**LAPORAN SIMULASI**

**ALJABAR LINEAR**



Modul 3

Ajewinata

240306009

Program Studi Teknologi Informasi

Fakultas Dakwah dan Ilmu Komunikasi

Universitas Islam Negeri Mataram

2025

|  |  |
| --- | --- |
| Pertemuan | 3 |
| Topik | Sistem Persamaan Linear |
| Repository | - |
| Tanggal | - |

1. **Tujuan:**

Simulasi ini dilakukan dengan tujuan:

1. Menguasai Metode Penyelesaian Sistem Persamaan Linear dengan NumPy
2. Memahami Penggunaan SymPy untuk Penyelesaian Simbolik
3. Menyelesaikan Sistem Persamaan Linear Secara Simbolik dengan SymPy
4. Membandingkan Solusi Numerik (NumPy) dan Simbolik (SymPy)
5. **Requirement**

1. Sistem Operasi yang digunakan : Windows 11 pro 64-bit (10.0, Build 26100)

2. Browser : Google Chrome, Versi 131.0.6778.265 (Build Resmi) (64 bit)

3. Bahasa pemrograman : Python 3.13.1

1. **Kompetensi yang diharapkan:**

Setelah mengikuti praktikum ini, peserta diharapkan mampu:

1. Mampu menggunakan fungsi-fungsi NumPy (seperti np.linalg.solve()) untuk mencari solusi sistem persamaan linear.
2. Menggunakan library SymPy untuk merepresentasikan variabel dan persamaan secara simbolik.
3. Menggunakan fungsi-fungsi SymPy (seperti sympy.solve()) untuk mendapatkan solusi sistem persamaan linear dalam bentuk simbolik.
4. Membandingkan hasil penyelesaian sistem persamaan linear menggunakan pendekatan numerik (NumPy) dan simbolik (SymPy), serta memahami kelebihan dan kekurangan masing-masing metode.
5. **Durasi Pertemuan: 50 menit**
6. **Bahan Simulasi:**

Hitunglah nilai variable dari sistem persamaan linear berikut!

1. 2x + 3y = 7

x - y = 1

1. x + 2y + z = 10

3x − y + 2z = 5

−2x + 3y − z = −9

1. **Hasil Percobaan**
2. Source code
3. SPLDV

* Numpy

|  |
| --- |
| import numpy as np  Matriks\_A = np.array([[2,3],  [1,-1]])  Matriks\_B = np.array([7,1])  hasil = np.linalg.solve (Matriks\_A,Matriks\_B)  print ("Hasil SPLDV adalah:")  print ("x =",hasil[0],"y =",hasil[1]) |

* Sympy

|  |
| --- |
| import sympy as sp  Matriks\_A = sp.Matrix([[2,3],  [1,-1]])  Matriks\_B = sp.Matrix([7,1])  hasil = Matriks\_A.solve (Matriks\_B)  print ("Hasil SPLDV adalah:")  print ("x =",hasil[0],"y =",hasil[1]) |

1. SPLTV

* Numpy

|  |
| --- |
| import numpy as np  Matriks\_A = np.array([[1, 2, 1],  [3, -1, 2],  [-2, 3, -1]])  Matriks\_B = np.array([10, 5, -9])  hasil = np.linalg.solve(Matriks\_A, Matriks\_B)  print("Hasil SPLTV adalah:")  print("x =", hasil[0], ", y =", hasil[1], ", z =", hasil[2]) |

* Sympy

|  |
| --- |
| import sympy as sp  Matriks\_A = sp.Matrix([[1, 2, 1],  [3, -1, 2],  [-2, 3, -1]])  Matriks\_B = sp.Matrix([10, 5, -9])  if Matriks\_A.det() == 0:  print ("Matriks tidak memiliki solusi")  else:  hasil = Matriks\_A.solve(Matriks\_B)  print("Hasil SPLTV adalah:")  print("x =", hasil[0], ", y =", hasil[1], ", z =", hasil[2]) |

1. Hasil Program
2. SPLDV

* Numpy

|  |
| --- |
|  |

* Sympy

|  |
| --- |
|  |

1. SPLTV

* Numpy

|  |
| --- |
|  |

* Sympy

|  |
| --- |
|  |

1. **Analisis Data**
2. SPLDV

* Numpy

|  |
| --- |
| import numpy as np |

Mengimpor pustaka numpy dan memberikan alias np.

|  |
| --- |
| matriks\_a = np.array([[2,3],  [1-1]]) |

Membuat list bersarang yang mewakili matriks 2x2, kemudian mengubahnya menjadi array numpy dan menyimpannya ke dalam variabel matriks\_A.

|  |
| --- |
| matriks\_b = np.array([7,1]) |

Membuat list yang mewakili matriks 1x2, kemudian mengubahnya menjadi array numpy dan menyimpannya ke dalam variabel matriks\_B.

|  |
| --- |
| hasil = np.linalg.solve (Matriks\_A,Matriks\_B) |

Melakukan penyelesaian sistem persamaan linear antara nilai dari Variabel Matriks\_A dan Matriks\_B dengan menggunakan fungsi dari numpy yaitu np.linalg.solve(). yang digunakan untuk mencari solusi dari sistem persamaan linear dalam bentuk matriks. Kemudian menyimpannya ke dalam variabel hasil.

|  |
| --- |
| print ("Hasil SPLDV adalah:") |

Mencetak teks “Hasil SPLDV adalah”

|  |
| --- |
| print ("x =",hasil[0],"y =",hasil[1]) |

Menampilkan teks “x =” diikuti oleh nilai dari indeks ke-0 pada array hasil, serta teks “y =” diikuti oleh nilai dari indeks ke-1

* Sympy

|  |
| --- |
| import sympy as sp |

Mengimpor pustaka sympy dan memberikan alias sp.

|  |
| --- |
| matriks\_a = sp.Matriks([[2,3],  [1-1]]) |

Membuat list bersarang yang mewakili matriks 2x2, kemudian mengubahnya menjadi objek matriks dan menyimpannya ke dalam variabel matriks\_A.

|  |
| --- |
| matriks\_B = sp.Matriks([7,1]) |

Membuat list yang mewakili matriks 1x2, kemudian mengubahnya menjadi objek matriks dan menyimpannya ke dalam variabel matriks\_B.

|  |
| --- |
| hasil = Matriks\_A,.solve (Matriks\_B) |

Melakukan penyelesaian sistem persamaan linear antara nilai dari Variabel Matriks\_A dan Matriks\_B dengan menggunakan fungsi dari sympy yaitu .solve(), yang digunakan untuk mencari solusi dari sistem persamaan linear dalam bentuk matriks. Maupun umum Kemudian menyimpannya ke dalam variabel hasil.

|  |
| --- |
| print ("Hasil SPLDV adalah:") |

Mencetak teks “Hasil SPLDV adalah”

|  |
| --- |
| print ("x =",hasil[0],"y =",hasil[1]) |

Menampilkan teks “x =” diikuti oleh nilai dari indeks ke-0 pada array hasil, serta teks “y =” diikuti oleh nilai dari indeks ke-1

1. SPLTV

* Numpy

|  |
| --- |
| import numpy as np |

Mengimpor pustaka numpy dan memberikan alias np.

|  |
| --- |
| matriks\_a = np.array([[1,2,1],  [3,-1,2],  [-2,3,-1]]) |

Membuat list bersarang yang mewakili matriks 3x3, kemudian mengubahnya menjadi array numpy dan menyimpannya ke dalam variabel matriks\_A.

|  |
| --- |
| matriks\_b = np.array([10, 5,-9]) |

Membuat list yang mewakili matriks 1x2, kemudian mengubahnya menjadi array numpy dan menyimpannya ke dalam variabel matriks\_B.

|  |
| --- |
| hasil = np.linalg.solve (Matriks\_A,Matriks\_B) |

Melakukan penyelesaian sistem persamaan linear antara nilai dari Variabel Matriks\_A dan Matriks\_B dengan menggunakan fungsi dari numpy yaitu np.linalg.solve(). yang digunakan untuk mencari solusi dari sistem persamaan linear dalam bentuk matriks. Kemudian menyimpannya ke dalam variabel hasil.

|  |
| --- |
| print ("Hasil SPLDV adalah:") |

Mencetak teks “Hasil SPLDV adalah”

|  |
| --- |
| print ("x =", hasil[0], "y =", hasil[1], ”z =”, hasil[2]) |

Menampilkan teks “x =” diikuti oleh nilai dari indeks ke-0 pada array hasil, teks “y =” diikuti oleh nilai dari indeks ke-1 dan teks “z =” diikuti oleh nilai dari indeks ke-2

* Sympy

|  |
| --- |
| import sympy as sp |

Mengimpor pustaka sympy dan memberikan alias sp.

|  |
| --- |
| matriks\_a = sp.Matriks([[1,2,1],  [3,-1,2],  [-2,3,-1]]) |

Membuat list bersarang yang mewakili matriks 3x3, kemudian mengubahnya menjadi objek matriks dan menyimpannya ke dalam variabel matriks\_A.

|  |
| --- |
| matriks\_B = sp.Matriks([10,5,-9]) |

Membuat list yang mewakili matriks 1x2, kemudian mengubahnya menjadi objek matriks dan menyimpannya ke dalam variabel matriks\_B.

|  |
| --- |
| If Matriks\_A.det() == 0: |

Mengecek apakah determinan variabel matriks\_A adalah 0, jika benar blok code di bawah akan dijalankan

|  |
| --- |
| print(“Tidak ada solusi”) |

Mencetak teks “Tidak ada solusi”

|  |
| --- |
| else: |

Mengecek apakah kondisi sebelumnya tidak terpenuhi kemudian menjalankan blok kode di bawahnya.

|  |
| --- |
| hasil = Matriks\_A,.solve (Matriks\_B) |

Melakukan penyelesaian sistem persamaan linear antara nilai dari Variabel Matriks\_A dan Matriks\_B dengan menggunakan fungsi dari sympy yaitu .solve(), yang digunakan untuk mencari solusi dari sistem persamaan linear dalam bentuk matriks. Maupun umum Kemudian menyimpannya ke dalam variabel hasil.

|  |
| --- |
| print ("Hasil SPLDV adalah:") |

Mencetak teks “Hasil SPLDV adalah”

|  |
| --- |
| print ("x =",hasil[0],"y =",hasil[1]) |

Menampilkan teks “x =” diikuti oleh nilai dari indeks ke-0 pada array hasil, teks “y =” diikuti oleh nilai dari indeks ke-1 dan teks “z =” diikuti oleh nilai dari indeks ke-2

1. **Kesimpulan**

Simulasi ini menunjukkan bahwa NumPy sangat efisien untuk menyelesaikan sistem persamaan linear secara numerik dengan np.linalg.solve(), sedangkan SymPy memungkinkan penyelesaian secara simbolik melalui .solve(), menghasilkan solusi dalam bentuk eksak. Meskipun hasil yang diperoleh dari kedua metode sama, pendekatan dan kegunaannya berbeda. NumPy lebih cocok untuk perhitungan cepat, sedangkan SymPy lebih tepat untuk analisis simbolik. Pemeriksaan determinan pada SymPy juga penting untuk memastikan adanya solusi unik. Secara keseluruhan, praktikum ini membantu memahami penerapan kedua metode dalam menyelesaikan sistem persamaan linear.

**DAFTAR PUSTAKA**

https://classroom.google.com/u/2/c/NzU2NDM0Njg1NzY4/m/NzcyNDM3NzcwMjU0/details