

LIBRARY ALJABAR LINEAR DALAM BAHASA JAVA DENGAN FUNGSI SISTEM PERSAMAAN LINIER, DETERMINAN, DAN APLIKASINYA

LAPORAN TUGAS BESAR

**Disusun untuk memenuhi salah satu tugas besar
mata kuliah Aljabar Linier dan Geometri
IF2123-03**



Disusun Oleh :
JavaScript

Maulvi Ziadinda Maulana (13522122)
Mohamad Akmal Ramadan (13522161)
Atqiya Haydar Luqman (13522163)

**Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
INSTITUT TEKNOLOGI BANDUNG
2023**

Daftar Isi

Daftar Isi.....	2
Bab 1.....	3
Bab 2.....	4
Bab 3.....	9
Bab 4.....	21
Bab 5.....	33
Daftar Referensi.....	35

Bab 1

Deskripsi Masalah

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Anda sudah mempelajari berbagai metode untuk menyelesaikan SPL, termasuk menghitung determinan matriks. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan ($x = A^{-1}b$), dan kaidah *Cramer* (khusus untuk SPL dengan n peubah dan n persamaan). Solusi sebuah SPL mungkin tidak ada, banyak (tidak berhingga), atau hanya satu (unik/tunggal).

$$\begin{bmatrix} 0 & \mathbf{2} & 1 & -1 \\ 0 & 0 & \mathbf{3} & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & \mathbf{1} & 0 & -\frac{2}{3} \\ 0 & 0 & \mathbf{1} & \frac{1}{3} \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Gambar 1. Eliminasi Gauss dilakukan dengan matriks eselon baris dan eliminasi Gauss-Jordan dengan matriks eselon baris tereduksi.

Di dalam Tugas Besar 1 ini, Anda diminta membuat satu atau lebih *library* aljabar linier dalam Bahasa Java. Library tersebut berisi fungsi-fungsi seperti eliminasi Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung determinan, kaidah *Cramer* (kaidah *Cramer* khusus untuk SPL dengan n peubah dan n persamaan). Selanjutnya, gunakan *library* tersebut di dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi. Penjelasan tentang interpolasi dan regresi adalah seperti di bawah ini.

Beberapa tulisan cara membuat library di Java:

1. <https://www.programcreek.com/2011/07/build-a-java-library-for-yourself/>
2. <https://developer.ibm.com/tutorials/j-javalibrary/>
3. <https://stackoverflow.com/questions/3612567/how-to-create-my-own-java-libraryapi>

Bab 2

Teori Dasar

Metode Eliminasi Gauss

Eliminasi Gauss adalah suatu metode untuk mengoperasikan nilai-nilai di dalam matriks sehingga menjadi matriks yang lebih sederhana lagi. Dengan melakukan operasi baris sehingga matriks tersebut menjadi matriks baris. Ini dapat digunakan sebagai salah satu metode penyelesaian persamaan linear dengan menggunakan matriks. Caranya dengan mengubah persamaan linear tersebut ke dalam matriks teraugmentasi dan mengoperasikaninya. Setelah menjadi matriks baris, lakukan substitusi balik untuk mendapatkan nilai dari variabel-variabel tersebut. Ciri - ciri metode gauss adalah :

1. Jika suatu baris tidak semua nol, maka bilangan pertama yang tidak nol adalah 1 (1 utama)
2. Baris nol terletak paling bawah
3. 1 utama baris berikutnya berada di kanan 1 utama baris di atasnya
4. Dibawah 1 utama harus nol

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{bmatrix}$$

Metode Eliminasi Gauss-Jordan

Eliminasi Gauss-Jordan adalah pengembangan dari eliminasi Gauss yang hasilnya lebih sederhana lagi. Caranya adalah dengan meneruskan operasi baris dari eliminasi Gauss sehingga menghasilkan matriks yang Eselon-baris. Ini juga dapat digunakan sebagai salah satu metode penyelesaian persamaan linear dengan menggunakan matriks.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Metode ini digunakan untuk mencari invers dari sebuah matriks. Prosedur umum untuk metode eliminasi Gauss-Jordan ini adalah :

1. Ubah sistem persamaan linier yang ingin dihitung menjadi matriks augmentasi.
2. Lakukan operasi baris elementer pada matriks augmentasi (A|b) untuk mengubah matriks A menjadi dalam bentuk baris eselon yang tereduksi

Determinan

Determinan adalah sebuah nilai yang berkaitan dengan matriks persegi atau matriks bujur sangkar. Determinan matriks persegi sangat diperlukan sebagai dasar untuk menghitung invers dari matriks tersebut dan aplikasi matriks seperti dalam menyelesaikan masalah kontekstual misalnya mencari penyelesaian sistem persamaan linier dua variabel atau sistem persamaan linier tiga variabel. Nilai determinan suatu matriks persegi dapat diperoleh dengan cara tertentu, tergantung dari ordo setiap matriks. Determinan dari suatu matriks A dinotasikan dengan $\det A$ atau $|A|$. Jika sebuah matriks memiliki determinan yang nilainya 0, maka matriks tersebut disebut matriks singular. Jadi, jika $|A| = 0$, maka matriks A adalah matriks singular. Determinan matriks yang akan kita bahas disini adalah determinan matriks persegi berordo 1 x 1, 2 x 2, dan 3 x 3 saja.

$$\text{Jika } A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \text{ maka determinan matriks}$$
$$|A| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

$$\text{Jika } A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \text{ maka determinan matriks dapat ditentukan}$$

sebagai berikut:

$$|A| = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = aei + bfg + cdh - ceg - afh - bdi$$

Invers Matriks

Invers matriks adalah kebalikan (invers) dari sebuah matriks yang apabila matriks tersebut dikalikan dengan inversnya, akan menjadi matriks identitas. Invers matriks dilambangkan dengan A^{-1} . Suatu matriks dikatakan memiliki invers jika determinan dari matriks tersebut tidak sama dengan nol.

Untuk menentukan invers dari sebuah matriks, terdapat dua aturan berdasarkan ordonya, yaitu ordo 2x2 dan ordo 3x3.

1. Invers Matriks Ordo 2x2

Invers matriks persegi dengan ordo 2x2 dapat dicari dengan cara sebagai berikut:

$$A^{-1} = \frac{1}{|A|} \times \text{Adj } A, \text{ dengan syarat } |A| \neq 0$$

$$\text{Jika } A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \text{ maka } A^{-1} = \frac{1}{ad-bc} \times \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}, \text{ dengan } |A| \neq 0$$

2. Invers Matriks Ordo 3x3

Untuk mencari invers matriks pada ordo 3x3, dapat digunakan metode eliminasi Gauss Jordan. Secara sistematis, eliminasi Gauss Jordan dapat dinyatakan sebagai berikut

$$[A|I] \rightarrow [I|A^{-1}]$$

Matriks persegi A dieliminasi menggunakan operasi aljabar sampai membentuk matriks identitas. Operasi yang dilakukan pada matriks A juga dilakukan pada matriks identitas sehingga jika matriks A sudah menjadi matriks identitas, maka matriks identitas akan berubah menjadi invers dari matriks A.

Matriks Kofaktor

Matriks kofaktor adalah matriks yang terdiri dari kofaktor-kofaktor matriks itu sendiri. Apabila terdapat suatu matriks A, maka matriks kofaktor A merupakan matriks yang akan terdiri dari kofaktor-kofaktor matriks A. Susunan elemen matriks kofaktor akan mengikuti susunan (letak) kofaktor-kofaktornya. Sebagai contoh :

$$A = \begin{bmatrix} 3 & 1 & -1 \\ 2 & -2 & 0 \\ 1 & 2 & -1 \end{bmatrix}$$

$$\text{Kofaktor dari } 3 = 3 = A_{11} = \begin{vmatrix} -2 & 0 \\ 2 & -1 \end{vmatrix} = 2$$

$$\text{Kofaktor dari } 1 = 1 = A_{12} = \begin{vmatrix} 2 & 0 \\ 1 & -1 \end{vmatrix} = 2$$

$$\text{Kofaktor dari } -1 = -1 = A_{13} = \begin{vmatrix} 2 & -2 \\ 1 & 2 \end{vmatrix} = 6$$

$$\text{Kofaktor dari } 2 = 2 = A_{21} = \begin{vmatrix} 1 & -1 \\ 2 & -1 \end{vmatrix} = -1$$

$$\text{Kofaktor dari } -2 = -2 = A_{22} = \begin{vmatrix} 3 & -1 \\ 1 & -1 \end{vmatrix} = -2$$

$$\text{Kofaktor dari } 0 = 0 = A_{23} = \begin{vmatrix} 3 & 1 \\ 1 & 2 \end{vmatrix} = -5$$

$$\text{Kofaktor dari } 1 = 1 = A_{31} = \begin{vmatrix} 1 & -1 \\ -2 & 0 \end{vmatrix} = -2$$

$$\text{Kofaktor dari } 2 = 2 = A_{32} = \begin{vmatrix} 3 & -1 \\ 2 & 0 \end{vmatrix} = -2$$

$$\text{Kofaktor dari } -1 = -1 = A_{33} = \begin{vmatrix} 3 & 1 \\ 2 & -2 \end{vmatrix} = -8$$

$$\text{Kofaktor dari matrix A adalah } [A_{ij}] = \begin{bmatrix} 2 & 2 & 6 \\ -1 & -2 & -5 \\ -2 & -2 & -8 \end{bmatrix}$$

Matriks Adjoin

Adjoin dari suatu matriks persegi $A = [a_{ij}]$ nxn didefinisikan sebagai transpos dari matriks $[A_{ij}]$ nxn di mana A_{ij} adalah kofaktor dari elemen a_{ij} . Adjoin dari matriks A dilambangkan dengan $\text{adj } A$. Untuk mencari adjoin dari sebuah matriks, pertama-tama cari kofaktor dari matriks yang diberikan. Kemudian temukan transpos dari matriks kofaktor tersebut. Kalau kamu masih bingung dengan istilah di atas, maka kita akan coba membahasnya secara satu per satu.

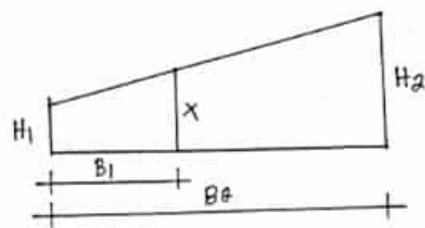
Kaidah Cramer

Dalam aljabar linear, kaidah Cramer adalah rumus yang dapat digunakan untuk menyelesaikan sistem persamaan linear dengan banyak persamaan sama dengan banyak variabel, dan berlaku ketika sistem tersebut memiliki solusi yang tunggal. Rumus ini menyatakan solusi dengan menggunakan determinan matriks koefisien (dari sistem persamaan) dan determinan matriks lain yang diperoleh dengan mengganti salah satu kolom matriks koefisien dengan vektor yang berada sebelah kanan persamaan. Metode ini dinamai dari matematikawan Swiss Gabriel Cramer (1704–1752), yang pada tahun 1750 menerbitkan kaidah ini untuk sebarang banyaknya variabel, walau Colin Maclaurin juga menerbitkan kasus khusus dari kaidah ini pada tahun 1748 (dan mungkin ia sudah mengetahuinya sejak 1729).

$$Z = \frac{A^1Y^2X^3 - A^1Y^3X^2 - A^2Y^1X^3 + A^2Y^3X^1 + A^3Y^1X^2 - A^3Y^2X^1}{Z^1Y^2X^3 - Z^1Y^3X^2 - Z^2Y^1X^3 + Z^2Y^3X^1 + Z^3Y^1X^2 - Z^3Y^2X^1}$$

Interpolasi Polinomial

Interpolasi polinomial merupakan teknik interpolasi dengan mengasumsikan pola data yang kita miliki mengikuti pola polinomial baik berderajat satu (linier) maupun berderajat tinggi. Interpolasi dengan metode ini dilakukan dengan terlebih dahulu membentuk persamaan polinomial. Persamaan polinomial yang terbentuk selanjutnya digunakan untuk melakukan interpolasi dari nilai yang diketahui atau ekstrapolasi (prediksi) dari nilai diluar rentang data yang diketahui.



$$\text{RUMUS : } X = H_1 - \frac{B_1}{B_2} \times (H_1 - H_2)$$

Interpolasi Bicubic

Interpolasi bicubic merupakan sebuah metode interpolasi yang menggunakan 16 pixel dalam pixel 4x4 tetangga terdekat pada citra aslinya. Dengan menggunakan metode interpolasi bicubic ini dapat membuat tepi-tepi citra hasil lebih halus. Sehingga metode interpolasi bicubic sering digunakan dalam pengeditan perangkat lunak dan banyak kamera digital lainnya. Seiring

perkembangannya maka mulai dibentuk berbagai metode interpolasi, salah satunya adalah interpolasi bicubic basis spline, yang mana metode ini juga memanfaatkan 16 pixel tetangga terdekatnya.

Konsep metode interpolasi bicubic basis spline dan representasinya dalam matematika pertama kali dideskripsikan oleh Schoenberg pada tahun 1946. Menurut definisinya, metode interpolasi bicubic basis spline dimaksudkan sebagai potongan-potongan polinomial dengan potongan-potongan yang lebih halus dan digabungkan secara bersamaan. Berikut ditampilkan fungsi dari bicubic basis spline untuk $n = 3$.

$$\beta^3(x) = \begin{cases} \left(\frac{2}{3}\right) - 0.5 |x|^2 (2 - |x|) & 0 \leq |x| < 1 \\ \left(\frac{1}{6}\right) * (2 - |x|)^3 & 1 \leq |x| < 2 \\ 0 & 2 \leq |x| \end{cases}$$

Regresi Linier Berganda

Regresi Linier Berganda merupakan model regresi yang melibatkan lebih dari satu variabel independen. Analisis regresi linear berganda dilakukan untuk mengetahui arah dan seberapa besar pengaruh variabel independen terhadap variabel dependen (Ghozali, 2018).

Dalam contoh penelitian ini uji regresi linear berganda dilakukan untuk mendapatkan gambaran bagaimana variabel independen yang meliputi CSR, likuiditas, capital intensity, dan inventory intensity mempengaruhi variabel dependen yaitu agresivitas pajak dengan tingkat signifikansi sebesar 0,05 (Ghozali, 2018).

Bab 3

Implementasi dan Program dalam Java

Pada program Java kami, terdapat 16 class yang dibagi menjadi 1 folder utama dan 2 sub-folder source code, yaitu :

1. src (folder apps)

Folder apps berisi implementasi dari algoritma-algoritma yang akan dilakukan pada matriks. Folder ini berisi 11 class :

- Balikan_Adjoin.java
- Balikan_GaussJ.java
- BicubiSI.java
- Determinan_EK.java
- Determinan_ReduksiBaris.java
- ImageScaler.java
- Interpolasi_Polinom.java
- SPL_Balikan.java
- SPL_Cramer.java
- SPL_Gauss.java
- SPL_GaussJ.java
- RegresiLinearBerganda.java
- driver.java

1.1 Balikan_Adjoin.java

Class ini berisi tentang algoritma matriks balikan menggunakan matriks adjoin.

- Method

Method(Parameter)	Description
public static void adjoin(MatriksADT matrix)	Method untuk menghitung matriks adjoin berdasarkan matriks input.

1.2 Balikan_Gauss.java

Class ini berisi tentang algoritma matriks balikan menggunakan metode Gauss-Jordan.

- Method

Method(Parameter)	Description
-------------------	-------------

public static MatrixADT inverseGaussJ(MatrixADT m)	Method ini digunakan untuk menghitung invers dari matriks m menggunakan metode Gauss-Jordan.
---	--

1.3 BicubicSI.java

Class ini berisi metode untuk menghitung interpolasi bicubic berdasarkan matriks m dan koordinat x dan y.

- Method

Method(Parameter)	Description
public static double bicubic(MatrixADT m, double x, double y)	Method utama yang menghitung hasil interpolasi bicubic berdasarkan matriks m dan koordinat x dan y.
public static MatrixADTkoefisien(MatrixADT m, MatrixADT xMatrix)	Method ini menghitung koefisien interpolasi bicubic berdasarkan matriks m dan matriks xMatrix.
public static MatrixADT matrixY(MatrixADT m)	Method ini mengubah matriks 4x4 menjadi matriks 16x1 berdasarkan matriks m.
public static MatrixADTmatrixX()	Method ini membuat matriks 16x16 sesuai dengan spesifikasi tugas.
public static inversematrixX()	Method ini menggunakan invers Gauss-Jordan dari Balikan_GaussJ untuk menghasilkan invers dari matriks matrixX.

1.4 Determinan_EK.java

Class ini berisi metode untuk menghitung determinan dari matriks persegi dengan ekspansi kofaktor.

- Method

Method(Parameter)	Description
public static double determinant(MatrixADT matrix)	Method utama yang menghitung determinan matriks berdasarkan matriks input matrix.

1.5 Determinan_ReduksiBaris.java

Class ini berisi metode untuk menghitung determinan dari matriks persegi dengan reduksi baris.

- Attribute

Attributes	Description
private static String filename	Variabel untuk menyimpan nama file

- Method

Method(Parameter)	Description
static double performOBE(MatrixADT matrix)	Method untuk melakukan eliminasi Gauss-Jordan pada matriks matrix dan mengembalikan determinannya.
static void swap_row(MatrixADT matrix, int i, int j)	Method untuk menukar dua baris matriks pada indeks i dan j.
static MatrixADT in putFile(String namaFile)	Method untuk membaca matriks dari file dengan nama namaFile dan mengembalikan matriks tersebut dalam objek MatrixADT.
static void outputFile(Double answer)	Method untuk menyimpan hasil determinan ke dalam file dengan nama yang sudah ditentukan.
public static void main(String[] args)	Method utama yang digunakan untuk menjalankan program.

1.6 ImageScaler.java

Class ini berisi algoritma untuk melakukan penskalaan gambar.

- Attribute

Attributes	Description
static Scanner sc	Variabel untuk membaca input dari pengguna

- Method

Method(Parameter)	Description
public static void Scaling	Method utama untuk melakukan penskalaan gambar.
public static String imagePath(String fileName)	Method untuk mengambil path lengkap dari file gambar berdasarkan nama file yang diberikan.
public static void printImage(String fileName, BufferedImage img)	Method untuk mencetak gambar hasil penskalaan ke file dengan nama yang ditentukan.
public static int rgbAlpha	Method untuk mengekstrak nilai saluran alpha dari sebuah piksel.
public static int rgbGreen	Method untuk mengekstrak nilai saluran hijau dari sebuah piksel.

1.7 Interpolasi_Polinom.java

Class ini berisi algoritma untuk melakukan interpolasi polinomial berdasarkan titik-titik yang diberikan.

- Attribute

Attributes	Description
private static String fileName	Variabel untuk menyimpan nama file yang digunakan dalam operasi interpolasi

- Method

Method(Parameter)	Description
static void backSubstitute(MatrixADT matri)	Method untuk melakukan substitusi balik pada matriks setelah proses eliminasi Gauss-Jordan selesai.
static void swap-row(MatrixADT matrix, int i, int j)	Method untuk menukar dua baris dalam matriks.
static void performElimination(MatrixADT matrix)	Method untuk melakukan eliminasi Gauss-Jordan pada matriks augmented yang digunakan dalam interpolasi.
static void createAugmented(MatrixADT matrix, MatrixADT matrixPoint)	Method untuk membuat matriks augmented dari matriks titik input dan

	mengisinya sesuai dengan persamaan polinomial.
static void findAnswer(MatrixADT matrix, double x)	Method untuk mencari nilai fungsi interpolasi pada titik x dan menampilkan hasilnya.
static MatrixADT inputFile(String namaFile)	Method untuk membaca matriks titik input dari file dan mengembalikan matriks tersebut dalam bentuk objek MatrixADT.
static Double inputNumber(String fileName, int numRows)	Method untuk membaca nilai x dari file input.
static void outputFile(MatrixADT matrix, double answer, double inputNum)	Method untuk menyimpan hasil interpolasi dalam file.
public static void main(String[] args)	Method utama program yang mengatur alur eksekusi untuk memasukkan input dari file atau manual, melakukan interpolasi, dan menyimpan hasilnya dalam file jika diperlukan.

1.8 SPL_Balikan

Class ini berisi algoritma untuk menyelesaikan Sistem Persamaan Linear (SPL) dengan metode matriks balikan.

- Attribute

Attributes	Description
private class String fileName	Digunakan untuk menyimpan nama file yang akan digunakan untuk menyimpan solusi SPL.

- Method

Method(Parameter)	Description
static MatrixADT inputFile(String namaFile)	Membaca data matriks dari file eksternal dan mengembalikan objek MatrixADT yang berisi data matriks.
static void swap_row(MatrixADT, int i, int j)	Menukar baris ke-i dengan baris ke-j dalam matriks.

static void backSubstitute(Matrix matrix, MatrixADT matrixIdentity)	Melakukan substitusi mundur (back substitution) pada matriks dan matriks identitas setelah eliminasi Gauss-Jordan.
Static double performOBE(MatrixADT matrix, MatrixADT matrixIdentity)	Melakukan eliminasi Gauss-Jordan pada matriks koefisien dan matriks identitas untuk menghitung determinan.
public static void findSolution(MatrixADT matrix)	Menyelesaikan SPL dengan metode matriks balikan dan menampilkan solusi atau pesan kesalahan jika SPL tidak dapat diselesaikan.
static void outputFile(MatrixADT komponenB, MatrixADT answer)	Menyimpan solusi SPL ke dalam file eksternal.
public static void main(String[] args)	Metode utama program yang memungkinkan pengguna untuk memilih sumber input (file atau manual) dan menampilkan solusi SPL.

1.9 SPL_Cramer.java

Class ini berisi algoritma untuk menyelesaikan Sistem Persamaan Linear (SPL) dengan metode Cramer.

- Method

Method(Parameter)	Description
public static void cramer(MatrixADT matrix)	Metode utama yang digunakan untuk menyelesaikan SPL dengan metode Cramer berdasarkan matriks input.

1.10 SPL_Gauss.java

Class ini berisi algoritma untuk menyelesaikan Sistem Persamaan Linear (SPL) dengan metode eliminasi Gauss.

- Attribute

Attribute	Description
public static String fileName	Variabel statik yang digunakan untuk menyimpan nama file input atau output untuk penyimpanan hasil solusi SPL. Variabel ini dapat diakses oleh seluruh

	metode dalam class.
--	---------------------

- Method

Method(Parameter)	Description
static void performInconsistent(MatrixADT matrix)	Metode ini digunakan untuk mengevaluasi dan menampilkan solusi parametrik jika SPL bersifat inkonsisten.
static void performElimination(MatrixADT matrix)	Metode ini digunakan untuk mengevaluasi dan menampilkan solusi SPL setelah eliminasi Gauss.
static void swap_row(MatrixADT matrix, int i, int j)	Metode ini digunakan untuk menukar dua baris dalam matriks.
static int forwardElim(MatrixADT matrix)	Metode ini mengimplementasikan langkah eliminasi Gauss untuk menuju bentuk matriks eselon. Mengembalikan indeks baris pertama yang nol jika SPL singular, atau -1 jika SPL tidak singular.
static MatrixADT inputFile(String namaFile)	Metode ini digunakan untuk membaca matriks dari file input dan mengembalikan matriks dalam bentuk MatrixADT.
public static void consistentSolution(MatrixAdt matrix)	Metode ini digunakan untuk menampilkan solusi SPL yang konsisten.
static void inconsistentSolution(MatrixADT matrix)	Metode ini digunakan untuk menampilkan solusi parametrik jika SPL bersifat inkonsisten.
static void outputFileInconsistent(MatrixADT matrix)	Metode ini digunakan untuk menyimpan solusi SPL yang inkonsisten ke dalam file.
static void outputFileconsistent(MatrixADT matrix)	Metode ini digunakan untuk menyimpan solusi SPL yang konsisten ke dalam file.
public static void main(String[] args)	Metode main dari program yang digunakan untuk interaksi dengan pengguna, meminta input dari pengguna, dan menampilkan hasil solusi SPL.

1.11 SPL_GaussJ.java

Class ini berisi algoritma untuk menyelesaikan Sistem Persamaan Linear (SPL) dengan metode eliminasi Gauss-Jordan

- Method

Method(Parameter)	Description
public static void SPLGaussJordan(MatrixADT matrix)	Metode ini digunakan untuk mengevaluasi dan menampilkan solusi SPL menggunakan metode eliminasi Gauss-Jordan.
public static boolean checkUniqueSolution(MatrixADT matrix)	Metode ini digunakan untuk memeriksa apakah matriks hasil eliminasi Gauss-Jordan memiliki solusi unik atau tidak. Mengembalikan true jika ada solusi unik dan false jika tidak.

1.12 RegresiLinearBerganda.java

Class ini berisi algoritma untuk melakukan regresi linear berganda berdasarkan data yang dimasukkan oleh pengguna.

- Method

Method(Parameter)	Description
public static void RegresiLinearBerganda(MatrixADT matrix)	

1.13 driver.java

Class ini berfungsi sebagai program utama untuk menjalankan semua operasi matriks yang ada.

- Attribute

Attributes	Description
scanner	Objek Scanner untuk membaca input dari pengguna.
matriks	Objek matriks yang digunakan dalam berbagai perhitungan numerik.

- Method

Method(Parameter)	Description
public static void main(String[] args)	Metode utama yang menjalankan program utama. Ini menyediakan menu, membaca input pengguna, dan mengarahkan ke berbagai fitur terkait matriks dan perhitungan numerik.

2. Matrix (Abstract Data Type)

Pada subfolder ini terdapat 2 file yaitu :

- MatrixADT.java
- OBE.java

2.1 MatrixADT.java

Class ini berisi ADT (*Abstract Data Type*) yang digunakan pada fungsi-fungsi operasi matriks.

- Attribute

Attributes	Description
public double[][] data	Matriks data yang disimpan dalam array dua dimensi.
public int numRows	Jumlah baris matriks.
public numCols	Jumlah kolom matriks.
Scanner scanElmt	Objek Scanner untuk membaca elemen-elemen matriks.

- Method

Method(Parameter)	Description
public MatrixADT(int nRows, int nCols)	Konstruktur untuk membuat objek MatrixADT dengan jumlah baris dan kolom yang ditentukan.
public int getRows()	Mengembalikan jumlah baris matriks.
public int getCols()	Mengembalikan jumlah kolom matriks.
public double getElmt(int i, int j)	Mengembalikan elemen matriks pada

	baris i dan kolom j.
public void setElmt(int i, int j, double value)	Mengatur nilai elemen matriks pada baris i dan kolom j dengan nilai value.
public boolean isMatrixIdxValid(int i, int j)	Memeriksa apakah indeks baris dan kolom matriks valid.
public int getLastIdxRow()	Mengembalikan indeks terakhir dari baris matriks.
public int getLastIdxCol()	Mengembalikan indeks terakhir dari kolom matriks.
public boolean isIdxEff(int i, int j)	Memeriksa apakah indeks baris dan kolom efektif.
public double getElmtDiagonal(int i)	Mengembalikan elemen diagonal pada baris i.
public void displayMatrix()	Menampilkan matriks ke layar.
public MatrixADT matrikspersegi()	Membaca matriks persegi dari input pengguna.
public MatrixADT matriksSPL()	Membaca matriks untuk sistem persamaan linier dari input pengguna.
public void readMatrix(int row, int col)	Membaca matriks dengan ukuran row x col dari input pengguna.
public void readMatrix1()	Membaca matriks dengan ukuran tetap 4x4 dari input pengguna.
public static MatrixADT matriksBicubic()	Mengembalikan matriks untuk interpolasi bicubic.
public MatrixADT copyMatrix()	Menghasilkan salinan matriks.
public void setCol()	Mengatur jumlah kolom matriks.
public void resize(int row, int col)	Mengubah ukuran matriks ke row x col.
public MatrixADT copy(int Row1, int Row2, int Col1, int Col2)	Menghasilkan salinan sebagian dari matriks.
public boolean isRowZero(int idxRow)	Memeriksa apakah baris tertentu matriks adalah baris nol.
public static MatrixADT	Mengembalikan hasil perkalian dua

<code>multiplyMatrix(MatrixADT m1, MatrixADT m2)</code>	matriks m1 dan m2.
<code>public void createIdentity()</code>	Mengisi matriks dengan matriks identitas.
<code>public MatrixADT multiplyMatrix(MatrixADT matrix2)</code>	Mengembalikan hasil perkalian matriks dengan matriks matrix2.

2.2 OBE.java

Class ini berisi algoritma untuk melakukan eliminasi Gauss-Jordan pada sebuah matriks.

- Method

Method(Parameter)	Description
<code>public static MatrixADT gaussJordan(MatrixADT m)</code>	Metode statis yang mengambil objek MatrixADT m sebagai argumen dan melakukan eliminasi Gauss-Jordan pada matriks tersebut.

3. IO

Pada subfolder ini terdapat 2 file :

- Parser.java
- text.java

3.1 Parser.java

Pada class ini berisi metode-metode yang digunakan untuk membaca dan membuat file berdasarkan matriks.

- Method

Method(Parameter)	Description
<code>public static MatrixADT FiletoMatrix()</code>	Metode ini membaca data matriks dari sebuah file yang dimasukkan oleh pengguna melalui keyboard.
<code>publis static void createFile(MatrixADT matrix, String fileName)</code>	Metode ini digunakan untuk membuat file baru dan menulis matriks ke dalam file tersebut.

3.2 text.java

Pada class ini terdapat metode-metode yang digunakan untuk membaca dan membuat file berdasarkan matriks.

- Method

Method(Parameter)	Description
public static void matrixToFile(MatrixADT matriks)	Metode ini membaca data matriks dari sebuah file yang dimasukkan oleh pengguna melalui keyboard.
publix static MatrixADT pilihan1()	Metode ini digunakan untuk membuat file baru dan menulis matriks ke dalam file tersebut.

Bab 4

Eksperimen

- Temukan Solusi SPL $Ax = b$, berikut :

a.

$$A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$$

Metode Gauss :

```
Masukkan nomor program yang ingin dijalankan: 1
1. Input melalui file
2. Input melalui keyboard
Masukkan pilihan: 2

Jumlah baris: 4
Jumlah kolom: 5

Masukkan matrix:
1 1 -1 -1 1
2 5 -7 -5 -2
2 -1 1 3 4
5 2 -4 2 6
SPL tersebut tidak memiliki solusi.

Tekan enter untuk kembali ke menu awal
```

Metode Matriks Balikan :

```
Masukkan nomor program yang ingin dijalankan: 3
1. Input melalui file
2. Input melalui keyboard
Masukkan pilihan: 2

Jumlah baris: 4
Jumlah kolom: 5

Masukkan matrix:
1 1 -1 -1 1
2 5 -7 -5 -2
2 -1 1 3 4
5 2 -4 2 6
SPL tidak bisa diselesaikan dengan metode Inverse.

Tekan enter untuk kembali ke menu awal
```

b.

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$$

Metode Gauss :

```
1 -1 0 0 1 3
1 1 0 -3 0 6
2 -1 0 1 -1 5
-1 2 0 -2 -1 -1

SPL tersebut memiliki solusi parametrik.

x5 = 0.000000
x4 = -1.000000 + 1.000000x5
x2 = 2.333333 + 2.333333x4 - 0.333333x5
x1 = 2.500000 + 0.500000x2 - 0.500000x4 + 0.500000x5
x3 = bil. real

Apakah ingin menyimpan solusi dalam file?
1. Yes
2. No
```

Metode Matriks Balikan :

```
Masukkan nomor program yang ingin dijalankan: 3

1. Input melalui file
2. Input melalui keyboard
Masukkan pilihan: 2

Jumlah baris: 4
Jumlah kolom: 6

Masukkan matrix:
1 -1 0 0 1 3
1 1 0 -3 0 6
2 -1 0 1 -1 5
-1 2 0 -2 -1 -1
SPL tidak bisa dicari dengan metode Inverse.

Tekan enter untuk kembali ke menu awal
```

C.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$

Metode Gauss :

```
0 1 0 0 1 0 2
0 0 0 1 1 0 -1
0 1 0 0 0 1 1

SPL tersebut memiliki solusi parametrik.

x5 = 1.000000 + 1.000000x6
x4 = -1.000000 - x5
x2 = 2.000000 - x5
x1 = bil. real
x3 = bil. real
x6 = bil. real

Apakah ingin menyimpan solusi dalam file?
1. Yes
2. No
```

Metode Matriks Balikan :

```
Masukkan nomor program yang ingin dijalankan: 3

1. Input melalui file
2. Input melalui keyboard
Masukkan pilihan: 2

Jumlah baris: 3
Jumlah kolom: 7

Masukkan matrix:
0 1 0 0 1 0 2
0 0 0 1 1 0 -1
0 1 0 0 0 1 1
SPL tidak bisa dicari dengan metode Inverse.

Tekan enter untuk kembali ke menu awal
```

d.

$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \dots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \dots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \dots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \dots & \frac{1}{2n+1} \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

H adalah matriks *Hilbert*. Cobakan untuk $n = 6$ dan $n = 10$.

Metode Gauss :

```
1.000000    0.500000    0.333333    0.250000    0.200000    0.166667 1
0.500000    0.333333    0.250000    0.200000    0.166667    0.142857 0
0.333333    0.250000    0.200000    0.166667    0.142857    0.125000 0
0.250000    0.200000    0.166667    0.142857    0.125000    0.111111 0
0.200000    0.166667    0.142857    0.125000    0.111111    0.100000 0
0.166667    0.142857    0.125000    0.111111    0.100000    0.090909 0
```

```
Solusi dari SPL:
x1 = 11.540412
x2 = 46.616694
x3 = -1130.944969
x4 = 3969.618023
x5 = -5045.250153
x6 = 2158.663954
```

```
Apakah ingin menyimpan solusi dalam file?
1. Yes
2. No
█
```

```
0.100000    0.090909    0.083333    0.076923    0.071429    0.066667
```

```
Solusi dari SPL:
x1 = 19.525627
x2 = -159.545670
x3 = 20.759112
x4 = 1983.940776
x5 = -5084.976735
x6 = 5660.147295
x7 = -4622.220507
x8 = 3257.960172
x9 = -623.836012
x10 = -456.355282
```

```
Apakah ingin menyimpan solusi dalam file?
1. Yes
2. No
█
```


Metode Matriks Balikan :

```
Masukkan matrix:
1.000000 0.500000 0.333333 0.250000 0.200000 0.166667 1
0.500000 0.333333 0.250000 0.200000 0.166667 0.142857 0
0.333333 0.250000 0.200000 0.166667 0.142857 0.125000 0
0.250000 0.200000 0.166667 0.142857 0.125000 0.111111 0
0.200000 0.166667 0.142857 0.125000 0.111111 0.100000 0
0.166667 0.142857 0.125000 0.111111 0.100000 0.090909 0
Nilai x1 = 11.54
Nilai x2 = 46.62
Nilai x3 = -1130.94
Nilai x4 = 3969.62
Nilai x5 = -5045.25
Nilai x6 = 2158.66

Apakah ingin menyimpan solusi dalam file?
1. Yes
2. No
█
```

```
0.125000 0.111111 0.100000 0.090909 0.083333 0.076923
0.111111 0.100000 0.090909 0.083333 0.076923 0.071429
0.100000 0.090909 0.083333 0.076923 0.071429 0.066667

Nilai x1 = 19.53
Nilai x2 = -159.55
Nilai x3 = 20.76
Nilai x4 = 1983.94
Nilai x5 = -5084.98
Nilai x6 = 5660.15
Nilai x7 = -4622.22
Nilai x8 = 3257.96
Nilai x9 = -623.84
Nilai x10 = -456.36

Apakah ingin menyimpan solusi dalam file?
1. Yes
2. No
█
```

- SPL berbentuk matriks *augmented*

a.

$$\begin{bmatrix} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{bmatrix}.$$

Metode Gauss :

```
Masukkan matrix:
1 -1 2 -1 -1
2 1 -2 -2 -2
-1 2 -4 1 1
3 0 0 -3 -3

SPL tersebut memiliki solusi parametrik.

x2 = 0.00 + 2.000000x3
x1 = -1.000000 + 1.000000x4
x3 = bil. real
x4 = bil. real

Apakah ingin menyimpan solusi dalam file?
1. Yes
2. No
█
```

Metode Matriks Balikan :

```
Masukkan nomor program yang ingin dijalankan: 3
1. Input melalui file
2. Input melalui keyboard
Masukkan pilihan: 2

Jumlah baris: 4
Jumlah kolom: 5

Masukkan matrix:
1 -1 2 -1 -1
2 1 -2 -2 -2
-1 2 -4 1 1
3 0 0 -3 -3
SPL tidak bisa diselesaikan dengan metode Inverse.

Tekan enter untuk kembali ke menu awal
█
```

b.

$$\begin{bmatrix} 2 & 0 & 8 & 0 & 8 \\ 0 & 1 & 0 & 4 & 6 \\ -4 & 0 & 6 & 0 & 6 \\ 0 & -2 & 0 & 3 & -1 \\ 2 & 0 & -4 & 0 & -4 \\ 0 & 1 & 0 & -2 & 0 \end{bmatrix}.$$

Metode Gauss :

```
0 1 0 4 6
-4 0 6 0 6
0 -2 0 3 -1
2 0 -4 -0 -4
0 1 0 -2 0

SPL tersebut memiliki solusi parametrik.

x4 = 1.000000
x3 = 1.000000
x2 = 0.00 + 2.000000x4
x1 = -2.000000 + 2.000000x3

Apakah ingin menyimpan solusi dalam file?
1. Yes
2. No
█
```

Metode Matriks Balikan :

```

1. Input melalui file
2. Input melalui keyboard
Masukkan pilihan: 2

Jumlah baris: 6
Jumlah kolom: 5

Masukkan matrix:
2 0 8 0 8
0 1 0 4 6
-4 0 6 0 6
0 -2 0 3 -1
2 0 -4 -0 -4
0 1 0 -2 0
SPL tidak bisa dicari dengan metode Inverse.

Tekan enter untuk kembali ke menu awal

```

- SPL Berbentuk

a.

$$\begin{aligned}
 8x_1 + x_2 + 3x_3 + 2x_4 &= 0 \\
 2x_1 + 9x_2 - x_3 - 2x_4 &= 1 \\
 x_1 + 3x_2 + 2x_3 - x_4 &= 2 \\
 x_1 + 6x_3 + 4x_4 &= 3
 \end{aligned}$$

Metode Gauss :

```

Jumlah kolom: 5

Masukkan matrix:
8 1 3 2 0
2 9 -1 -2 1
1 3 2 -1 2
1 0 6 4 3

Solusi dari SPL:
x1 = -0.224324
x2 = 0.182432
x3 = 0.709459
x4 = -0.258108

Apakah ingin menyimpan solusi dalam file?
1. Yes
2. No

```

Metode Matriks Balikan :

```
Jumlah baris: 4
Jumlah kolom: 5

Masukkan matrix:
8 1 3 2 0
2 9 -1 -2 1
1 3 2 -1 2
1 0 6 4 3
Nilai x1 = -0.22
Nilai x2 = 0.18
Nilai x3 = 0.71
Nilai x4 = -0.26

Apakah ingin menyimpan solusi dalam file?
1. Yes
2. No
█
```

b.

$$\begin{aligned}x_7 + x_8 + x_9 &= 13.00 \\x_4 + x_5 + x_6 &= 15.00 \\x_1 + x_2 + x_3 &= 8.00 \\0.04289(x_3 + x_5 + x_7) + 0.75(x_6 + x_8) + 0.61396x_9 &= 14.79 \\0.91421(x_3 + x_5 + x_7) + 0.25(x_2 + x_4 + x_6 + x_8) &= 14.31 \\0.04289(x_3 + x_5 + x_7) + 0.75(x_2 + x_4) + 0.61396x_1 &= 3.81 \\x_3 + x_6 + x_9 &= 18.00 \\x_2 + x_5 + x_8 &= 12.00 \\x_1 + x_4 + x_7 &= 6.00 \\0.04289(x_1 + x_5 + x_9) + 0.75(x_2 + x_6) + 0.61396x_3 &= 10.51 \\0.91421(x_1 + x_5 + x_9) + 0.25(x_2 + x_4 + x_6 + x_8) &= 16.13 \\0.04289(x_1 + x_5 + x_9) + 0.75(x_4 + x_8) + 0.61396x_7 &= 7.04\end{aligned}$$

Metode Gauss :

```
SPL tersebut memiliki solusi parametrik.

x9 = 5.014692
x8 = 9.157072 - 0.631050x9
x7 = 13.000000 - x8 - x9
x6 = 19.282608 - 0.063298x7 - 1.063298x8 - 0.755670x9
x5 = 22.201277 - 0.688193x6 - 0.688193x8 - 1.376387x9
x4 = 15.000000 - x5 - x6
x3 = 18.000000 - x6 - x9
x2 = 12.000000 - x5 - x8
x1 = 8.000000 - x2 - x3

Apakah ingin menyimpan solusi dalam file?
1. Yes
2. No
█
```

Metode Matriks Balikan :

```
Jumlah baris: 10
Jumlah kolom: 10

Masukkan matrix:
0 0 0 0 0 0 1 1 1 13
0 0 0 1 1 1 0 0 0 15
1 1 1 0 0 0 0 0 0 8
0 0 0.04289 0 0.04289 0.75 0.04289 0.75 0.61396 14.79
0 0 1 0 0 1 0 0 1 18
0 1 0 0 1 0 0 1 0 12
1 0 0 1 0 0 1 0 0 6
0.04289 0.74 0.61396 0 0.04289 0.75 0 0 0.04289 10.51
0.91421 0.25 0 0.25 0.91421 0.25 0 0.25 0.91421 16.13
0.04289 0 0 0.75 0.04289 0 0.61396 0.75 0.04289 7.04
SPL tidak bisa dicari dengan metode Inverse.

Tekan enter untuk kembali ke menu awal
█
```

- Sistem Reaktor

$$A: m_{A_{in}} + Q_{BA}x_B - Q_{AB}x_A - Q_{AC}x_A = 0$$

$$B: Q_{AB}x_A - Q_{BA}x_B - Q_{BC}x_B = 0$$

$$C: m_{C_{in}} + Q_{AC}x_A + Q_{BC}x_B - Q_{Cout}x_C = 0$$

Tentukan solusi x_A , x_B , x_C dengan menggunakan parameter berikut : $Q_{AB} = 40$, $Q_{AC} = 80$, $Q_{BA} = 60$, $Q_{BC} = 20$ dan $Q_{Cout} = 150 \text{ m}^3/\text{s}$ dan $m_{A_{in}} = 1300$ dan $m_{C_{in}} = 200 \text{ mg/s}$.

Metode Gauss :

```
Jumlah baris: 3
Jumlah kolom: 4

Masukkan matrix:
-120 60 0 1300
40 -80 0 0
80 20 -150 200

Solusi dari SPL:
x1 = -14.444444
x2 = -7.222222
x3 = -10.000000

Apakah ingin menyimpan solusi dalam file?
1. Yes
2. No
█
```

Metode Matriks Balikan :

```
2. Input melalui keyboard
Masukkan pilihan: 2

Jumlah baris: 3
Jumlah kolom: 4

Masukkan matrix:
-120 60 0 1300
40 -80 0 0
80 20 -150 200
Nilai x1 = -14.44
Nilai x2 = -7.22
Nilai x3 = -10.00

Apakah ingin menyimpan solusi dalam file?
1. Yes
2. No
█
```

- Studi Kasus Interpolasi

```
2
Masukkan nilai N (derajat maksimal polinom):
6
Masukkan Matriks, pisahkan baris dengan newline dan kolom dengan spasi
0.1 0.003
0.3 0.067
0.5 0.148
0.7 0.248
0.9 0.370
1.1 0.518
1.3 0.697
Masukkan nilai x:
0.2
f(x) = - 0.0000x^6 + 0.0000x^5 + 0.0260x^4 + 0.0000x^3 + 0.1974x^2 + 0.2400x - 0.0230 , f(0.200000) = 0.032961.
Apakah ingin menyimpan solusi dalam file?
1. Yes
2. No
█
```

```
2
Masukkan nilai N (derajat maksimal polinom):
6
Masukkan Matriks, pisahkan baris dengan newline dan kolom dengan spasi
0.1 0.003
0.3 0.067
0.5 0.148
0.7 0.248
0.9 0.370
1.1 0.518
1.3 0.697
Masukkan nilai x:
1.28
f(x) = - 0.0000x^6 + 0.0000x^5 + 0.0260x^4 + 0.0000x^3 + 0.1974x^2 + 0.2400x - 0.0230 , f(1.280000) = 0.677542.
Apakah ingin menyimpan solusi dalam file?
1. Yes
2. No
█
```

- Studi Kasus Regresi Linear Berganda

Masukkan jumlah data: 20

Masukkan jumlah variabel independen (termasuk konstanta): 3

Masukkan data:

```
72.4 76.3 29.18 0.90
41.6 70.3 29.35 0.91
34.3 77.1 29.24 0.96
35.1 68.0 29.27 0.89
10.7 79.0 29.78 1.00
12.9 67.4 29.39 1.10
8.3 66.8 29.69 1.15
20.1 76.9 29.48 1.03
72.2 77.7 29.09 0.77
24.0 67.7 29.60 1.07
23.2 76.8 29.38 1.07
47.4 86.6 29.35 0.94
31.5 76.9 29.63 1.10
10.6 86.3 29.56 1.10
11.2 86.0 29.48 1.10
73.3 76.3 29.40 0.91
75.4 77.9 29.28 0.87
96.6 78.7 29.29 0.78
107.4 86.6 29.03 0.82
54.9 70.9 29.37 0.95
```

Hasil koefisien:

$x_0 = -0.0003518$

$x_1 = -0.0032168$

$x_2 = 0.0008526$

$x_3 = 0.0355119$

Persamaan regresi: $-0.0003518 + -0.0032168x_1 + 0.0008526x_2 + 0.0355119x_3$

Masukkan nilai x_1, x_2, \dots, x_n :

50 76 29.30

Nilai y : 0.94

- Studi Kasus Interpolasi Bicubic Spine

```
Masukkan Matriks!
```

```
21 98 125 153
```

```
51 101 161 59
```

```
0 42 72 210
```

```
16 12 81 96
```

```
Masukkan nilai yang ingin dicari:
```

```
x: 0
```

```
Masukkan nilai yang ingin dicari:
```

```
y: 0
```

```
Nilai dari  $f(0.0, 0.0)$  adalah: 101.0
```

```
Apakah ingin mencari nilai lain? (y/n)
```

```
y
```

```
Masukkan nilai yang ingin dicari:
```

```
x: 0.5
```

```
Masukkan nilai yang ingin dicari:
```

```
y: 0.5
```

```
Nilai dari  $f(0.5, 0.5)$  adalah: 97.7265625
```

```
Apakah ingin mencari nilai lain? (y/n)
```

```
y
```

```
Masukkan nilai yang ingin dicari:
```

```
x: 0.25
```

```
Masukkan nilai yang ingin dicari:
```

```
y: 0.75
```

```
Masukkan nilai yang ingin dicari:
```

```
x: 0.25
```

```
Masukkan nilai yang ingin dicari:
```

```
y: 0.75
```

```
Nilai dari  $f(0.25, 0.75)$  adalah: 133.7977294921875
```

```
Apakah ingin mencari nilai lain? (y/n)
```

```
y
```

```
Masukkan nilai yang ingin dicari:
```

```
x: 0.1
```

```
Masukkan nilai yang ingin dicari:
```

```
y: 0.9
```

```
Nilai dari  $f(0.1, 0.9)$  adalah: 152.13772649999999
```

```
Apakah ingin mencari nilai lain? (y/n)
```


Bab 5

Kesimpulan

5.1 Kesimpulan

Sistem Persamaan Linier (SPL) memiliki sejumlah metode penyelesaian yang beragam, seperti eliminasi Gauss, Gauss-Jordan, matriks balikan, dan kaidah Cramer. Selain itu, SPL juga memiliki aplikasi yang luas, termasuk interpolasi polinom dan analisis regresi linear berganda. Dalam proyek tugas besar kami, kami berhasil mengimplementasikan kalkulator matriks dalam bahasa Java dengan menggunakan algoritma-algoritma tersebut. Meskipun demikian, terdapat beberapa aspek yang masih memerlukan perbaikan dalam implementasi kami, yang sebagian besar disebabkan oleh kendala waktu dan keterbatasan pemahaman kami terhadap bahasa pemrograman Java.

5.2 Saran

- Sebaiknya dari awal telah menentukan tipe data yang akan digunakan agar tidak kebingungan antar anggota
- Seharusnya pengerjaan dimulai dari saat tugas besar ini dikeluarkan dan tidak dikerjakan mepet deadline sehingga masih banyak waktu untuk memperbaiki bug dan menyelesaikan setiap fungsi yang ada di spesifikasi tugas besar.

5.3 Komentar

Sebagai siswa yang mengerjakan tugas besar ini, kami merasa bahwa proyek ini adalah pengalaman yang sangat berharga. Meskipun kami menghadapi beberapa tantangan dan hambatan di sepanjang jalan, tugas besar ini telah memberikan kami kesempatan untuk mengembangkan keterampilan pemrograman kami dalam bahasa Java, memahami konsep-konsep matriks dan algoritma-algoritma yang terkait, serta meningkatkan kemampuan dalam perencanaan dan manajemen waktu.

5.4 Refleksi

Refleksi dari tugas ini mengajarkan kami beberapa hal berharga. Pertama, kami menyadari pentingnya perencanaan yang matang, seperti pembuatan timeline kerja, untuk memastikan bahwa proyek berjalan dengan lancar dan ada kemajuan yang konsisten setiap harinya. Kami juga menyadari perlunya peningkatan dalam pengaturan waktu, karena kami sering kali menyelesaikan tugas mendekati deadline, yang dapat menimbulkan tekanan ekstra.

Selain itu, kami belajar bahwa menentukan tipe data yang akan digunakan dari awal proyek sangat penting. Ini membantu mencegah kebingungan di antara anggota

tim dan memastikan bahwa semua aspek teknis terkendali dengan baik. Kami juga menyadari bahwa memulai pengerjaan tugas besar sejak awal ketika tugas diberikan adalah pendekatan yang lebih bijak. Dengan demikian, kami memiliki lebih banyak waktu untuk memperbaiki bug, menyempurnakan fungsi-fungsi, dan menghindari penyelesaian mendekati batas waktu.

Secara keseluruhan, tugas ini memberikan kami pengalaman berharga dalam pengembangan program dengan bahasa Java dan mengajarkan kami pentingnya perencanaan dan manajemen waktu yang efisien dalam proyek-proyek mendatang.

Daftar Referensi

<https://adhityafaika.wordpress.com/2018/10/24/metode-eliminasi-gauss-dan-gauss-jordan/>
<https://www.maretong.com/2019/06/determinan-dan-invers-matriks.html>
<https://akupintar.id/info-pintar/-/blogs/matriks-pengertian-operasi-determinan-invers-dan-contoh-soal#:~:text=Invers%20matriks%20adalah%20kebalikan%20>
<https://www.kelaspintar.id/blog/tips-pintar/adjoin-matriks-11543>
<https://rpubs.com/ummiku/841844>
<http://e-journal.uajy.ac.id/19481/4/TF079013.pdf>
[https://accounting.binus.ac.id/2021/08/12/memahami-analisis-regresi-linear-berganda/#:~:text=Regresi%20linear%20berganda%20merupakan%20model,dependen%20\(Ghozali%202018\).](https://accounting.binus.ac.id/2021/08/12/memahami-analisis-regresi-linear-berganda/#:~:text=Regresi%20linear%20berganda%20merupakan%20model,dependen%20(Ghozali%202018).)

GitHub Repository

<https://github.com/akmalrmn/Algeo01-22122>