

# MODUL 1

## PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

Nama : Afiq Tri Nugraha

NIM : L200180080

Kelas : C

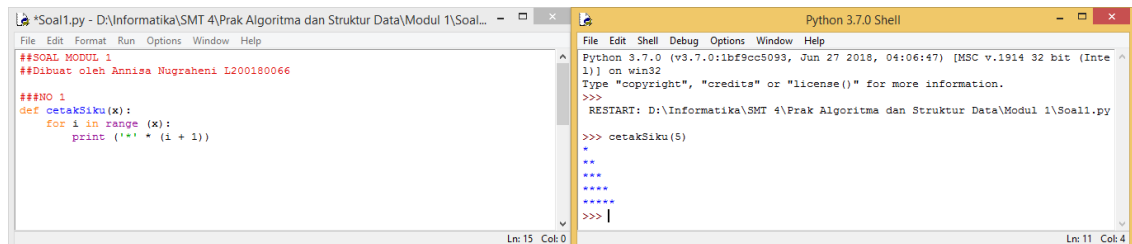
Soal – Soal Untuk Mahasiswa

1. Buatlah suatu fungsi cetakSiku(5) yang akan mencetak yang berikut:

```
*  
**  
***  
****  
*****
```

Nilai x menunjukkan tingi segitiga itu (gambar diatas berarti bisa didapatkan dari menjalankan cetakSiku(5)).

**Jawab:**



```
def cetakSiku(x):
```

```
    for i in range (x):
```

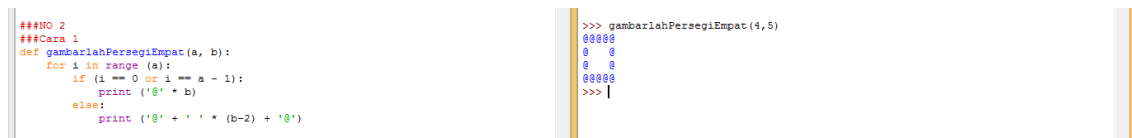
```
        print ('*' * (i + 1))
```

2. Buatlah sebuah fungsi yang menerima dua integer positif, yang akan menggambar bentuk persegi empat. Contoh pemanggilan:

```
>>> gambarlahPersegiEmpat(4,5)  
00000  
0 0  
0 0  
0 0  
00000
```

**Jawab:**

Cara 1:



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

Cara 2:

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

```
RESTART: D:\Informatika\SMT 4\Prak Algoritma dan Struktur Data\Modul 1\Soal1.py
>>> gambarlahPersegiEmpat(4,5)
@@@@@
@  @
@  @
@@@@@
>>> |
```

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

3. Berikut ini adalah dua soal yang saling berkaitan

a. Buatlah sebuah fungsi yang menerima sebuah string dan mengembalikan sebuah list yang terdiri dari dua integer. Dua integer kembalian ini adalah jumlah huruf di string itu dan jumlah huruf vocal (huruf vocal adalah huruf hidup) di string itu. Contoh pemanggilan:

```
>>> k = jumlahHurufVokal('Surakarta')
>>> k
(9, 4)
```

**Jawab:**

```
####NO 3
####3A
def jumlahHurufVokal(ch) :
    b = len(ch)
    a = 0
    for i in ch :
        if (i=='A' or i=='a' or i=='E' or i=='e' or i=='I' or i=='i' or i=='O' or i=='o' or i=='U' or i=='u') :
            a += 1
    return b, a
```

```
>>>
RESTART: D:\Informatika\SMT 4\Prak Algoritma dan Struktur Data\Modul 1\Soal1.py
>>> k = jumlahHurufVokal('Surakarta')
>>> k
(9, 4)
>>>
```

```
def gambarlahPersegiEmpat(a,b) :
    for i in range(a) :
        if ((i+1) == 1) :
            print(b*"@")
        elif ((i+1) == a) :
```

```
print(b*"@")
```

else :

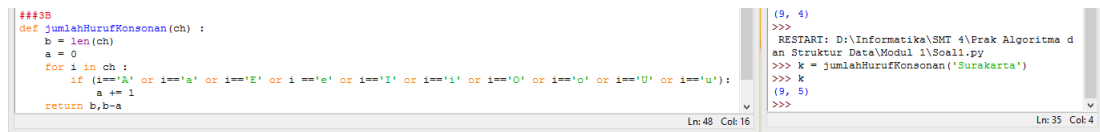
```
print("@"+" "*(b-2)+"@")
```

- b. Sama dengan soal (a) di atas, tapi sekarang yang dihitung adalah huruf konsonan.

Hanya ada satu baris yang berbeda di dalam kodenya! Contoh pemanggilan:

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

**Jawab:**



```
def jumlahHurufKonsonan(ch) :
    b = len(ch)
    a = 0
    for i in ch :
        if (i=='A' or i=='a' or i=='E' or i=='e' or i=='I' or i=='i' or i=='O' or i=='o' or i=='U' or i=='u') :
            a += 1
    return b,b-a
```

Ln:48 Col:16

```
(9, 4)
>>>
RESTART: D:\Informatika\SMT 4\Prak Algoritma dan Struktur Data\Modul 1\Soal1.py
>>> k = jumlahHurufKonsonan('Surakarta')
>>> k
(9, 5)
>>>
```

Ln:35 Col:4

```
def jumlahHurufKonsonan(ch) :
```

```
    b = len(ch)
```

```
    a = 0
```

```
    for i in ch :
```

```
        if (i=='A' or i=='a' or i=='E' or i=='e' or i=='I' or i=='i' or i=='O' or i=='o' or
i=='U' or i=='u'):
```

```
            a += 1
```

```
    return b,b-a
```

4. Buatlah sebuah fungsi yang menghitung rerata sebuah array yang berisi bilangan. Rerata mempunyai rumus

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

Namun ingatlah bahwa Python memulai index dari 0. Fungsi itu harus mempunyai bentuk rerata (x), dengan x adalah list berisi bilangan yang ingin dihitung reratanya. Jadi, pekerjaanmu akan mempunyai bentuk:

- Buat suatu file dengan isi seperti ini:

```
def rerata(b) :
    #Di sini letak
    #programmu
    #...
    return hasil
```

- Larikan program itu dengan memencet F5, lalu panggil program itu seperti ini

```
rerata([1,2,3,4,5])    #hasilnya 3
```

```
g = [3,4,5,4,3,4,5,2,2,10,11,23]
```

```
rerata(g)
```

**Jawab:**

```
##NO 4
def rerata(x) :
    a = 0
    b = 0
    for i in x :
        a += 1
        b = b + i
    a = float(a)
    b = float(b)
    return (b/a)
```

```
>>> rerata([1,2,3,4,5])
3.0
>>> g = [3,4,5,4,3,4,5,2,2,10,11,23]
>>> rerata(g)
6.333333333333333
>>> |
```

```
def rerata(x) :
```

```
    a = 0
```

```
    b = 0
```

```
    for i in x :
```

```
        a += 1
```

```
        b = b + i
```

```
    a = float(a)
```

```
    b = float(b)
```

```
    return(b/a)
```

5. Buatlah suatu fungsi untuk menentukan apakah suatu bilangan bulat adalah bilangan prima atau bukan.

Setelah selesai, larikan program di atas dan lalu tes di Python Shell:

```
apakahPrima(17)
```

```
apakahPrima(97)
```

```
apakahPrima(123)
```

**Jawab:**

```
##NO 5
def apakahPrima(a) :
    x = 0
    for i in range(a) :
        if a % (i+1) == 0 :
            x += 1
    if x == 2 :
        print("YA")
    else :
        print("TIDAK")
```

```
>>> apakahPrima(17)
YA
>>> apakahPrima(97)
YA
>>> apakahPrima(123)
TIDAK
>>> |
```

```
def apakahPrima(a) :
```

```
    x = 0
```

```
    for i in range(a) :
```

```
        if a % (i+1) == 0 :
```

```
            x += 1
```

```
    if x == 2 :
```

```
        print("YA")
```

```
    else :
```

```
        print("TIDAK")
```

6. Buatlah suatu program yang mencetak semua bilangan prima dari 2 sampai 1000. Kamu tidak harus memanfaatkan fungsi diatas.

**Jawab:**

```
Soal1.py - D:\Informatika\SMT 4\Prak Algoritma dan Struktur Data\Modul 1\Soal1.py
File Edit Format Run Options Window Help

def cekPrima() :
    y = range(1001)
    for i in range(2,1001) :
        x = 0
        for j in range(1) :
            if i % (j+1) == 0 :
                x += 1
        if x == 2 :
            print(i)

Ln: 84 Col: 20

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help

RESTART: D:\Informatika\SMT 4\Prak Algoritma dan Struktur Data\Modul 1\Soal1.py

>>> cekPrima()
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
151
157

Ln: 182 Col: 4
```

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help

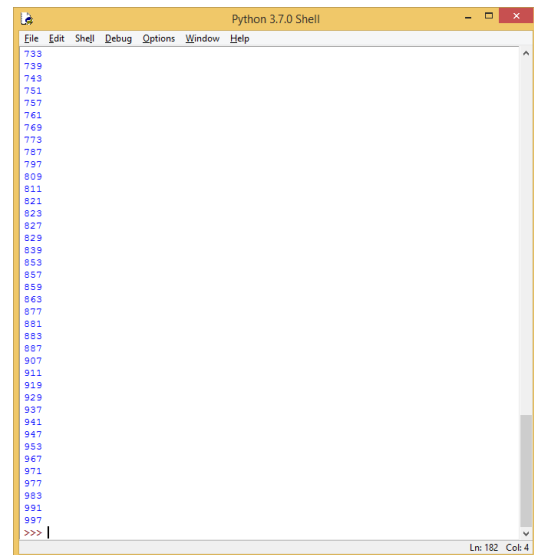
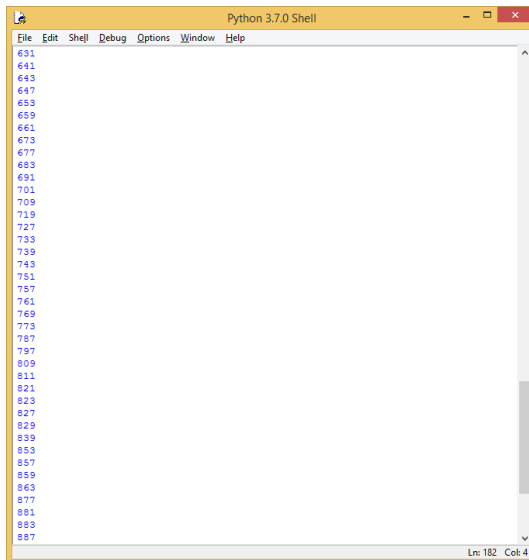
157
163
167
173
179
181
191
193
197
199
211
223
227
229
233
239
241
251
257
263
269
271
277
281
283
293
307
311
313
317
331
337
347
349
353
359
367
373
379
383

Ln: 182 Col: 4

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help

383
389
397
401
409
419
421
431
433
439
443
449
457
461
463
467
479
487
491
499
503
509
521
523
541
547
557
563
569
571
577
587
593
599
601
607
613
617
619
631

Ln: 182 Col: 4
```

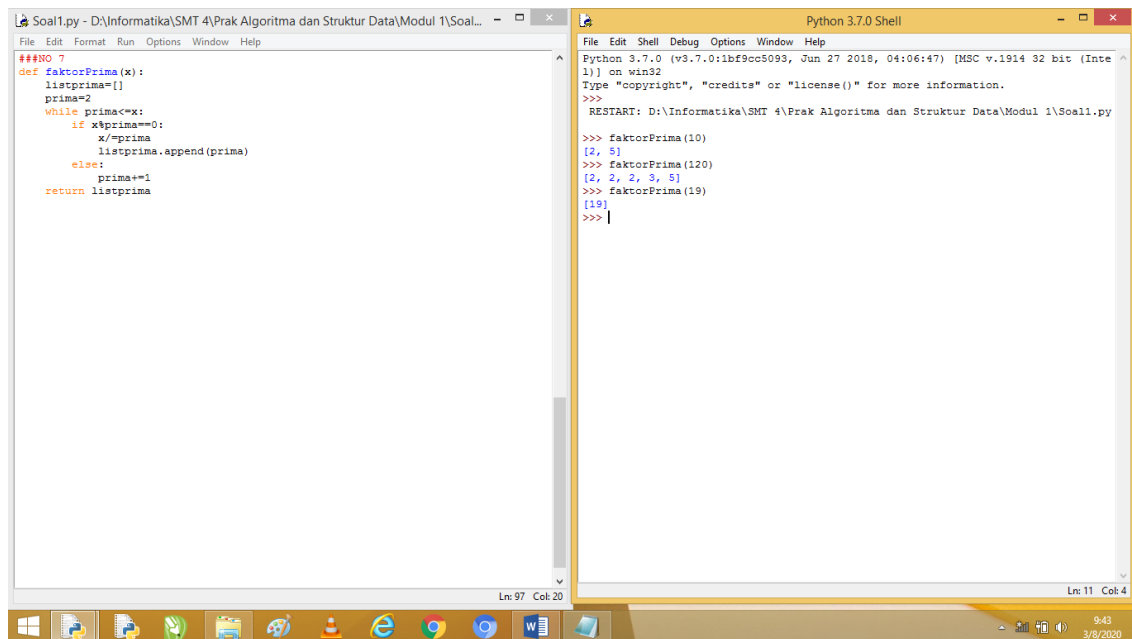


```
def cekPrima() :
    y = range(1001)
    for i in range(2,1001) :
        x = 0
        for j in range(i) :
            if i % (j+1) == 0 :
                x += 1
        if x == 2 :
            print(i)
```

7. Buatlah suatu program yang menerima bilangan bulat positif dan memberikan faktorisasi primanya. Faktorisasi prima adalah pemfaktoran suatu bilangan bulat ke dalam bilangan-bilangan prima yang menjadi konstituennya. Contoh:

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

**Jawab:**



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

8. Buat suatu fungsi apakahTerkandung(a,b) yang menerima dua string a dan b, lalu menentukan apakah string a terkandung dalam string b. Eksekusinya seperti ini:

```
>>> h = 'do'
>>> k = 'Indonesia tanah air beta'
>>> apakahTerkandung(h,k)
True
>>> apakahTerkandung('pusaka',k)
False
>>>
```

**Jawab:**

```
###NO 8
def apakahTerkandung(a,b) :
    if a in b :
        return True
    else :
        return False
```

```
>>> h = 'do'
>>> k = 'Indonesia tanah air beta'
>>> apakahTerkandung(h,k)
True
>>> apakahTerkandung('pusaka',k)
False
>>>
```

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

if a in b :

return True

else :

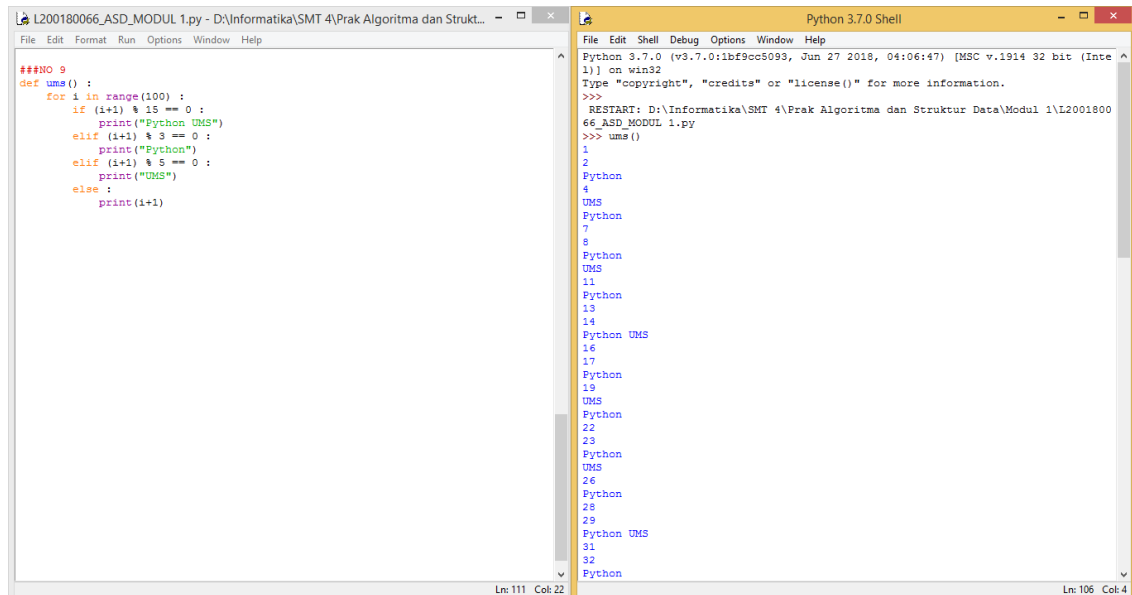
return False

9. Buat program untuk mencetak angka dari 1 sampai 100. Kalau angkanya pas kelipatan 3, cetak 'Python'. Kalau pas kelipatan 5, cetak 'UMS'. Kalau pas kelipatan 3 sekaligus kelipatan 5, cetak 'Python UMS'. Jadi hasilnya:

```
1
2
Python
4
UMS
Python
7
8
Python
UMS
11
Python
13
14
Python UMS
16
17
```

...

**Jawab:**



```
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\Informatika\SMT 4\Prak Algoritma dan Struktur Data\Modul 1\L200180066_ASD_MODUL 1.py
>>> ums()
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
```



```
def ums() :
    for i in range(100) :
        if (i+1) % 15 == 0 :
            print("Python UMS")
        elif (i+1) % 3 == 0 :
            print("Python")
        elif (i+1) % 5 == 0 :
            print("UMS")
        else :
            print(i+1)
```

10. Buat modifikasi pada contoh 1.4 agar bisa menangkap kasus di mana determinannya kurang dari nol. Jika ini terjadi, tampilkan peringatan di layar seperti ini:

```
>>> selesaikanABC(1,2,3)
Determinannya negatif. Persamaan tidak mempunyai akar real.
>>> |
```

**Jawab:**

Berikut adalah program yang saya buat:

```
def selesaikanABC(a,b,c) :
    det = (b**2) - (4*a*c)
```

if det == 0 :

print("Determinannya nol. Persamaan mempunyai satu akar kembar.")

elif det > 0 :

print("Determinannya positif. Persamaan mempunyai akar real dan berlainan.")

elif det < 0 :

print("Determinannya negatif. Persamaan tidak mempunyai akar real.")

11. Buat suatu fungsi apakahKabisat() yang menerima suatu angka (tahun). Jika tahun itu kabisat, kembalikan True. Jika bukan kabisat, kembalikan False.

Tahun kabisat – tahun yang memiliki tanggal 29 Februari – adalah tahun yang habis dibagi 4, kecuali dia habis dibagi 100 (maka dia bukan tahun kabisat). Tapi kalau dia habis dibagi 400, dia adalah tahun kabisat (meski habis dibagi 100).

Berikut adalah beberapa contoh:

- 1896 tahun kabisat (habis dibagi 4)
- 1897 bukan tahun kabisat (sudah jelas)
- 1900 bukan tahun kabisat (meski habis dibagi 4 tapi habis dibagi 100 dan tidak habis dibagi 400)
- 2000 tahun kabisat (habis dibagi 400)
- 2004, 2008, 2012, 2016, ..., 2096 tahun kabisat
- 2100, 2200, 2300 bukan tahun kabisat
- 2400 tahun kabisat

**Jawab:**

```
##NO.11
def apakahKabisat() :
    thn = int(input("Masukkan Tahun : "))
    if thn % 4 == 0 :
        if thn % 100 == 0 :
            if thn % 400 == 0 :
                return True
            else :
                return False
        else :
            return True
    else :
        return False

>>> apakahKabisat()
Masukkan Tahun : 1896
True
>>> apakahKabisat()
Masukkan Tahun : 1897
False
>>> apakahKabisat()
Masukkan Tahun : 1900
False
>>> apakahKabisat()
Masukkan Tahun : 2000
True
>>> apakahKabisat()
Masukkan Tahun : 2004
True
>>> apakahKabisat()
Masukkan Tahun : 2008
True
>>> apakahKabisat()
Masukkan Tahun : 2012
True
>>> apakahKabisat()
Masukkan Tahun : 2016
True
>>> apakahKabisat()
Masukkan Tahun : 2096
True
>>> apakahKabisat()
Masukkan Tahun : 2100
False
>>> apakahKabisat()
Masukkan Tahun : 2400
True
>>>
```

```

def apakahKabisat() :
    thn = int(input("Masukkan Tahun : "))
    if thn % 4 == 0 :
        if thn % 100 == 0 :
            if thn % 400 == 0 :
                return True
            else :
                return False
        else :
            return True
    else :
        return
False

```

12. Program permainan tebak angka. Buat program yang alurnya secara global seperti ini.

- Komputer membangkitkan bilangan bulat random antara 1 sampai 100. Nilainya disimpan di suatu variable dan tidak ditampilkan ke pengguna.
- Pengguna diminta menebak angka itu, diinputkan lewat keyboard.
- Jika angka yang diinputkan terlalu kecil atau terlalu besar, pengguna mendapatkan umpan balik dari computer (“Angka itu terlalu kecil. Coba lagi”)
- Proses diulangi sampai angka itu tertebak atau sampai sekian tebakan meleset.

Ketika programnya dilarikan, prosesnya kurang lenih seperti di bawah ini:

Permainan tebak angka.

Saya menyimpan sebuah angka bulat antara 1 sampai 100. Coba tebak

Masukkan tebakan ke-1:> 50

Itu terlalu kecil. Coba lagi

Masukkan tebakan ke-2:> 75

Itu terlalu besar. Coba lagi

Masukkan tebakan ke-3:> 58

Ya. Anda benar.

**Jawab:**

The screenshot shows a Python IDE with two windows. The left window, titled 'L200180066\_ASD\_MODUL 1.py', contains the following code:

```
#####NO 12
def tebak() :
    a = random.randrange(1,101)
    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 right window, titled 'Python 3.7.0 Shell', shows the execution output:

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\Informatika\SMT 4\Prak Algoritma dan Struktur Data\Modul 1\L200180066_ASD_MODUL 1.py
>>> tebak()
Permainan tebak angka.
Saya menyimpan sebuah angka bulat antara 1 sampai 100. Coba tebak
Masukkan tebakan ke-1:> 50
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-2:> 75
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-3:> 58
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-4:> 70
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-5:> 71
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-6:> 72
Ya. Anda benar.
>>>
```

```
def tebak() :
    a = random.randrange(1,101)
    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
```

13. Buat suatu fungsi katakana() yang menerima bilangan bulat positif dan mengembalikan suatu string yang merupakan pengucapan angka itu dalam Bahasa Indonesia. Contoh:

```
>>> katakan(3125750)
'tiga juta seratus dua puluh lima ribu tujuh ratus lima puluh '
```

**Jawab:**

The screenshot shows a Python IDE with two windows. The left window, titled 'L200180066\_ASD\_MODUL 1.py', contains a function `katakan(x)` that converts a number to its Indonesian word representation. The right window, titled 'Python 3.7.0 Shell', shows the execution of the function for various inputs, including 11, 17, 20, 45, 189, 206, 5467, 690500, and 690500000.

```
def katakan(x):
    satuan = [' ', 'satu', 'dua', 'tiga', 'empat', 'lima', 'enam', 'tujuh', 'delapan', 'sembilan', 'sepuluh', 'sebelas']
    hasil = ""
    if x < 0:
        hasil += 'Bilangan Haruslah Positif\ndan Bilangan Asli'
    elif x < 12:
        hasil += satuan[x]
    elif x < 20:
        hasil += katakan(x-10) + " belas "
    elif x < 100:
        hasil += katakan(int(x/10)) + " puluh " + katakan(x%10)
    elif x < 200:
        hasil += "seratus " + katakan(x-100)
    elif x < 1000:
        hasil += katakan(int(x/100)) + " ratus " + katakan(x%100)
    elif x < 2000:
        hasil += "seribu " + katakan(x-1000)
    elif x < 1000000:
        hasil += katakan(int(x/1000)) + " ribu " + katakan(x%1000)
    elif x < 1000000000:
        hasil += katakan(int(x/1000000)) + " juta " + katakan(x%1000000)
    elif x >= 1000000000:
        hasil += katakan(int(x/1000000000)) + " milyar " + katakan(x%1000000000)
    return hasil
```

```
Python 3.7.0 Shell
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> RESTART: D:\Informatika\SMT 4\Prak Algoritma dan Struktur Data\Modul 1\L200180066_ASD_MODUL 1.py
>>> katakan(11)
'tiga puluh satu'
>>> katakan(17)
'tujuh belas'
>>> katakan(20)
'dua puluh'
>>> katakan(45)
'empat puluh lima'
>>> katakan(189)
'seratus delapan puluh sembilan'
>>> katakan(206)
'dua ratus enam'
>>> katakan(5467)
'lima ribu empat ratus enam puluh tujuh'
>>> katakan(690500)
'enam ratus sembilan puluh ribu lima ratus'
>>>
```

```
def katakan(x):

    satuan = [' ', 'satu', 'dua', 'tiga', 'empat', 'lima', 'enam', 'tujuh', 'delapan', 'sembilan', 'sepuluh', 'sebelas']

    hasil = ""

    if x < 0:

        hasil += 'Bilangan Haruslah Positif\ndan Bilangan Asli'

    elif x < 12 :

        hasil += satuan[x]

    elif x < 20 :

        hasil += katakan(x-10) + " belas "

    elif x < 100:

        hasil += katakan(int(x/10)) + " puluh " + katakan(x%10)

    elif x < 200 :

        hasil += "seratus " + katakan(x-100)

    elif x < 1000 :

        hasil += katakan(int(x/100)) + " ratus " + katakan(x%100)

    elif x < 2000 :

        hasil += "seribu " + katakan(x-1000)

    elif x < 1000000 :

        hasil += katakan(int(x/1000)) + " ribu " + katakan(x%1000)

    elif x < 1000000000 :

        hasil += katakan(int(x/1000000)) + " juta " + katakan(x%1000000)
```

elif x >= 1000000000 :

hasil += katakan(int(x/1000000000)) + " milyar " + katakan(x%1000000000)

return hasil

14. Buat suatu fungsi formatRupiah() yang menerima suatu bilangan bulat positif dan mengembalikan suatu string yang merupakan bilangan itu tapi dengan 'format rupiah'.

Contoh:

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

Jawab:

```
##NO 14
def formatRupiah(a) :
    a = list(str(a))
    b = len(a)
    if b % 3 == 0 :
        b = int(b/3) - 1
    else :
        b = int(b/3)
    n = 0
    for i in range(b) :
        x = -3*(i+1)
        a.insert(int(x)+n, ".")
        n = n - 1
    a = "".join(a)
    print("Rp "+a)
```

```
RESTART: D:\Informatika\SMI 4\Prak Algoritma dan Struktur Data\Modul 1\L2001800
66 ASD MODUL 1.py
>>> formatRupiah(1500)
Rp 1.500
>>> formatRupiah(2560000)
Rp 2.560.000
>>> formatRupiah(721000)
Rp 721.000
>>> formatRupiah(75500000)
Rp 75.500.000
>>>
```

def formatRupiah(a) :

a = list(str(a))

b = len(a)

if b % 3 == 0 :

b = int(b/3) - 1

else :

b = int(b/3)

n = 0

for i in range(b) :

x = -3\*(i+1)

a.insert(int(x)+n, ".")

n = n - 1

a = "".join(a)

print("Rp "+a)