Col: 4

Nomor 11



The screenshot shows a Python IDE window titled "Modul_1.py - C:\Python27\Modul_1.py (2.7.15)". The code defines a function `apakahKabisat(tahun)` that checks if a year is a leap year based on the rules: divisible by 4 but not 100, or divisible by 400. The shell window shows multiple test cases with years ranging from 1896 to 2400, demonstrating the function's logic.

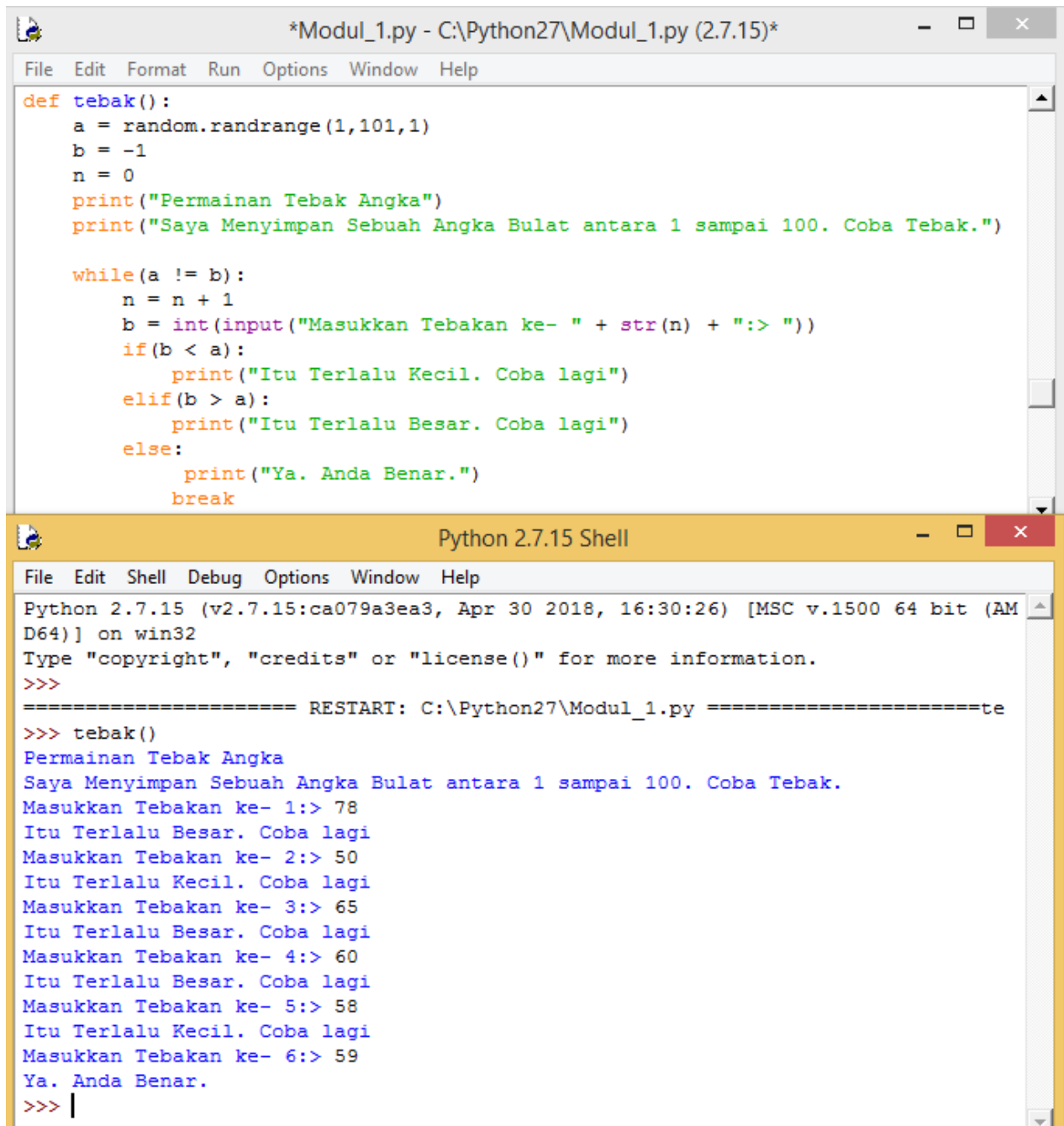
```
def apakahKabisat(tahun): #Nomor11
    hasil = False
    if(tahun%4==0 and tahun%100!=0 and tahun%400!=0):
        hasil = True
    elif(tahun%100==0 and tahun%400!=0):
        hasil = False
    elif(tahun%400==0):
        hasil = True
    else:
        hasil = False
    return hasil
```

```
===== RESTART: C:\Python27\Modul_1.py =====
>>> apakahKabisat(1896)
True
>>> apakahKabisat(1897)
False
>>> apakahKabisat(1900)
False
>>> apakahKabisat(2000)
True
>>> apakahKabisat(2004)
True
>>> apakahKabisat(2100)
False
>>> apakahKabisat(2400)
True
>>>
```

Ln: 153 Col: 16

Ln: 15 Col: 0

Nomor 12



The image shows a screenshot of a Python IDE with two windows. The top window, titled '*Modul_1.py - C:\Python27\Modul_1.py (2.7.15)*', contains a Python script for a number guessing game. The script defines a function 'tebak()' that generates a random number 'a' between 1 and 100, initializes 'b' to -1 and 'n' to 0, and prints a welcome message. It then enters a 'while' loop where it prompts the user to guess a number. If the guess is too low, it says 'Itu Terlalu Kecil. Coba lagi'. If too high, it says 'Itu Terlalu Besar. Coba lagi'. If correct, it says 'Ya. Anda Benar.' and breaks the loop.

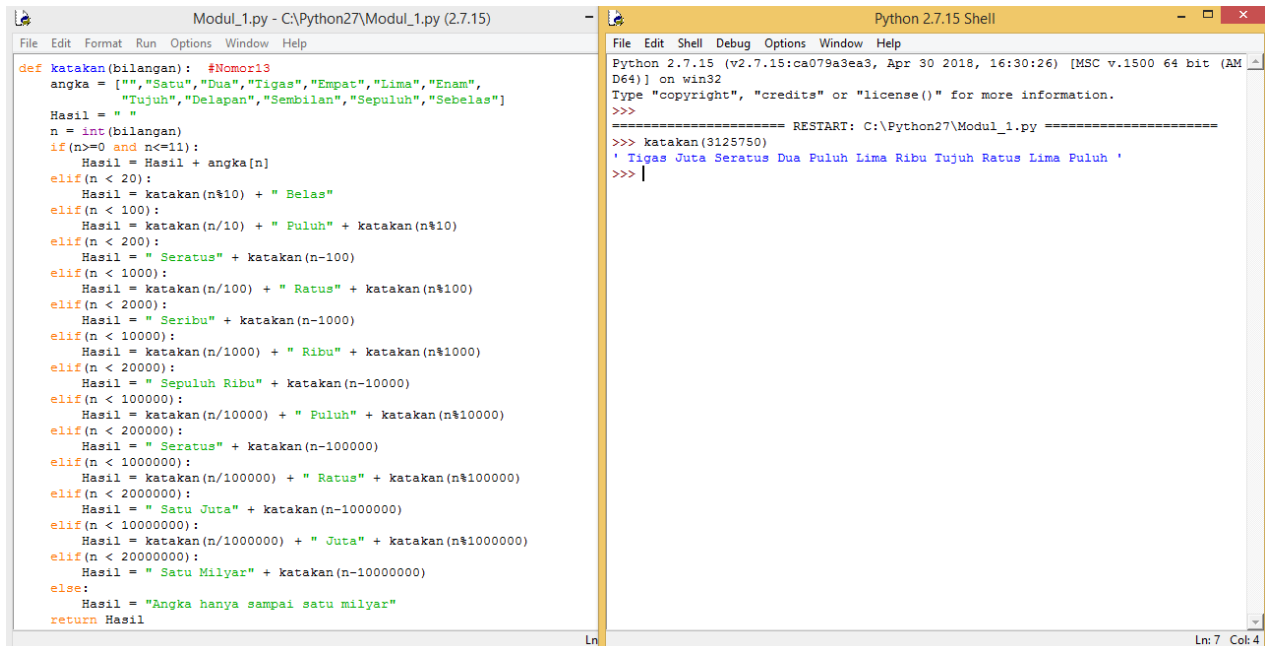
```
def tebak():
    a = random.randrange(1,101,1)
    b = -1
    n = 0
    print("Permainan Tebak Angka")
    print("Saya Menyimpan Sebuah Angka Bulat antara 1 sampai 100. Coba Tebak.")

    while(a != b):
        n = n + 1
        b = int(input("Masukkan Tebakan ke- " + str(n) + " :> "))
        if(b < a):
            print("Itu Terlalu Kecil. Coba lagi")
        elif(b > a):
            print("Itu Terlalu Besar. Coba lagi")
        else:
            print("Ya. Anda Benar.")
            break
```

The bottom window, titled 'Python 2.7.15 Shell', shows the execution of the script. It displays the version information, a prompt to type 'copyright', 'credits', or 'license()', and then the execution of 'tebak()'. The output shows the game's progress: the user guesses 78 (too high), 50 (too low), 65 (too high), 60 (too high), 58 (too low), and finally 59 (correct). The shell ends with a prompt ' >>> |'.

```
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:\Python27\Modul_1.py =====te
>>> tebak()
Permainan Tebak Angka
Saya Menyimpan Sebuah Angka Bulat antara 1 sampai 100. Coba Tebak.
Masukkan Tebakan ke- 1:> 78
Itu Terlalu Besar. Coba lagi
Masukkan Tebakan ke- 2:> 50
Itu Terlalu Kecil. Coba lagi
Masukkan Tebakan ke- 3:> 65
Itu Terlalu Besar. Coba lagi
Masukkan Tebakan ke- 4:> 60
Itu Terlalu Besar. Coba lagi
Masukkan Tebakan ke- 5:> 58
Itu Terlalu Kecil. Coba lagi
Masukkan Tebakan ke- 6:> 59
Ya. Anda Benar.
>>> |
```


Nomor 13

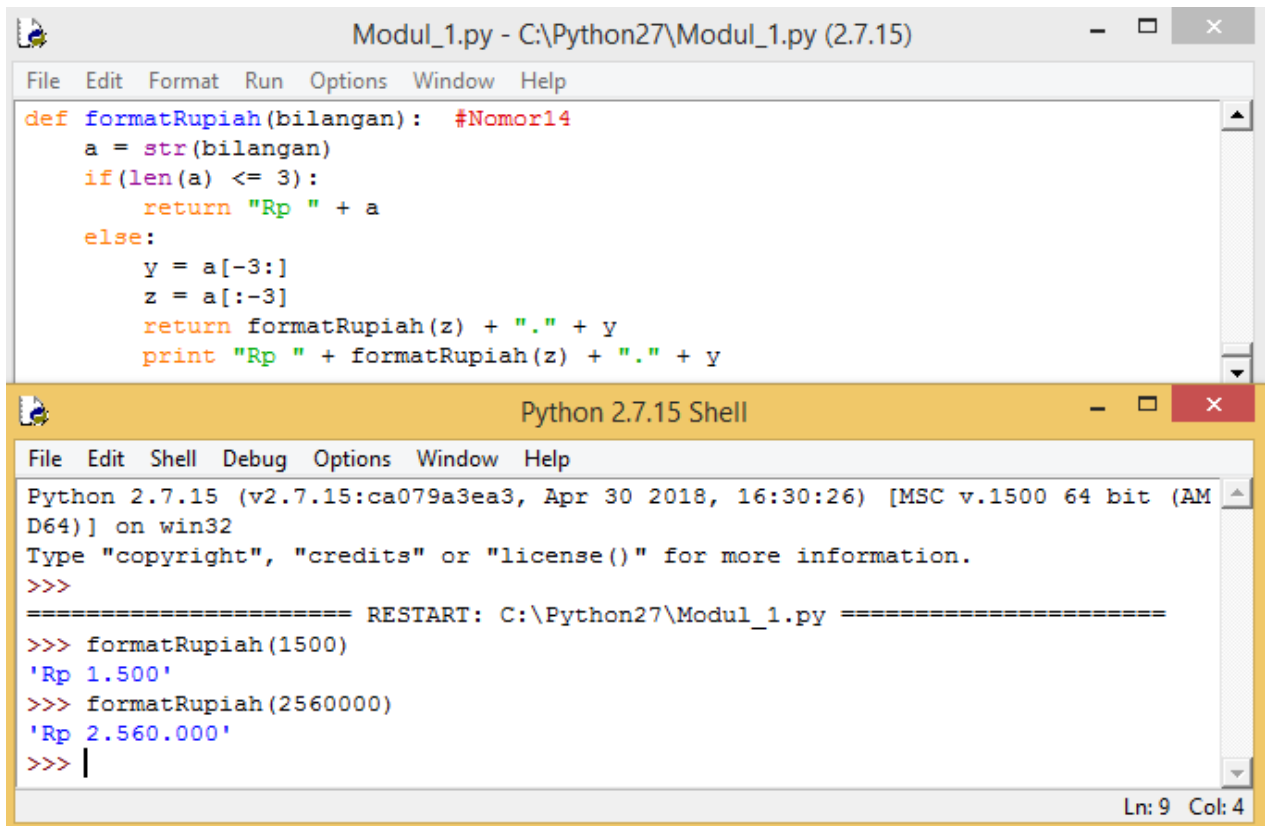


```
Modul_1.py - C:\Python27\Modul_1.py (2.7.15)
File Edit Format Run Options Window Help

def katakan(bilangan): #Nomor13
    angka = ["", "Satu", "Dua", "Tiga", "Empat", "Lima", "Enam",
             "Tujuh", "Delapan", "Sembilan", "Sepuluh", "Sebelas"]
    Hasil = ""
    n = int(bilangan)
    if(n>0 and n<=11):
        Hasil = 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 = " Seratus" + katakan(n-100)
    elif(n < 1000):
        Hasil = katakan(n/100) + " Ratus" + katakan(n%100)
    elif(n < 2000):
        Hasil = " Seribu" + katakan(n-1000)
    elif(n < 10000):
        Hasil = katakan(n/1000) + " Ribu" + katakan(n%1000)
    elif(n < 20000):
        Hasil = " Sepuluh Ribu" + katakan(n-10000)
    elif(n < 100000):
        Hasil = katakan(n/10000) + " Puluh" + katakan(n%10000)
    elif(n < 200000):
        Hasil = " Seratus" + katakan(n-100000)
    elif(n < 1000000):
        Hasil = katakan(n/100000) + " Ratus" + katakan(n%100000)
    elif(n < 2000000):
        Hasil = " Satu Juta" + katakan(n-1000000)
    elif(n < 10000000):
        Hasil = katakan(n/1000000) + " Juta" + katakan(n%1000000)
    elif(n < 20000000):
        Hasil = " Satu Milyar" + katakan(n-10000000)
    else:
        Hasil = "Angka hanya sampai satu milyar"
    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:\Python27\Modul_1.py =====
>>> katakan(3125750)
'Tiga Juta Seratus Dua Puluh Lima Ribu Tujuh Ratus Lima Puluh '
>>> |
```

Nomor 14



```
Modul_1.py - C:\Python27\Modul_1.py (2.7.15)
File Edit Format Run Options Window Help

def formatRupiah(bilangan): #Nomor14
    a = str(bilangan)
    if(len(a) <= 3):
        return "Rp " + a
    else:
        y = a[-3:]
        z = a[:-3]
        return formatRupiah(z) + "." + y
    print "Rp " + formatRupiah(z) + "." + y

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:\Python27\Modul_1.py =====
>>> formatRupiah(1500)
'Rp 1.500'
>>> formatRupiah(2560000)
'Rp 2.560.000'
>>> |
```