

Nama : Annisa Charisma Wijayanti

NIM : 211220122140086

Mata Kuliah : Metode Numerik

Jurusan : Teknik Komputer

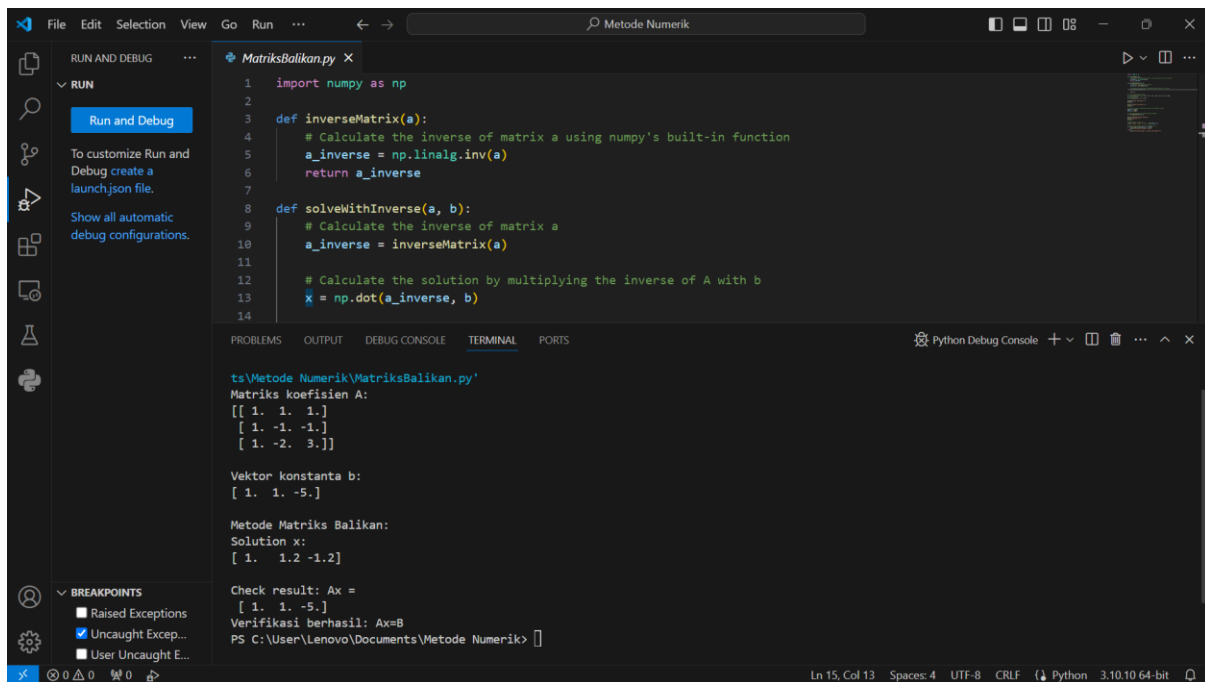
Implementasi Sistem Persamaan Linear

1. Metode Matriks Balikan

Metode matriks balikan digunakan untuk menyelesaikan sistem persamaan linear $Ax=b$ dengan mengalikan invers dari matriks koefisien A dengan vektor konstanta b . Penjelasan alur kode dengan menggunakan bahasa pemrograman python yaitu:

- a. Import NumPy
- b. Definisi Fungsi `inverseMatrix(a)`:
 - Fungsi ini menerima matriks A sebagai argumen.
 - Menggunakan fungsi `np.linalg.inv(a)` dari NumPy untuk menghitung invers dari matriks A .
 - Mengembalikan matriks invers A^{-1} .
- c. Definisi Fungsi `solveWithInverse(a, b)`:
 - Fungsi ini menerima dua argumen, yaitu matriks koefisien A dan vektor konstanta b .
 - Memanggil fungsi `inverseMatrix(a)` untuk menghitung invers dari matriks A .
 - Menghitung solusi x dengan mengalikan invers dari matriks A dengan vektor b menggunakan fungsi `np.dot(a_inverse, b)`.
 - Mengembalikan vektor solusi x .
- d. Inisialisasi Matriks koefisien A dan vektor konstanta b dengan nilai-nilai tertentu.
- e. Matriks koefisien asli A dan vektor konstanta asli b disalin untuk verifikasi hasil nantinya.
- f. Pemanggilan Fungsi `solveWithInverse(a,b)` sebagai argumen untuk mendapatkan solusi dari sistem persamaan linear $Ax=b$.
- g. Cetak Solusi x
- h. Verifikasi kebenaran solusi yang ditemukan, hasil dari perkalian matriks koefisien asli A dengan solusi x (yaitu Ax) dihitung menggunakan `np.dot(aOrig, x)`, lalu hasilnya dibandingkan dengan vektor konstanta asli b . Jika hasilnya mendekati nol

atau sangat kecil (dengan menggunakan `np.allclose()`), maka verifikasi berhasil. Jika tidak, maka verifikasi gagal.



```
1 import numpy as np
2
3 def inverseMatrix(a):
4     # Calculate the inverse of matrix a using numpy's built-in function
5     a_inverse = np.linalg.inv(a)
6     return a_inverse
7
8 def solveWithInverse(a, b):
9     # Calculate the inverse of matrix a
10    a_inverse = inverseMatrix(a)
11
12    # Calculate the solution by multiplying the inverse of A with b
13    x = np.dot(a_inverse, b)
14
15
16 ts\Metode Numerik\MatriksBalikan.py'
17 Matriks koefisien A:
18 [[ 1.  1.  1.]
19 [ 1. -1. -1.]
20 [ 1. -2.  3.]]
21
22 Vektor konstanta b:
23 [ 1.  1. -5.]
24
25 Metode Matriks Balikan:
26 Solution x:
27 [ 1.  1.2 -1.2]
28
29 Check result: Ax =
30 [ 1.  1. -5.]
31 Verifikasi berhasil: Ax=B
32 PS C:\User\Lenovo\Documents\Metode Numerik>
```

2. Metode Dekomposisi LU Gauss

Metode dekomposisi LU Gauss adalah salah satu metode dalam aljabar linier yang digunakan untuk menyelesaikan sistem persamaan linear $Ax=b$. Metode ini melibatkan faktorisasi matriks koefisien A menjadi dua matriks segitiga, yaitu matriks segitiga bawah L dan matriks segitiga atas U , di mana $A=LU$. Pada metode ini, proses dekomposisi dilakukan dengan melakukan eliminasi Gauss pada matriks koefisien A . Faktorisasi dilakukan dengan menemukan matriks segitiga atas U terlebih dahulu, diikuti dengan pencarian matriks segitiga bawah L yang memungkinkan $A=LU$. Penjelasan alur kode dengan menggunakan bahasa pemrograman python yaitu:

a. Definisi Fungsi `gaussElim(a, b):`

- Fungsi ini menerima dua argumen, yaitu matriks koefisien A dan vektor konstanta b .
- Variabel n diinisialisasi dengan panjang vektor b , yang juga merupakan ukuran matriks koefisien A .
- Fase eliminasi dilakukan menggunakan nested loop: iterasi pertama untuk setiap baris (kecuali baris terakhir), dan iterasi kedua untuk setiap baris di bawah baris saat ini.

- Pada setiap iterasi, dilakukan pengecekan apakah elemen diagonal utama a_{kk} tidak nol. Jika nol, langkah eliminasi dilewati karena akan menghasilkan pembagian dengan nol.
 - Jika elemen diagonal utama tidak nol, faktor pengali λ dihitung dan digunakan untuk memperbarui matriks A dan vektor b .
 - Tahap substitusi mundur dilakukan untuk menghitung solusi x .
 - Vektor solusi x dikembalikan.
- b. Inisialisasi Matriks koefisien A dan vektor konstanta b diinisialisasi dengan nilai-nilai tertentu.
 - c. Menyalin Matriks koefisien asli A dan vektor konstanta asli b untuk verifikasi hasil nantinya.
 - d. Panggil Fungsi `gaussElim(a, b)` dengan matriks koefisien A dan vektor konstanta b sebagai argumen untuk mendapatkan solusi dari sistem persamaan linear $Ax = b$.
 - e. Cetak Matriks A dan Solusi x untuk ditampilkan kepada pengguna.
 - f. Verifikasi Hasil: Hasil dari Ax dihitung menggunakan `np.dot(aOrig, x)`, lalu hasilnya dibandingkan dengan vektor konstanta asli b . Jika hasilnya mendekati nol atau sangat kecil (dengan menggunakan `np.allclose()`), maka verifikasi berhasil. Jika tidak, maka verifikasi gagal.

The screenshot shows a VS Code editor window titled 'Metode Numerik' with a Python file named 'LUGauss.py'. The code implements a function `gaussElim(a, b)` for LU decomposition. It includes comments for the elimination phase and the calculation of the multiplier λ . The script also prints the original matrix A and the solution vector x .

```

1
2 import numpy as np
3 def gaussElim(a,b):
4     n = len(b)
5     # Elimination phase
6     for k in range(0,n-1):
7         for i in range(k+1,n):
8             if a[i,k] != 0.0:
9                 #if not null define λ
10                lam = a[i,k]/a[k,k]
11                #we calculate the new row of the matrix
12                a[i,k+1:n] = a[i,k+1:n] - lam*a[k,k+1:n]
13                #we update vector b
14                b[i] = b[i] - lam*b[k]

```

The terminal output shows the execution of the script, displaying the matrix A and the solution vector x :

```

PS C:\User\Lenovo\Documents\Metode Numerik> & 'c:\Users\LENOVO\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\LENOVO\.vscode\extensions\ms-python.debugpy-2024.6.0-win32-x64\bundle\libs\debugpy\adapter\..\..\debugpy\launcher' '62354' '--' 'C:\User\Lenovo\Documents\Metode Numerik\LUGauss.py'

Metode Dekomposisi LU Gauss:
Matriks koefisien A:
[[ 1.  1.  1.]
 [ 1. -2. -2.]
 [ 1. -3.  5.]]
x =
[ 1.  1.2 -1.2]

Check result: [a]{x} - x =
[2.22044605e-16 0.00000000e+00 0.00000000e+00]
PS C:\User\Lenovo\Documents\Metode Numerik>

```

3. Metode Dekomposisi Crout

Metode Dekomposisi Crout merupakan metode yang memfaktorkan matriks koefisien A dari sistem persamaan $AX = Y$ menjadi hasil kali suatu matriks segitiga bawah L yang elemen diagonal utamanya bernilai tak nol dan matriks segitiga atas U yang elemen diagonal utamanya bernilai satu. Dengan demikian sistem persamaan linear akan berubah menjadi $AX = LUX = Y$. Pada metode ini, proses dekomposisi dilakukan dengan pendekatan Crout, di mana elemen-elemen matriks L dan U dihitung secara bergantian. Faktorisasi dilakukan dengan menemukan matriks segitiga bawah L terlebih dahulu, diikuti dengan pencarian matriks segitiga atas U yang memungkinkan $A=LU$. Penjelasan alur kode dengan menggunakan bahasa pemrograman python yaitu:

a. Definisi Fungsi `croutDecomposition(a)`:

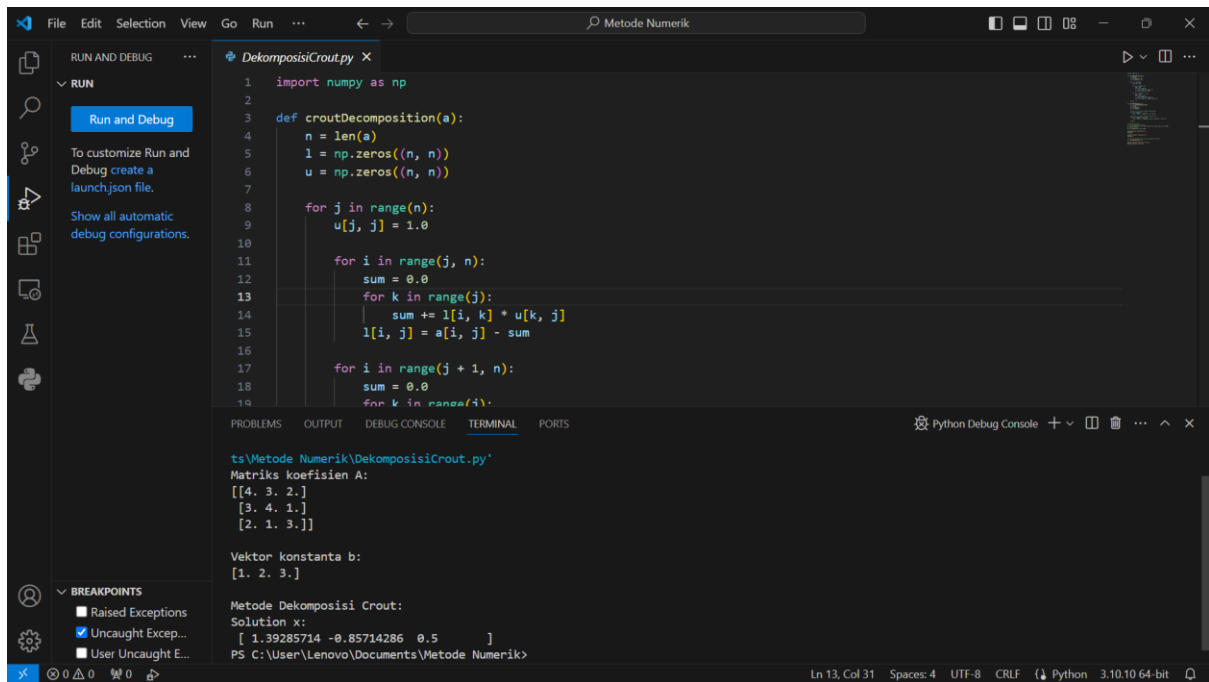
- Fungsi ini menerima matriks koefisien A sebagai argumen.
- Membuat matriks L dan U dengan ukuran yang sama dengan matriks koefisien A , dan diinisialisasi dengan nilai nol.
- Melakukan dekomposisi Crout dengan menggunakan nested loop:
 - Iterasi pertama untuk setiap kolom. Iterasi kedua untuk setiap baris, dimulai dari indeks kolom saat ini hingga indeks terakhir.
 - Pada setiap iterasi, menghitung elemen L dan U menggunakan rumus dekomposisi Crout.
- Mengembalikan matriks L dan U .

b. Definisi Fungsi `solveWithCrout(a, b)`:

- Fungsi ini menerima matriks koefisien A dan vektor konstanta b sebagai argumen.
- Memanggil fungsi `croutDecomposition(a)` untuk mendapatkan matriks L dan U .
- Menghitung solusi x menggunakan substitusi maju dan mundur.
 - Substitusi maju dilakukan untuk menyelesaikan sistem $Ly = b$ dengan matriks segitiga bawah L .
 - Substitusi mundur dilakukan untuk menyelesaikan sistem $Ux = y$ dengan matriks segitiga atas U .
- Mengembalikan vektor solusi x .

c. Inisialisasi Matriks koefisien A dan vektor konstanta b dengan nilai-nilai tertentu.

- d. Panggil Fungsi `solveWithCrout(a, b)` dengan matriks koefisien A dan vektor konstanta b sebagai argumen untuk mendapatkan solusi dari sistem persamaan linear $Ax = b$.
- e. Cetak Matriks A dan Solusi x



The screenshot shows a Python IDE with a file named `DekomposisiCrout.py`. The code implements a Crout decomposition function. The output window displays the results of running the program.

```
1 import numpy as np
2
3 def croutDecomposition(a):
4     n = len(a)
5     l = np.zeros((n, n))
6     u = np.zeros((n, n))
7
8     for j in range(n):
9         u[j, j] = 1.0
10
11         for i in range(j, n):
12             sum = 0.0
13             for k in range(j):
14                 sum += l[i, k] * u[k, j]
15             l[i, j] = a[i, j] - sum
16
17         for i in range(j + 1, n):
18             sum = 0.0
19             for k in range(i):
```

Output:

```
ts\Metode Numerik\DekomposisiCrout.py'
Matriks koefisien A:
[[4. 3. 2.]
 [3. 4. 1.]
 [2. 1. 3.]]

Vektor konstanta b:
[1. 2. 3.]

Metode Dekomposisi Crout:
Solution x:
[ 1.39285714 -0.85714286  0.5       ]
PS C:\User\Lenovo\Documents\Metode Numerik>
```