LAPORAN TUGAS BESAR 1

IF2123 - ALJABAR LINIER DAN GEOMETRI

“Sistem Persamaan Linier, Determinan, dan Aplikasinya”

Dosen:

Ir. Rila Mandala, M.Eng., Ph.D.

Arrival Dwi Sentosa, S.Kom., M.T.

Kelompok 04:

Attara Majesta Ayub (13522139)

Yasmin Farisah Salma (3522140)

Ikhwan Al Hakim (13522147)

PROGRAM STUDI TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

SEMESTER I TAHUN 2023/2024

BAB 1
DESKRIPSI MASALAH

Sistem Persamaan Linier (SPL) merupakan model dan rumus pemecahan masalah yang banyak ditemui dalam ragam multidisiplin. Dalam silabus mata kuliah Aljabar Linier dan Geometri, kami telah memperdalam berbagai metode yang digunakan untuk menyelesaikan SPL, termasuk teknik perhitungan determinan matriks serta metode eliminasi seperti eliminasi Gauss, eliminasi Gauss-Jordan, metode matriks balikan (x = A-1b), dan juga implementasi kaidah Cramer. Kemampuan komprehensif terkait metode tersebut menjadi fundamental dalam menghadapi berbagai masalah kompleks yang berkaitan dengan SPL di dunia nyata.

Dalam Tugas Besar I ini, kami menyusun sebuah *library* aljabar linier dalam Bahasa Java. Library ini mencakup berbagai fungsi, mulai dari operasi sederhana matriks, eliminasi Gauss, eliminasi Gauss-Jordan, metode matriks balikan, kaidah Cramer, dan sebagainya. Selanjutnya, library ini akan digunakan untuk menyelesaikan berbagai persoalan yang dapat dimodelkan dalam bentuk SPL, serta persoalan interpolasi dan regresi.

BAB 2
TEORI SINGKAT

2.1. Elminasi Gauss

Eliminasi Gauss adalah metode yang digunakan untuk menyelesaikan sistem persamaan linier (SPL) dengan mengurangkan atau menambahkan baris SPL sehingga matriks SPL yang diberikan dapat disederhanakan menjadi bentuk segitiga atas (*upper triangular form*). Dalam proses ini, elemen-elemen matriks utama digunakan untuk menghilangkan koefisien yang ada di bawahnya, sehingga sistem dapat dipecahkan dengan mudah menggunakan metode substitusi mundur (*backward substitution*).

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} & b_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & b_n \end{bmatrix} \rightarrow \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} & b_2 \\ 0 & 0 & a_{33} & \cdots & a_{3n} & b_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \cdots \\ 0 & 0 & 0 & 0 & a_{nn} & b_n \end{bmatrix}$$

Gambar 2.1 Operasi eliminasi Gauss

2.2. Eliminasi Gauss-Jordan

Eliminasi Gauss-Jordan adalah metode yang bertujuan untuk mendapatkan bentuk matriks SPL yang disederhanakan menjadi bentuk eselon tereduksi (*reduced row echelon form*) dengan semua elemen diagonal utama menjadi 1 dan semua elemen di atas dan di bawah diagonal utama menjadi 0. Ini adalah bentuk yang sangat terstruktur dan dapat memberikan solusi yang unik jika SPL memenuhi kriteria. Metode ini melibatkan operasi baris elementer, seperti menukar satu baris dengan nilai dari baris lainnya.

$$\left[\begin{array}{cccccc} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} & b_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} & b_n \end{array}\right] \Rightarrow \left[\begin{array}{cccccc} 1 & 0 & 0 & \dots & 0 & d_1 \\ 0 & 1 & 0 & \dots & 0 & d_2 \\ 0 & 0 & 1 & \dots & 0 & d_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & d_n \end{array}\right]$$

Gambar 2.2 Operasi eliminasi Gauss-Jordan

2.3. Determinan

Determinan matriks adalah suatu nilai skalar yang dapat dihitung dari matriks persegi (n x n) dan menyediakan informasi tentang sifat geometris dan transformasi linear matriks tersebut. Pendekatan determinan bisa melalui berbagai cara, dan yang saat ini diperdalam adalah ekspansi kofaktor dengan menggunakan minor dan kofaktor, juga metode reduksi baris.

2.4. Matriks Balikan

Matriks balikan dapat diperoleh dengan dua metode. Metode pertama adalah dengan terlebih dahulu mencari determinan matriks dan adjoin nya. Sehingga, eksistensi matriks invers bergantung pada nilai determinannya. Jika determinannya bernilai nol, maka disimpulkan bahwa matriks tidak mempunyai invers (matriks singular).

$$A^{-1} = \frac{1}{|A|} Adj(A)$$

Metode kedua pencarian invers matriks menggunakan metode matriks balikan. Hal ini dilakukan dengan menambahkan matriks identitas di sebelah kanan. Kemudian operasikan sedemikian rupa sehingga matriks pada bagian kiri termodifikasi menjadi matriks identitas, sedangkan matriks sebelah kanan bernilai.

$$A = \left[\begin{array}{cccc|cccc} 1 & -2 & 1 & -2 & 1 & 0 & 0 & 0 \\ 2 & 4 & 3 & -1 & 0 & 1 & 0 & 0 \\ 1 & 3 & 1 & 0 & 0 & 0 & 1 & 0 \\ -3 & 1 & 4 & 3 & 0 & 0 & 0 & 1 \end{array}\right] \Rightarrow A = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & -\frac{1}{3} & -\frac{11}{6} & \frac{5}{2} & -\frac{5}{6} \\ 0 & 1 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 1 & 0 & -\frac{2}{3} & \frac{5}{6} & -\frac{3}{2} & -\frac{1}{6} \\ 0 & 0 & 0 & 1 & -\frac{4}{3} & -\frac{5}{6} & \frac{1}{2} & -\frac{5}{6} \end{array}\right]$$

Gambar 2.4 Invers dengan metode matriks balikan

2.5. Matriks Kofaktor

Matriks kofaktor merupakan suatu matriks yang terbentuk dari kofaktor-kofaktor matriks asalnya. Dalam prosedurnya, kita harus terlebih dahulu mencari minor dari setiap elemen matriks. Untuk setiap elemen matriks A baris ke-i dan kolom ke-j, minor nya adalah determinan submatriks A yang diperoleh dengan menghilangkan elemen baris ke-i dan kolom ke-j. Setelah diperoleh matriks minor A, kita dapat menentukan nilai atau harga dari kofaktor. Nilai kofaktor tersebut disusun sesuai tata letak masing-masing. Susunan ini akan menghasilkan sebuah matriks baru yang dinamakan dengan matriks kofaktor.

2.6. Matriks Adjoin

Matriks adjoin adalah matriks yang diperoleh dengan membentuk matriks kofaktor kemudian melakukan operasi transpose pada matriks tersebut. Misalnya, kita memiliki matriks A sebagai berikut,

$$A = \begin{bmatrix} 2 & 3 & 5 \\ 4 & 1 & 6 \\ 1 & 4 & 0 \end{bmatrix}$$

Lalu, modifikasi dilakukan sedemikian rupa sehingga kita mendapatkan matriks kofaktor A, yang akan dinamakan matriks C, Maka, matriks adjoin dari matriks A adalah transpose dari matriks kofaktor atau C’,

$$C^T = \begin{bmatrix} -24 & 20 & 13 \\ 6 & -5 & 8 \\ 15 & -5 & -10 \end{bmatrix}$$

2.7. Kaidah Cramer

Aturan Cramer atau disebut juga Kaidah Cramer, diusulkan pertama kali oleh Gabriel Cramer (1704–1752). Metode Cramer berlaku jika jumlah variabel yang tidak diketahui berjumlah sama dengan persamaannya. Maka, jika jumlah persamaannya lebih banyak akan menyebabkan determinan penyebutnya bernilai 0. Sedangkan, jika jumlah variabel lebih sedikit dapat menyebabkan matriks tidak berbentuk persegi, membuat tidak bisa dihitung determinan.

2.8. Interpolasi Polinom

Interpolasi polinomial adalah teknik interpolasi dengan mengasumsikan bahwa 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.

2.9. Interpolasi *bicubic spline*

Interpolasi ini memberikan kurva tergabung halus. Hal ini karena setiap bagian individu diwakili oleh kurva kubik, atau polinomial derajat 3. Sehingga, masing-masing bagian individu juga dapat dianalisis sebagai kurva kubik. Dengan asumsi terdapat n titik data untuk interpolasi, kita akan mendefinisikan Si sebagai fungsi polinomial kubik yang mewakili kurva pada domain $[xi \; ; \; xi + 1]$. Kemudian untuk n titik observasi, ada $(n - 1)$ interpolasi polinomial kubik.

2.10. Regresi Linier Berganda

Regresi linear berganda adalah model regresi yang melibatkan multivariabel independen. Mengetahui arah dan besar pengaruh variabel independen terhadap variabel yang dependen adalah tujuan analisis regresi linear dilakukan. (Ghozali, 2018).

BAB 3
IMPLEMENTASI PUSTAKA

Program Java kami mengumpulkan implementasi algoritma pada folder source code, yang di dalamnya terdapat file Main.java juga 3 packages dengan total 13 class, yaitu:

1. Functions

    Functions berisi implementasi dari algoritma pencarian solusi dengan input berupa matriks. Algoritma tersebut dikemas dalam 3 class sebagai berikut:
    - Interpolation.java
    - Regression.java
    - SPL.java

1.1. Interpolation.java

Class ini berisi implementasi dari algoritma interpolasi polinomial dan interpolasi bicubic.

- Attributes

Attribute	**Description**
int derajat	Berisi derajat dari interpolasi polinom
double[][] matriks	Menyimpah data matriks *augmented* input
double hasil	Merepresentasikan nilai taksiran sebagai solusi dari interpolasi polinom

- Constructor

Constructor	Description
-	-

- Methods

Method(Parameter)	Description
public static void Polynomial(double[][] titik, double a)	Membentuk persamaan polinomial kemudian menginterpolasi nilai taksiran
public static void BicubicSpline(double[][] input, double a, double b)	Mengestimasi nilai di antara titik-titik data yang sudah diketahui dengan tingkat presisi yang tinggi.

1.2. Regression.Java

Pada class Regression.java, regresi matriks dicari dengan *Normal Estimation Equation for Multiple Linear Regression* kemudian diselesaikan dengan eliminasi Gauss.

- Attributes

Attribute	Description
int m	Merepresentasikan jumlah baris matriks
int n	Merepresentasikan jumlah kolom matriks
double est	Merepresentasikan hasil estimasi nilai f(xk)
double[] solutions	Menampung hasil x1, x2, …,

	xk
double[][] result	Menampung data SPL sebelum eliminasi Gauss

- Constructor

Constructor	**Description**
-	-

- Methods

Method(Parameter)	**Description**
public static void AppendRegresi (double[][] result, double[][] B, double sumY, int persrow, int n)	Memasukkan SPL baris ke-*persrow* ke matriks *result*
public static void RegresiMatrix (double[][] matrix)	Mengoperasikan regresi pada matriks

1.3. SPL.java

Pada class SPL, terdapat 4 metode untuk mencari nilai variabel Sistem Persamaan Linier: eliminasi Gauss, eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah *Cramer.*

- Attributes

Attribute	**Description**
int row	Merepresentasikan jumlah baris matriks

int col	Merepresentasikan jumlah kolom matriks
double[][] matrix	Menampung data angka double yang merupakan isi matriks
double[] solutions	Menampung hasil x1, x2, …, xk
double[][] result	Menampung data angka double hasil metode matriks balikan

- Methods

Method(Parameter)	**Description**
public static boolean CheckSolution (double[][] matrix)	Mengecek apakah matriks memiliki solusi atau tidak
public static double[] Gauss (double[][] matrix, double[] solutions)	Mencari nilai X matriks dengan operasi eliminasi Gauss
public static double[][] DeleteZeros (double[][] matrix)	Menghapus row matriks yang semua elemennya terdiri dari angka 0
public static void GaussJordan(double[][] matrix)	Mencari nilai X matriks dengan operasi eliminasi Gauss-Jordan
public static void MatriksBalikan(double[][] matrix)	Mencari nilai X dengan Metode Matriks Balikan
public static void Cramer(double[][] matrix)	Mencari nilai X matriks dengan Kaidah Cramer
public static double recursiveCalculation(int	Membantu perhitungan persamaan matriks secara

topLimit, int bottomLimit, int currentRow, int variableColumn, double[] resultArray, String[] expressionArray, double[][] matrix)	rekursif
public static void Parametrik(double[][] matrix)	Membuat persamaan parametrik sebagai output jika matriks memiliki banyak solusi

2. Matrices

Matrices berisi implementasi dari algoritma yang memodifikasi sebuah matriks. Algoritma tersebut dikemas dalam 8 class sebagai berikut:

- Cofactor.java
- Determinant.java
- Echelon.java
- Inverse.java
- MatricesIO.java
- PrintMatrices.java
- ReadMatrices.java
- Transpose.java

2.1. Cofactor.java

Pada class Cofactor, terdapat 2 metode yang saling komplemen untuk mencari kofaktor sebuah matriks.

- Attributes

Attribute	**Description**
int m	Merepresentasikan jumlah baris matriks

int n	Merepresentasikan jumlah kolom matriks
double[][] result	Menyimpan data angka double hasil matriks kofaktor

- Constructor

Constructor	**Description**
-	-

- Methods

Method(Parameter)	**Description**
public static double SingleCofactor(double[][] matriks, int baris, int kolom)	Mengubah elemen di posisi matriks tertentu menjadi kofaktornya
public static double[][] MatricesCofactor(double[][] matriks)	Mengubah matriks input menjadi matriks kofaktor

2.2. Determinant.java

Pada class Determinant, terdapat metode untuk mencari determinan dari suatu matriks.

- Attributes

Attribute	**Description**
int m	Merepresentasikan jumlah baris matriks

int n	Merepresentasikan jumlah kolom matriks
double determinan	Merepresentasikan hasil perhitungan determinan dari suatu matriks

- Constructor

Constructor	**Description**
-	-

- Methods

Method(Parameter)	**Description**
public static double CofactorExp(double[][] matriks)	Fungsi yang menghitung nilai determinan matriks inputnya
public static double RowReduction(double[][] matriks)	Metode pencarian determinan dengan menggunakan reduksi baris

2.3. Echelon.java

Pada class Echelon, terdapat 3 metode yang digunakan untuk mengubah matriks menjadi bentuk eselon baris dan eselon baris tereduksi.

- Attributes

Attribute	**Description**
int m	Merepresentasikan jumlah baris

	matriks
int n	Merepresentasikan jumlah kolom matriks

- Constructor

Constructor	**Description**
-	-

- Methods

Method(Parameter)	**Description**
public static void SwapRow(double[][] matriks, int row1, int row2)	Fungsi untuk menukar baris-baris matriks jika …
public static double[][] RowEchelon(double[][] matriks)	Fungsi untuk memodifikasi matriks input menjadi matriks eselon baris
public static double[][] ReducedRowEchelon(double[][] matriks)	Fungsi untuk memodifikasi matriks input menjadi matriks eselon baris tereduksi (hanya angka 0 yang ada di atas dan bawah angka 1)

2.4. Inverse.java

Pada class Inverse, terdapat 3 metode yang komplemen untuk mencari invers matriks: Adjoin, Cofactor, dan OBE.

- Attributes

Attribute	Description
int row	Merepresentasikan jumlah baris matriks
int col	Merepresentasikan jumlah kolom matriks
double[][] adj	
double[][] inv	
double[][] hasil	

- Constructor

Constructor	Description
-	-

- Methods

Method(Parameter)	Description
public static double[][] Adjoin(double[][] matrix)	Memodifikasi matriks awal menjadi matriks adjoin
public static double[][] Cofactor(double[][] matrix)	Memodifikasi nilai matriks awal menjadi kofaktornya
public static double[][] OBE(double[][] matriks)	Memodifikasi nilai matriks awal menjadi inversnya

2.5. MatricesIO.java

Class MatricesIO berisi implementasi algoritma untuk operasi *input* dan *output* matriks ke file.

- Attributes

Attribute	Description
String outputFileName	String yang merepresentasikan nama file
String outputPath	Untuk menyimpan file di folder Test
File file	File yang digunakan untuk menyimpan data

- Constructor

Constructor	Description
-	-

- Methods

Method(Parameter)	Description
public static void SaveMatrixToFile(double[][] matrix, String filename)	Fungsi yang menyimpan data matriks ke dalam file
public static void SaveStringToFile(String content, String filename)	Fungsi yang menyimpan data string ke dalam file
public static double[][] FileToMatrix(String filename)	Fungsi yang memanggil matriks dari file untuk dioperasikan

2.6. PrintMatrices.java

Pada class PrintMatrices, terdapat metode Print untuk memberi luaran matriks pada terminal.

- Attributes

Attribute	Description
int n	Merepresentasikan jumlah baris matriks
int m	Merepresentasikan jumlah kolom matriks

- Constructor

Constructor	Description
-	-

- Methods

Method(Parameter)	Description
public static void Print(double[][] matrix)	Fungsi yang digunakan untuk mencetak matriks ke terminal.
public static double[][] CreateHilbert(int n)	Fungsi untuk membuat matriks Hilbert

2.7. ReadMatrices.java

Pada class ReadMatrices, terdapat metode untuk membaca matriks di layar dan menyimpan nilainya.

- Attributes

Attribute	Description

int m	Merepresentasikan jumlah baris matriks
int n	Merepresentasikan jumlah kolom matriks
double[][] matrix	Menyimpan data angka double yang dibaca dari input user

- Constructor

Constructor	**Description**
-	-

- Methods

Method(Parameter)	**Description**
public static double[][] Keyboard()	Fungsi yang membaca input user berupa baris, kolom, dan isi matriks

2.8. Transpose.java

Pada class Transpose, terdapat metode untuk operasi transpose (menukar kolom dan baris) pada matriks.

- Attributes

Attribute	**Description**
int row	Merepresentasikan jumlah baris matriks
int col	Merepresentasikan jumlah

	kolom matriks
double[][] trans	Menyimpan data angka double hasil transpose pada matriks

- Constructor

Constructor	**Description**
-	-

- Methods

Method(Parameter)	**Description**
public static double[][] TransposeMatrix(double[][] matrix)	Fungsi yang memodifikasi matriks input menjadi matriks transposenya

3. Operations

Operations berisi implementasi dari algoritma yang mengoperasikan dua matriks. Dibagi ke dalam 2 class sebagai berikut:

- Airthmetics.java
- Multiplies.java

3.1. Arithmetics.java

Pada class Arithmetics, terdapat 3 metode yang terdiri atas penjumlahan, pengurangan, dan perhitungan *trace* matriks

- Attributes

Attribute	**Description**
int baris	Merepresentasikan jumlah baris

	matriks
int kolom	Merepresentasikan jumlah kolom matriks
double[][] result	Menyimpan data angka double hasil penjumlahan atau pengurangan matriks
double trace	Merepresentasikan hasil perhitungan *trace* matriks

- Constructor

Constructor	**Description**
-	-

- Methods

Method(Parameter)	**Description**
public static double[][] Addition(double[][] matrix1, double[][] matrix2)	Fungsi penambahan dua matriks
public static double[][] Subtraction(double[][] matrix1, double[][] matrix2)	Fungsi pengurangan dua matriks
public static double Trace(double[][] matrix)	Fungsi untuk menghitung *trace* matriks

3.2. Multiplies

Pada class Multiplies, terdapat 3 metode operasi perkalian matriks: perkalian dua matriks, perkalian matriks dengan konstanta, dan perkalian matriks dengan modulo.

- Attributes

Attribute	Description
int m	Merepresentasikan jumlah baris kedua matriks
int n	Merepresentasikan jumlah kolom matriks 1
int p	Merepresentasikan jumlah kolom matriks 2
double[][] result	Menyimpan data hasil perkalian matriks

- Constructor

Constructor	Description
-	-

- Methods

Method(Parameter)	Description
public static double[][] MultiplyMatrix(double[][] matrix1, double[][] matrix2)	Fungsi yang memberi hasil perkalian dua matriks input
public static double[][] MultiplyMatrixByConstant(double[][] matrix, double constant)	Fungsi yang memberikan hasil perkalian suatu matriks dengan suatu konstanta input
public static double[][] MultiplyMatrixWithMod(doubl	Fungsi yang mengoperasikan hasil kali dua matriks dengan

e[][] matrix1, double[][] matrix2, double mod)	modulo

BAB 4
EKSPERIMEN

Menggunakan studi kasus yang terdapat pada file "Spesifikasi Tugas Besar 1 IF2123 Algeo 2023/2024", didapatkan luaran sebagai berikut:

1. Temukan solusi SPL *Ax = b,* berikut:
   a. SPL berbentuk

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

Menggunakan eliminasi Gauss:

Menggunakan eliminasi Gauss-Jordan:

Menggunakan metode matriks balikan:

Menggunakan Kaidah *Cramer*:

   b. SPL berbentuk

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

Menggunakan eliminasi Gauss:

Menggunakan eliminasi Gauss-Jordan:

Menggunakan metode matriks balikan:

Menggunakan Kaidah *Cramer*:

c. SPL berbentuk

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

Menggunakan eliminasi Gauss:

Menggunakan eliminasi Gauss-Jordan:

Menggunakan metode matriks balikan:

Menggunakan Kaidah *Cramer*:

d. SPL berbentuk

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

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

Menggunakan eliminasi Gauss:

Menggunakan eliminasi Gauss-Jordan:

Menggunakan metode matriks balikan:

Menggunakan Kaidah *Cramer*:

2. SPL berbentuk matriks *augmented*

    a. SPL matriks *augmented* berbentuk

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

    Menggunakan eliminasi Gauss:

    Menggunakan eliminasi Gauss-Jordan:

    Menggunakan metode matriks balikan:

    Menggunakan Kaidah *Cramer*:

    b. SPL matriks *augmented* berbentuk

b.

$$\left[\begin{array}{ccccc} 2 & 0 & 8 & 0 & 8 \\ 0 & 1 & 0 & 4 & 6 \\ -4 & 0 & 6 & 0 & 6 \\ 0 & -2 & 0 & 3 & -1 \\ 2 & 0 & -4 & 0 & -4 \\ 0 & 1 & 0 & -2 & 0 \end{array}\right].$$

Menggunakan eliminasi Gauss:

Menggunakan eliminasi Gauss-Jordan:

Menggunakan metode matriks balikan:

Menggunakan Kaidah *Cramer*:

3. SPL berbentuk

   a.

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

Menggunakan eliminasi Gauss:

Menggunakan eliminasi Gauss-Jordan:

Menggunakan metode matriks balikan:

Menggunakan Kaidah *Cramer*:

b.

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

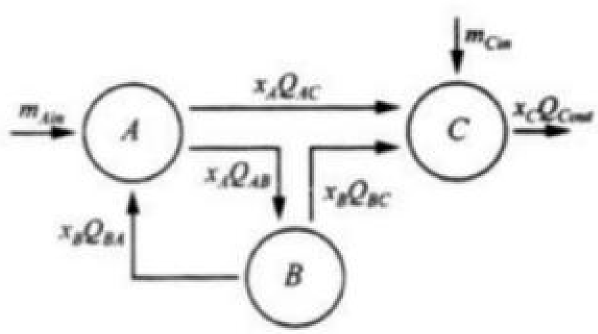
Menggunakan eliminasi Gauss:

Menggunakan eliminasi Gauss-Jordan:

Menggunakan metode matriks balikan:

Menggunakan Kaidah *Cramer*:

4. Lihatlah sistem reaktor pada gambar berikut.



Dengan laju volume Q dalam m^3/s dan input massa min dalam mg/s. Konservasi massa pada tiap inti reaktor adalah sebagai berikut:

$$
\begin{aligned}
&\text{A:} \quad m_{A_{in}} + Q_{BA}x_B - Q_{AB}x_A - Q_{AC}x_A = 0 \\
&\text{B:} \quad Q_{AB}x_A - Q_{BA}x_B - Q_{BC}x_B = 0 \\
&\text{C:} \quad m_{C_{in}} + Q_{AC}x_A + Q_{BC}x_B - Q_{C_{out}}x_C = 0
\end{aligned}
$$

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 $Q_{Cout} = 150$ m^3/s dan $m_{Ain} = 1300$ dan $m_{Cin} = 200$ mg/s.

Studi Kasus Interpolasi

a. Gunakan tabel di bawah ini untuk mencari polinom interpolasi dari pasangan titik-titik yang terdapat dalam tabel. Program menerima masukan nilai x yang akan dicari nilai fungsi $f(x)$.

x	0.1	0.3	0.5	0.7	0.9	1.1	1.3
$f(x)$	0.003	0.067	0.148	0.248	0.370	0.518	0.697

Lakukan pengujian pada nilai-nilai berikut:

$x = 0.2$ $f(x) = ?$
$x = 0.55$ $f(x) = ?$
$x = 0.85$ $f(x) = ?$
$x = 1.28$ $f(x) = ?$

Jumlah kasus positif baru Covid-19 di Indonesia semakin fluktuatif dari hari ke hari. Di bawah ini diperlihatkan jumlah kasus baru Covid-19 di Indonesia mulai dari tanggal 17 Juni 2022 hingga 31 Agustus 2022:

Tanggal	Tanggal (desimal)	Jumlah Kasus Baru
17/06/2022	6,567	12.624
30/06/2022	7	21.807
08/07/2022	7,258	38.391
14/07/2022	7,451	54.517
17/07/2022	7,548	51.952
26/07/2022	7,839	28.228
05/08/2022	8,161	35.764
15/08/2022	8,484	20.813
22/08/2022	8,709	12.408
31/08/2022	9	10.534

Tanggal (desimal) adalah tanggal yang sudah diolah ke dalam bentuk desimal 3 angka di belakang koma dengan memanfaatkan perhitungan sebagai berikut:

Tanggal (desimal) = bulan + (tanggal / jumlah hari pada bulan tersebut)

Sebagai contoh, untuk tanggal 17/06/2022 (dibaca: 17 Juni 2022) diperoleh tanggal(desimal) sebagai berikut:

Tanggal (desimal) = 6 + (17/30) = 6,567

Gunakanlah data di atas dengan memanfaatkan **interpolasi polinomial** untuk melakukan prediksi jumlah kasus baru Covid-19 pada tanggal-tanggal berikut:

a. 16/07/2022
b. 10/08/2022
c. 05/09/2022
d. Masukan user lainnya berupa **tanggal (desimal) yang sudah diolah** dengan asumsi prediksi selalu dilakukan untuk tahun 2022.

c. Sederhanakan fungsi $f(x)$ yang memenuhi kondisi

$$f(x) = \frac{x^2 + \sqrt{x}}{e^x + x}$$

dengan polinom interpolasi derajat n di dalam selang [0, 2].
Sebagai contoh, jika $n = 5$, maka titik-titik x yang diambil di dalam selang [0, 2] berjarak $h = (2 - 0)/5 = 0.4$.

Studi Kasus Regresi Linear Berganda

Diberikan sekumpulan data sesuai pada tabel berikut ini.

Table 12.1: Data for Example 12.1

Nitrous Oxide, y	Humidity, x_1	Temp., x_2	Pressure, x_3	Nitrous Oxide, y	Humidity, x_1	Temp., x_2	Pressure, x_3
0.90	72.4	76.3	29.18	1.07	23.2	76.8	29.38
0.91	41.6	70.3	29.35	0.94	47.4	86.6	29.35
0.96	34.3	77.1	29.24	1.10	31.5	76.9	29.63
0.89	35.1	68.0	29.27	1.10	10.6	86.3	29.56
1.00	10.7	79.0	29.78	1.10	11.2	86.0	29.48
1.10	12.9	67.4	29.39	0.91	73.3	76.3	29.40
1.15	8.3	66.8	29.69	0.87	75.4	77.9	29.28
1.03	20.1	76.9	29.48	0.78	96.6	78.7	29.29
0.77	72.2	77.7	29.09	0.82	107.4	86.8	29.03
1.07	24.0	67.7	29.60	0.95	54.9	70.9	29.37

Source: Charles T. Hare, "Light-Duty Diesel Emission Correction Factors for Ambient Conditions," EPA-600/2-77-116. U.S. Environmental Protection Agency.

Gunakan *Normal Estimation Equation for Multiple Linear Regression* untuk mendapatkan regresi linear berganda dari data pada tabel di atas, kemudian estimasi nilai Nitrous Oxide apabila Humidity bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30.

Dari data-data tersebut, apabila diterapkan *Normal Estimation Equation for Multiple Linear Regression*, maka diperoleh sistem persamaan linear sebagai berikut.

$$20b_0 + 863.1b_1 + 1530.4b_2 + 587.84b_3 = 19.42$$

$$863.1b_0 + 54876.89b_1 + 67000.09b_2 + 25283.395b_3 = 779.477$$

$$1530.4b_0 + 67000.09b_1 + 117912.32b_2 + 44976.867b_3 = 1483.437$$

$$587.84b_0 + 25283.395b_1 + 44976.867b_2 + 17278.5086b_3 = 571.1219$$

Studi Kasus Interpolasi *Bicubic Spline*

Diberikan matriks input dengan bentuk sebagai berikut. Format matriks masukan bukan mewakili nilai matriks, tetapi mengikuti format masukan pada bagian "Spesifikasi Tugas" nomor 7.

$$\begin{pmatrix} 21 & 98 & 125 & 153 \\ 51 & 101 & 161 & 59 \\ 0 & 42 & 72 & 210 \\ 16 & 12 & 81 & 96 \end{pmatrix}$$

Tentukan nilai:

$f(0, 0) = ?$
$f(0.5, 0.5) = ?$
$f(0.25, 0.75) = ?$
$f(0.1, 0.9) = ?$

BAB 5
KESIMPULAN

1. Kesimpulan

   Matematika merupakan ilmu pengetahuan dasar yang dibutuhkan semua manusia dalam kehidupan sehari-hari. Salah satunya Sistem Persamaan Linier (SPL) yang secara eksplisit atau implisit sebenarnya adalah dasar dari kompleksitas permasalahan dunia yang kita hadapi. SPL perlu ditemukan solusinya dengan akurasi dan ketelitian tinggi, karena berbeda sedikit akan mempengaruhi keseluruhan nilai akhirnya. Contoh penerapan Sistem Persamaan Linear (SPL) dalam aljabar elementer yakni menggunakan matriks. Matriks yang merepresentasikan SPL dapat dihitung dengan banyak metode, seperti eliminasi Gauss, eliminasi Gauss-Jordan, metode matriks balikan (x = A-1b), dan juga implementasi kaidah Cramer. Kemampuan komprehensif terkait ilmu ini meningkat secara signifikan setelah kami menyelesaikan Tugas Besar I.

2. Saran
   - Menguji seluruh *testcase* jauh-jauh hari sehingga segala bentuk ketidaktelitian dapat direpresi secepat mungkin.
   - Lebih pandai mengatur waktu, jangan *deadliner*

3. Refleksi

   Mengerjakan Tugas Besar I menyadarkan kami bahwa bekerja secara sukarela dengan rasa keingintahuan yang tinggi akan meningkatkan produktivitas lingkungan kerja. Kami membuat tiga *deadline* untuk menyelesaikan tiga tahap pekerjaan: operasi dasar matriks, spesifikasi, dan *finishing*. Hal ini membuka kesempatan untuk saling membantu jika ada kesulitan.

BAB 6
REFERENSI

http://repositori.uin-alauddin.ac.id/16727/1/%2836%29Yeyen%20Syafira%20L.%20H.pdf

https://www.lkouniv.ac.in/site/writereaddata/siteContent/202004032250571912siddharth_bhatt_engg_Interpolation.pdf