



Nama : Ricky Suryanto Parsaulian Samosir

NPM : 202010225254

Kelas : TF3A6

PRAKTIKUM METODE NUMERIK MODUL 2

1.1 METODE GAUSS JORDAN

```
1 # Ricky Suryanto Parsaulian Samosir 202010225254 TF3A6
2 import numpy as np
3 import sys
4
5 n = int(input('Masukkan jumlah variabel: '))
6
7 # Membuat array berukuran n x n+1 dan menginisiasi
8 # Matriks augmented Aib
9 a = np.zeros((n,n+1))
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

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

```
PS C:\Users\HP\Downloads> & 'C:\Users\HP\AppData\Local\Programs\Python\Python17\python.exe' 'C:\Users\HP\AppData\Local\Programs\Python\Python17\python.exe' 'C:\Users\HP\Downloads\gaussjordan.py'
Masukkan jumlah variabel: 3
Masukkan koefisien matriks augmented:
a[0][0]=5
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]=10
a[2][2]=-1.4

Solusi yang dibutuhkan:
XB = -5.704962 X1 = -2.453246 X2 = -1.715600
```



Nama : Ricky Suryanto Parsaulian Samosir

NPM : 202010225254

Kelas : TF3A6

Codingnya:

```
# Ricky Suryanto Parsaulian Samosir 202010225254 TF3A6
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('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)+'']='))

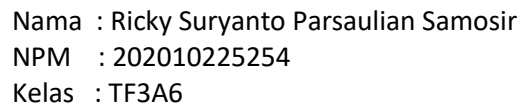
# Implementasi Eliminasi Gauss Jordan
for i in range(n):
    if a[i][i] == 0.0:
        sys.exit('Divide by zero detected!')

    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[1][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')
```



The screenshot shows a Visual Studio Code editor with a Python file named `faktorisasi LU.py`. The script defines a matrix `A` and a vector `b`, then uses NumPy to solve the system `Ax = b`. The output shows the matrices and the solution vector.

```

1  # Ricky Suryanta Persaulian Samsir 202018223254 TP5ad
2  import sys
3  from scipy.linalg import lu, lu_factor, lu_solve
4  import numpy as np
5
6  # Definisi matriks A
7  A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
8
9  # Definisi vektor b
10 b = np.array([7.85, -19.3, 71.4])
11
12 # Solusi yang diberikan lu dan b
13
14 # Output
15
16 # Definisi matriks A
17 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
18
19 # Definisi vektor b
20 b = np.array([7.85, -19.3, 71.4])
21
22 # Solusi yang diberikan lu dan b
23
24 # Output
25
26 # Definisi matriks A
27 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
28
29 # Definisi vektor b
30 b = np.array([7.85, -19.3, 71.4])
31
32 # Solusi yang diberikan lu dan b
33
34 # Output
35
36 # Definisi matriks A
37 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
38
39 # Definisi vektor b
40 b = np.array([7.85, -19.3, 71.4])
41
42 # Solusi yang diberikan lu dan b
43
44 # Output
45
46 # Definisi matriks A
47 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
48
49 # Definisi vektor b
50 b = np.array([7.85, -19.3, 71.4])
51
52 # Solusi yang diberikan lu dan b
53
54 # Output
55
56 # Definisi matriks A
57 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
58
59 # Definisi vektor b
60 b = np.array([7.85, -19.3, 71.4])
61
62 # Solusi yang diberikan lu dan b
63
64 # Output
65
66 # Definisi matriks A
67 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
68
69 # Definisi vektor b
70 b = np.array([7.85, -19.3, 71.4])
71
72 # Solusi yang diberikan lu dan b
73
74 # Output
75
76 # Definisi matriks A
77 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
78
79 # Definisi vektor b
80 b = np.array([7.85, -19.3, 71.4])
81
82 # Solusi yang diberikan lu dan b
83
84 # Output
85
86 # Definisi matriks A
87 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
88
89 # Definisi vektor b
90 b = np.array([7.85, -19.3, 71.4])
91
92 # Solusi yang diberikan lu dan b
93
94 # Output
95
96 # Definisi matriks A
97 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
98
99 # Definisi vektor b
100 b = np.array([7.85, -19.3, 71.4])
101
102 # Solusi yang diberikan lu dan b
103
104 # Output
105
106 # Definisi matriks A
107 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
108
109 # Definisi vektor b
110 b = np.array([7.85, -19.3, 71.4])
111
112 # Solusi yang diberikan lu dan b
113
114 # Output
115
116 # Definisi matriks A
117 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
118
119 # Definisi vektor b
120 b = np.array([7.85, -19.3, 71.4])
121
122 # Solusi yang diberikan lu dan b
123
124 # Output
125
126 # Definisi matriks A
127 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
128
129 # Definisi vektor b
130 b = np.array([7.85, -19.3, 71.4])
131
132 # Solusi yang diberikan lu dan b
133
134 # Output
135
136 # Definisi matriks A
137 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
138
139 # Definisi vektor b
140 b = np.array([7.85, -19.3, 71.4])
141
142 # Solusi yang diberikan lu dan b
143
144 # Output
145
146 # Definisi matriks A
147 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
148
149 # Definisi vektor b
150 b = np.array([7.85, -19.3, 71.4])
151
152 # Solusi yang diberikan lu dan b
153
154 # Output
155
156 # Definisi matriks A
157 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
158
159 # Definisi vektor b
160 b = np.array([7.85, -19.3, 71.4])
161
162 # Solusi yang diberikan lu dan b
163
164 # Output
165
166 # Definisi matriks A
167 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
168
169 # Definisi vektor b
170 b = np.array([7.85, -19.3, 71.4])
171
172 # Solusi yang diberikan lu dan b
173
174 # Output
175
176 # Definisi matriks A
177 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
178
179 # Definisi vektor b
180 b = np.array([7.85, -19.3, 71.4])
181
182 # Solusi yang diberikan lu dan b
183
184 # Output
185
186 # Definisi matriks A
187 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
188
189 # Definisi vektor b
190 b = np.array([7.85, -19.3, 71.4])
191
192 # Solusi yang diberikan lu dan b
193
194 # Output
195
196 # Definisi matriks A
197 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
198
199 # Definisi vektor b
200 b = np.array([7.85, -19.3, 71.4])
201
202 # Solusi yang diberikan lu dan b
203
204 # Output
205
206 # Definisi matriks A
207 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
208
209 # Definisi vektor b
210 b = np.array([7.85, -19.3, 71.4])
211
212 # Solusi yang diberikan lu dan b
213
214 # Output
215
216 # Definisi matriks A
217 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
218
219 # Definisi vektor b
220 b = np.array([7.85, -19.3, 71.4])
221
222 # Solusi yang diberikan lu dan b
223
224 # Output
225
226 # Definisi matriks A
227 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
228
229 # Definisi vektor b
230 b = np.array([7.85, -19.3, 71.4])
231
232 # Solusi yang diberikan lu dan b
233
234 # Output
235
236 # Definisi matriks A
237 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
238
239 # Definisi vektor b
240 b = np.array([7.85, -19.3, 71.4])
241
242 # Solusi yang diberikan lu dan b
243
244 # Output
245
246 # Definisi matriks A
247 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
248
249 # Definisi vektor b
250 b = np.array([7.85, -19.3, 71.4])
251
252 # Solusi yang diberikan lu dan b
253
254 # Output
255
256 # Definisi matriks A
257 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
258
259 # Definisi vektor b
260 b = np.array([7.85, -19.3, 71.4])
261
262 # Solusi yang diberikan lu dan b
263
264 # Output
265
266 # Definisi matriks A
267 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
268
269 # Definisi vektor b
270 b = np.array([7.85, -19.3, 71.4])
271
272 # Solusi yang diberikan lu dan b
273
274 # Output
275
276 # Definisi matriks A
277 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
278
279 # Definisi vektor b
280 b = np.array([7.85, -19.3, 71.4])
281
282 # Solusi yang diberikan lu dan b
283
284 # Output
285
286 # Definisi matriks A
287 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
288
289 # Definisi vektor b
290 b = np.array([7.85, -19.3, 71.4])
291
292 # Solusi yang diberikan lu dan b
293
294 # Output
295
296 # Definisi matriks A
297 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
298
299 # Definisi vektor b
300 b = np.array([7.85, -19.3, 71.4])
301
302 # Solusi yang diberikan lu dan b
303
304 # Output
305
306 # Definisi matriks A
307 A = np.array([[1., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])
308
309 # Definisi vektor b
310 b = np.array([7.85, -19.3, 71.4])
311
312 # Solusi yang diberikan lu dan b
313
314 # Output
315
316 # Defin
```



Nama : Ricky Suryanto Parsaulian Samosir

NPM : 202010225254

Kelas : TF3A6

Codingnya:

```
# Ricky Suryanto Parsaulian Samosir 202010225254 TF3A6
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, -19.3, 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)
```



Nama : Ricky Suryanto Parsaulian Samosir

NPM : 202010225254

Kelas : TF3A6

1.3 METODE GAUSS SEIDEL

```
1 # Ricky Suryanto Parsaulian Samosir 202010225254 TF3A6
2 # Iterasi Gauss Seidel
3
4 # Definisi Persamaan yang akan diselesaikan
5 # Dalam bentuk matriks secara diagonal
6 f1 = lambda x,y,z: (7.85+0.1*y+0.2*z)/3
7 f2 = lambda x,y,z: (-19.3-0.1*x+0.3*z)/7
8 f3 = lambda x,y,z: (71.4-0.5*x+0.2*y)/10
9
10 # Inisialisasi awal
11 x0 = 1
12 y0 = 2
13
14 # Iterasi Gauss Seidel
15 for i in range(1, 100000):
16     x = f1(x0, y0, z0)
17     y = f2(x, y0, z0)
18     z = f3(x, y, z0)
19     x0, y0, z0 = x, y, z
20
21 # Output
22 print("Solusi: x=3.0000, y=-2.5000 and z = 7.0000")
```

Output:

```
PS C:\Users\HP\Downloads> cd "C:\Users\HP\Downloads"; & "C:\Users\HP\AppData\Local\Programs\Python\Python77\python.exe"
" C:\Users\HP\vscode\extensions\ms-python.python-2021.10.1361161279\pythonFiles\lib\python\deleguy\launcher" "52223" "--"
"C:\Users\HP\Downloads\GaussSeidel.py"
Input Toleransi error: 0.00001

Step  x      y      z
1      2.8167 -2.7117 7.0013
2      2.9930 -2.4996 7.0002
3      3.0000 -2.5000 7.0000
4      3.0000 -2.5000 7.0000

Solusi: x=3.0000, y=-2.5000 and z = 7.0000
```



Nama : Ricky Suryanto Parsaulian Samosir

NPM : 202010225254

Kelas : TF3A6

Codingnya:

```
# Ricky Suryanto Parsaulian Samosir 202010225254 TF3A6
# 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
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%.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))
```