

Nama : Aditya Putra Prastyo

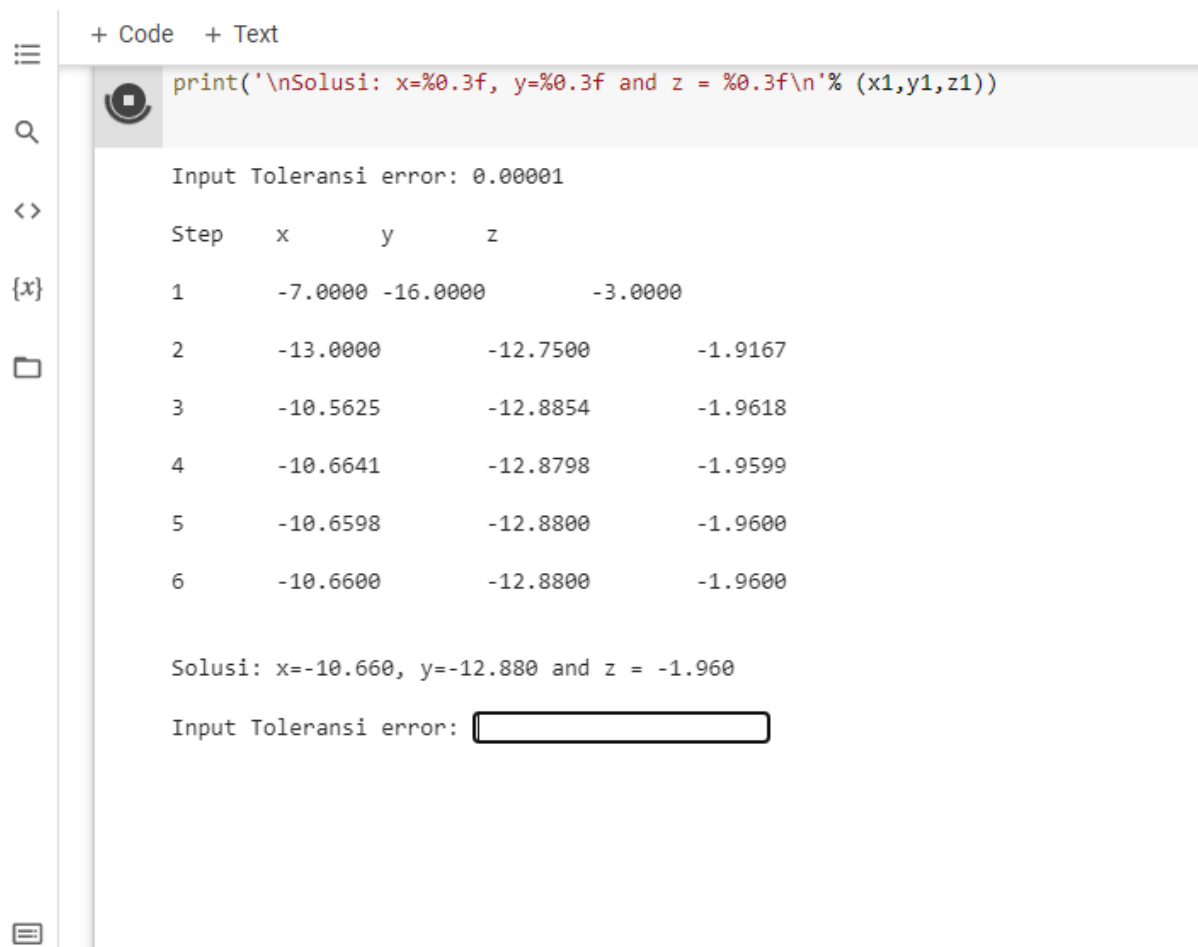
Npm : 202010225259

Kelas : TF 3 A6

---

## PRAKTIKUM 2 METNUM

### Lat Gaus Seidel



The screenshot shows a Python code editor with a sidebar on the left containing icons for file explorer, search, and code execution. The main editor area has a title bar with '+ Code' and '+ Text'. The code being executed is a Python script for the Gauss-Seidel method. It starts with a print statement for the solution, followed by an input for the tolerance error. A table displays the iteration results for 6 steps, showing the values of x, y, and z. The final solution is printed, and there is an input field for the tolerance error.

```
print('\nSolusi: x=%0.3f, y=%0.3f and z = %0.3f\n'% (x1,y1,z1))
```

Input Toleransi error: 0.00001

Step	x	y	z
1	-7.0000	-16.0000	-3.0000
2	-13.0000	-12.7500	-1.9167
3	-10.5625	-12.8854	-1.9618
4	-10.6641	-12.8798	-1.9599
5	-10.6598	-12.8800	-1.9600
6	-10.6600	-12.8800	-1.9600

Solusi: x=-10.660, y=-12.880 and z = -1.960

Input Toleransi error:

# Iterasi Gauss Seidel

# Definisikan Persamaan yang akan diselesaikan

# Dalam bentuk dominan secara diagonal

# Iterasi Gauss Seidel

```

# Definisikan Persamaan yang akan diselesaikan
# Dalam bentuk dominan secara diagonal
f1 = lambda x,y,z: (-4+3*y-0*z)/4
f2 = lambda x,y,z: (40-2*x+5*z)/-4
f3 = lambda x,y,z: (14+0*x+2*y)/6

# Inisial awal
x0 = 2
y0 = -8
z0 = 2
step = 1

# Input nilai galat/error
e = float(input('Input Toleransi error: '))

# Implementasi iterasi Gauss Seidel
print('\nStep\tx\ty\tz\n')

condition = True

while condition:
    x1 = f1(x0,y0,z0)
    y1 = f2(x1,y0,z0)
    z1 = f3(x1,y1,z0)
    print('%d\t%.4f\t%.4f\t%.4f\n' %(step, x1,y1,z1))
    e1 = abs(x0-x1);
    e2 = abs(y0-y1);
    e3 = abs(z0-z1);

    step +=1
    x0 = x1
    y0 = y1
    z0 = z1

    condition = e1>e and e2>e and e3>e
print('\nSolusi: x=%.3f, y=%.3f and z = %.3f\n' % (x1,y1,z1))

# Inisial awal
x0 = 1
y0 = 2
z0 = 2
step = 1

# Input nilai galat/error

```

```

e = float(input('Input Toleransi error: '))

# Implementasi iterasi Gauss Seidel
print('\nStep\tx\ty\tz\n')

condition = True

while condition:
    x1 = f1(x0,y0,z0)
    y1 = f2(x1,y0,z0)
    z1 = f3(x1,y1,z0)
    print('%d\t%0.4f\t%0.4f\t%0.4f\n' %(step, x1,y1,z1))
    e1 = abs(x0-x1);
    e2 = abs(y0-y1);
    e3 = abs(z0-z1);

    step +=1
    x0 = x1
    y0 = y1
    z0 = z1

    condition = e1>e and e2>e and e3>e
print('\nSolusi: x=%0.3f, y=%0.3f and z = %0.3f\n' % (x1,y1,z1))

# Inisial awal
x0 = 1
y0 = 2
z0 = 2
step = 1

# Input nilai galat/error
e = float(input('Input Toleransi error: '))

# Implementasi iterasi Gauss Seidel
print('\nStep\tx\ty\tz\n')

condition = True

while condition:
    x1 = f1(x0,y0,z0)
    y1 = f2(x1,y0,z0)
    z1 = f3(x1,y1,z0)
    print('%d\t%0.4f\t%0.4f\t%0.4f\n' %(step, x1,y1,z1))
    e1 = abs(x0-x1);

```

```
e2 = abs(y0-y1);  
e3 = abs(z0-z1);
```

```
step += 1
x0 = x1
y0 = y1
z0 = z1
```

```
condition = e1>e and e2>e and e3>e
print('\nSolusi: x=%0.3f, y=%0.3f and z = %0.3f\n' % (x1,y1,z1))
```

```
# Inisial awal
x0 = 1
y0 = 2
z0 = 2
step = 1
```

```
# Input nilai galat/error
e = float(input('Input Toleransi error: '))
```

```
# Implementasi iterasi Gauss Seidel
print('\nStep\tx\ty\tz\n')
```

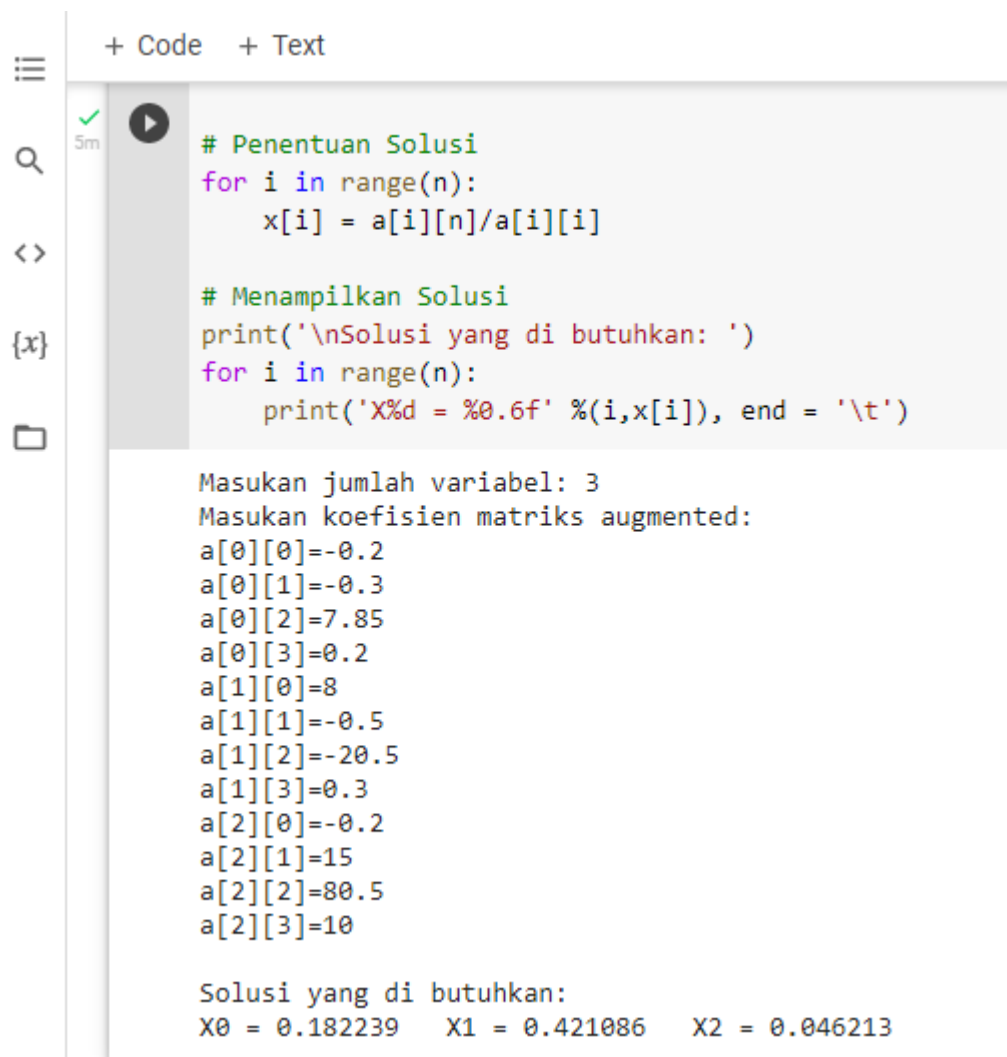
```
condition = True
```

```
while condition:
    x1 = f1(x0,y0,z0)
    y1 = f2(x1,y0,z0)
    z1 = f3(x1,y1,z0)
    print('%d\t%0.4f\t%0.4f\t%0.4f\n' %(step, x1,y1,z1))
    e1 = abs(x0-x1);
    e2 = abs(y0-y1);
    e3 = abs(z0-z1);
```

```
step += 1
x0 = x1
y0 = y1
z0 = z1
```

```
condition = e1>e and e2>e and e3>e
print('\nSolusi: x=%0.3f, y=%0.3f and z = %0.3f\n' % (x1,y1,z1))
```

## Gauss Jordan



```
+ Code + Text

# Penentuan Solusi
for i in range(n):
    x[i] = a[i][n]/a[i][i]

# Menampilkan Solusi
print('\nSolusi yang di butuhkan: ')
for i in range(n):
    print('X%d = %.6f' %(i,x[i]), end = '\t')
```

Masukan jumlah variabel: 3  
Masukan koefisien matriks augmented:  
a[0][0]=-0.2  
a[0][1]=-0.3  
a[0][2]=7.85  
a[0][3]=0.2  
a[1][0]=8  
a[1][1]=-0.5  
a[1][2]=-20.5  
a[1][3]=0.3  
a[2][0]=-0.2  
a[2][1]=15  
a[2][2]=80.5  
a[2][3]=10

Solusi yang di butuhkan:  
X0 = 0.182239    X1 = 0.421086    X2 = 0.046213

```
import numpy as np
```

```
import sys
```

```
n = int (input('Masukan jumlah variabel: '))
```

```
# Membuat array berukuran n x n+1 dan menginisiasi
```

```
# Menyimpan matriks augmented A | b
```

```
a = np.zeros((n,n+1))
```

```
# Membuat array berukuran n dan menginisiasi
```

```
# Vektor solusi
```

```
x = np.zeros(n)
```

```
# Membaca kofisien matrik augmented
```

```
print('Masukan koefisien matriks augmented: ')
```

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

```
for j in range(n+1):  
    a[i][j] = float(input( 'a[' +str(i)+'']['+str(j)+'']='))
```

```
# Implementasi Eliminasi Gaus Jordan
```

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

```
    if a[i][j] == 0.0:
```

```
        sys.exit('Divide by zero detected!:
```

```
for j in range(n):
```

```
    if i != j:
```

```
        ratio = a[j][i]/a[i][i]
```

```
        for k in range(n+1):
```

```
            a[j][k] = a [j][k] - ratio * a[i][k]
```

```
# Penentuan Solusi
```

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

```
    x[i] = a[i][n]/a[i][i]
```

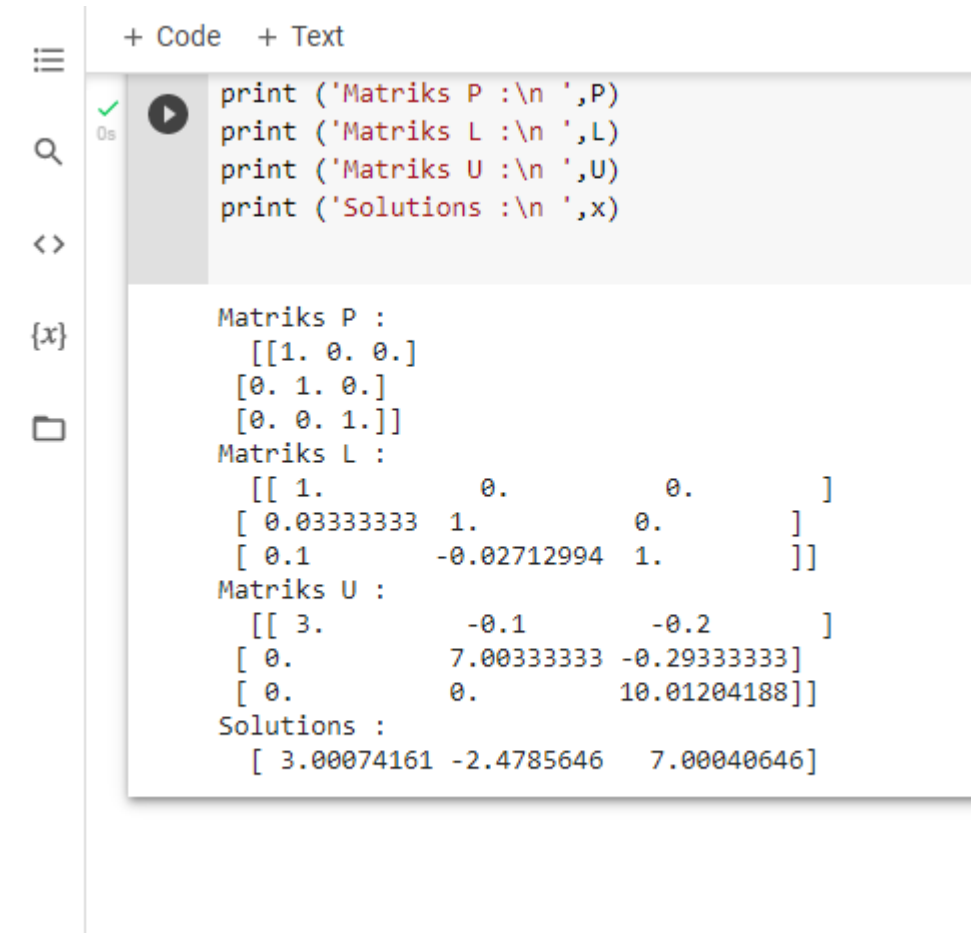
```
# Menampilkan Solusi
```

```
print('\nSolusi yang di butuhkan: ')
```

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

```
    print('X%d = %0.6f' %(i,x[i]), end = '\t')
```

## Faktorisasi



The screenshot shows a Jupyter Notebook with a code cell and its output. The code cell contains four print statements for matrices P, L, U, and the solution vector x. The output cell displays the numerical results for each matrix and the solution vector.

```
+ Code + Text
```

```
print ('Matriks P :\n ',P)
print ('Matriks L :\n ',L)
print ('Matriks U :\n ',U)
print ('Solutions :\n ',x)
```

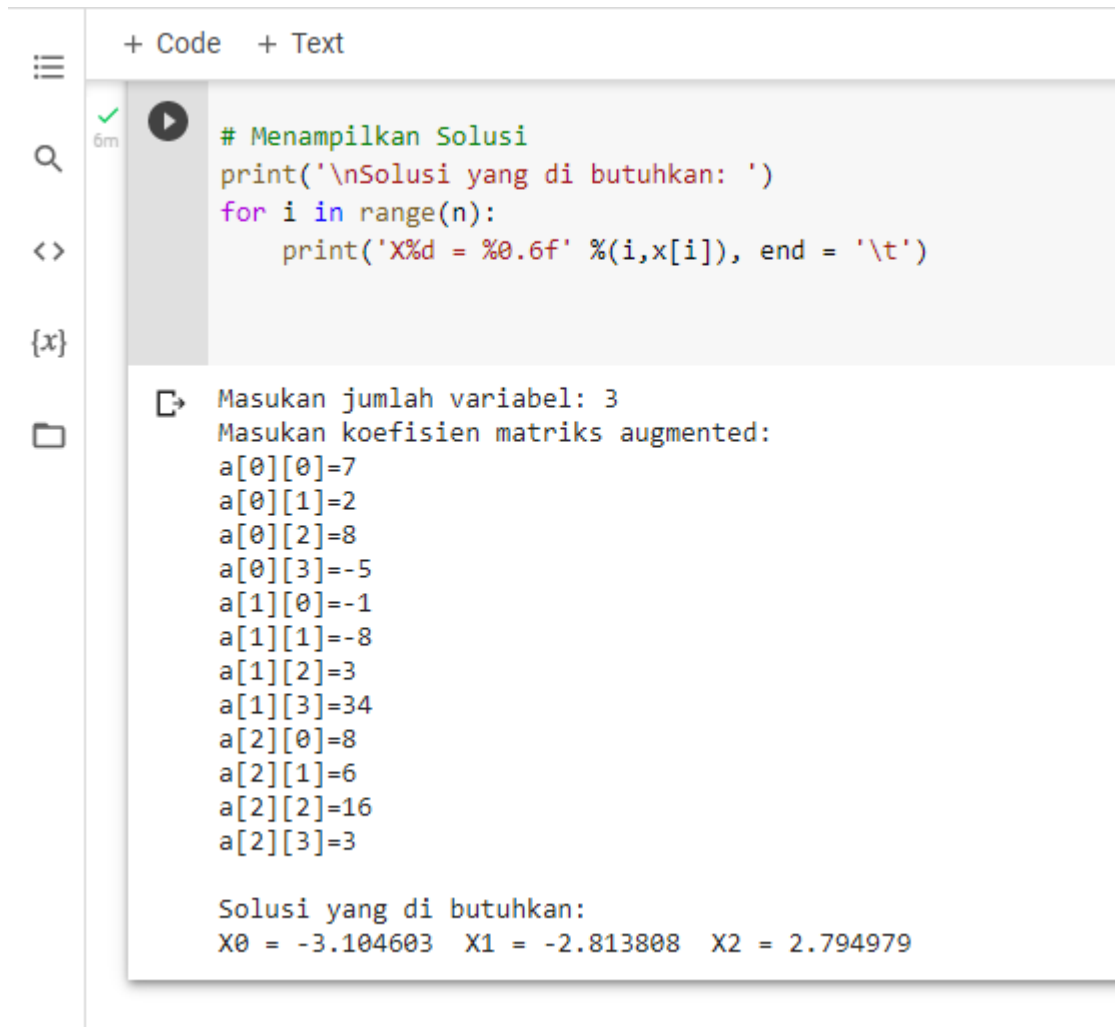
```
Matriks P :
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
Matriks L :
[[ 1.          0.          0.         ]
 [ 0.03333333  1.          0.         ]
 [ 0.1         -0.02712994  1.         ]]
Matriks U :
[[ 3.          -0.1         -0.2        ]
 [ 0.          7.00333333 -0.29333333]
 [ 0.          0.          10.01204188]]
Solutions :
[ 3.00074161 -2.4785646  7.00040646]
```

```
import scipy
from scipy.linalg import lu, lu_factor, lu_solve
import numpy as np

# Definisikan matriks A
A = np.array([[3., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])

# Definisikan vektor b
b = np.array([7.85, -19.15, 71.4])
# Solusi yang diberikan Lu dan b
P, L, U = lu(A)
lu, piv = lu_factor(A)
x = lu_solve((lu, piv),b)
print ('Matriks P :\n ',P)
print ('Matriks L :\n ',L)
print ('Matriks U :\n ',U)
print ('Solutions :\n ',x)
```

## Lat Gaus Jordan



The screenshot shows a Jupyter Notebook interface. On the left is a sidebar with icons for a menu, search, expand/collapse, variable {x}, and a file explorer. The top bar has '+ Code' and '+ Text' buttons. The main area contains a code cell with a green checkmark and a play button icon, labeled '6m'. The code cell contains Python code for solving a system of linear equations using Gaussian elimination. Below the code cell is the output, which shows the input values and the resulting solution vector.

```
+ Code + Text

# Menampilkan Solusi
print('\nSolusi yang di butuhkan: ')
for i in range(n):
    print('X%d = %0.6f' %(i,x[i]), end = '\t')
```

Masukan jumlah variabel: 3  
Masukan koefisien matriks augmented:  
a[0][0]=7  
a[0][1]=2  
a[0][2]=8  
a[0][3]=-5  
a[1][0]=-1  
a[1][1]=-8  
a[1][2]=3  
a[1][3]=34  
a[2][0]=8  
a[2][1]=6  
a[2][2]=16  
a[2][3]=3

Solusi yang di butuhkan:  
X0 = -3.104603 X1 = -2.813808 X2 = 2.794979

```
import numpy as np
```

```
import sys
```

```
n = int (input('Masukan jumlah variabel: '))
```

```
# Membuat array berukuran n x n+1 dan menginisiasi
```

```
# Menyimpan matriks augmented A | b
```

```
a = np.zeros((n,n+1))
```

```
# Membuat array berukuran n dan menginisiasi
```

```
# Vektor solusi
```

```
x = np.zeros(n)
```



```
# Membaca koefisien matrik augmented
print('Masukan koefisien matriks augmented: ')
for i in range(n):
    for j in range(n+1):
        a[i][j] = float(input( 'a[' +str(i)+'']['+str(j)+'']='))
```

```
# Implementasi Eliminasi Gaus Jordan
for i in range (n):
    if a[i][j] == 0.0:
        sys.exit('Divide by zero detected!:
```

```

    for j in range(n):
        if i != j:
            ratio = a[j][i]/a[i][i]

            for k in range(n+1):
                a[j][k] = a [j][k] - ratio * a[i][k]
```

```
# Penentuan Solusi
for i in range(n):
    x[i] = a[i][n]/a[i][i]
```

```
# Menampilkan Solusi
print('\nSolusi yang di butuhkan: ')
for i in range(n):
    print('X%d = %0.6f' %(i,x[i]), end = '\t')
```

## Gaus Seidel

```
+ Code + Text

condition = e1>e and e2>e and e3>e
print('\nSolusi: x=%0.3f, y=%0.3f and z = %0.3f\n'% (x1,y1,z1))

Input Toleransi error: 0.00001

Step    x        y        z
1       2.8167   -2.7117   7.0013
2       2.9930   -2.4998   7.0002
3       3.0000   -2.5000   7.0000
4       3.0000   -2.5000   7.0000

Solusi: x=3.000, y=-2.500 and z = 7.000
```

```
# Iterasi Gauss Seidel
```

```
# Definisikan Persamaan yang akan diselesaikan
```

# Dalam bentuk dominan secara diagonal

```
f1 = lambda x,y,z: (7.85+0.1*y+0.2*z)/3
```

```
f2 = lambda x,y,z: (-19.3-0.1*x+0.3*z)/7
```

```
f3 = lambda x,y,z: (71.4-0.3*x+0.2*y)/10
```

```
# Inisial awal
```

$$x_0 = 1$$
$$y_0 = 2$$
$$z_0 = 2$$

step = 1

```
# Input nilai galat/error
```

```
e = float(input('Input Toleransi error: '))
```

### # Implementasi iterasi Gauss Seidel

```
print('\nStep\tx\ty\tz\n')
```

```
condition = True
```

while condition:

```

x1 = f1(x0,y0,z0)
y1 = f2(x1,y0,z0)
z1 = f3(x1,y1,z0)
print('%d\t%0.4f\t%0.4f\t%0.4f\n' %(step, x1,y1,z1))
e1 = abs(x0-x1);
e2 = abs(y0-y1);
e3 = abs(z0-z1);

step +=1
x0 = x1
y0 = y1
z0 = z1

condition = e1>e and e2>e and e3>e
print('\nSolusi: x=%0.3f, y=%0.3f and z = %0.3f\n' % (x1,y1,z1))

```