

NAMA : OZZI ARDHIYANTO

KELAS : TF3B

NPM 202010225030

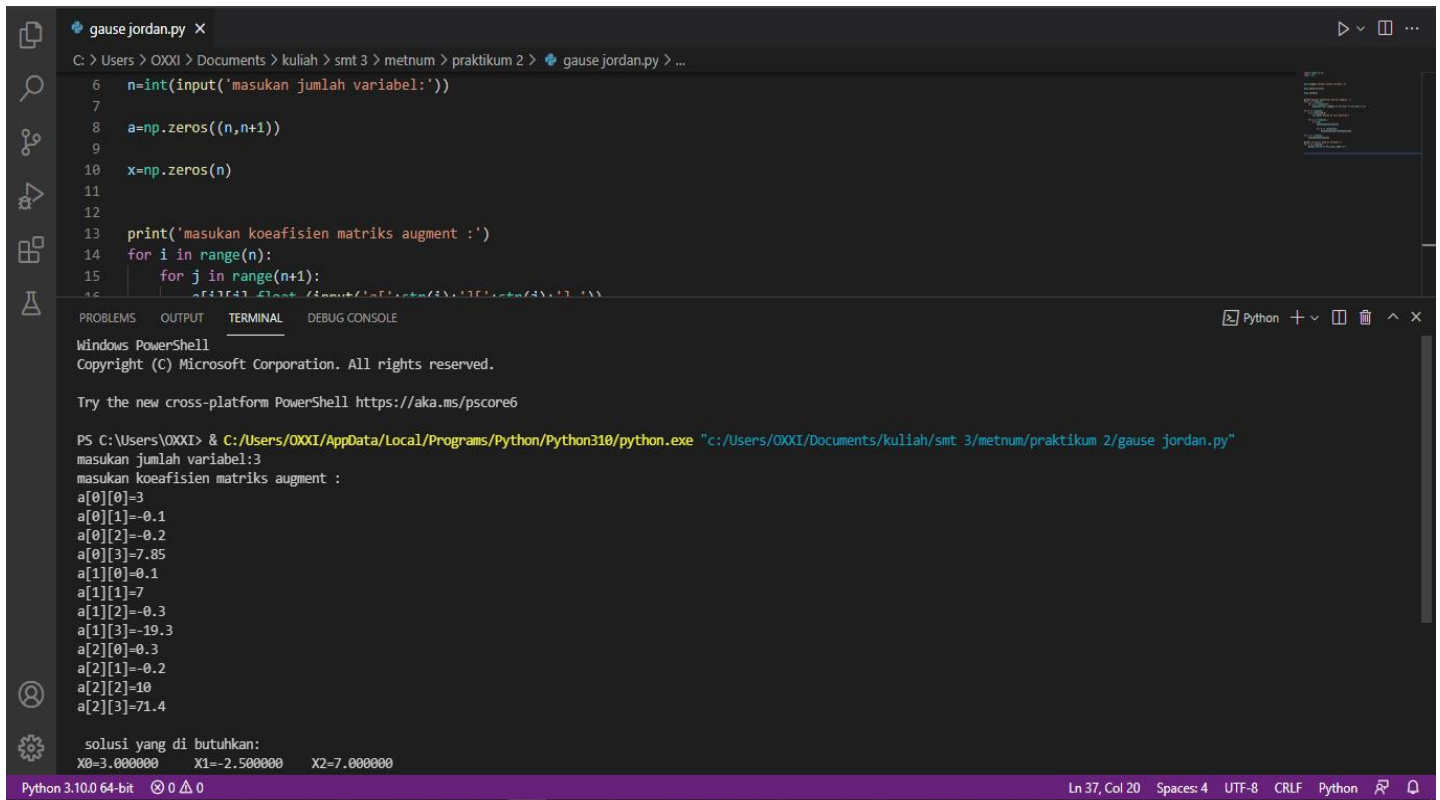
TUGAS PRAKTIKUM 2

CONTOH

1. GAUS JORDAN

```
2. import numpy as np
3. import sys
4.
5. n=int(input('masukan jumlah variabel:'))
6.
7. a=np.zeros((n,n+1))
8.
9. x=np.zeros(n)
10.
11.print('masukan koefisien matriks augment :')
12.for i in range(n):
13.    for j in range(n+1):
14.        a[i][j]=float (input('a['+str(i)+'']['+str(j)+'']='))
15.
16.for i in range(n):
17.    if a [i][i]==0.0:
18.        sys.exit('divide by zero detected')
19.
20.    for j in range(n) :
21.        if i!=j:
22.            ratio=a[j][i]/a[i][i]
23.
24.            for k in range(n+1):
25.                a[j][k]=a[j][k]-ratio*a[i][k]
26.
27.for i in range(n):
28.    x[i]=a[i][n]/a[i][i]
29.
30.print('\n solusi yang di butuhkan:')
31.for i in range(n):
32.    print('X%d=%0.6f'%(i,x[i]),end='\t')
```

33.



The screenshot shows a VS Code editor with a file named `gause_jordan.py` open. The file contains a Python script for Gaussian elimination. The terminal window shows the execution of the script, which prompts the user to input the number of variables (3) and then the coefficients of the augmented matrix. The output displays the matrix elements and the solution for the system of linear equations.

```
gause_jordan.py X
C: > Users > OXXI > Documents > kuliah > smt 3 > metnum > praktikum 2 > gause_jordan.py > ...

6  n=int(input('masukan jumlah variabel:'))
7
8  a=np.zeros((n,n+1))
9
10 x=np.zeros(n)
11
12
13 print('masukan koefisien matriks augment :')
14 for i in range(n):
15     for j in range(n+1):
16         a[i][j]=float(input('masukan koefisien matriks augment :'))

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Users\OXXI> & C:/Users/OXXI/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/OXXI/Documents/kuliah/smt 3/metnum/praktikum 2/gause_jordan.py"
masukan jumlah variabel:3
masukan koefisien matriks augment :
a[0][0]=3
a[0][1]=-0.1
a[0][2]=-0.2
a[0][3]=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 di butuhkan:
X0=3.000000 X1=-2.500000 X2=7.000000

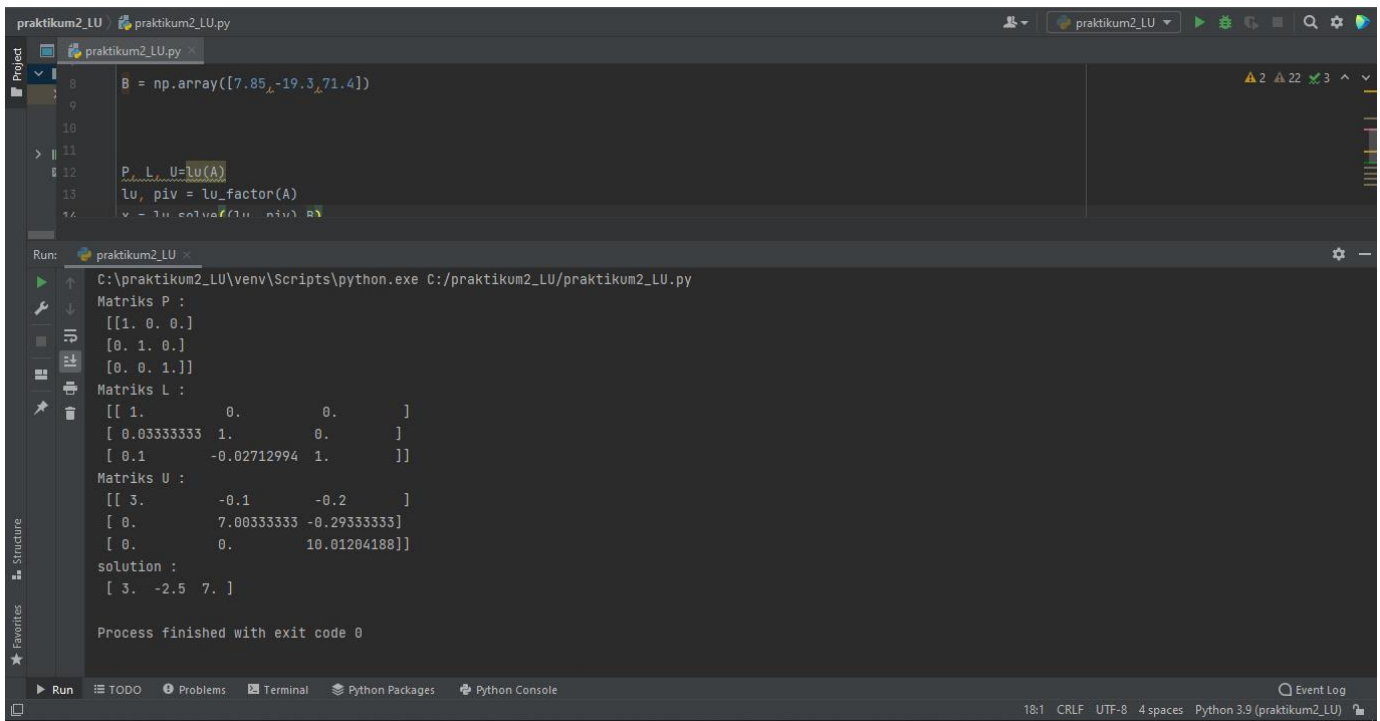
Python 3.10.0 64-bit 0 0 0 Ln 37, Col 20 Spaces: 4 UTF-8 CRLF Python
```

2. LU

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

A = np.array([[3., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
B = np.array([7.85, -19.3, 71.4])

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 ('solution :\n',x)
```



The screenshot shows an IDE window with a Python file named `praktikum2_LU.py`. The code defines a matrix `B` and performs LU decomposition using `lu_factor` and `lu_solve`. The Run console displays the output, showing the matrices `P`, `L`, and `U`, and the solution vector.

```
praktikum2_LU.py
8 B = np.array([7.85,-19.3,71.4])
9
10
11
12 P, L, U = lu_factor(A)
13 lu, piv = lu_factor(A)
14 x = lu_solve((lu, piv), b)
```

Run: praktikum2_LU x

```
C:\praktikum2_LU\venv\Scripts\python.exe C:/praktikum2_LU/praktikum2_LU.py
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]]
solution :
[ 3. -2.5  7. ]

Process finished with exit code 0
```

3. GAUSE SEIDEL

```
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

x0=1
y0=2
z0=2
step=1

e=float(input('input toleransi error:'))

print('\nstep\tx\ty\tz\n')

condition = True

while condition:
    x1 = f1 (x0,y0,z0)
```

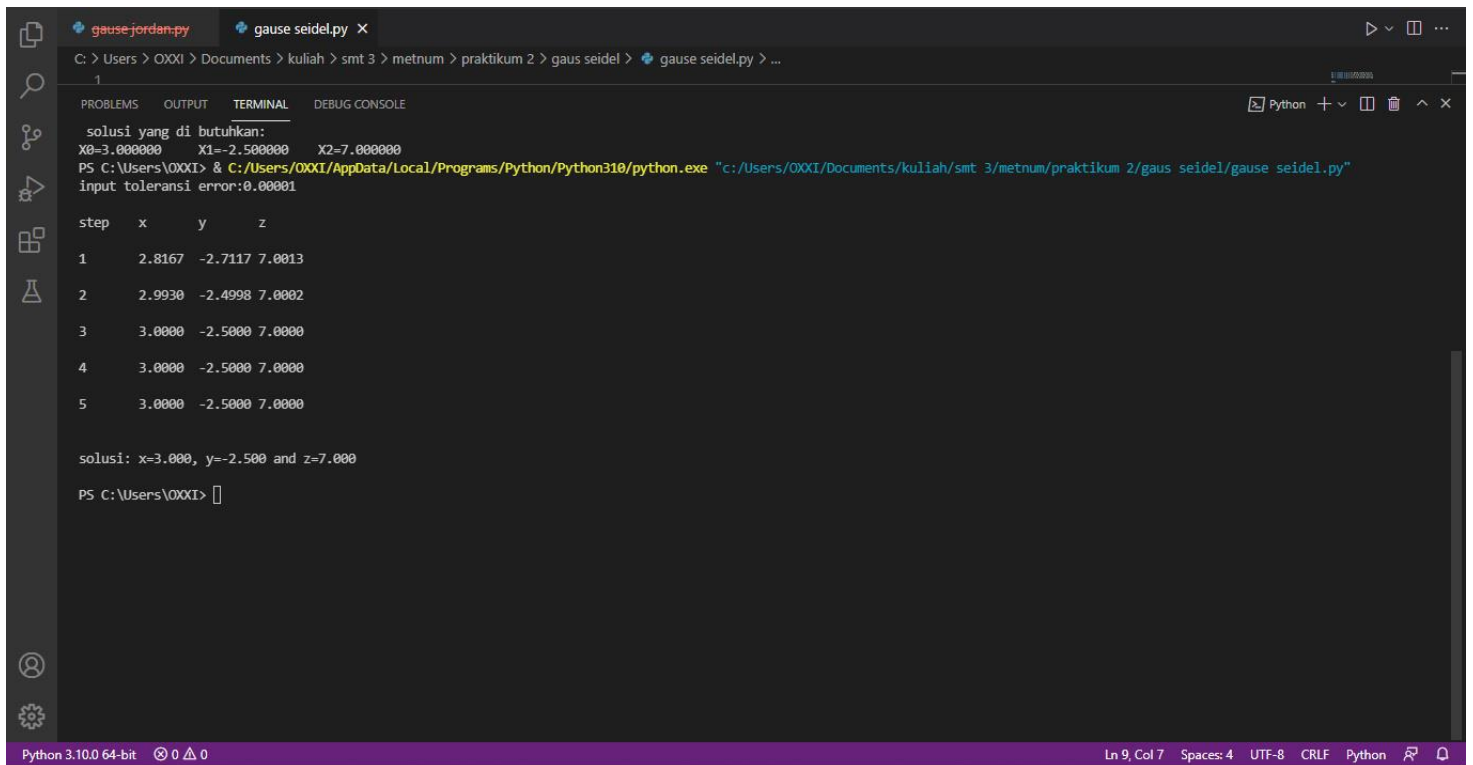
```

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(x0 - y1);
e3 = abs(x0 - 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))

```



The screenshot shows a Python IDE with two tabs: 'gausse-jordan.py' and 'gausse-seidel.py'. The active tab is 'gausse-seidel.py'. The terminal window displays the following output:

```

1
solusi yang di butuhkan:
X0=3.000000 X1=-2.500000 X2=7.000000
PS C:\Users\OOXI> & C:/Users/OOXI/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/OOXI/Documents/kuliah/smt 3/metnum/praktikum 2/gaus seidel/gause seidel.py"
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
5      3.0000 -2.5000 7.0000

solusi: x=3.000, y=-2.500 and z=7.000
PS C:\Users\OOXI>

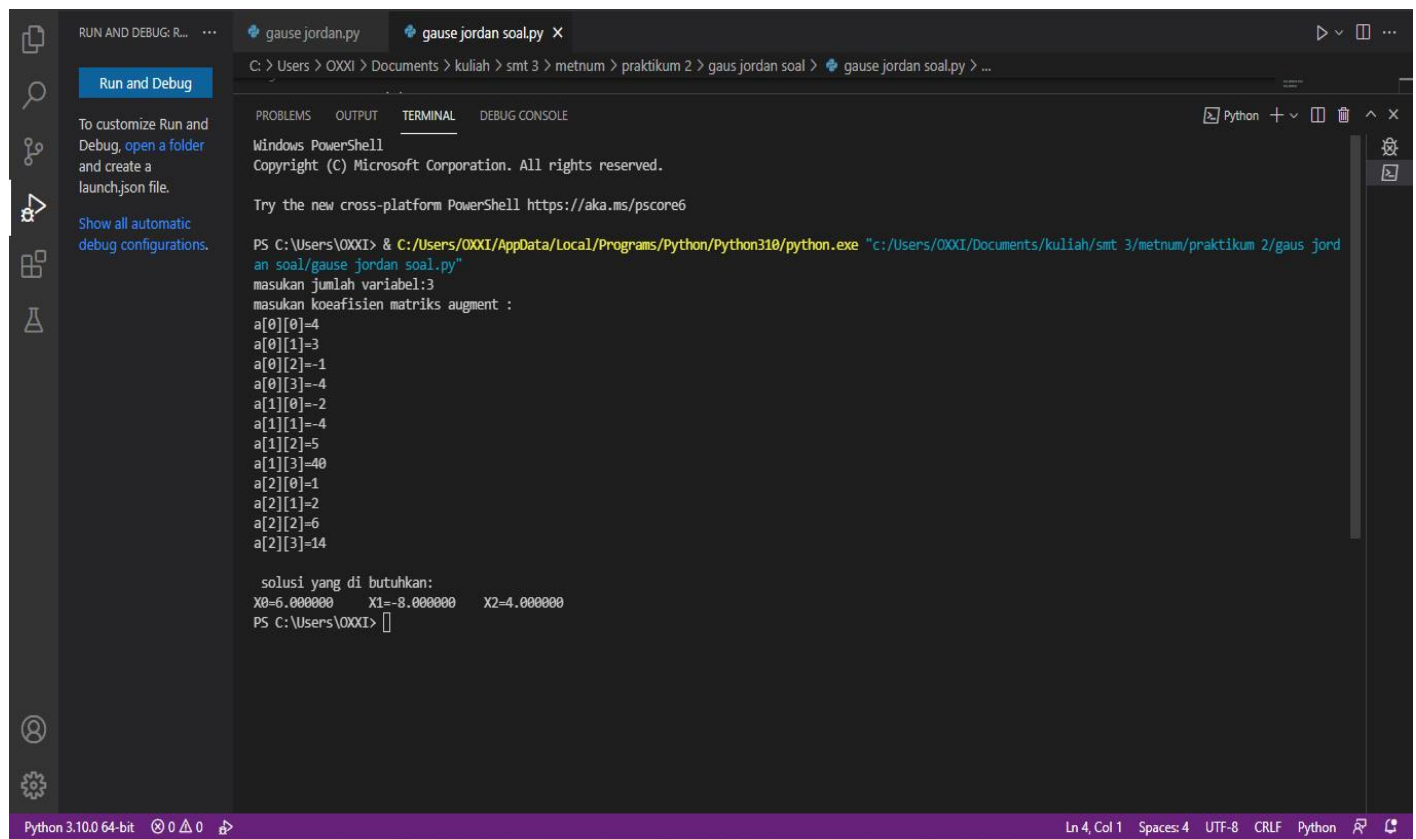
```

The status bar at the bottom indicates 'Python 3.10.0 64-bit', 'Ln 9, Col 7', 'Spaces: 4', 'UTF-8', 'CRLF', and 'Python'.

SOAL LATIHAN

1. GAUSE JORDAN

```
2. import numpy as np
3. import sys
4.
5. n=int(input('masukan jumlah variabel:'))
6.
7. a=np.zeros((n,n+1))
8.
9. x=np.zeros(n)
10.
11.print('masukan koefisien matriks augment :')
12.for i in range(n):
13.    for j in range(n+1):
14.        a[i][j]=float (input('a['+str(i)+'']['+str(j)+'']='))
15.
16.for i in range(n):
17.    if a [i][i]==0.0:
18.        sys.exit('divide by zero detected')
19.
20.    for j in range(n) :
21.        if i!=j:
22.            ratio=a[j][i]/a[i][i]
23.
24.            for k in range(n+1):
25.                a[j][k]=a[j][k]-ratio*a[i][k]
26.
27.for i in range(n):
28.    x[i]=a[i][n]/a[i][i]
29.
30.print('\n solusi yang dibutuhkan:')
31.for i in range(n):
32.    print('X%d=%0.6f'%(i,x[i]),end='\t')
```

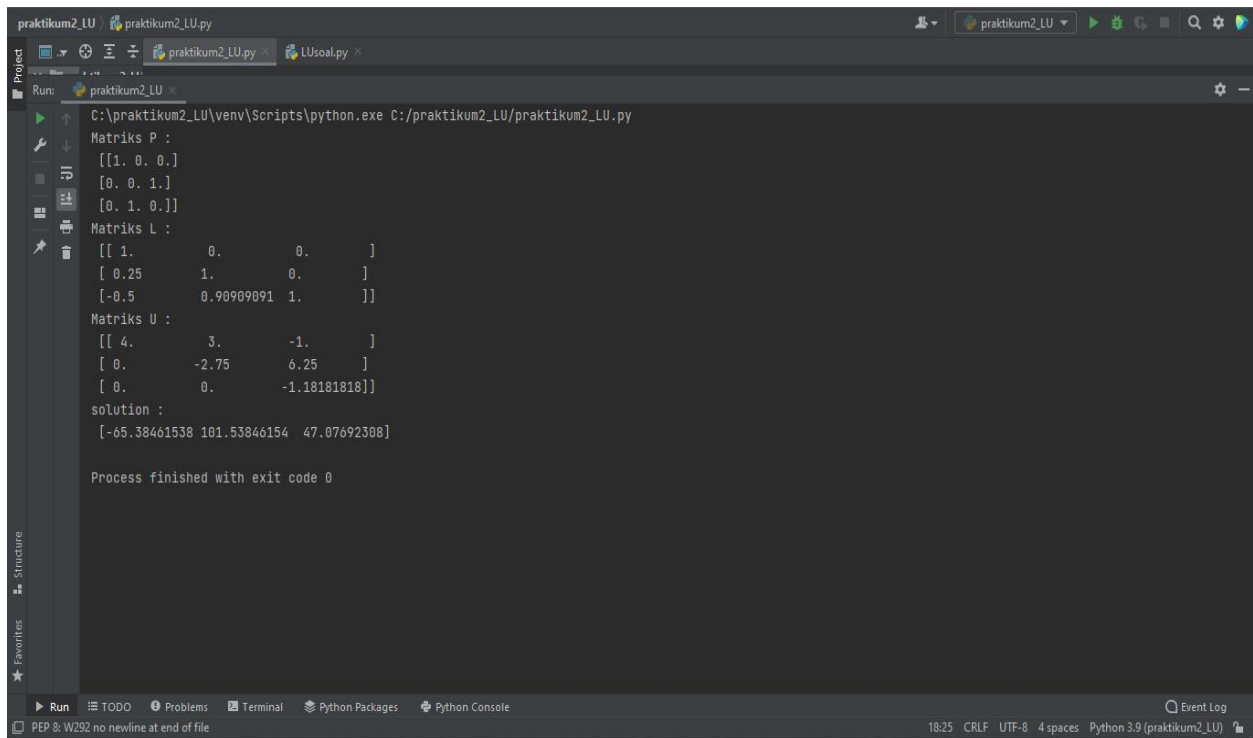


2. LU

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

A = np.array([[4., 3, -1], [-2, -4, 5], [1, -2, 6]])
B = np.array([-4, -40, 14])

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 ('solution :\n',x)
```



```
praktikum2_LU / praktikum2_LU.py
praktikum2_LU.py x LUsoal.py x
Run: praktikum2_LU x
C:\praktikum2_LU\venv\Scripts\python.exe C:/praktikum2_LU/praktikum2_LU.py
Matriks P :
[[1. 0. 0.]
 [0. 0. 1.]
 [0. 1. 0.]]
Matriks L :
[[ 1.      0.      0.      ]
 [ 0.25    1.      0.      ]
 [-0.5     0.90909091 1.      ]]
Matriks U :
[[ 4.      3.      -1.      ]
 [ 0.      -2.75    6.25    ]
 [ 0.      0.      -1.18181818]]
solution :
[-65.38461538 101.53846154 47.07692308]

Process finished with exit code 0
```

3.GAUSE SEIDEL

```
f1 = lambda x,y,z:(-4+3*y+-1*z)/4
f2 = lambda x,y,z:(40+-2*x+5*z)/-4
f3 = lambda x,y,z:(14+1*x+2*y)/6

x0=1
y0=2
z0=2
step=1

e=float(input('input toleransi error:'))

print('\nstep\ttx\ty\ttz\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(x0 - y1);
e3 = abs(x0 - 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))

```

The screenshot shows a Python IDE with a file named `gause seidel soal.py` open. The terminal window displays the output of the program, which is a table of values for steps 3 through 17. The final output is `solusi: x=-8.472, y=-10.862 and z=-2.699`.

PROBLEMS	OUTPUT	TERMINAL	DEBUG CONSOLE
3	-9.6146	-10.3802	-2.7292
4	-8.1029	-10.6400	-2.5638
5	-8.3390	-10.9648	-2.7114
6	-8.5457	-10.8836	-2.7188
7	-8.4830	-10.8430	-2.6948
8	-8.4585	-10.8607	-2.6967
9	-8.4714	-10.8649	-2.7002
10	-8.4736	-10.8616	-2.6995
11	-8.4713	-10.8613	-2.6990
12	-8.4713	-10.8619	-2.6992
13	-8.4716	-10.8618	-2.6992
14	-8.4716	-10.8618	-2.6992
15	-8.4715	-10.8618	-2.6992
16	-8.4715	-10.8618	-2.6992
17	-8.4715	-10.8618	-2.6992

solusi: x=-8.472, y=-10.862 and z=-2.699