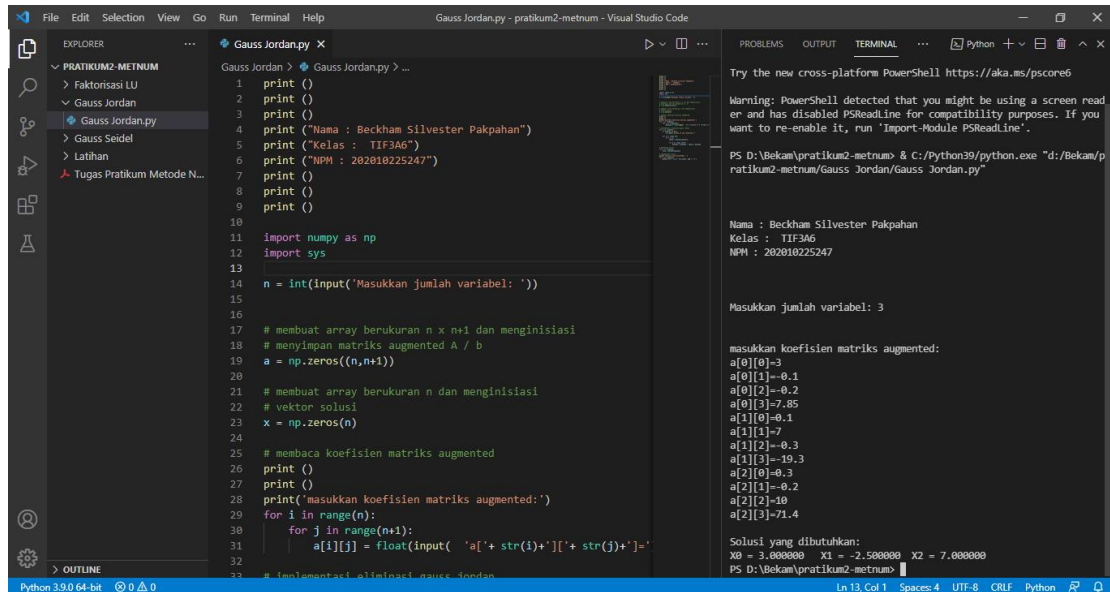


Nama : Beckham Silvester Pakpahan
NPM : 202010225247
Kelas : TIF3A6

METODE NUMERIK

Tugas Python Pratikum 2

Script Dan Hasil Gauss Jordan



```
1 print ()
2 print ()
3 print ()
4 print ("Nama : Beckham Silvester Pakpahan")
5 print ("Kelas : TIF3A6")
6 print ("NPM : 202010225247")
7 print ()
8 print ()
9 print ()
10
11 import numpy as np
12 import sys
13
14 n = int(input("Masukkan jumlah variabel: "))
15
16
17 # membuat array berukuran n x n+1 dan menginisiasi
18 # menyimpan matriks augmented A / b
19 a = np.zeros((n,n+1))
20
21 # membuat array berukuran n dan menginisiasi
22 # vektor solusi
23 x = np.zeros(n)
24
25 # membaca koefisien matriks augmented
26 print ()
27 print ()
28 print('masukkan koefisien matriks augmented:')
29 for i in range(n):
30     for j in range(n+1):
31         a[i][j] = float(input(' a['+ str(i)+'']['+ str(j)+'']= '))
32
33 # implementasi eliminasi gauss jordan
```

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

PS D:\Bekam\pratikum2-metnum> & C:/Python39/python.exe "d:/Bekam/pratikum2-metnum/Gauss Jordan/Gauss Jordan.py"

Nama : Beckham Silvester Pakpahan
Kelas : TIF3A6
NPM : 202010225247

Masukkan jumlah variabel: 3

masukkan koefisien matriks augmented:

a[0][0]=-0.1
a[0][1]=-0.2
a[0][2]=7.85
a[1][0]=0.1
a[1][1]=7
a[1][2]=-0.3
a[1][3]=-19.3
a[2][0]=0.3
a[2][1]=0.2
a[2][2]=10
a[2][3]=-71.4

Solusi yang dibutuhkan:
X0 = 3.000000 X1 = -2.500000 X2 = 7.000000

PS D:\Bekam\pratikum2-metnum>

```
print ()
print ()
print ()
print ("Nama : Beckham Silvester Pakpahan")
print ("Kelas : TIF3A6")
print ("NPM : 202010225247")
print ()
print ()
print ()
```

```
import numpy as np
import sys
```

```
n = int(input("Masukkan 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 matriks augmented
```

```

print ()
print ()
print('masukkan 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 gauss jordan

```

for i in range(n):
    if a[i][i] == 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 dibutuhkan: ")
for i in range(n):
    print('X%d = %0.6f' %(i,x[i]), end = '\t')

```

Script Dan Hasil Faktorisasi LU

The screenshot shows a Visual Studio Code window with a file named 'Faktorisasi LU.py'. The script uses NumPy and SciPy to perform LU factorization on a 3x3 matrix. The output in the terminal shows the matrix A, the LU factorization result (P, L, U), and the solution vector x.

```

1 print ()
2 print ()
3 print ()
4 print ("Nama : Beckham Silvester Pakpahan")
5 print ("Kelas : TIF3A6")
6 print ("NPM : 202010225247")
7 print ()
8 print ()
9 print ()
10
11
12 import scipy
13 from scipy.linalg import lu, lu_factor, lu_solve
14 import numpy as np
15
16 # Definisikan matriks A
17 A = np.array([[3., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 1.]])
18
19 # Definisikan vektor b
20 b = np.array([7.85, -19.3, 71.4])
21
22 # Solusi yang diberikan lu dan b
23 P, L, U = lu(A)
24 lu, piv = lu_factor(A)
25 x = lu_solve((lu, piv),b)
26 print ("Matriks P : \n",P)
27 print ("Matriks L : \n",L)
28 print ("Matriks U : \n",U)
29 print ("Solutions : \n",x)

```

Terminal Output:

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

PS D:\Bekam\pratikum2-metnum> & C:/Python39/python.exe "d:/Bekam/pratikum2-metnum/Faktorisasi LU/Faktorisasi LU.py"

Nama : Beckham Silvester Pakpahan
Kelas : TIF3A6
NPM : 202010225247

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.         7.03333333 -0.29333333 ]
 [ 0.          0.         10.01204188]]
Solutions :
[ 3. -2.5  7. ]
PS D:\Bekam\pratikum2-metnum>

```

```

print ()
print ()
print ()
print ("Nama : Beckham Silvester Pakpahan")

```



```
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))
```

Latihan Pratikum 2

Selesaikan :

$$4x_1 + 3x_2 - x_3 = -4$$

$$-2x_1 - 4x_2 + 5x_3 = 40$$

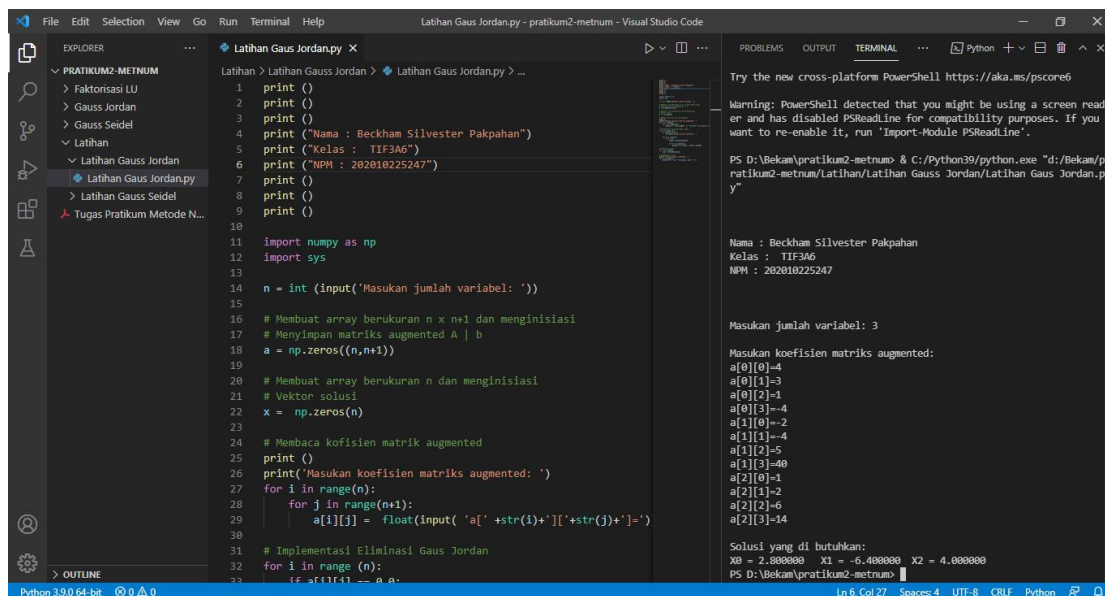
$$x_1 + 2x_2 + 6x_3 = 14$$

Sistem persamaan linear di atas dengan Metode :

1. Eliminasi Gauss Jordan
2. Iterasi Gauss-Sidel dengan nilai awal (2,-8,2), dan solusi sejatinya berdasarkan perhitungan (a).
3. Berdasarkan jawaban 1 dan 2, apa yang dapat kalian simpulkan?

Jawaban

Eliminasi Gauss Jordan



```
1 print ()
2 print ()
3 print ()
4 print ("Nama : Beckham Silvester Pakpahan")
5 print ("Kelas : TIF3A6")
6 print ("NPM : 282018225247")
7 print ()
8 print ()
9 print ()
10
11 import numpy as np
12 import sys
13
14 n = int(input("Masukan jumlah variabel: "))
15
16 # Membuat array berukuran n x n+1 dan menginisiasi
17 # Menyimpan matriks augmented A | b
18 a = np.zeros((n,n+1))
19
20 # Membuat array berukuran n dan menginisiasi
21 # Vektor solusi
22 x = np.zeros(n)
23
24 # Membaca koefisien matriks augmented
25 print ()
26 print("Masukan koefisien matriks augmented: ")
27 for i in range(n):
28     for j in range(n+1):
29         a[i][j] = float(input("a[" + str(i) + "][" + str(j) + "]="))
30
31 # Implementasi Eliminasi Gauss Jordan
32 for i in range(n):
33     if a[i][i] == 0.0:
```

```
print ()
print ()
print ()
print ("Nama : Beckham Silvester Pakpahan")
print ("Kelas : TIF3A6")
```

```

print ("NPM : 202010225247")
print ()
print ()
print ()

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 ()
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][i] == 0.0:
        sys.exit('Divide by zero detected!: ')

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

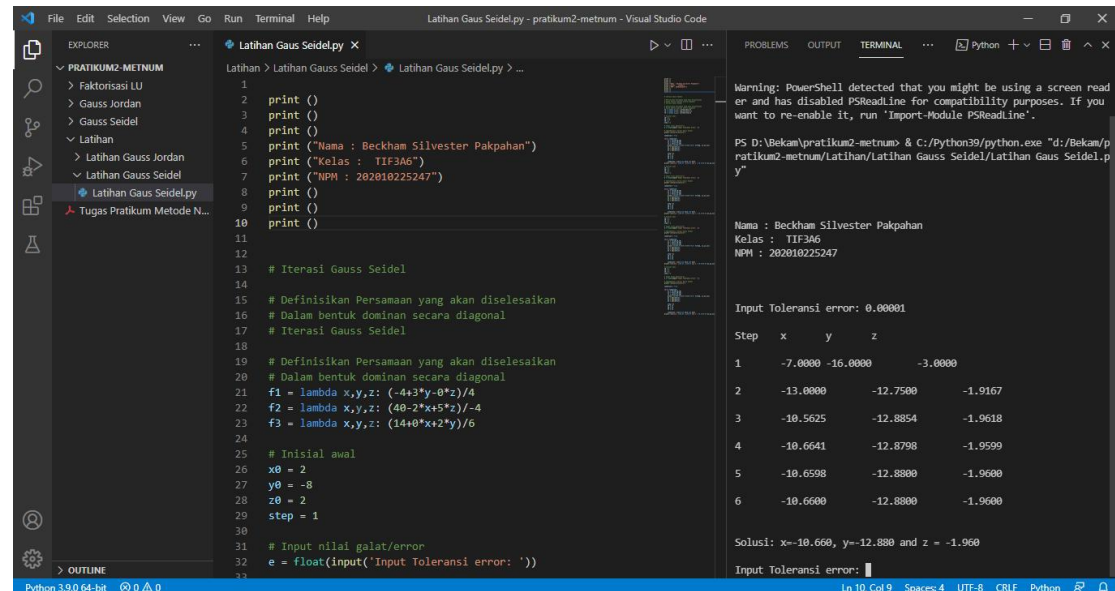
            for k in range(n+1):
                a[j][k] = a [j][k] - ratio * a[i][k]

# 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')

```

Iterasi Gauss-Sidel



```
1 print ()
2 print ()
3 print ()
4 print ()
5 print ("Nama : Beckham Silvester Pakpahan")
6 print ("Kelas : TIF3A6")
7 print ("NPM : 202010225247")
8 print ()
9 print ()
10 print ()
11
12
13 # Iterasi Gauss Seidel
14
15 # Definisikan Persamaan yang akan diselesaikan
16 # Dalam bentuk dominan secara diagonal
17 # Iterasi Gauss Seidel
18
19 # Definisikan Persamaan yang akan diselesaikan
20 # Dalam bentuk dominan secara diagonal
21 f1 = lambda x,y,z: (-4+3*y-0*z)/4
22 f2 = lambda x,y,z: (40-2*x+5*z)/-4
23 f3 = lambda x,y,z: (14+0*x+2*y)/6
24
25 # Inisial awal
26 x0 = 2
27 y0 = -8
28 z0 = 2
29 step = 1
30
31 # Input nilai galat/error
32 e = float(input('Input Toleransi error: '))
33
```

Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

PS D:\Bekam\pratikum2-metnum & C:/Python39/python.exe "d:/Bekam/pratikum2-metnum/Latihan/Latihan Gauss Seidel/Latihan Gauss Seidel.py"

Nama : Beckham Silvester Pakpahan
Kelas : TIF3A6
NPM : 202010225247

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:

```
print ()
print ()
print ()
print ("Nama : Beckham Silvester Pakpahan")
print ("Kelas : TIF3A6")
print ("NPM : 202010225247")
print ()
print ()
print ()
```

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%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))

# 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))
```