Phil Gandeng 11417001 11 Februari 2021

1. By using **bisection method**, find the root of the following function(s):

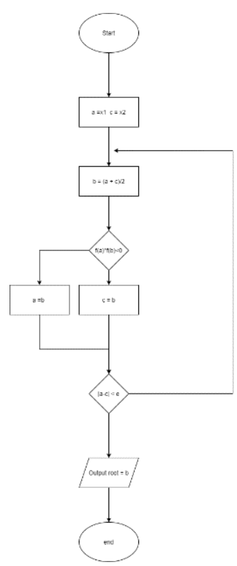
- $f(x) = x^2 - 3x - 2$
- $f(x) = x^3 + x^2 - 3x - 2$

2. A ball at 1200 K is allowed to cool down in air at an ambient temperature of 300 K. Assuming heat is lost only due to radiation, the differential equation for the temperature of the ball is given by:

$$\frac{d\theta}{dt} = -2.2067 \times 10^{-12} \left(\theta - 8.1 \times 10^{9}\right); \quad \theta(0) = 1200\text{K}$$

where θ is in K and t in seconds. Find the temperature at t = 480 s using RungeKutta 4th order method. Assume a step size of h are 30 s, 60 s, 120 s, dan 240 s.

1. Algoritma (Bisection Method)



2. Output Keluaran (Bisection Method)

```
import math


# Fungi yang akan digunakan
def f(x):
    return x**2 - 3*x -2


# Penentuan metode bisection
def bisection(x0,x1,e):
    step = 1


    condition = True
    while condition:
        x2 = (x0 + x1)/2
        print('Iterasi ke -%d, x2 = %0.6f dan f(x2) = %0.6f' % (step, x2, f(x2)))


        if f(x0) * f(x2) < 0:
            x1 = x2
        else:
            x0 = x2


        step = step + 1
        condition = abs(f(x2)) > e


    print('\nAkar yang dibutuhkan : %0.8f' % x2)



# Baris untuk memasukkan nilai
x0 = input('Titik Awal: ')
```

```
x1 = input('Titik Kedua: ')
e = input('Batasan error: ')

# Merubah nilai input menjadi float
x0 = float(x0)
x1 = float(x1)
e = float(e)

# Pengecekan awal nilai bisection
if f(x0) * f(x1) > 0.0:
    print('Nilai yang dimasukkan tidak masuk dalam kurungan.')
    print('Coba dengan nilai baru.')
else:
    bisection(x0,x1,e)
```

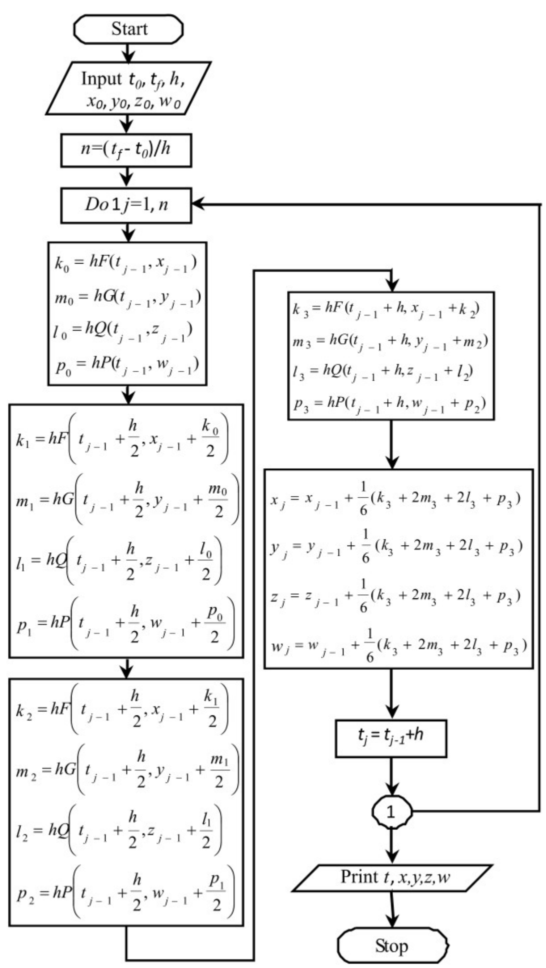
Untuk persamaan pertama outputnya:

```
Titik Awal: 2
Titik Kedua: 8
Batasan error: 10e-5
Iterasi ke -1, x2 = 5.000000 dan f(x2) = 8.000000
Iterasi ke -2, x2 = 3.500000 dan f(x2) = -0.250000
Iterasi ke -3, x2 = 4.250000 dan f(x2) = 3.312500
Iterasi ke -4, x2 = 3.875000 dan f(x2) = 1.390625
Iterasi ke -5, x2 = 3.687500 dan f(x2) = 0.535156
Iterasi ke -6, x2 = 3.593750 dan f(x2) = 0.133789
Iterasi ke -7, x2 = 3.546875 dan f(x2) = -0.060303
Iterasi ke -8, x2 = 3.570312 dan f(x2) = 0.036194
Iterasi ke -9, x2 = 3.558594 dan f(x2) = -0.012192
Iterasi ke -10, x2 = 3.564453 dan f(x2) = 0.011967
Iterasi ke -11, x2 = 3.561523 dan f(x2) = -0.000121
Iterasi ke -12, x2 = 3.562988 dan f(x2) = 0.005921
Iterasi ke -13, x2 = 3.562256 dan f(x2) = 0.002899
Iterasi ke -14, x2 = 3.561890 dan f(x2) = 0.001389
Iterasi ke -15, x2 = 3.561707 dan f(x2) = 0.000634
Iterasi ke -16, x2 = 3.561615 dan f(x2) = 0.000256
Iterasi ke -17, x2 = 3.561569 dan f(x2) = 0.000068

Akar yang dibutuhkan : 3.56156921
```

Untuk persamaan kedua outputnya:

```
Titik Awal: 0
Titik Kedua: 2
Batasan error: 10e-5
Iterasi ke -1, x2 = 1.000000 dan f(x2) = -3.000000
Iterasi ke -2, x2 = 1.500000 dan f(x2) = -0.875000
Iterasi ke -3, x2 = 1.750000 dan f(x2) = 1.171875
Iterasi ke -4, x2 = 1.625000 dan f(x2) = 0.056641
Iterasi ke -5, x2 = 1.562500 dan f(x2) = -0.431396
Iterasi ke -6, x2 = 1.593750 dan f(x2) = -0.193024
Iterasi ke -7, x2 = 1.609375 dan f(x2) = -0.069614
Iterasi ke -8, x2 = 1.617188 dan f(x2) = -0.006844
Iterasi ke -9, x2 = 1.621094 dan f(x2) = 0.024809
Iterasi ke -10, x2 = 1.619141 dan f(x2) = 0.008960
Iterasi ke -11, x2 = 1.618164 dan f(x2) = 0.001052
Iterasi ke -12, x2 = 1.617676 dan f(x2) = -0.002897
Iterasi ke -13, x2 = 1.617920 dan f(x2) = -0.000923
Iterasi ke -14, x2 = 1.618042 dan f(x2) = 0.000065

Akar yang dibutuhkan : 1.61804199
```

1. Algortima (Runge – Kutta)



2.

```
# RK-4 method python program

# function to be solved
def f(x,y):
    return (-2.2067 * (10**-12) * (y**4-81*(10**9)))

# or
# f = lambda x: x+y

# RK-4 method
def rk4(x0,y0,xn,n):

    # Calculating step size
    h = (xn-x0)/n

    print('\n--------SOLUTION--------')
    print('-------------------------')
    print('x0\ty0\tyn')
    print('-------------------------')
    for i in range(n):
        k1 = h * (f(x0, y0))
        k2 = h * (f((x0+h/2), (y0+k1/2)))
        k3 = h * (f((x0+h/2), (y0+k2/2)))
        k4 = h * (f((x0+h), (y0+k3)))
        k = (k1+2*k2+2*k3+k4)/6
        yn = y0 + k
        print('%.4f\t%.4f\t%.4f'% (x0,y0,yn) )
        print('-------------------------')
```

```
        y0 = yn
        x0 = x0+h

    print('\nAt x=%.4f, y=%.4f' %(xn,yn))

# Inputs
print('Enter initial conditions:')
x0 = float(input('x0 = '))
y0 = float(input('y0 = '))

print('Enter calculation point: ')
xn = float(input('xn = '))

print('Enter number of steps:')
step = int(input('Number of steps = '))

# RK4 method call
rk4(x0,y0,xn,step)
```

Hasil Output :

- Ketika step = 30

```
Enter initial conditions:
x0 = 0
y0 = 1200
Enter calculation point:
xn = 480
Enter number of steps:
Number of steps = 30


At x=480.0000, y=684.7349
```

- Ketika step = 120

```
Enter initial conditions:
x0 = 0
y0 = 1200
Enter calculation point:
xn = 480
Enter number of steps:
Number of steps = 120


At x=480.0000, y=533.8006
```