

AININ MAYASYIFA ALDA
L200180195
G

MODUL 1

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

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

- Kode Program

```
#1  
def cetakSiku(x):  
    for i in range (1,x+1):  
        print ('*' * i)
```

- Hasil jika di Run Module

```
>>> cetakSiku(5)  
*  
**  
***  
****  
*****
```

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

```
>>> gambarlahPersegiEmpat(4,5)      #tombol <enter> dipencet
```

```
@ @ @ @ @  
@       @  
@       @  
@ @ @ @ @
```

- Kode Program

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

- Hasil jika di Run Module :

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

3. Berikut ini adalah dua soal yang saling berkaitan

- a. Berikut adalah 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 vokal (huruf vokal adalah huruf hidup) di string itu.

- Kode Program

```
#3a
def jumlahHurufVokal (huruf):
    vokal = 'aiueoAIUEO'
    a = 0
    hasil = 0
    for i in huruf:
        if i in vokal:
            a += len(i)
        else:
            a += 0
    hasil = len(huruf), a
    return hasil
```

- Hasil jika di Run Module : #dimana sembilan adalah jumlah huruf dan 4 adalah jumlah huruf vokalnya

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

- b. Sama dengan soal (a) di atas, tapi sekarang yang dihitung adalah huruf konsonan. Hanya ada satu baris yang berbeda di dalam kodenya!

- Kode Program

```
#3b
def jumlahHurufKonsonan (huruf):
    konsonan = 'bcd fghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ'
    b = 0
    hasil = 0
    for i in huruf:
        if i in konsonan:
            b +=len(i)
        else:
            b += 0
    hasil = len(huruf),b
    return hasil
```

- Hasil jika di Run Module

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

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

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 :

- Kode Program

```
#4
def rerata(b):
    return sum(b)/len(b)
```

- Hasil jika di Run Module

```
>>> 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
```

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

- Kode Program

```
#5
from math import sqrt as sq
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
```

- Hasil jika di Run Module

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

6. Buatlah suatu program yang mencetak semua bilangan prima dari 2 sampai 1000. Manfaatkan fungsi apakahPrima() pada nomor diatas.

- Kode Program

```
#6
def bilanganPrima(n):
    for i in range(2,n):
        prima = True
        for j in range(2,i):
            if(i%j==0):
                prima = False
        if (prima):
            print(i)
```

- Hasil jika di Run Module

```
>>> 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
149
151
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
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
641
```

```
643
647
653
659
661
673
677
683
691
701
709
719
727
733
739
743
751
757
761
769
773
787
797
809
811
821
823
827
829
839
853
857
859
863
877
881
883
887
907
```

```
911
919
929
937
941
947
953
967
971
977
983
991
997
>>> |
```

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

- Kode Program

```
#7
def faktorPrima(x):
    bilanganList = []
    loop = 2
    while loop <= x:
        if x%loop == 0:
            x/=loop
            bilanganList.append(loop)
        else:
            loop +=1
    return bilanganList
```

- Hasil jika di Run Module

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

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

- Kode Program

```
#8
def apakahTerkandung(a,b):
    x = True
    for i in range(len(b)):
        if a in b:
            x = True
        else:
            x = False
    return x
```

- Hasil jika di Run Module

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

9. Buatlah 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'.

- Kode Program

```
#9
def kelipatan(x):
    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)
```

- Hasil jika di Run Module

```
>>> kelipatan(17)
1
2
Python
4
UMS
Python
7
8
Python
UMS
11
Python
13
14
Python UMS
16
```

10. Buat modifikasi pada contoh 1.4, agar bisa menangkap kasus dimana determinannya kurang dari nol. Jika ini terjadi, tampilkan peringatan di layar seperti ini: Determinannya negatif. Persamaan tidak mempunyai akar real.

- Kode Program

```
#10
from math import sqrt as akar
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 = "Determinannya negatif, persamaan tidak mempunyai akar real."
        return hasil
    else:
        x1 = (-b + akar(D))/(2*a)
        x2 = (-b - akar(D))/(2*a)
        hasil = (x1,x2)
        return hasil
```

- Hasil jika di Run Module

```
>>> selesaikanABC(1,2,3)
'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).

- Kode Program

```
#11
def apakahKabisat(tahun):
    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
```

- Hasil jika di Run Module

```
>>> apakahKabisat(1896)
True
>>> apakahKabisat(1900)
False
```

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

- Komputer membangkitkan bilangan bulat random antara 1 sampai 100. Nilainya di simpan disuatu variabel 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 komputer (“ Angka itu terlalu kecil. Coba lagi”)
- Proses diulangi sampai angka itu tertebak atau sampai sekian tebakan meleset⁸.

- Kode Program

```
#12
import random
def tebak():
    max = 7
    start = 1
    x = random.randrange(1,100,1)
    while (start <= max):
        s = 'Masukkan tebakan ke- ' +str(start)+ ':> '
        i = input(s)
        if(i == x):
            print ('Ya, Anda Benar')
        elif(i > x):
            print ('Itu Terlalu Besar, Coba Lagi')
        elif(i < x):
            print ('Itu Terlalu Kecil, Coba Lagi')
        start +=1
```

- Hasil jika di Run Module

```
>>> tebak()
masukan tebakan ke- 1:> 50
Itu Terlalu Besar. Coba lagi
masukan tebakan ke- 2:> 75
Itu Terlalu Besar. Coba lagi
masukan tebakan ke- 3:> 58
Itu Terlalu Besar. Coba lagi
masukan tebakan ke- 4:> 40
Itu Terlalu Besar. Coba lagi
masukan tebakan ke- 5:> 20
Itu Terlalu Besar. Coba lagi
masukan tebakan ke- 6:> 10
Itu Terlalu Besar. Coba lagi
masukan tebakan ke- 7:> 5
ya, Anda Benar
>>>
```

13. Buat suatu fungsi katakan() yang menerima bilangan bulat positif dan mengembalikan suatu string yang merupakan pengucapan angka itu dalam Bahasa Indonesia. Batasi inputnya agar lebih kecil dari satu milyar. *Extra credit*: gunakan rekursi.

- Kode Program

```
#13
def katakan(bilangan):
    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 < 100000000):
        Hasil = ' Satu Milyar' + katakan(n % 100000000)
    else:
        Hasil = 'Angka hanya sampai satu milyar'
    return Hasil
```

- Hasil jika di Run Module

```
>>> katakan(3125750)
' Tiga Juta Seratus Dua Puluh Lima Ribu Tujuh Ratus Lima Puluh '
```

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

- Kode Program

```
#14
def formatRupiah(bilangan):
    y = str(bilangan)
    if len(y) <= 3:
        return ('Rp ' + y)
    else:
        p = y[-3:]
        q = y[:-3]
        return formatRupiah(q) + '.' + p
    print ('Rp' + formatRupiah(q) + '.' + p)
```

- Hasil jika di Run Module

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