



Pemrograman Berbasis Fungsi - RA

TA Genap 2021/2022

Lecturer: Riksa Meidy Karim , S.Kom. , M.Si., M.Sc.

NAMA : Shahnaz Salsabila Ishak

NIM : 120450065

Tugas Exercise

>> Exercise 1 >>

1.a. Bantulah user tersebut dengan membuatkan sebuah program yang secara otomatis mengubah password yang di input menjadi terenkripsi sesuai aturan tersebut!

```
1 def start_program():
2     while True:
3         selection = input("""Silahkan input value sesuai kebutuhan anda\nInput 'E' untuk melakukan enkripsi\nInput 'D' untuk melakukan dekripsi\nInput 'Q' untuk menghentikan progra
4         if selection == 'E':
5             inpass = input("Masukkan Password yang ingin dienkrpsi: ")
6             if len(inpass) <= 100:
7                 print(encrypt(inpass))
8             else:
9                 print('Tidak boleh melebihi 100 karakter')
10                continue
11        elif selection == 'D':
12            inpass = input("Masukkan Password yang ingin didekripsi: ")
13            try:
14                print(decrypt(inpass))
15            except:
16                print('Password anda tidak tidak sesuai, dimohon untuk periksa kembali')
17                continue
18        elif selection == 'Q':
19            break
20        else:
21            print('Invalid Input')
22            continue
23
24 def joinlist(list):
25     return ''.join([i for i in list])
26
27 def lol_to_list(list): #elemen list dalam list digabung jadi list biasa
28     return [j for i in list for j in i]
29
30 def chartoascii(pwd):
31     listchar = list(pwd)
32     return [ord(char) for char in listchar]
```

```

33
34 def asciitochar(pwd):
35     return [[chr(pwd[i]),chr(pwd[i+1]), pwd[i+2]] for i in range(0,len(pwd),3)]
36
37 def val_calc(num):
38
39     firstval = num//26 + 80
40     secondval = num%26 + 80
41     if firstval > secondval:
42         thirdval = '+'
43     else:
44         thirdval = '-'
45
46     return [firstval, secondval, thirdval]
47
48 def encrypt(passwordd):
49     asciilist = char_to_ascii(passwordd)
50     firstencrypt = list(map(val_calc, asciilist))
51     encryptednum_list = lol_to_list(firstencrypt)
52     encryptedpass_list = asciitochar(encryptednum_list)
53
54     return joinlist(lol_to_list(encryptedpass_list))
55
56 -----
57
58 def split3val(pwd):
59     return [[pwd[i],pwd[i+1]] for i in range(0, len(pwd), 3)]
60
61 def val_decalc(numlist):
62     return [26 * (ord(i[0]) - 80) + (ord(i[1]) - 80) for i in numlist]
63
64 def decrypt(passwordd):
65     splittedpwd = split3val(passwordd)
66     asciipass = val_decalc(splittedpwd)
67
68     return joinlist([chr(i) for i in asciipass])
69
70 start_program()

```

1.b. Apa output yang dihasilkan dari program tersebut jika input password adalah 'anakanakcerdas2020' ?

Hasil enkripsi : Sc-TS+T]-Sc-TV-Sc-TS+Se-Sg-TZ-Sf-Sc-T[-Qh-Qf-Qh-Qf-

```

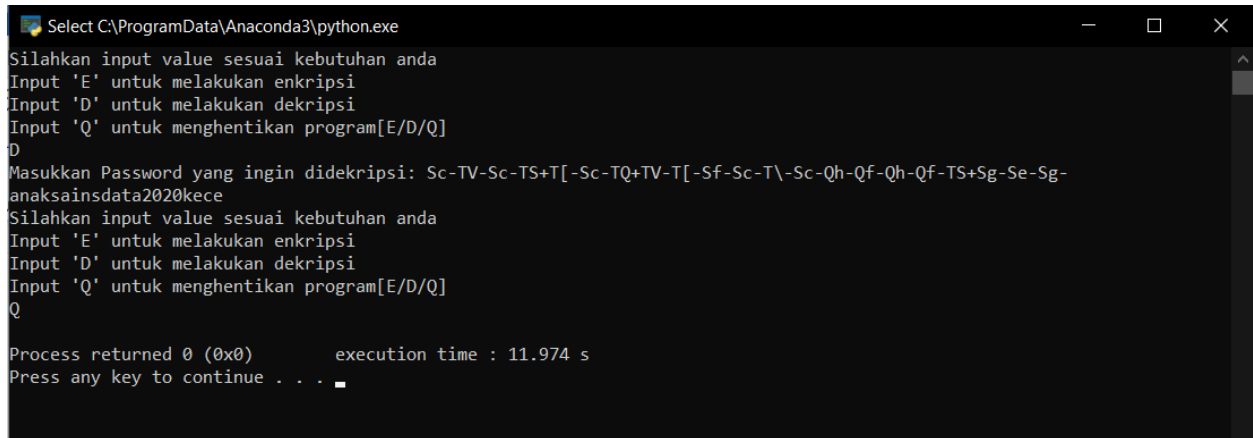
C:\ProgramData\Anaconda3\python.exe
Silahkan input value sesuai kebutuhan anda
Input 'E' untuk melakukan enkripsi
Input 'D' untuk melakukan dekripsi
Input 'Q' untuk menghentikan program[E/D/Q]
E
Masukkan Password yang ingin dienkrpsi: akuanakcerdas2020
Sc-TS+T]-Sc-TV-Sc-TS+Se-Sg-TZ-Sf-Sc-T[-Qh-Qf-Qh-Qf-
Silahkan input value sesuai kebutuhan anda
Input 'E' untuk melakukan enkripsi
Input 'D' untuk melakukan dekripsi
Input 'Q' untuk menghentikan program[E/D/Q]
Q
Process returned 0 (0x0)          execution time : 8.614 s
Press any key to continue . . .

```

1.c. (Bonus) User tersebut lupa password asli yang dia inputkan ke dalam program tersebut, password setelah dienkrpsi adalah 'Sc-TV-Sc-

TS+T[-Sc-TQ+TV-T[-Sf-Sc-T\ -Sc-Qh-Qf-Qh-Qf-TS+Sg-Se-Sg-'. Bantulah user tersebut mendapatkan password asli nya!

Hasil dekripsi : anaksainsdata2020kece



```
Select C:\ProgramData\Anaconda3\python.exe
Silahkan input value sesuai kebutuhan anda
Input 'E' untuk melakukan enkripsi
Input 'D' untuk melakukan dekripsi
Input 'Q' untuk menghentikan program[E/D/Q]
D
Masukkan Password yang ingin didekripsi: Sc-TV-Sc-TS+T[-Sc-TQ+TV-T[-Sf-Sc-T\ -Sc-Qh-Qf-Qh-Qf-TS+Sg-Se-Sg-
anakainsdata2020kece
Silahkan input value sesuai kebutuhan anda
Input 'E' untuk melakukan enkripsi
Input 'D' untuk melakukan dekripsi
Input 'Q' untuk menghentikan program[E/D/Q]
Q
Process returned 0 (0x0)          execution time : 11.974 s
Press any key to continue . . .
```

>> Exercise 2 >>

Selesaikan penjumlahan 100 digit tersebut dengan membuat program dengan langkah sebagai berikut:

1. Buat file txt untuk bilangan pertama
2. Buat file txt untuk bilangan kedua
3. Input file txt bilangan pertama dan bilangan kedua
4. Buatlah program untuk menghitung penjumlahan kedua bilangan tersebut

15. Implementasikan konsep fungsi yang telah dipelajari ke dalam pembuatan program tersebut! Sertakan juga screenshot hasil program nya!

```
num1.txt
1 9502561694858652150281747994108545943651521215096841995237040384498740803993469376602031341619585763
2

num2.txt
1 2116068642696162934965789080530992805391900568978958496201555855833896833372295507803936243187061092
2
```

```
1 firstnum = open('num1.txt').read()
2 secnum = open('num2.txt').read()
3
4 print((lambda x,y: int(x)+int(y))(firstnum,secnum))
5
```

Output Program :

```
C:\ProgramData\Anaconda3\python.exe
11618630337554815085247537074639538749043421784075800491438596240332637637365764884405967584806646855
Process returned 0 (0x0)      execution time : 0.071 s
Press any key to continue . . .
```

>> Exercise 3 >>

1. Apakah modul solver.py dapat digunakan untuk menyelesaikan persamaan differensial orde 2 selain kasus non linear pendulum? Jelaskan mengapa!

Jawab : Ya, karena metode yang digunakan untuk menghitung persamaan differensial orde 2 sama saja hanya berbeda di persamaan dan nilai yang digunakan.

2. Implementasikan solver tersebut dengan cara membuat file solver.py

```
solver.py
1 def euler(t,h,y,dy,Func):
2     d2y = Func(t,y,dy)
3     y_next = y + (h * dy)
4     dy_next = dy + (h * d2y)
5     return (y_next, dy_next)
6
7
8 def cauchy_euler(params, Func):
9     # initial condition
10    t0 = params['t0']
11    t_akhir = params['t_akhir']
12    h = params['h']
13    y0 = params['y0']
14    dy0 = params['dy0']
15
```

```
16 res_euler = []
17 t = []
18 step = int((t_akhir - t0) / h)
19
20 for i in range(step):
21     tm = (i + 1) * h
22     (y_next, dy_next) = euler(tm, h, y0, dy0, Func)
23     res_euler.append(y_next)
24     t.append(tm)
25     y0 = y_next
26     dy0 = dy_next
27
28 return(t, res_euler)
29
```

3. Untuk menyelesaikan persamaan dengan solver.py, bentuk fungsi harus diubah menjadi:

$$\frac{d^2\alpha}{dt^2} = -\frac{g}{L} * \sin(\alpha)$$

```
import math

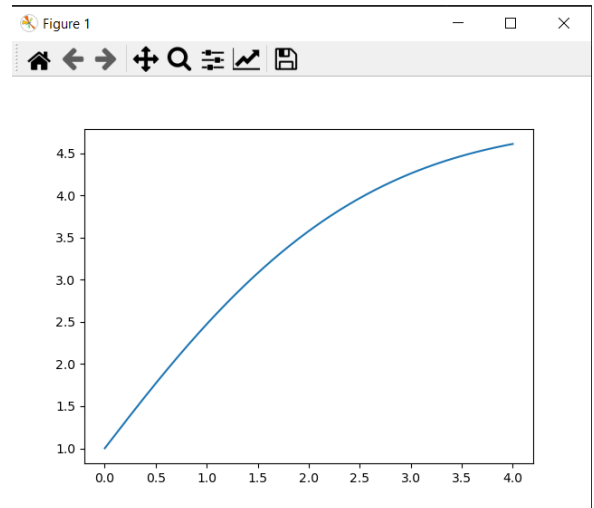
def pendulumfunc(g, L, a):
    return -(g/L) * math.sin(a)
```

4. Menggunakan Parameter Parameter yang ada dalam tabel diatas, buatlah program yang menggunakan solver.py untuk menemukan solusi persamaan diferensial non linear tersebut! Hint(Solusi Akhir berupa plot)

tugas3_pd1.py

```
1 from solver import *
2 import matplotlib.pyplot as plt
3 import math
4
5 def pendulumfunc(g, l, a):
6     return -(g/l) * math.sin(a)
7
8 parameter = {
9     'g' : 9.8,
10    'y0' : 1,
11    't0' : 0,
12    't_akhir' : 4,
13    'h' : 0.001,
14    'dy0' : 0.5 * 3.14
15 }
16
17
18 t, res = cauchy_euler(parameter, pendulumfunc)
19
20 plt.plot(t,res)
21 plt.show()
22
```

Output:

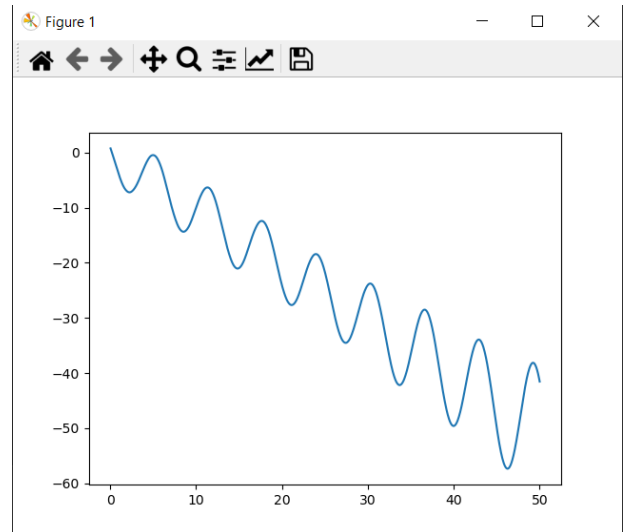


5. Perhatikan Persamaan Diferensial Diatas! Buatlah program untuk menyelesaikan PD tersebut dengan menggunakan solver.py sebagai modul dengan parameter berikut!

tugas3_pd2.py

Output:

```
1 import matplotlib.pyplot as plt
2 from solver import *
3 import math
4
5 def pd2(y, dy, x):
6     return -y - dy - (math.sin(x) ** 2)
7
8 parameter1 = {
9     't0' : 0,
10    't_akhir' : 50,
11    'h' : 0.05,
12    'y0' : 1,
13    'dy0' : -9/2
14 }
15
16 x, y = cauchy_euler(parameter1, pd2)
17
18 plt.plot(x,y)
19 plt.show()
20
```



>> Exercise 4 >>

1. Identifikasikan Variabel Global yang dibutuhkan !

Variabel Global yang dibutuhkan:

- a. Time/waktu waktu permainan (t)
- b. Skor dari Tim A (A_score)
- c. Skor dari Tim B (B_score)
- d. Status bola untuk mengetahui letak keberadaan bola (ball)
- e. Status permainan untuk menandakan permainan sedang berlangsung atau memulai pertandingan baru (status)

2. Identifikasikan Event yang terjadi !

- a. Pertarungan antara kedua Midfielder dari masing-masing tim, Midfielder dengan poin skill paling tinggi yang akan berhasil merebut bola
- b. Pertarungan antara Attacker tim penyerang dan Defender bertahan tim. Jika skill dribble Attacker lebih besar daripada skill tackle Defender, maka bola akan tetap dipegang oleh Attacker, begitupun sebaliknya.
- c. Pertarungan antara Defender tim penyerang dan Attacker tim bertahan. Jika skill Intercept Attacker lebih besar daripada skill Passing Defender, maka bola akan berhasil direbut oleh Attacker, jika sebaliknya, maka Defender berhasil melewati Attacker dan mengoper bola ke Midfielder tim penyerang.
- d. Pertarungan antara Attacker tim penyerang dan Goalkeeper tim bertahan. Jika skill shooting Attacker lebih besar daripada skill save Goalkeeper, maka Attacker akan berhasil mencetak goal. Sebaliknya jika skill save Goalkeeper lebih besar, maka Goalkeeper akan berhasil menangkap bola dan mengopernya ke Defender.

3. Buatlah simulasi program tersebut menggunakan konsep paradigma fungsional dengan kondisi sebagai berikut:

a. Jumlah supporter Tim A = 100.000 orang

b. Jumlah supporter Tim B = 115.000 orang

c. Komposisi Pemain sebagai berikut:

Tim A			
Role	Mentality	Skill	Value
GK	80	Save	81
DF	79	Tackle	79
		Passing	78
MD	78	Tackle	60
		Dribble	76
ATK	77	Dribble	80
		Intercept	85
		Shoot	92

Tim B			
Role	Mentality	Skill	Value
GK	77	Save	86
DF	78	Tackle	80
		Passing	81
MD	79	Tackle	70
		Dribble	70
ATK	80	Dribble	81
		Intercept	86
		Shoot	90

Done! Dalam tugas4.py

4. Bagaimanakah hasil pertandingan tersebut? Berapa skor akhir pertandingan?

```
C:\ProgramData\Anaconda3\python.exe
Bola diterima oleh Midfielder A
Terjadi pertarungan antara Midfielder A dan Midfielder B
Bola berhasil dipertahankan oleh Midfielder A dan dioper menuju Attacker A
Attacker A berhasil melewati Defender B
Attacker A akan melakukan shoot
Attacker A berhasil melakukan shoot dan mencetak gol
Score sementara adalah
Team A : 1
Team B : 0

Bola diterima oleh Midfielder B
Terjadi pertarungan antara Midfielder A dan Midfielder B
Bola berhasil dipertahankan oleh Midfielder B dan dioper menuju Attacker B
Attacker B gagal melewati Defender A
Defender A berhasil merebut bola!
Defender A berhasil melakukan pass ke Midfielder A!
Bola diterima oleh Midfielder A
Terjadi pertarungan antara Midfielder A dan Midfielder B
Bola berhasil dipertahankan oleh Midfielder A dan dioper menuju Attacker A
Attacker A gagal melewati Defender B
Defender B berhasil merebut bola!
Defender B gagal melakukan pass ke Midfielder B!
Bola direbut Attacker A dan bersiap melakukan shoot
Attacker A berhasil melakukan shoot dan mencetak gol
Score sementara adalah
Team A : 2
Team B : 0
```

```
Bola diterima oleh Midfielder B
Terjadi pertarungan antara Midfielder A dan Midfielder B
Bola berhasil dipertahankan oleh Midfielder B dan dioper menuju Attacker B
Attacker B gagal melewati Defender A
Defender A berhasil merebut bola!
Defender A gagal melakukan pass ke Midfielder A!
Bola direbut Attacker B dan bersiap melakukan shoot
Attacker B berhasil melakukan shoot dan mencetak gol
Score sementara adalah
Team A : 2
Team B : 1
```

```
Bola diterima oleh Midfielder A
Terjadi pertarungan antara Midfielder A dan Midfielder B
Bola gagal dipertahankan oleh Midfielder A
Bola diambil alih oleh Midfielder B dan dioper menuju Attacker B
Attacker B berhasil melewati Defender A
Attacker B akan melakukan shoot
Attacker B berhasil melakukan shoot dan mencetak gol
Score sementara adalah
Team A : 2
Team B : 2
```

```
Bola diterima oleh Midfielder A
Terjadi pertarungan antara Midfielder A dan Midfielder B
Bola berhasil dipertahankan oleh Midfielder A dan dioper menuju Attacker A
Attacker A berhasil melewati Defender B
Attacker A akan melakukan shoot
Keeper B berhasil menangkap bola dan bola dioper menuju Defender B
Defender B gagal melakukan pass ke Midfielder B!
Bola direbut Attacker A dan bersiap melakukan shoot
Keeper B berhasil menangkap bola dan bola dioper menuju Defender B
Bola diterima oleh Midfielder B
Terjadi pertarungan antara Midfielder A dan Midfielder B
Bola berhasil dipertahankan oleh Midfielder B dan dioper menuju Attacker B
Attacker B berhasil melewati Defender A
Attacker B akan melakukan shoot
Attacker B berhasil melakukan shoot dan mencetak gol
Score sementara adalah
Team A : 2
Team B : 3
```

```
Bola diterima oleh Midfielder A
Terjadi pertarungan antara Midfielder A dan Midfielder B
Bola gagal dipertahankan oleh Midfielder A
Bola diambil alih oleh Midfielder B dan dioper menuju Attacker B
Attacker B gagal melewati Defender A
Defender A berhasil merebut bola!
Defender A gagal melakukan pass ke Midfielder A!
Bola direbut Attacker B dan bersiap melakukan shoot
Attacker B berhasil melakukan shoot dan mencetak gol
Score sementara adalah
Team A : 2
Team B : 4
```

```
Pertandingan telah berakhir!
-----
Score akhir adalah
Team A : 2
Team B : 4
Hasil akhir dari pertandingan kali ini dimenangkan oleh Team B
```

```
Process returned 0 (0x0)          execution time : 0.150 s
Press any key to continue . . .
```

Skor Akhirnya adalah 2 : 4

>> Exercise 7 >>

1. Given List P = ['a', 'k', 'u', 'l', 'u', 'p', 'a']

We want to make list of tuples of P like this

P' = [(1, 'a'), (3, 'k'), (5, 'u'), (7, 'l'), (9, 'u'), (11, 'p'), (13, 'a')]

```
[57]: def enumerasi(sequence):  
        return list(map(lambda i, l: (i, l), range(len(sequence)), list(sequence)))  
  
        enumerasi('akulupa')  
  
        [(0, 'a'), (1, 'k'), (2, 'u'), (3, 'l'), (4, 'u'), (5, 'p'), (6, 'a')]
```

2. Terdapat bilangan B

B = 24

Petakan B menjadi list faktor nya!

B' = [1, 2, 3, 4, 6, 8, 12, 24]

```
[83]: def factorof(angka):  
        return list(filter(lambda x: angka%x == 0, range(1, angka+1)))  
  
        factorof(24)  
  
        [1, 2, 3, 4, 6, 8, 12, 24]
```

3. Diketahui matriks A,B,C sebagai berikut

$A = \begin{bmatrix} 3 & 4 \\ 5 & 6 \end{bmatrix}$

$B = \begin{bmatrix} 1 & 2 \\ 7 & 8 \end{bmatrix}$

$C = AB$

Buatlah program untuk menghitung determinan matriks C menggunakan HOF map!

```
[82]: A = [ [3,4], [5,6]]
      B = [ [1,2], [7,8]]

      def normal_mult(A, B):
          return list(map(lambda ri, rj: list(map(lambda rii, rjj: rii+rjj, ri, rj)), A, B))

      def det(matrix):
          return matrix[0][0] * matrix[1][1] - matrix[0][1] * matrix[1][0]

      C = normal_mult(A,B)
      det(C)
```

-16