



# Pemrograman Berbasis Fungsi - RA

## TA Genap 2021/2022

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

NAMA : SANDY WIJAYA

NIM : 120450047

---

## Tugas Exercise

---

### >> Exercise 1 >>

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

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

1c. (Bonus) User tersebut lupa password asli yang dia inputkan ke dalam program tersebut, password setelah dienkripsi 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!

JAWAB =

The screenshot shows a Jupyter Notebook titled "PBF\_EXERCISE 1.ipynb". The code defines a function `encrypt_password` that takes a password and a shift value (26) and returns an encrypted password. It then demonstrates the encryption of "anakanakcerdas2020" and the decryption of the resulting ciphertext "Sc-TV-Sc-TS+T[-Sc-TQ+TV-T[-Sf-Sc-T\Sc-Qh-Qf-Qh-Qf-TS+5g-Se-Sg-". The output shows the original password, the encrypted password, and the decrypted password.

```
[ ] nilai = 26 * kesatu + kedua
    asciivalue.append(nilai)

    password = ''
    for i in asciivalue:
        char = chr(i)
        password = password + char

    return password

print("Enkripsi Password:")
paswor = "anakanakcerdas2020"
print ("Original Password : " + paswor)
print ("Encrypted Password : " + encrypted_password(paswor))
# Output : Original Password : anakanakcerdas2020
# Output : Encrypted Password : Sc-TV-Sc-TS+T[-Sc-TQ+TV-T[-Sf-Sc-T\Sc-Qh-Qf-Qh-Qf-TS+5g-Se-Sg-

print("Get Original Password:")
password = "Sc-TV-Sc-TS+T[-Sc-TQ+TV-T[-Sf-Sc-T\Sc-Qh-Qf-Qh-Qf-TS+5g-Se-Sg-"
print ("Encrypted Password : " + password)
print ("Original Password : " + original_password(password))
# Output : Encrypted Password : Sc-TV-Sc-TS+T[-Sc-TQ+TV-T[-Sf-Sc-T\Sc-Qh-Qf-Qh-Qf-TS+5g-Se-Sg-
# Output : Original Password : anaksainsdata2020kece

Enkripsi Password:
Original Password : anakanakcerdas2020
Encrypted Password : Sc-TV-Sc-TS+T[-Sc-TQ+TV-T[-Sf-Sc-T\Sc-Qh-Qf-Qh-Qf-TS+5g-Se-Sg-
Get Original Password:
Encrypted Password : Sc-TV-Sc-TS+T[-Sc-TQ+TV-T[-Sf-Sc-T\Sc-Qh-Qf-Qh-Qf-TS+5g-Se-Sg-
Original Password : anaksainsdata2020kece
```

## >> 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

```
import numpy as np
a = open("text1.txt", 'w')
a.write("9502561694858652150281747994108545943651521215096841995237040384498740803993469376602031341619585763")
a.close()
aa = open("text1.txt", 'r')
f1=np.array(aa.read().split("\n")).astype(float)
ff1 = int(f1)

b = open("text2.txt", 'w')
b.write("2116068642696162934965789080530992805391900568978958496201555855833896833372295507803936243187061092")
b.close()
bb = open("text2.txt", 'r')
f2=np.array(bb.read().split("\n")).astype(float)
ff2 = int(f2)

f3 = ff1 + ff2
print(f3)
```

11618630337554814826348086790285854587429944447132304038458358061424824076107783714807302076307079168

## >> Exercise 3 >>

Code tersebut adalah metode euler yang dapat digunakan untuk menyelesaikan persamaan differensial non linear pendulum. Misal metode tersebut disimpan di modul bernama solver.py

Soal:

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

= Dapat, karena pada modul solver.py langkah awal untuk menyimpan fungsi sebelum main dan metode\_eurel.py di run, jika salah maka program tidak dapat berjalan atau error dan karena rumus diferensial orde 2 dapat menyelesaikan permasalahan selain non linear pendulum.

mengapa!

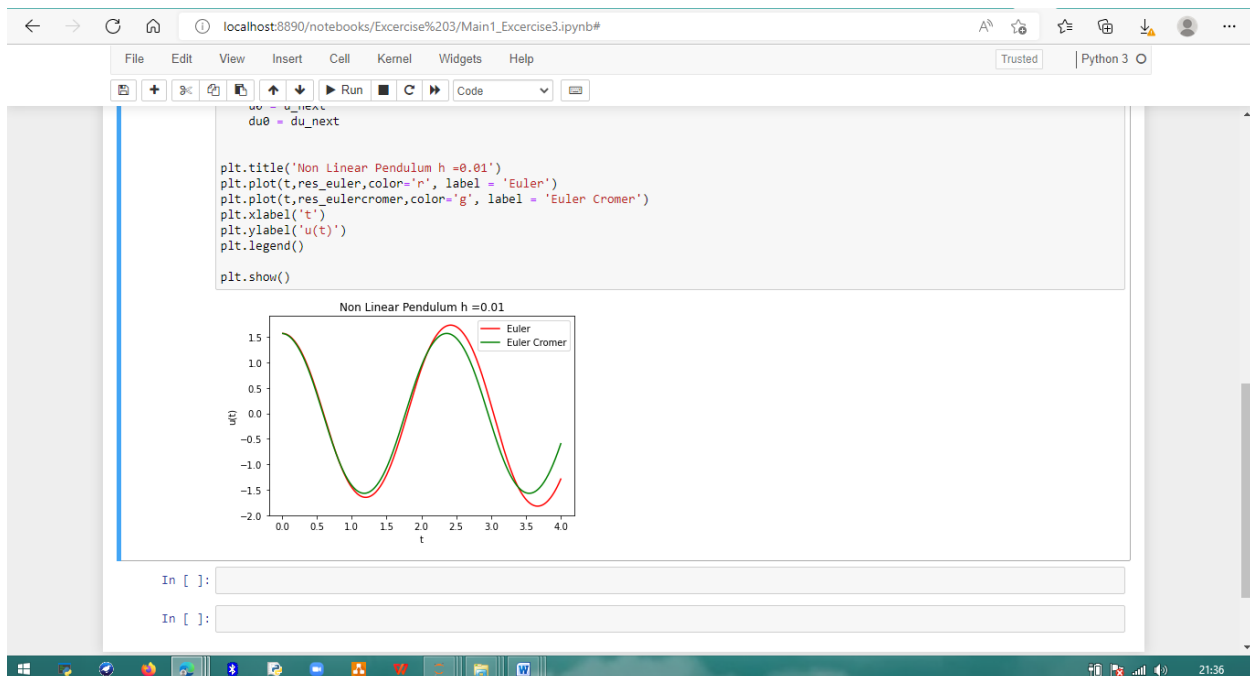
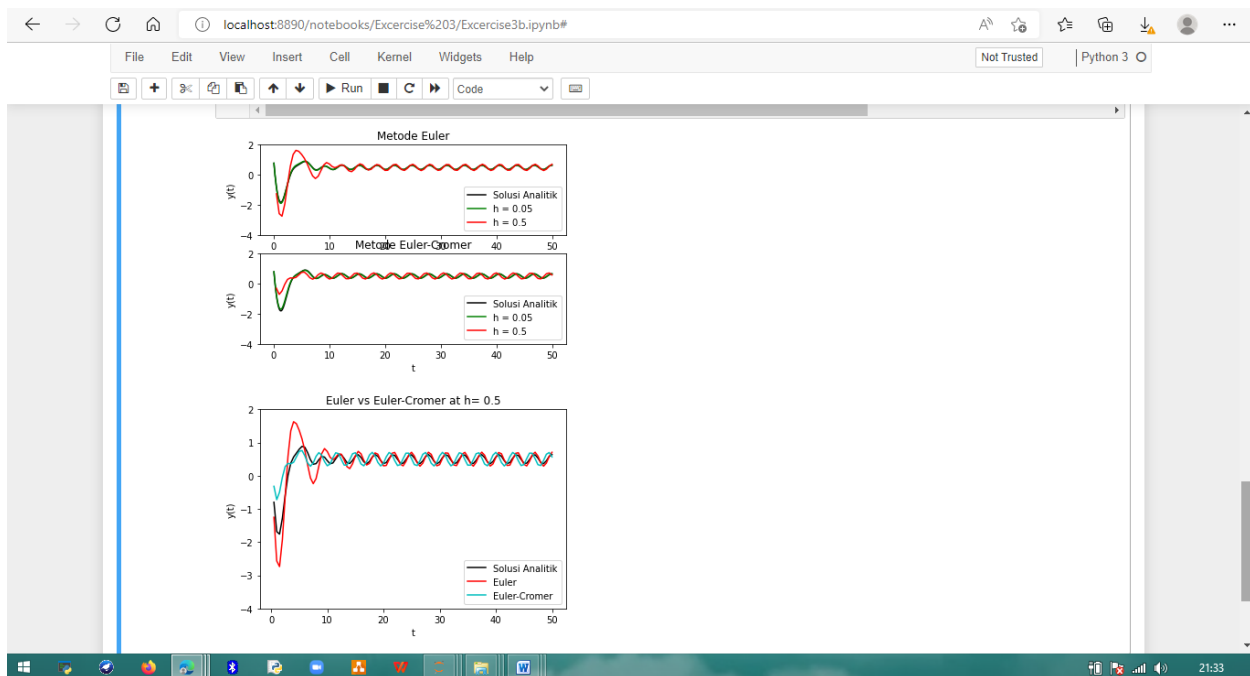
2. Implementasikan solver tersebut dengan cara

membuat file solver.py

```
1  
2 #!/usr/bin/env python  
3 # coding: utf-8  
4  
5 # In[4]:  
6  
7  
8 def euler(t,h,y,dy, Func):  
9     d2y = Func(t,y,dy)  
10    y_next = y + (h * dy)  
11    dy_next = dy + (h * d2y)  
12    return (y_next, dy_next)  
13  
14 def euler_cromer(t,h,y,dy,Func):  
15     d2y = Func(t, y, dy)  
16     dy_next = dy + (h * d2y)  
17     y_next = y + (h * dy_next)  
18     return (y_next, dy_next)  
19  
20  
21 def cauchy_euler(params, Func):  
22     #initial condition  
23     t0 = params['t0']  
24     t_akhir = params ['t_akhir']  
25     h = params['h']  
26     y0 = params['y0']  
27     dy0 = params ['dy0']  
28  
29     res_euler = []  
30     t = []  
31     step = int ((t-akhir - t0)/h)  
32
```

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

```
1 from solver import *  
2  
3  
4 def cauchy_euler(params,Func):  
5     # Initial Condition  
6     t0 = params['t0']  
7     t_akhir = params['t_akhir']  
8     h = params['h']  
9     y0 = params['y0']  
10    dy0 = params['dy0']  
11  
12    res_euler = []  
13    t = []  
14    step = int((t_akhir - t0) / h)  
15  
16    for i in range(step):  
17        tm = (i + 1) * h  
18        (y_next, dy_next) = euler(tm, h, y0, dy0, Func)  
19        res_euler.append(y_next)  
20        t.append(tm)  
21        y0 = y_next  
22        dy0 = dy_next  
23  
24    return (t,res_euler)  
25  
26 def cauchy_eulercromer(params,Func):  
27     # Initial Condition  
28     t0 = params['t0']  
29     t_akhir = params['t_akhir']  
30     h = params['h']  
31     y0 = params['y0']  
32     dy0 = params['dy0']
```



>> Exercise 4 >>

1. Identifikasikan Variabel Global yang dibutuhkan !
2. Identifikasikan Event yang terjadi. 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:
3. Bagaimanakah hasil pertandingan tersebut? Berapa skor akhir pertandingan?

```
for x in range(leagueSize):
    #Lots of formatting
    print(teamNames[x]+(15-len(teamNames[x]))*" "+" Skill: "+str(teamSkill[x])+(5-len(str(teamSkill[x]))*" "+" Points: "+str(teamPoints[x])+(5-len(str(teamPoints[x]))*" "+" For: '
    teamPoints.sort()
    print(teamPoints)
```

Enter number of teams in league: 2  
Enter team 1 name: Alfa  
Enter team 2 name: Beta  
Enter Alfa skill: 98  
Enter Beta skill: 97  
=====

Alfa's home games:  
=====

Alfa 2 - 1 Beta

=====

Beta's home games:  
=====

Beta 0 - 0 Alfa

Final table:

Alfa	Skill: 98	Points: 4	For: 2	Against: 1	Goal difference: 1	Wins: 1	Draws: 1	Losses: 0
Beta	Skill: 97	Points: 1	For: 1	Against: 2	Goal difference: -1	Wins: 0	Draws: 1	Losses: 1
	[1, 4]							

Note:

# new Page untuk setiap pertemuan

# Gunakan Header 2 untuk setiap soal , dan Header 1 untuk setiap pertemuan

# Jika ingin tulis tangan , sertakan gambar sesuai soal

# Selamat mengerjakan tugas