LAPORAN TUGAS BESAR

PROGRAM MATRIKS MENGGUNAKAN BAHASA PYTHON

Untuk memenuhi salah satu tugas besar "Aljabar Geometri "

Dosen Pengampu : Ahmad Zamaksyari Sidiq,M.T.

Di susun oleh :

Kelompok 3

Lukmanul Hakim	10222042
Maulia Astuti Suryaningrum	10222052
Bizan Afzani Fasya	10222164
M. Faisal Mubarok	10222039
Riki Nugraha	10222059

SEKOLAH TINGGI TEKNOLOGI CIPASUNG

PROGRAM STUDI INFORMATIKA

TASIKMALAYA

2023

KATA PENGANTAR

Puji syukur kita panjatkan kehadirat Allah Swt. yang telah memberikan rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan tugas yang berjudul "Laporan Program Matriks Menggunakan Bahasa Pyhton " ini tepat pada waktunya. Adapun tujuan dari penulisan dari laporan ini adalah untuk memenuhi tugas pada mata kuliah Aljabar Geometri. Selain itu, laporan ini juga bertujuan untuk menambah wawasan mengenai program Matriks Menggunakan Bahasa Pyhton di kehidupan sehari-hari bagi para pembaca dan juga bagi penulis.

Terlebih dahulu, saya mengucapkan terima kasih kepada Bapak Ade Zamakhsyari Sidiq, MT, selaku Dosen Aljabar Geometri yang telah memberikan tugas ini sehingga dapat menambah pengetahuan dan wawasan ini. Saya juga mengucapkan terima kasih kepada semua pihak yang tidak dapat saya sebutkan semua, terima kasih atas bantuannya sehingga sehingga saya dapat menyelesaikan tugas ini.

Kemudian, saya menyadari bahwa tugas yang saya tulis ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun kami butuhkan demi kesempurnaan laporan ini.

Tasikmalaya, 20 Desember 2023

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

DESKRIPSI MASALAH

Matriks dan operasinya merupakan hal yang erat kaitannya dengan bidang aljabar linier, dimana matriks itu sendiri adalah kumpulan bilangan, simbol, berbentuk persegi atau persegi panjang yang disusun menurut baris dan kolom. Bilangan-bilangan yang terdapat di suatu matriks disebut dengan entri atau anggota dari matriks.

Matriks *fuzzy* adalah matriks yang entri-entrinya berada pada selang tutup [0,1]. Semua matriks *fuzzy* merupakan matriks, akan tetapi untuk sebarang matriks belum tentu merupakan matriks *fuzzy*. Penjumlahan, perkalian skalar pada matriks *fuzzy* berbeda dengan matriks pada umumnya. Penjumlahan pada matriks *fuzzy* didefinisikan sebagai maksimum dari entri-entri yang bersesuaian dan perkalian pada matriks *fuzzy* didefinisikan sebagai minimun dari matriks *fuzzy* tersebut.

Suatu matriks memiliki invers jika matriks tersebut berukuran $n \times n$ dan determinan matriks tersebut tidak sama dengan 0, untuk mencari invers dari matriks selain berukuran $n \times n$ digunakan generalisasi invers. Matriks X dikatakan generalisasi invers (g-invers) dari matriks A jika matriks X ada sedemikian sehingga AXA = A. Matriks fuzzy yang memiliki g-invers disebut matriks *fuzzy* 1 regular. Matriks *fuzzy* regular A memiliki banyak g-invers, himpunan semua g-invers dari matriks A dinotasikan $Af1g$.

Skripsi ini akan menjelaskan kembali sebagian isi dari buku *Fuzzy Matrix Theory and Application* oleh A. R. Meenakshi, yang mengacu kepada sifat-sifat dari matriks *fuzzy* regular.

BAB III

TEORI SINGKAT SISTEM PERSAMAAN LINIER

3.1 Metode Eliminasi Gaus

Metode Eliminasi Gauss adalah algoritma yang digunakan untuk menyelesaikan sistem persamaan linear. Algoritma ini terdiri dari serangkaian operasi yang dilakukan pada matriks koefisien dari sistem persamaan tersebut. Walau akan mengubah bentuk matriks, operasi-operasi tersebut tidak akan mengubah solusi dari sistem persamaan. Hal ini memungkinkan matriks koefisien dibentuk menjadi sebuah matriks segitiga atas, sehingga solusi sistem persamaan dapat ditentukan dengan cukup melakukan eliminasi variabel secara berulang. Eliminasi Gauss juga dapat digunakan untuk menghitung rank dari matriks, determinan dari matriks persegi, dan invers dari matriks nonsingular. Metode ini dinamai dari matematikawan Carl Friedrich Gauss (1777–1855), walaupun beberapa kasus khusus dari metode ini — tapi tanpa dilengkapi bukti — sudah dikenal oleh matematikawan Tionghoa semenjak tahun 179 M.

Metode ini berangkat dari kenyataan bahwa bila matriks *A* berbentuk *segitiga atas* seperti sistem persamaan berikut ini

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ 0 & a_{22} & a_{23} & \dots & a_{2n} \\ 0 & 0 & a_{33} & \dots & a_{3n} \\ \vdots & & & & \\ 0 & 0 & 0 & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

maka solusinya dapat dihitung dengan **teknik penyulihan mundur** (*backward substitution*):

$$a_{mm}x_n = b_n \rightarrow x_n = b_n/a_{nn}$$

$$a_{n-1,n-1}x_{n-1} + a_{n-1,n}\,x_n = b_{n-1} \rightarrow x_{n-1} = \frac{b_{n-1} - a_{n-1,\ n}x_n}{a_{n-1,\ n-1}}$$

$$a_{n-2,\ n-2}x_{n-2} + a_{n-2,\ n-1}x_{n-1} + a_{n-2,\ n}x_n = b_{n-2} \rightarrow x_{n-2} = \frac{b_{-1} - a_{n-2,\ n-1}x_{n-1} - a_{n-2,\ n}x_n}{a_{n-2,\ n-2}}$$

Sekali x_n, x_{n-1}, x_{n-2}, ..., x_{k-1} diketahui, maka nilai x_k dapat dihitung dengan

$$X_k = \frac{b_k - \sum_{j=k+1}^{n} a_{kj} x_j}{a_{kk}} \mid k = n-1, n-2, \dots, 1 \; dan \; a_{kk} \neq 0$$

Kondisi $a_{kk} \neq 0$ sangat penting, sebab bila $a_{kk} = 0$, jika kondisi tersebut tidak terkenuhi maka SPL tidak mempunyai jawaban.

Contoh:

Selesaikan sistem persamaan lanjar berikut dengan teknik penyulihan mundur

$4x1 - x2 + 2x3 + 3x4 = 20$

$-2x2 + 7x3 - 4x4 = -7$

$6x3 + 5x4 = 4$

$3x4 = 6$

Penyelesaian:

$x4 = 6/3 = 2$

$$x3 = \frac{(4 - 5(2)) = -1}{6}$$

$$x2 = \frac{-7 - 7(-1) + 4 = -4}{-2}$$

$$x1 = \frac{20 + 1(-4) - 2(-1) - 3(2) = 3}{4}$$

Jadi, solusinya adalah x = (3, -4, -1, 2)T.

Metode eliminasi Gauss pada prinsipnya bertujuan mentransformasi sistem $Ax = b$ menjadi sistem U_x= y . Dengan U adalah matriks *segitiga atas*. Selanjutnya solusi x dapat dihitung dengan teknik penyulihan mundur. Contohnya pada sistem dengan 4 persamaan lanjar berikut (Elemen matriks A dan vektor kolom b disatukan dalam bentuk satu bentuk matriks):

$$\underset{[A, b]}{\left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ a_{21} & a_{22} & a_{23} & a_{24} & b_2 \\ a_{31} & a_{32} & a_{33} & a_{34} & b_3 \\ a_{41} & a_{42} & a_{43} & a_{44} & b_4 \end{array}\right]} \xrightarrow[\text{menjadi } [U, y]]{\text{dieliminasi}} \underset{[U, y]}{\left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & b_2^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & b_3^{(2)} \\ 0 & 0 & 0 & a_{44}^{(3)} & b_4^{(3)} \end{array}\right]}$$

Tanda pangkat (1), (2), (3) menunjukkan bahwa elemen matriks A telah berubah satu kali, dua kali, dan tiga kali.

Proses eliminasi terdiri atas tiga operasi baris elementer:

1. *Pertukaran* : Urutan dua persamaan dapat ditukar karena pertukaran tersebut tidak mempengaruhi solusi akhir.
2. *Penskalaan* : Persamaan dapat dikali dengan konstanta bukan nol, karena perkalian tersebut tidak mempengaruhi solusi akhir.
3. *Penggantian* : Persamaan dapat diganti dengan penjumlahan persamaan itu dengan gandaan persamaan lain. Misalnya persamaan diganti dengan selisih persamaan itu dengan dua kali persamaan lain; yaitu

   $$baris_r := baris_r - m_{p,r}\, baris_p$$

Nilai $a_{r,r}$ pada posisi (r, r) yang digunakan untuk mengeliminasi x_r pada baris $r + 1, r + 2, ..., N$ dinamakan elemen *pivot* dan persamaan pada baris ke-r disebut **persamaan *pivot*** [MAT92]. Ada kemungkinan *pivot* bernilai nol sehingga pembagian dengan nol tidak dapat dielakkan. Tata-ancang eliminasi yang tidak mempedulikan nilai *pivot* adalah tatancang yang naif (*naive*) atau sederhana. Metode eliminasi Gauss seperti ini dinamakan **metode eliminasi Gauss naif** (*naive Gaussian elimination*), karena metodenya tidak melakukan pemeriksaan kemungkinan pembagian dengan nol. Pada metode eliminasi Gauss naif tidak ada operasi pertukaran baris dalam rangka menghindari *pivot* yang bernilai nol itu.

Contoh: Selesaikan sistem persamaan lanjar dengan metode eliminasi Gauss naif:

$2x_1 + 3x_2 - x_3 = 5$

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

$-2x_1 + 3x_2 - x_3 = 1$

Penyelesaian:

$$\begin{bmatrix} 2 & 3 & -1 & |5 \\ 4 & 4 & -3 & |3 \\ -2 & 3 & -1 & |1 \end{bmatrix} \underset{R_3 - \frac{-2}{2} R1}{\overset{R_2 - \frac{4}{2} R_1}{\sim}} \begin{bmatrix} 2 & 3 & -1 & |5 \\ 0 & -2 & -1 & |-7 \\ 0 & 6 & -2 & |6 \end{bmatrix} \overset{R_3 - \frac{6}{-2} R_2}{\sim} \begin{bmatrix} 2 & 3 & -1 & |5 \\ 0 & -2 & -1 & |-7 \\ 0 & 0 & -5 & |-15 \end{bmatrix}$$

Keterangan:

(i) elemen yang dicetak tebal menyatakan *pivot*.

(ii) simbol "~" menyatakan operasi baris elementer .

(iii) R_i menyatakan baris (*row*) ke-*i*

(iv) $R2$ - 4/2 $R1$ artinya elemen-elemen pada baris kedua dikurangi dengan dua kali elemen-elemen pada baris ke satu.

$$\begin{array}{rcl} R_2 & : & 4 \quad 4 \quad -3 \quad 3 \\ 2R_1 & : & 4 \quad 6 \quad -2 \quad 10 \ - \\ \hline R_2 - \frac{4}{2}R_1 & : & 0 \ -2 \quad -1 \quad -7 \end{array}$$ (Menjadi elemen baris ke -2)

Solusi sistem di peroleh dengan teknik penyulihan mundur sebagai berikut:

$-5x_3 = -15 \quad \rightarrow \quad x_3 = 3$

-5 $-2x_2 - x_3 = -7 \quad \rightarrow \quad x_2 = \frac{-7+3}{-2} = 2$

$2x_1 + 3x_2 - x_3 = 5 \quad \rightarrow \quad x_1 = (5 + 3 - 6) = 1$

Jadi solusinya adalah $x = (1,2,3)^T$

Kelemahan eliminasi Gauss naif

Jika *pivot app* = 0, baris ke-*k* tidak dapat digunakan untuk memgeliminasi elemen pada kolom *p*, karena terjadinya pembagian dengan nol. Oleh karena itu, *pivot* yang bernilai nol harus dihindari dengan tata-ancang (*strategy*) *pivoting.*

3.2 Eliminasi Gauss Jordan

Metode eliminasi Gauss-Jordan merupakan variasi dari metode eliminasi Gauss. Dalam hal ini, matriks *A* dieliminasi menjadi matriks identitas *I*. Di sini tidak diperlukan lagi teknik penyulihan mundur untuk memperoleh solusi SPL. Solusinya langsung diperoleh dari vektor kolom *b* hasil proses eliminasi.

Ax = *b* □□*Ix* = *b*'

Dalam bentuk matriks, eliminasi Gaus-Jordan ditulis sebagai

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13}\cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23}\cdots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33}\cdots & a_{3n} & b_3 \\ \vdots & & & & \vdots \\ a_{n1} & a_{n2} & a_{n3}\cdots & a_{nn} & b_n \end{array}\right] - \left[\begin{array}{cccccc|c} 1 & 0 & 0 & 0 & \vdots & 0 & b_1{}' \\ 0 & 1 & 0 & 0 & \vdots & 0 & b_2{}' \\ 0 & 0 & 1 & 1 & \vdots & 0 & b_3{}' \\ \vdots & & & & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \vdots & 1 & b_n{}' \end{array}\right]$$

Solusinya:

$x_1 = b_1'$

$x_2 = b_2'$

... ...

$x_n = b_n'$

Seperti pada metode eliminasi Gauss naif, metode eliminasi Gauss-Jordan naif tidak menerapkan tata-ancang pivoting dalam proses eliminasinya. Halnya metode eliminasi Gauss, tata-ancang *pivoting* dan penskalaan juga dapat diterapkan pada metoda ini untuk memperkecil galat pembulatan.

Contoh: Selesaikan sistem persamaan lanjar di bawah ini dengan metode eliminasi Gauss- Jordan.

$3x1 - 0.1x_2 - 0.2x_3 = 7.85$

$0.1x_1 + 7x_2 - 0.3x_3 = -19.3$

$0.3x_1 - 0.2x_2 + 10x_3 = 71.4$

Penyelesaian:

$$\left[\begin{array}{ccc|c} 3 & -0.1 & -0.2 & 7.85 \\ 0.1 & 7 & -0.3 & -19.3 \\ 0.3 & -0.2 & 10 & 71.7 \end{array}\right] \underset{\sim}{\frac{R_1}{3}} \left[\begin{array}{ccc|c} 1 & -0.0333333 & -0.0666667 & 2.61667 \\ 0.1 & 7 & -0.3 & -19.3 \\ 0.3 & -0.2 & 10 & 71.4 \end{array}\right]$$

$$\begin{array}{c} R_2 - 0.1\,R1 \\ R_3 - 0.3\,R1 \\ \sim \end{array} \left[\begin{array}{cc|cc} 1 & -0.0333333 & -0.0666667 & 2.61667 \\ 0 & 7.00333 & -0.2933333 & -19.5617 \\ 0 & -0.190000 & 10.0200 & 70.6150 \end{array}\right]$$

$$R_2/7.00333 \left[\begin{array}{ccc|c} 1 & -0.0333333 & -0.0666667 & 2.61667 \\ 0 & 1 & -0.0418848 & -2.79320 \\ 0 & -0.190000 & 10.01200 & 70.6150 \end{array}\right]$$

$$\begin{array}{c} R_1 - (-0.003333)R_2 \\ R_3 - (-0.190000)R_2 \end{array} \left[\begin{array}{ccc|c} 1 & 0 & -0.0680629 & 2.52356 \\ 0 & 1 & -0.0418848 & -2.79320 \\ 0 & 0 & 10.01200 & 70.0843 \end{array}\right]$$

$$R_3/10.0200 \left[\begin{array}{ccc|c} 1 & 0 & -0.0680629 & 2.52356 \\ 0 & 1 & -0.0418848 & -2.79320 \\ 0 & 0 & 1 & 7.00003 \end{array}\right]$$

$$\begin{array}{c} R_1 - (-0.0680629)R_3 \\ R_2 - (-0.0418848)R_2 \end{array} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 3.00000 \\ 0 & 1 & 0 & -0.50001 \\ 0 & 0 & 1 & 7.00003 \end{array}\right]$$

Solusi:

$x_1 = 3.00000$

$x_2 = -2.50001$

$x_3 = 7.00003$

Penyelesaian SPL dengan metode eliminasi Gauss-Jordan membutuhkan jumlah komputasi yang lebih banyak daripada metode eliminasi Gauss. Karena alasan itu, metode eliminasi Gauss sudah cukup memuaskan untuk digunakan dalam penyelesaian SPL.

3.3 Metoda Gauss Seidel

Kecepatan konvergen pada lelaran Jacobi dapat dipercepat bila setiap harga x_i yang baru dihasilkan segera dipakai pada persamaan berikutnya untuk menentukan harga x_{i+1} yang lainnya.

Lelaran pertama:

$$x_1^{(1)} = \frac{b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)} - a_{14}x_4^{(0)}}{a_{11}}$$

$$x_2^{(1)} = \frac{b_1 - a_{21}x_1^{(0)} - a_{23}x_3^{(0)} - a_{24}x_4^{(0)}}{a_{22}}$$

$$x_3^{(1)} = \frac{b_3 - a_{31}x_1^{(1)} - a_{32}x_2^{(1)} - a_{34}x_4^{(0)}}{a_{44}}$$

$$x_4^{(1)} = \frac{b_4 - a_{41}x_1^{(1)} - a_{42}x_2^{(1)} - a_{43}x_3^{(1)}}{a_{44}}$$

Lelaran kedua:

$$x_1^{(2)} = \frac{b_1 - a_{12}x_2^{(1)} - a_{13}x_3^{(1)} - a_{14}x_4^{(1)}}{a_{11}}$$

$$x_2^{(2)} = \frac{b_1 - a_{21}x_1^{(2)} - a_{23}x_3^{(1)} - a_{24}x_4^{(1)}}{a_{22}}$$

$$x_3^{(2)} = \frac{b_3 - a_{31}x_1^{(2)} - a_{32}x_2^{(2)} - a_{34}x_4^{(1)}}{a_{44}}$$

$$x_4^{(2)} = \frac{b_4 - a_{41}x_1^{(2)} - a_{42}x_2^{(2)} - a_{43}x_3^{(2)}}{a_{44}}$$

Rumus umum:

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{n} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k}}{a_{ii}}, k = 0,1,2, ...$$

Contoh:

Tentukan Solusi SPL

$4x - y + z = 7$

$4x - 8y + z = -21$

$-2x + y + 5z = 15$

dengan nilai awal $P0 = (x_0, y_0, z_0) = (1, 2, 2)$.

(Solusi sejatinya adalah (2, 4, 3))

Penyelesaian:

(a) Metode lelaran Jacobi

Persamaan lelarannya:

$$x_{r+1} = \frac{7 + y_r - z_r}{4}$$

$$y_{r+1} = \frac{21 + 4x_r - z_r}{8}$$

$$z_{r+1} = \frac{15 + 2x_r - y_r}{5}$$

Lelarannya:

$$x_1 = \frac{7 + 2 - 2}{4} = 1.75$$

$$y_1 = \frac{21 + 4(1) + 2}{8} = 3.375$$

$$z_1 = \frac{15 + 2(1) - 2}{5} = 3.000$$

$$x_2 = \frac{7 + 3.375 - 3.00}{4} = 1.84375$$

$$y_2 = \frac{21 + 4(3.375) - 3.00}{8} = 3.875$$

$$z_2 = \frac{15 + 2(1.75) - 3.375}{5} = 3.025$$

x_{19}=2.00000000

y_{19}=4.00000000

z_{19}=3.00000000

(b) Metode lelaran Gauss-Seidel Persamaan lelarannya,

$$x_{r+1} = \frac{7 + y_r - z_r}{4}$$

$$y_{r+1} = \frac{21 + 4x_r - z_r}{8}$$

$$z_{r+1} = \frac{15 + 2x_r - y_r}{5}$$

Lelarannya:

$$x_1 = \frac{7+2-2}{4} = 1.75$$

$$y_1 = \frac{21+4(1.75)+2}{8} = 3.75$$

$$z_1 = \frac{15+2(1.75)-3.75}{5} = 3.000$$

$$x_2 = \frac{7+3.75-2.95}{4} = 1.95$$

$$y_2 = \frac{7+3.75-2.95}{8} = 3.96875$$

$$z_2 = \frac{15+2(1.95)-3.96875}{5} = 2.98625$$

x_{10}= 2.00000000

y_{10}= 4.00000000

z_{10}= 3.00000000

Jadi, solusi SPL adalah x = 2.00000000, y = 4.00000000, z = 3.00000000

3.4 Determinan

Determinan matriks adalah bilangan tunggal yang diperoleh dari semua permutasi elemen matriks bujur sangkar.Jika subskrip permutasi elemen matriks adalah genap (inversi genap) diberi tanda positif (+) sebaliknya jika subskrip permutasi elemen matriks adalah ganjil (inversi ganjil) diberi tanda negative (-).Inversi terjadi jika bilangan yang lebih besar mendahului bilangan yang lebih kecil dalam urutan subskrip permutasi elemen matriks.

Determinan matriks hanya didefinisikan pada matriks bujursangkar (matriks kuadrat).

Notasi determinan matriks A: Jika diketahui matriks

$$\det(A) = |A| \text{atau} \det A = |A|$$

jika di ketahui matriks A :

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1i} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2i} & \cdots & a_{2n} \\ \vdots & \vdots & & & & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ii} & \cdots & a_{in} \\ \vdots & \vdots & & & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{ni} & \cdots & a_{nn} \end{bmatrix}$$

maka determinan matriks A :

$$\det(A) = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1i} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2i} & \cdots & a_{2n} \\ \vdots & \vdots & & & & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ii} & \cdots & a_{in} \\ \vdots & \vdots & & & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{ni} & \cdots & a_{nn} \end{bmatrix}$$

3.5 Matriks balikan Orde 2x2

JIka A dan B matriks bujur sangkar sedemikian rupa sehingga A B = B A = I , maka B disebut balikan atau *invers* dari A dan dapat dituliskan $B = A^{-1}$ (B sama dengan *invers* A). Matriks B juga mempunyai *invers* yaitu A maka dapat dituliskan $A = B^{-1}$. Jika tidak ditemukan matriks B, maka A dikatakan **matriks tunggal** (singular). Jika matriks B dan C adalah *invers* dari A maka B = C. Matriks A = $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ dapat di-*invers* apabila ad - bc ≠ 0

Dengan Rumus

$$A^{-1} = \frac{1}{det(A)}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \begin{bmatrix} \frac{d}{ad-bc} & -\frac{b}{ad-bc} \\ -\frac{c}{ad-bc} & \frac{a}{ad-bc} \end{bmatrix}$$

Apabila A dan B adalah matriks seordo dan memiliki balikan maka AB dapat di-*invers* dan $(AB)^{-1} = B^{-1}A^{-1}$

Contoh 1:

Matriks A = $\begin{bmatrix} 2 & -5 \\ -1 & 3 \end{bmatrix}$ dan B = $\begin{bmatrix} 3 & 5 \\ 1 & 2 \end{bmatrix}$

AB = $\begin{bmatrix} 2 & -5 \\ -1 & 3 \end{bmatrix}\begin{bmatrix} 3 & 5 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ = I (matriks identitas)

BA = $\begin{bmatrix} 3 & 5 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} 2 & -5 \\ -1 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ = I (matriks identitas)

Maka dapat dituliskan bahwa $B = A^{-1}$ (B Merupakan *invers* dari A)

Contoh 2:

MatriksA = $\begin{bmatrix} 1 & 1 \\ 3 & 4 \end{bmatrix}$ dan B = $\begin{bmatrix} 2 & 5 \\ 3 & 4 \end{bmatrix}$

AB = $\begin{bmatrix} 1 & 1 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} 2 & 5 \\ 3 & 4 \end{bmatrix}$ = $\begin{bmatrix} 3 & 4 \\ 6 & 8 \end{bmatrix}$

BA = $\begin{bmatrix} 2 & 5 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 3 & 4 \end{bmatrix}$ = $\begin{bmatrix} 17 & 21 \\ 15 & 19 \end{bmatrix}$

Karena AB ≠ BA ≠ I maka matriks A dan matriks B disebut **matriks tunggal**.

Contoh 3:

MatriksA = $\begin{bmatrix} 3 & 1 \\ 5 & 2 \end{bmatrix}$

Tentukan Nilai dari A^{-1}

Jawab:

$$A^{-1} = \frac{1}{(3)(2)-(5)(1)}\begin{bmatrix} 2 & -1 \\ -5 & 3 \end{bmatrix} = \frac{1}{6-5}\begin{bmatrix} 2 & -1 \\ -5 & 3 \end{bmatrix} = \frac{1}{1}\begin{bmatrix} 2 & -1 \\ -5 & 3 \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ -5 & 3 \end{bmatrix}$$

Contoh 4:

Matriks A = $\begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix}$, B = $\begin{bmatrix} 3 & 2 \\ 2 & 2 \end{bmatrix}$, AB = $\begin{bmatrix} 7 & 6 \\ 9 & 8 \end{bmatrix}$

Dengan menggunakan rumus, maka didapatkan

$A^{-1} = \begin{bmatrix} 3 & -2 \\ -1 & 1 \end{bmatrix}$, $B^{-1} = \begin{bmatrix} 1 & -1 \\ -1 & \frac{3}{2} \end{bmatrix}$, $(AB)^{-1} = \begin{bmatrix} 4 & -3 \\ -\frac{9}{2} & 7 \end{bmatrix}$

Maka $B^{-1}A^{-1} = \begin{bmatrix} 1 & -1 \\ -1 & \frac{3}{2} \end{bmatrix}\begin{bmatrix} 3 & -2 \\ -1 & 1 \end{bmatrix}$ = $\begin{bmatrix} 4 & -3 \\ -\frac{9}{2} & 7 \end{bmatrix}$

Ini membuktikan bahwa

$(AB)^{-1} = B^{-1}A^{-1}$**Orde 3x3**A = $\begin{bmatrix} 3 & 2 & -1 \\ 1 & 6 & 3 \\ 2 & -4 & 0 \end{bmatrix}$

kemudian hitung kofaktor dari matrix A

C_{11} = 12 C_{12} = 6 C_{13} = -16C_{21} = 4 C_{22} = 2 C_{23} = 16C_{31} = 12 C_{32} = -10 C_{33} = 16 menjadi matrix kofaktor

$$\begin{bmatrix} 12 & 6 & -16 \\ 4 & 2 & 16 \\ 12 & -10 & 16 \end{bmatrix}$$

cari *adj* dari matrix kofaktor tadi dengan mentranspose matrix kofaktor di atas, sehingga menjadi

$$A^{-1} = \frac{1}{det(A)} adj(A)$$

= adj(A) = $\begin{bmatrix} 12 & 4 & 12 \\ 6 & 2 & -10 \\ -16 & 16 & 16 \end{bmatrix}$

dengan metode Sarrus, kita dapat menghitung determinan dari matrix A

BAB IV

IMPLEMENTASI PROGRAM

4.1 Program

Disini kami menggunakan bahasa pemrograman *python*, untuk menyelesaikan tugas besar aljabar geometri. Kami memilih bahasa pemrograman dikarenakan bahasa ini seperti mudah dipelajari dari bahasa pemrograman lain, mudah dibaca, dan lain-lain.

4.1.1 *Python*

Python merupakan bahasa pemrograman komputer yang biasa dipakai untuk membangun situs, *software*/aplikasi, mengotomatiskan tugas dan melakukan analisis data. Bahasa pemrograman ini termasuk bahasa tujuan umum. Artinya, ia bisa digunakan untuk membuat berbagai program berbeda, bukan khusus untuk masalah tertentu saja.Di dalam *python* terdapat banyak *library* yang di sediakan seperti *NumPy*, Pandas, TensorFlow, dll. *Library* yang kami pilih adalah *NumPy*. *NumPy* adalah salah satu *library* paling populer dalam bahasa pemrograman *python* yang digunakan unutk komputasi numerik.

4.1.2 *NumPy*

NumPy menyediakan struktur data array multidimensi yang efisien. *Array* ini dapat berupa array 1D, 2D, 3D atau lebih dengan tipe data yang seragam disetiap elemennya. Menyediakan fungsi-fungsi yang dioptimalkan untuk melakukan operasi pada array. *Numpy* dapat diintegrasikan dengan baik dengna berbagai *library* dan *framwork* dalam ekosistem Python, seperti Pandas (untuk analisis data), Matplotlib (visualisasi data). Dan numpy juga sering digunakan dalam nernagai aplikasi ilmu data, pengolahan sinyal, komputasi saintifik, pemrosesan gambar, dan banyak bidang lainnya karena kecepetan, kehandalan, dan kemampuannya dalam melakukan komputasi pada array dengan ukuran besar.

4.2 Atribut

Variabel	Menyimpan nilai atau data
Fungsi	Blok code yang dapat dipanggil dengan nama untuk menjalankan tugas tertentu
Kelas	Struktur untuk membuat objek yang menggabungkan data dan data
Methode	Fungsi yang terkait dengan objek tertentu
Modul	File Python yang berisi definisi variable, fungsi, kelas yang dapat di gunakan dalam program lain
Pernyataan	Intruksi Yang Melakukan Tindakan Tertentu
Operator	Simbol yang melakukan operasi pada satu atau lebih nilai.
List	Struktur data untuk menyimpan sejumlah nilai
Tuple	Mirip dengan list, tetapi bersifat tidak dapat diubah setelah dibuat.
Dictionary	Struktur data yang menyimpan pasangan kunci-nilai.
Size	menyimpan banyaknya elemen dalam matriks
Elements	List yang menyimpan elemen-elemen matriks

Tabel 1 Atribut Program

4.3 Metode

np.array( )	Membuat array NumPy
np.add( )	Pertambahan elemen array
np.subtract( )	Pengurangan elemen array
np.transpose( )	Pengembalian transpose dari array
np.linalg.det( )	Menghitung determinan dari array
np.linalg.inv( )	Menghitung invers dari array
np.linalg.solve( )	Menyelesaikan persamaan linear

Tabel 2 Metode Program

4.4 Garis Besar

Fungsi-fungsi matriks	Ada fungsi-fungsi yang melakukan operasi matriks dasar seperti penjumlahan,pengurangan,transpose,invers,determinan,dan SPL
Input elemen matriks	Terdapat fungsi get_elements( ) unutk mengumpulkan elemen-elemen matriks dari pengguna
Menu Interaktif	Program menampilkan menu dengan 6 pilihan operasi matriks
Loop utama	Program berjalan sampai pengguna memilih untuk keluar
Memproses pilihan pengguna	Berdasarkan pilihan pengguna, program meminta input yang diperlukan unutk menjalankan fungsi yang sesuai pada matriks yang diinginkan

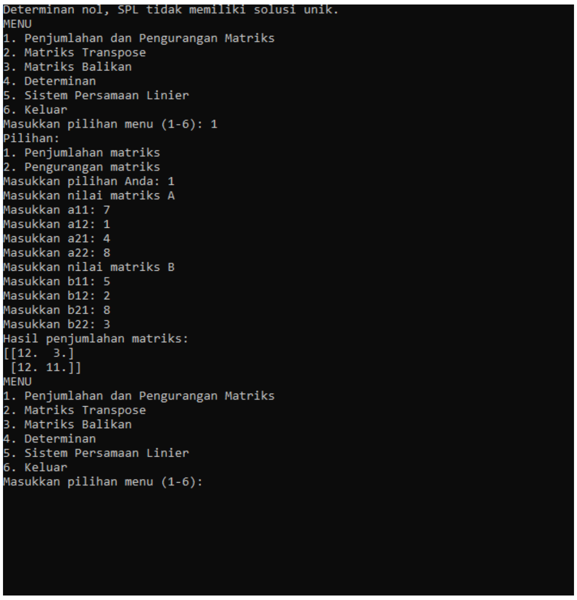
Tabel 3 Garis Besar

BAB V

PENGUJIAN PROGRAM

Analisis hasil eksekusi??

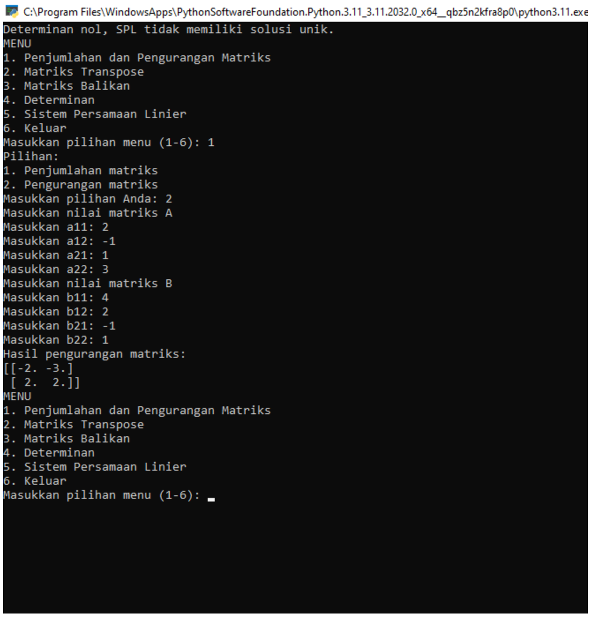
1. Uji Hasil Penjumlahan Matriks



```
Determinan nol, SPL tidak memiliki solusi unik.
MENU
1. Penjumlahan dan Pengurangan Matriks
2. Matriks Transpose
3. Matriks Balikan
4. Determinan
5. Sistem Persamaan Linier
6. Keluar
Masukkan pilihan menu (1-6): 1
Pilihan:
1. Penjumlahan matriks
2. Pengurangan matriks
Masukkan pilihan Anda: 1
Masukkan nilai matriks A
Masukkan a11: 7
Masukkan a12: 1
Masukkan a21: 4
Masukkan a22: 8
Masukkan nilai matriks B
Masukkan b11: 5
Masukkan b12: 2
Masukkan b21: 8
Masukkan b22: 3
Hasil penjumlahan matriks:
[[12.  3.]
 [12. 11.]]
MENU
1. Penjumlahan dan Pengurangan Matriks
2. Matriks Transpose
3. Matriks Balikan
4. Determinan
5. Sistem Persamaan Linier
6. Keluar
Masukkan pilihan menu (1-6):
```

Gambar 1 Hasil Uji Penjumlahan Matriks

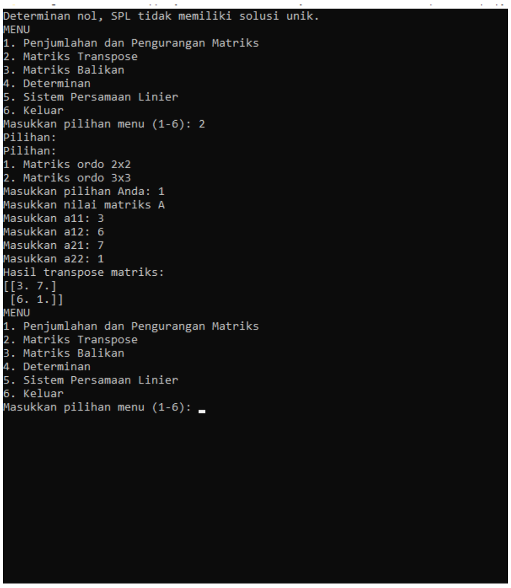
2. Hasil Uji Pengurangan Matriks



```
C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.11_3.11.2032.0_x64__qbz5n2kfra8p0\python3.11.exe
Determinan nol, SPL tidak memiliki solusi unik.
MENU
1. Penjumlahan dan Pengurangan Matriks
2. Matriks Transpose
3. Matriks Balikan
4. Determinan
5. Sistem Persamaan Linier
6. Keluar
Masukkan pilihan menu (1-6): 1
Pilihan:
1. Penjumlahan matriks
2. Pengurangan matriks
Masukkan pilihan Anda: 2
Masukkan nilai matriks A
Masukkan a11: 2
Masukkan a12: -1
Masukkan a21: 1
Masukkan a22: 3
Masukkan nilai matriks B
Masukkan b11: 4
Masukkan b12: 2
Masukkan b21: -1
Masukkan b22: 1
Hasil pengurangan matriks:
[[-2. -3.]
 [ 2.  2.]]
MENU
1. Penjumlahan dan Pengurangan Matriks
2. Matriks Transpose
3. Matriks Balikan
4. Determinan
5. Sistem Persamaan Linier
6. Keluar
Masukkan pilihan menu (1-6):
```

Gambar 2 Hasil Uji Pengurangan Matriks

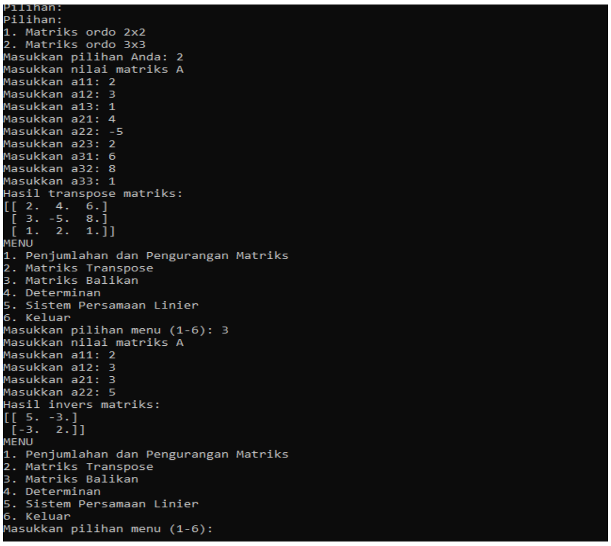
3. Hasil Uji Matriks Transpose 2x2



```
Determinan nol, SPL tidak memiliki solusi unik.
MENU
1. Penjumlahan dan Pengurangan Matriks
2. Matriks Transpose
3. Matriks Balikan
4. Determinan
5. Sistem Persamaan Linier
6. Keluar
Masukkan pilihan menu (1-6): 2
Pilihan:
Pilihan:
1. Matriks ordo 2x2
2. Matriks ordo 3x3
Masukkan pilihan Anda: 1
Masukkan nilai matriks A
Masukkan a11: 3
Masukkan a12: 6
Masukkan a21: 7
Masukkan a22: 1
Hasil transpose matriks:
[[3. 7.]
 [6. 1.]]
MENU
1. Penjumlahan dan Pengurangan Matriks
2. Matriks Transpose
3. Matriks Balikan
4. Determinan
5. Sistem Persamaan Linier
6. Keluar
Masukkan pilihan menu (1-6):
```

Gambar 3 Hasil Uji Matriks Transpose Ordo 2x2

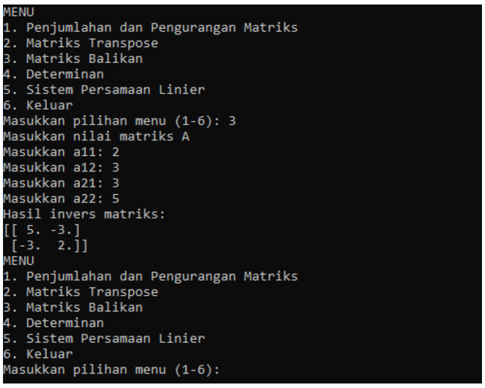
4. Hasil Uji Matriks Transpose Ordo 3x3



```
Pilihan:
Pilihan:
1. Matriks ordo 2x2
2. Matriks ordo 3x3
Masukkan pilihan Anda: 2
Masukkan nilai matriks A
Masukkan a11: 2
Masukkan a12: 3
Masukkan a13: 1
Masukkan a21: 4
Masukkan a22: -5
Masukkan a23: 2
Masukkan a31: 6
Masukkan a32: 8
Masukkan a33: 1
Hasil transpose matriks:
[[ 2.  4.  6.]
 [ 3. -5.  8.]
 [ 1.  2.  1.]]
MENU
1. Penjumlahan dan Pengurangan Matriks
2. Matriks Transpose
3. Matriks Balikan
4. Determinan
5. Sistem Persamaan Linier
6. Keluar
Masukkan pilihan menu (1-6): 3
Masukkan nilai matriks A
Masukkan a11: 2
Masukkan a12: 3
Masukkan a21: 3
Masukkan a22: 5
Hasil invers matriks:
[[ 5. -3.]
 [-3.  2.]]
MENU
1. Penjumlahan dan Pengurangan Matriks
2. Matriks Transpose
3. Matriks Balikan
4. Determinan
5. Sistem Persamaan Linier
6. Keluar
Masukkan pilihan menu (1-6):
```

Gambar 4 Hasil Uji Matriks Transpose Ordo 3x3

5. Hasil Uji Matriks Balikan/Invers

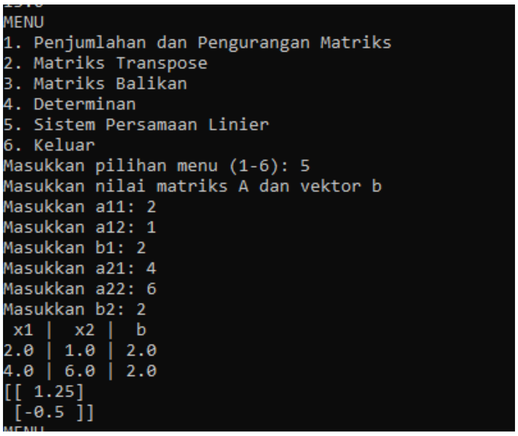


Gambar 5 Hasil Uji Matriks Balikan/Invers

6. Hasil Uji Matriks Determinan

Gambar 6 Hasil Uji Matriks Determinan

7. Hasil Uji Matriks SPL



Gambar 7 Hasil Uji Matriks SPL

BAB VI
PENUTUP

A. Kesimpulan

Matriks dan operasi merupakan hal erat kaitannya dengan bidang aljabar linier, dimana matriks itu sendiri adalah kumpulan bilangan, simbol, berbentuk persegi atau persegi panjang yang disusun menurut baris dan kolom. Matriks *fuzzy* adalah matriks yang entri-entrinya berada pada selangtutup [0,1]. Matriks *fuzzy* merupakan matriks, akan tetapi untuk sebarang matriks belum tentu merupakan matriks fuzzy. Program ini menggunakan *Python* untuk membuat peta geometris berskala besar.

Python adalah bahasa pemrograman yang banyak digunakan untuk membuat lingkungan, perangkat lunak/aplikasi, analisis data, dan banyak lagi. Ini mencakup perpustakaan seperti NumPy, Pandas, TensorFlow, dan dll. NumPy menyediakan larik data multidimensi dengan fungsi yang dioptimalkan untuk operasi larik. Itu dapat diintegrasikan dengan perpustakaan Python lain seperti Pandas dan Matplotlib.

Numpy juga digunakan dalam visualisasi data, analisis data, dan aplikasi lainnya. Program memiliki beberapa atribut, seperti variabel, fungsi, modul, input, output, dan fungsi utama. Program ini menggunakan sintaks 'matriks' Python untuk membuat array angka, dan fungsi 'matriks' untuk membuat matriks angka. Program ini juga memiliki fungsi 'gerbang' untuk menampilkan hasilnya.

B. Saran

Selama mengembangkan program perhitungan matriks ini, kami menyadari pentingnya pemahaman konsep matriks dan efisiensi operasi. Penggunaan NumPy sangat membantu dalam menyerderhanakan implementasi. Saat mengembangkan program perhitungan matriks dasar, pemilihan struktur data yang efisien dan pemisahan tugas kedalam fungsi-fungsi terpisah memberikan kemudahan dalam pengelola kode, validasi input, penangganan kesalahan, dan dokumentasi yang jelas juga meningkatkan keandalan dan keterbacaan program.

C. Refleksi

Kami menyadari pentingnya eleminasi gaus jordan, matriks balikan dan lainnya dalam konteks matriks. Contoh perhitungan invers matriks memberikan gambaran tentang bagaimana konsep ini dapat di terapkan dalam menyelesaikan masalah nyata.

Dalam pengujian program, kesuksesan dalam menyelesaikan operasi matriks menegaskan bahwa program bisa menjadi alat yang berguna dalam memproses data dan menganalisis matematika.

Saat kami membahas metode eliminasi gauss dan gauss jordan, kami memahami bahwa pemahaman tentang eliminasi sistem persamaan linear dapat di impelentasikan program dengan python, khususnya menggunakan NumPy, membuktikan kehandalan Python dalam menangani operasi matriks secara efisien.

REFERENSI

http://team-aljabar.blogspot.com/2013/03/matriks-balikan-invers.html

http://eprints.binadarma.ac.id/4659/1/pert%207%20determinan.pdf

https://katadata.co.id/intan/lifestyle/63f8b44a67a6f/memahami-metode-eliminasi-gauss-dan-pembahasan-soal

https://online-stat-psu-edu.translate.goog/statprogram/reviews/matrix-algebra/gauss-jordan-elimination