

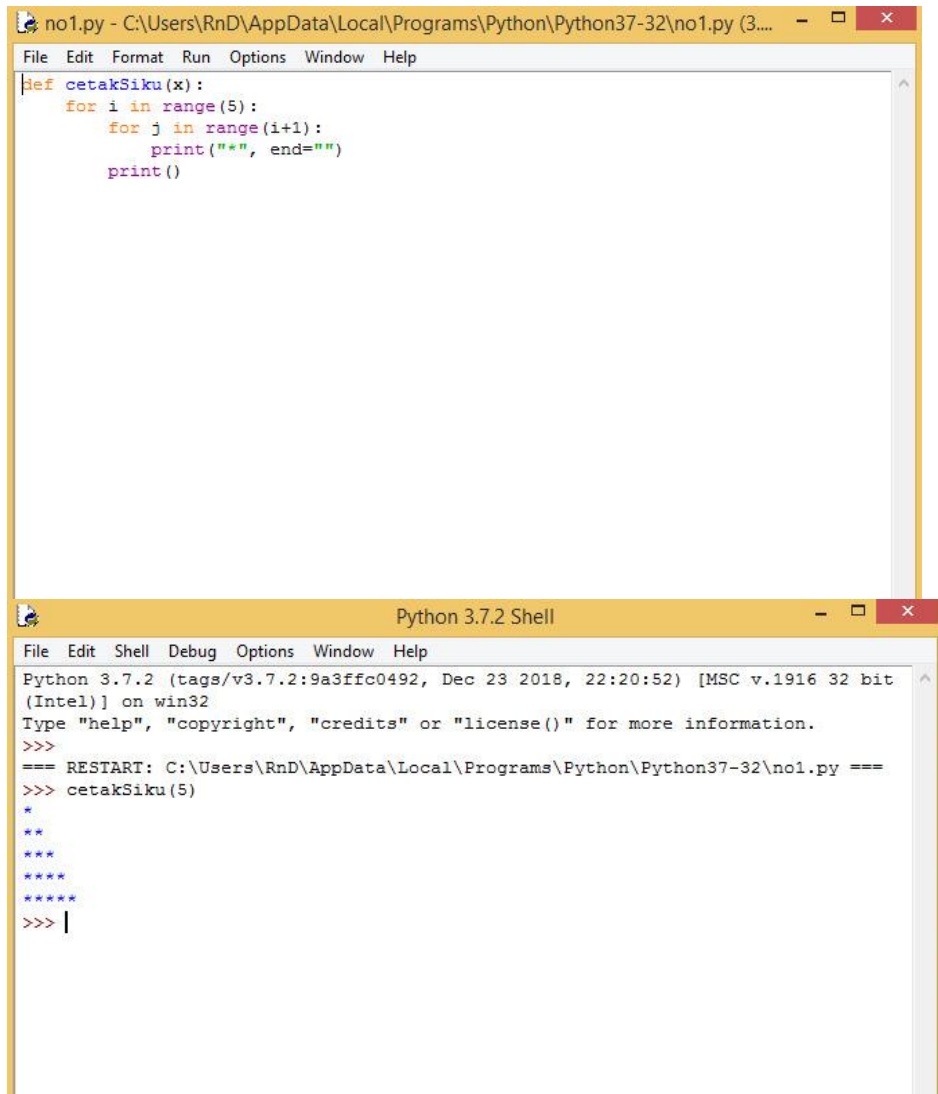
**NAMA** = **CORRY LUQMA ZUNIRA**

**KELAS** = **D**

**NIM** = **L200170152**

**MODUL** = **1**

1.



The image shows a screenshot of a Python IDE with two windows. The top window, titled 'no1.py', contains the following Python code:

```
def cetakSiku(x):  
    for i in range(5):  
        for j in range(i+1):  
            print("x", end="")  
            print()
```

The bottom window, titled 'Python 3.7.2 Shell', shows the execution of the code. It displays the following output:

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit  
(Intel)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
=== RESTART: C:\Users\RnD\AppData\Local\Programs\Python\Python37-32\nol.py ===  
>>> cetakSiku(5)  
x  
x  
x  
x  
x  
>>> |
```

2.

The screenshot shows a Python 2.7.13 IDE window with a file named `no2.py`. The code defines a function `PersegiEmpat(x,y)` that prints a grid of asterisks. The function is called with `PersegiEmpat(4,5)`. The output in the shell window shows a 4x5 grid of asterisks.

```
def PersegiEmpat(x,y):
    for i in range(x):
        if i==0 or i==x-1:
            print ("@"*y)
        else:
            print ("@"+" "*(y-2)+"@")
PersegiEmpat(4,5)
```

```
Python 2.7.13 (v2.7.13:a06454blafal, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: D:\KULIAH CORRY LUQMA\SEMESTER IV\PRAKTIKUM ALGOSTRUK\n02.py ===
@@@@@
@ @
@ @
@@@@@
>>>
```

3. a)

The screenshot shows a Python 2.7.13 IDE window with a file named `no3a.py`. The code defines a function `itung(x)` that counts the number of vowels in a string. The function is called with `itung('Surakarta')`. The output in the shell window shows the count 4.

```
def itung(x):
    vocal='aioeuAIOUE'
    c=0
    for i in x:
        if i in vocal:
            c+=1
    return (len(x),c)
print (itung('Surakarta'))
```

```
Python 2.7.13 (v2.7.13:a06454blafal, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: D:\KULIAH CORRY LUQMA\SEMESTER IV\PRAKTIKUM ALGOSTRUK\n03a.py ===
(9, 4)
>>> |
```

3. b)

The screenshot shows a Python 2.7.13 IDE window with a file named `no3b.py`. The code defines a function `itung(x)` that counts the number of vowels in a string. The function is called with `itung('Surakarta')`. The output in the shell window shows the count 5.

```
def itung(x):
    vocal='bcdfghjklmnpqrstvwxyzBCDFGHJKLMPQRSTVWXYZ'
    d=0
    for i in x:
        if i in vocal:
            d+=1
    return (len(x),d)
print (itung('Surakarta'))
```

```
Python 2.7.13 (v2.7.13:a06454blafal, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: D:\KULIAH CORRY LUQMA\SEMESTER IV\PRAKTIKUM ALGOSTRUK\n03b.py ===
(9, 5)
>>> |
```

4.

The screenshot shows a Python 2.7.13 IDE window with a file named `no4.py`. The code defines a function `rerata(b)` that calculates the average of a list. The function is called with `rerata([1,2,3,4,5])` and `rerata(g)` where `g` is a list of numbers. The output in the shell window shows the average 3 for the first call and 6 for the second call.

```
def rerata(b):
    a=sum(b)/len(b)
    print (a)
```

```
Python 2.7.13 (v2.7.13:a06454blafal, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: D:\KULIAH CORRY LUQMA\SEMESTER IV\PRAKTIKUM ALGOSTRUK\n04.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.

```

File Edit Format Run Options Window Help
no5.py - D:\KULIAH CORRY LUQMA\SEMESTER IV\PRAKTIKUM ALGOSTRUK\no5.py (2)
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
        return True

Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454blafal, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: D:\KULIAH CORRY LUQMA\SEMESTER IV\PRAKTIKUM ALGOSTRUK\no5.py ===
>>> apakahPrima(17)
True
>>> apakahPrima(97)
True
>>> apakahPrima(123)
False
>>>

```

6.

```

File Edit Format Run Options Window Help
no6.py - D:\KULIAH CORRY LUQMA\SEMESTER IV\PRAKTIKUM ALGOSTRUK\no6.py (2.7.13)
def prima(x,y):
    for i in range(2,1000,1):
        xy=2
        while i%xy != 0:
            if xy == i-1:
                print(i)
            xy = xy+1

Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454blafal, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: D:\KULIAH CORRY LUQMA\SEMESTER IV\PRAKTIKUM ALGOSTRUK\no6.py ===
>>> prima(2,1000)
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
151

```

7.

```

File Edit Format Run Options Window Help
no7.py - D:\KULIAH CORRY LUQMA\SEMESTER IV\PRAKTIKUM ALGOSTRUK\no7.py
def faktorPrima(x):
    faktor = []
    awal = 2
    while awal <= x:
        if x%awal==0:
            x/=awal
            faktor.append(awal)
        else:
            awal+=1
    return faktor

Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454blafal, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: D:\KULIAH CORRY LUQMA\SEMESTER IV\PRAKTIKUM ALGOSTRUK\no7.py ===
>>> faktorPrima(10)
[2, 5]
>>> faktorPrima(120)
[2, 2, 2, 3, 5]
>>> faktorPrima(19)
[19]
>>>

```

8.

```

Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454b1afaf, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: D:\KULIAH CORRY LUQMA\SEMESTER IV\PRAKTIKUM ALGOSTRUK\no8.py ===
True
False
>>> h = "do"
>>> k = "Indonesia tanah air beta"
>>> apakahTerkandung(h,k)
True
>>> apakahTerkandung('pusaka', k)
False
>>>

```

9.

```

Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454b1afaf, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: D:\KULIAH CORRY LUQMA\SEMESTER IV\PRAKTIKUM ALGOSTRUK\no9.py ===
>>> cetak()
1
2
python
4
UMS
python
7
8
python
UMS
11
python
13
14
pyton UMS
16
17
python
19
UMS
python
22
23
python
UMS
26
python
28
29
pyton UMS
31
32
python

```

10.

```

Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454b1afaf, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: D:\KULIAH CORRY LUQMA\SEMESTER IV\PRAKTIKUM ALGOSTRUK\no10.py ===
>>> print(selesaikanABC(1,2,3))
determinan negatif. Persamaan tidak mempunyai akar real
>>>

```

11.

The screenshot shows a Python IDE with a file named 'no11.py'. The code defines a function 'apakahKabisat(x)' that checks if a year is a leap year. The shell window shows the execution of the function for various years.

```

def apakahKabisat(x):
    if (x%400==0):
        return True
    if (x%100==0):
        return False
    if (x%4==0):
        return True
    return False

```

```

Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454b1afal, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: D:\KULIAH CORRY LUQMA\SEMESTER IV\PRAKTIKUM ALGOSTRUK\nol1.py ====
>>> apakahKabisat(1896)
True
>>> apakahKabisat(1897)
False
>>> apakahKabisat(1900)
False
>>> apakahKabisat(2000)
True
>>> apakahKabisat(2100)
False
>>> apakahKabisat(2200)
False
>>>

```

12.

The screenshot shows a Python IDE with a file named 'no12.py'. The code defines a function 'tebakAngka()' that generates a random number and prompts the user to guess it. The shell window shows the execution of the function, with the user entering guesses and receiving feedback.

```

import random
def tebakAngka():
    a=random.randrange(0, 100)
    while(True):
        b=int(input("Masukkan angka: "))
        if (b>a):
            print("Itu terlalu besar, coba lagi")
        elif (b<a):
            print("Itu terlalu kecil, coba lagi")
        else:
            print("Ya. Anda benar")
            break

```

```

Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454b1afal, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: D:\KULIAH CORRY LUQMA\SEMESTER IV\PRAKTIKUM ALGOSTRUK\nol2.py ====
>>> tebakAngka()
Masukkan Angka: 100
Itu terlalu besar, coba lagi
Masukkan Angka: 50
Itu terlalu besar, coba lagi
Masukkan Angka: 99
Itu terlalu besar, coba lagi
Masukkan Angka: 1
Itu terlalu kecil, coba lagi
Masukkan Angka: 35
Itu terlalu besar, coba lagi
Masukkan Angka: 58
Itu terlalu besar, coba lagi
Masukkan Angka: 55
Itu terlalu besar, coba lagi
Masukkan Angka: 78
Itu terlalu besar, coba lagi
Masukkan Angka: 90
Itu terlalu besar, coba lagi
Masukkan Angka: 99
Itu terlalu besar, coba lagi
Masukkan Angka: 23
Itu terlalu besar, coba lagi
Masukkan Angka: 45
Itu terlalu besar, coba lagi
Masukkan Angka: 12
Itu terlalu kecil, coba lagi
Masukkan Angka: 10
Itu terlalu kecil, coba lagi
Masukkan Angka: 99
Itu terlalu besar, coba lagi
Masukkan Angka: 95
Itu terlalu besar, coba lagi
Masukkan Angka: 90
Itu terlalu besar, coba lagi
Masukkan Angka:

```

13.

The screenshot shows a Python IDE with a file named 'no13.py' open. The script defines a function 'katakan' that takes a string 'a' and returns a string 'c' by concatenating words from a dictionary 'x' based on the digits in 'a'. The dictionary 'x' maps digits 1-9 to Indonesian words: 1: 'Se', 2: 'Dua', 3: 'Tiga', 4: 'Empat', 5: 'Lima', 6: 'Enam', 7: 'Tujuh', 8: 'Delapan', 9: 'Sembilan'. The function 'katakan' iterates through the string 'a' from right to left, building the result string 'c' by concatenating the corresponding word from 'x' and a space. The shell window shows the execution of 'katakan(10)', which outputs 'Sepuluh'.

```

def katakan(a):
    x={"0":"","1":"Se","2":"Dua ","3":"Tiga ","4":"Empat ","5":"Lima ","6":"Enam ","7":"Tujuh ","8":"Delapan ","9":"Sembilan "}
    y={"-1":"","-2":"puluh ","-3":"ratus ","-4":"ribu ","-5":"puluh ","6":"ratus ","7":"juta ","8":"puluhjuta "}
    b=str(a)
    c=""
    i=-1
    while i>= -len(b):
        c=x[b[i]]+y[i]+c
        i=i-1
    return c

>>>
=== RESTART: D:\KULIAH CORRY LUQMA\SEMESTER IV\PRAKTIKUM ALGOSTRUK\nol3.py ===
>>> katakan(10)
'Sepuluh '
>>>
  
```

14.

The screenshot shows a Python IDE with a file named 'no14.py' open. The script defines a function 'formatRupiah' that takes a string 'x' and returns a string 'y' by formatting the number in 'x' with commas as thousands separators. The function 'formatRupiah' iterates through the string 'x' from right to left, building the result string 'y' by concatenating the formatted number and a comma. The shell window shows the execution of 'formatRupiah(1500)', which outputs 'Rp 1.500', and 'formatRupiah(2560000)', which outputs 'Rp 2.560.000'.

```

def formatRupiah(x):
    y = str(x)
    if len(y) <= 3:
        return 'Rp ' + y
    else:
        a = y[-3:]
        b = y[:-3]
        return formatRupiah(b) + "," + a
    print ('Rp ' + formatRupiah(b) + "," + a)

>>>
=== RESTART: D:\KULIAH CORRY LUQMA\SEMESTER IV\PRAKTIKUM ALGOSTRUK\nol4.py ===
>>> formatRupiah(1500)
'Rp 1.500'
>>> formatRupiah(2560000)
'Rp 2.560.000'
>>>
  
```