LAPORAN TUGAS BESAR I

IF2123 ALJABAR LINIER DAN GEOMETRI

NOOGLER

ANGGOTA NOOGLER :

1. **JEFFREY CHOW (13521046)**
2. **WILSON TANSIL (13521054)**
3. **JIMLY FIRDAUS (13521102)**

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BAB I
DESKRIPSI MASALAH

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Kami 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).

Di dalam Tugas Besar 1 ini, kami 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.

BAB II
TEORI SINGKAT

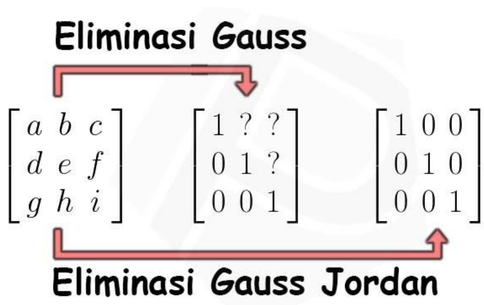
1. **Metode eliminasi Gauss**

   Merupakan salah satu metode yang dapat digunakan dalam menyelesaikan Sistem Persamaan Linear. Metode ini menggunakan eliminasi Gauss untuk menghasilkan matriks dengan bentuk eselon (*row echelon form*). Metode eliminasi Gauss merupakan pengembangan dari metode eliminasi biasa yang berusaha untuk menghilangkan / mengurangi jumlah variabel sehingga dapat diperoleh nilai dari suatu variabel bebas. Metode pengurangan / menghilangkan variabel tersebut dinamakan Operasi Baris Elementer (OBE).

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_n \end{bmatrix} \sim\text{OBE}\sim \begin{bmatrix} 1 & * & * & \dots & * & * \\ 0 & 1 & * & \dots & * & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \vdots & 1 & * \end{bmatrix}$$

2. **Metode eliminasi Gauss-Jordan**

   Eliminasi Gauss-Jordan merupakan salah satu metode yang umum digunakan dalam penyelesaian sistem persamaan linear yang kompleks. Metode ini merupakan metode lanjutan dari metode eliminasi gauss yang mengubah bentuk matriks eselon (*row echelon form*) menjadi bentuk eselon tereduksi (*reduction echelon form*). Sebuah matriks dapat dikatakan memiliki bentuk eselon tereduksi apabila setiap koefisien utama bernilai 1.



3. **Determinan**

   Determinan merupakan nilai dari perkalian diagonal utama dari sebuah matriks persegi. Terdapat 2 metode dalam mendapatkan nilai determinan dari suatu matriks persegi, yaitu :

   1. Dengan reduksi baris membentuk matriks segitiga.

Metode ini dapat menghasilkan matriks segitiga atas atau matriks segitiga bawah.

Matriks segitiga atas

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \Rightarrow \det(A) = a_{11}a_{22}\,a_{33}a_{44}$$

Matriks segitiga bawah

$$A = \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \Rightarrow \det(A) = a_{11}a_{22}\,a_{33}a_{44}$$

Matriks segitiga dapat diperoleh dengan menerapkan OBE kepada matriks persegi dengan aturan berikut:

$$[A] \overset{\text{OBE}}{\sim} [\text{matriks segitiga bawah}]$$

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \overset{\text{OBE}}{\sim} \begin{bmatrix} a'_{11} & a'_{12} & \dots & a'_{1n} \\ 0 & a'_{22} & \dots & a'_{2n} \\ \vdots & \vdots & \dots & a'_{3n} \\ 0 & 0 & 0 & a'_{nn} \end{bmatrix}$$

$$\text{maka } \det(A) = (-1)^p\, a'_{11}a'_{22} \dots a'_{nn}$$

p menyatakan banyaknya operasi pertukaran baris di dalam OBE

2. Dengan metode ekspansi kofaktor

   Misalkan A adalah matriks berukuran n x n dengan :

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

   Didefinisikan:

   M_{ij} = minor entri a_{ij}

   $C_{ij} = (-1)^{i+j} M_{ij}$, adalah kofaktor entri a_{ij}

Dengan tanda positif dan negatif untuk C_{ij} memperhatikan pola berikut:

$$\begin{bmatrix} + & - & + & - & \dots \\ - & + & - & + & \dots \\ + & - & + & - & \dots \\ - & + & - & + & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

Maka determinan dapat dihitung dengan menggunakan <u>salah satu</u> dari persamaan berikut:

$$\det(A) = a_{11}C_{11} + a_{12}C_{12} + \dots + a_{1n}C_{1n}$$
$$\det(A) = a_{21}C_{21} + a_{22}C_{22} + \dots + a_{2n}C_{2n}$$
$$\vdots$$
$$\det(A) = a_{n1}C_{n1} + a_{n2}C_{n2} + \dots + a_{nn}C_{nn}$$

Secara baris

$$\det(A) = a_{11}C_{11} + a_{21}C_{21} + \dots + a_{n1}C_{n1}$$
$$\det(A) = a_{12}C_{12} + a_{22}C_{22} + \dots + a_{n2}C_{n2}$$
$$\vdots$$
$$\det(A) = a_{1n}C_{1n} + a_{2n}C_{2n} + \dots + a_{nn}C_{nn}$$

Secara kolom

4. Matriks balikan

Matriks balikkan / invers matriks adalah matriks baru yang merupakan kebalikan dari matriks awal. Matriks balikkan dapat ditulis sebagai A^{-1} , dengan A adalah suatu matriks persegi. Matriks balikkan dapat digunakan untuk menyelesaikan suatu sistem persamaan linear. Penentuan matriks balikkan dari suatu matriks dapat dilakukan dengan menghitung determinan dari matriks terlebih dahulu atau menggunakan metode Eliminasi Gauss-Jordan. Penentuan dengan menggunakan determinan adalah jika determinan dari matriks tersebut tidak nol maka matriks tersebut mempunyai balikkan dan sebaliknya. Penentuan matriks balikkan dengan metode Eliminasi Gauss-Jordan adalah dengan menambah sebuah matriks identitas dengan ukuran yang sama lalu diterapkan OBE pada kedua matriks tersebut dengan cara $[A|I]$ ~ Gauss-Jordan ~ $[I|A^{-1}]$. Jika terdapat 1 baris atau lebih yang seluruh elemennya bernilai 0 maka dapat disimpulkan matriks tersebut tidak memiliki balikkan.

5. Matriks kofaktor

Matriks kofaktor merupakan matriks yang terdiri dari kofaktor-kofaktor matriks itu sendiri. Misalkan A adalah matriks n x n dan C_{ij} merupakan kofaktor entri dari a_{ij}, maka matriks kofaktor dari A adalah:

$$\begin{bmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ C_{n1} & C_{n2} & \dots & C_{nn} \end{bmatrix}$$

6. **Matriks adjoin**

Matriks adjoin merupakan hasil transpose dari matriks kofaktor. Penulisan adjoin dapat disingkat menjadi adj(A), dengan A adalah matriks. Adjoin matriks dapat digunakan untuk menentukan balikkan dari suatu matriks, dengan:

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

7. **Kaidah Cramer**

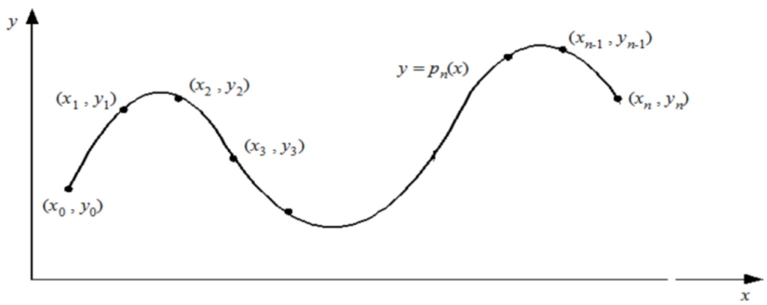
Kaidah cramer adalah metode yang dikembangkan untuk menyelesaikan suatu sistem persamaan linear dengan catatan banyak persamaan sama dengan banyak variabel. Kaidah cramer dalam penerapannya menggunakan determinan matriks koefisien dan determinan matriks lain yang diperoleh dengan mengganti salah satu kolom dalam matriks tersebut dengan kolom paling kanan augmented matrix secara bergantian.

$$x = \frac{\begin{vmatrix} d_1 & b_1 & c_1 \\ d_2 & b_2 & c_2 \\ d_3 & b_3 & c_3 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}}, \quad y = \frac{\begin{vmatrix} a_1 & d_1 & c_1 \\ a_2 & d_2 & c_2 \\ a_3 & d_3 & c_3 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}}, \text{ dan } z = \frac{\begin{vmatrix} a_1 & b_1 & d_1 \\ a_2 & b_2 & d_2 \\ a_3 & b_3 & d_3 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}}.$$

8. **Interpolasi polinom**

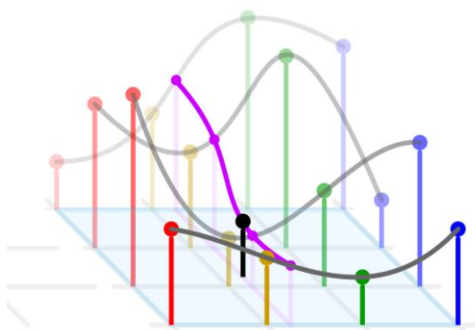
Interpolasi polinom merupakan interpolasi dari kumpulan data yang diberikan derajat terendah yang paling dimungkinkan, melewati titik-titik kumpulan data tersebut sehingga dapat dibuat kurva. Titik-titik yang diinterpolasi dengan sebuah kurva/fungsi berbentuk polinom, maka polinom tersebut dapat disebut sebagai polinom interpolasi.

$$p_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$$



9. Interpolasi bicubic

Interpolasi bicubic adalah teknik interpolasi yang bisa digunakan pada data 2D yang biasa digunakan dalam pembesaran citra dan untuk mendapatkan nilai hubungan paling memungkinkan pad kurva 3 dimensi. Interpolasi bicubic merupakan pengembangan dari interpolasi linear dan kubik.



Bicubic

10. Regresi linear berganda

Regresi linear berganda adalah model regresi yang melibatkan lebih dari satu variabel independen. Analisis regresi linear berganda pada umumnya ditujukan untuk memprediksi arah dan seberapa besar pengaruh variabel independen terhadap variabel dependen. Data yang biasa digunakan biasanya berskala interval atau rasio.

$$h_w(x) = w_0 + w_1x_1 + w_2x_2 + \cdots + w_mx_m$$

$$h_w(x) = w_0 + \sum_{i=0}^{m} w_ix_i$$

BAB III

IMPLEMENTASI PUSTAKA DAN PROGRAM

1. Class Matrix

- Attribute

Nama	Tipe	Deskripsi
rows	public int	Menyimpan jumlah baris
cols	public int	Menyimpan jumlah kolom
Mtrx	public double[][]	Menyimpan data matriks
sign	public int	Menyimpan data tanda pada determinan
has_inversed	public boolean	Digunakan untuk memproses hasil inverse
polynom_x	public double	Menyimpan nilai taksiran x untuk interpolasi polinom

- Methods

Nama	Tipe	Parameter	Deskripsi
Matrix	public Constructor	int rows int cols	Meinisialisasi matriks
check	public static Matrix	Matrix matrix	Memformat setiap elemen matriks yang bernilai negatif 0 menjadi 0
check	public static double[]	double[] x	Memformat setiap elemen array yang bernilai negatif 0 menjadi 0
copyMtrx	public static Matrix	Matrix matrix	Mengirimkan matriks baru yang sudah dicopy dari m

isSquare	public static boolean	Matrix matrix	Mengecek apakah matriks persegi atau tidak
switchCol	public static Matrix	Matrix m1 Matrix m2 int cols1 int cols2	Menukarkan kolom tertentu dari 2 matriks yang diberikan
timesDiagonal	public static double	Matrix matrix boolean isSquare	Mengalikan elemen diagonal utama pada matriks
transpose	public static Matrix	Matrix matrix	Mengembalikan salinan matriks yang sudah ditranspose

2. Class Param
    - Attributes

Nama	Tipe	Deskripsi
val	public double	Value untuk persamaan parametrik
params	public char[]	Varibel parametrik yang digunakan
valPar	public double[]	Konstanta pada variabel parametrik

3. Class Gauss
    - Methods

Nama	Tipe	Parameter	Deskripsi
gauss	public static Matrix	Matrix matrix	Mengembalikan salinan matriks yang

			sudah menjadi eselon baris
count0	public static int	Matrix matrix int row int limit	Menghitung jumlah 0 pada suatu baris
check0RemainingRows	public static boolean	Matrix matrix int current_row	Menghitung jumlah baris 0 pada sisa baris yang diberikan
switchRows	public static void	Matrix matrix int row1 int row2	Menukar kedua baris yang diberikan pada matriks
isNotFinalEchelon	public static boolean	Matrix matrix	Mengecek apakah hasil dari gauss sudah memenuhi syarat matriks baris eselon.
switchFor0TrailsFront	public static void	Matrix matrix	Menukar posisi jika jumlah 0 didepan 1 utama di *current row* lebih banyak dari jumlah 0 di depan *next row*.
switchRows	public static void	Matrix matrix int start_row	Menempatkan baris 0 menjadi baris paling bawah matriks

4. Class GaussJordan
    - Methods

Nama	Tipe	Parameter	Deskripsi
jordan	public static Matrix	Matrix matrix	Mengembalikan salinan matriks yang sudah menjadi eselon baris tereduksi

5. Class Cramer
    - Methods

Nama	Tipe	Parameter	Deskripsi
cramer	public static double[]	Matrix matrix	Mengembalikan array of double nilai hasil SPL menggunakan kaidah Cramer

6. Class determinant
    - Methods

Nama	Tipe	Parameter	Deskripsi
detCofactor	public static double	Matrix matrix int n boolean isSquare	Mengembalikan determinan dari matriks yang diberikan dengan metode kofaktor
getCofactor	public static void	Matrix matrix Matrix temp int permitted_row int permitted_col int n	Mendapatkan kofaktor dari suatu baris dan kolom dalam matriks

detGauss	public static double	Matrix matrix boolean isSquare	Mengembalikan determinan dari matriks yang diberikan dengan metode Gauss (segitiga atas)

7. Class inputSPL
    - Methods

Nama	Tipe	Parameter	Deskripsi
chooseMethods	public static int		Mengembalikan pilihan cara untuk menyelesaikan SPL
processMethods	public static void	int choice Matrix matrix	Memanggil fungsi/prosedur sesuai pilihan
printGauss	public static void	Matrix matrix Matrix real	Mengembalikan hasil penyelesaian SPL dengan metode Gauss
isNone	public static boolean	Matrix matrix	Mengecek apakah matrix augmented tidak memiliki solusi.
hasNaN	public static boolean	double[] unik	Mengecek apakah ada NaN pada suatu array double.
uniqueCase	public static double[]	Matrix matrix	Mengembalikan nilai x1,x2,x3,..,xn jika solusi SPL unik
infiniteCase	public static Param[]	Matrix matrix Matrix real	Mengembalikan nilai parametrik

			x1,x2,…,xn jika SPL solusi SPL tak berhingga
countNotZero	public static int	Matrix matrix int cols	Menghitung jumlah tidak 0 pada suatu kolom matriks
countTrue	public static int	boolean[] ans	Menentukan banyak variabel yang parametrik
firstNotZero	public static int	Matrix matrix Int rows	Mencari kolom pertama dalam baris yang tidak nol
findVal	public static int	int[] matrix int rows	Mencari index find dalam array of integer indexParam
printCramer	public static int	double[] ans	Mengoutput hasil dari Cramer
processInv	public static double[]	Matrix matrix Matrix inversed	Menghasikan solusi SPL dengan metode matriks balikkan
count0	public static int	Matrix matrix int cols	Menghitung jumlah 0 pada suatu kolom matriks

8. Class InverseCofactor
    - Methods

Nama	Tipe	Parameter	Deskripsi
inverse	public static Matrix	Matrix matrix	Mengembalikan salinan matriks yang sudah diinvers

adjoinInverse	public static void	Matrix matrix Matrix adjMtrx	Mencari adjoin dari Matrix olahan metode kofaktor

9. Class InverseGaussJordan
    - Methods

Nama	Tipe	Parameter	Deskripsi
inverse	public static Matrix	Matrix matrix	Mengembalikan salinan matriks yang sudah diinvers
count0	public static int	Matrix matrix int row	Mengembalikan jumlah elemen 0 pada baris tertentu

BAB IV

EKSPERIMEN

1. Solusi SPL Ax = b

<table>
<tr><td>a.</td><td>Metode Cramer</td></tr>
<tr><td>$$A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$$</td><td>Solusi dari SPL ini tidak ada atau tidak bisa menggunakan Metode Cramer.
Silakan coba pakai metode lain, seperti Gauss/Gauss-Jordan</td></tr>
<tr><td>b.</td><td>Metode Gauss-Jordan</td></tr>
<tr><td>$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$$</td><td>X1 = 3.00 + u
X2 = 2.00u
X3 = t
X4 = -1.00 + u
X5 = u</td></tr>
<tr><td>c.</td><td>Metode Matriks Balikan</td></tr>
<tr><td>$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$</td><td>Solusi dari SPL ini tidak ada atau tidak bisa menggunakan Metode Invers.
Silakan coba pakai metode lain, seperti Gauss/Gauss-Jordan</td></tr>
<tr><td colspan="2">d.</td></tr>
<tr><td colspan="2">$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n+1} \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$</td></tr>
</table>

Untuk n = 6	Metode Gauss X1 : 8.080835703347601 X2 : -23.558849267122838 X3 : -31.735397068465872 X4 : 108.01793918179214 X5 : -43.699409319623655 X6 : -17.9528549551518
Untuk n = 10	Metode Gauss X1 : 2.0367618783013066 X2 : 35.10463414289861 X3 : -133.7802481663315 X4 : 96.20421878744672 X5 : 6.354656793156975 X6 : -69.64125005285277 X7 : 111.46697189152786 X8 : 105.6322644423025 X9 : -91.12804344018338 X10 : -67.60750820945181

2. SPL berbentuk matriks *augmented*

a.	Metode Gauss-Jordan
$\begin{bmatrix} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{bmatrix}.$	X1 = -1.00 + u X2 = 2.00t X3 = t X4 = u

b.	Metode Gauss
$\begin{bmatrix} 2 & 0 & 8 & 0 & 8 \\ 0 & 1 & 0 & 4 & 6 \\ -4 & 0 & 6 & 0 & 6 \\ 0 & -2 & 0 & 3 & -1 \\ 2 & 0 & -4 & 0 & -4 \\ 0 & 1 & 0 & -2 & 0 \end{bmatrix}$.	SPL memiliki solusi unik. X1 : 0.0 X2 : 2.0 X3 : 1.0 X4 : 1.0

3. SPL berbentuk

a.	Metode Matriks Balikan
$\begin{aligned} 8x_1 + x_2 + 3x_3 + 2x_4 &= 0 \\ 2x_1 + 9x_2 - x_3 - 2x_4 &= 1 \\ x_1 + 3x_2 + 2x_3 - x_4 &= 2 \\ x_1 + 6x_3 + 4x_4 &= 3 \end{aligned}$	X1 : -0.22432432432432434 X2 : 0.18243243243243243 X3 : 0.7094594594594594 X4 : -0.25810810810810814
b.	**Metode Gauss**
$\begin{aligned} x_7 + x_8 + x_9 &= 13.00 \\ x_4 + x_5 + x_6 &= 15.00 \\ x_1 + x_2 + x_3 &= 8.00 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_6 + x_8) + 0.61396x_9 &= 14.79 \\ 0.91421(x_3 + x_5 + x_7) + 0.25(x_2 + x_4 + x_6 + x_8) &= 14.31 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_2 + x_4) + 0.61396x_1 &= 3.81 \\ x_3 + x_6 + x_9 &= 18.00 \\ x_2 + x_5 + x_8 &= 12.00 \\ x_1 + x_4 + x_7 &= 6.00 \\ 0.04289(x_1 + x_5 + x_9) + 0.75(x_2 + x_6) + 0.61396x_3 &= 10.51 \\ 0.91421(x_1 + x_5 + x_9) + 0.25(x_2 + x_4 + x_6 + x_8) &= 16.13 \\ 0.04289(x_1 + x_5 + x_9) + 0.75(x_4 + x_8) + 0.61396x_7 &= 7.04 \end{aligned}$	SPL tidak memiliki solusi.

4. Determinan

a.	Metode Gauss
$\begin{bmatrix} 3 & 6 & 9 & 3 \\ -1 & 0 & 1 & 0 \\ 1 & 3 & 2 & -1 \\ -1 & -2 & -2 & 1 \end{bmatrix}$	Determinan : -21.00000

b.	Metode Cofactor
$\begin{bmatrix} 3 & 5 & -2 & 6 \\ 1 & 2 & -1 & 1 \\ 2 & 4 & 1 & 3 \\ 3 & 7 & 5 & 3 \end{bmatrix}$	`Determinan : 0.00000`

5. Invers

a.	Metode Gauss-Jordan

$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 3 \\ 1 & 0 & 8 \end{bmatrix}$	`-40.00 16.00 9.00`
	`13.00 -5.00 -3.00`
	`5.00 -2.00 -1.00`

b.	Metode Cofactor
$\begin{bmatrix} 1 & 6 & 4 \\ 2 & 4 & -1 \\ -1 & 2 & 5 \end{bmatrix}$	`Matrix has no inverse`

6. Interpolasi Titik

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	0.4	0.7	0.11	0.14	0.17	0.2	0.23
f(*x*)	0.043	0.005	0. 058	0.072	0.1	0.13	0.147

x	0.2	0.55	0.85	1.28
f(x)	`f(0.2) = 0.13000`	`f(0.55) = 2.13757`	`f(0.85) = -66.26964`	`f(1.28) = -3485.14490`

7. Interpolasi Kasus Covid

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

Gunakanlah data di atas dengan memanfaatkan **polinom interpolasi** 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. beserta masukan user lainnya berupa **tanggal (desimal) yang sudah diolah** dengan asumsi prediksi selalu dilakukan untuk tahun 2022.

Tanggal	16/07/2022	10/08/2022	05/09/2022
Prediksi	`f(7.516) = 53537.80078` 53.537 kasus baru	`f(8.333) = 35765.38672` 35.765 kasus baru	`f(9.161) = -616860.64844` 0 kasus baru (asumsi jika <= 0 maka = 0)

8. Penyederhanaan Fungsi

Sederhanakan fungsi	Dengan n = 5 pada selang [0, 2] dan berjarak 0.4
$$f(x) = \frac{x^2 + \sqrt{x}}{e^x + x}$$	`f(x) = -0.00376X4+0.02421X3-0.15093X2+0.37831X1+0.29025` $f(x) = -0.00376x^4 + 0.02421x^3 - 0.15093x^2 + 0.37831x + 0.29025$

9. Interpolasi Bicubic

Diberikan matriks input:

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

Tentukan nilai:

f(0,0) = ?

f(0.5, 0.5) = ?

f(0.25, 0.75) = ?

f(0.1, 0.9) = ?

```
153 59 210 96
125 161 72 81
98 101 42 12
21 51 0 16
0 0
```

```
f(0.0,0.0) : 161.00000
```

```
153 59 210 96
125 161 72 81
98 101 42 12
21 51 0 16
0.5 0.5
```

```
f(0.5,0.5) : 97.72656
```

```
153 59 210 96
125 161 72 81
98 101 42 12
21 51 0 16
0.25 0.75
```

```
f(0.25,0.75) : 82.50208
```

```
153 59 210 96
125 161 72 81
98 101 42 12
21 51 0 16
0.1 0.9
```

```
f(0.1,0.9) : 74.69612
```

10. Regresi Linier Berganda

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.

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

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

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

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

```
f(x) = -3.50778-0.00262X1+0.00080X2+0.15416X3

Masukkan X1 : 50
Masukkan X2 : 76
Masukkan X3 : 29.30
Taksiran/Estimasi nilai : 0.9384
```

BAB V

KESIMPULAN, SARAN, DAN REFLEKSI

Berbagai metode dapat diterapkan untuk menyelesaikan suatu sistem persamaan linear, salah satunya adalah dengan menggunakan metode-metode yang diproses dalam bentuk matriks. Metode yang disediakan dengan tujuan menyelesaikan suatu sistem persamaan linear diantaranya adalah Eliminasi Gauss, Eliminasi Gauss-Jordan, Matriks Balikan dan Kaidah Cramer. Dalam penggunaan metode-metode tersebut, terdapat berbagai ketentuan dan aturan. Misalkan dalam penggunaan Kaidah Cramer, sistem persamaan linear yang diinput harus memiliki jumlah persamaan yang sama dengan jumlah variabel.

Adapun tools yang dapat digunakan untuk menjalankan metode-metode penyelesaian sistem persamaan tersebut. Tools yang disediakan berupa determinan, kofaktor, dan adjoin.

Solusi yang didapat dari penyelesaian sistem persamaan tersebut dapat diklasifikasikan menjadi 3 jenis yakni unique solution, multi solution, dan no solution. Unique solution merujuk ke solusi persamaan yang semua nilai variabel dalam suatu persamaan memiliki hanya 1 nilai atau unik. Multi solution didapatkan dalam keadaan yang mengakibatkan suatu variabel memiliki nilai bebas atau banyak nilai yang memenuhi, biasanya direpresentasikan dalam bentuk parametrik. No solution adalah kondisi yang menunjuk terhadap sebuah nilai variabel yang tidak dapat memenuhi sistem persamaan tersebut.

Interpolasi polinom adalah teknik interpolasi yang digunakan untuk memprediksi data sesuai kurva yang dibentuk menggunakan sekumpulan data atau sample. Interpolasi polinom merupakan salah satu contoh aplikasi dari sistem persamaan linear.

Interpolasi bikubik adalah teknik interpolasi yang dikembangkan dari interpolasi linear dan kubik untuk mendapatkan sebuah nilai prediksi dalam rentang tertentu. Dalam kasus yang diberikan, matriks 4x4 tersebut merupakan sebuah kumpulan set data atau sample data yang apabila digambarkan akan membentuk sebuah kurva alas xy dalam sebuah bidang kartesian 3 dimensi. Interpolasi bikubik dalam hal ini digunakan untuk memprediksi sebuah nilai hasil (yang berada pada bidang z) sesuai dengan kurva yang terbentuk.

Regresi linear berganda digunakan untuk mendapatkan sebuah kesimpulan dari data-data yang didapatkan. Dengan menggunakan regresi linear, estimasi sebuah nilai dari data tersebut dapat diperoleh dan diprediksi.

Permasalahan-permasalahan pada eksperimen dapat diselesaikan dengan menggunakan program java yang telah dibuat.

Saran pengembangan untuk tugas besar ini adalah dapat di-*upgrade interface* programnya menggunakan GUI atau tampilan di console yang lebih rapi. Selain itu, pemberian test-case yang disertai dengan solusi juga akan membantu proses *debugging*, terutama untuk matrix augmented yang memiliki dimensi besar.

Refleksi kami terhadap Tugas Besar ini adalah kami merasa dapat memperbaiki kerja sama dalam tim kami dari segi komunikasi, pembagian tugas, dan tahap pengerjaan. Dengan komunikasi yang lebih baik, akan mengurangi terjadinya miskomunikasi dan mempercepat tahap pengerjaan. Dengan pembagian tugas yang lebih baik, maka pengerjaan tugas akan lebih sistematis, dan masing-masing anggota dapat lebih fokus dengan pekerjaannya sendiri. Tahap pengerjaan seharusnya lebih cepat dan melakukan *debugging* lebih awal agar tidak *chaos* di akhir *deadline* tugas besar ini.

REFERENSI

https://www.programcreek.com/2011/07/build-a-java-library-for-yourself/

https://developer.ibm.com/tutorials/j-javalibrary/

https://stackoverflow.com/questions/3612567/how-to-create-my-own-java-libraryapi

https://accounting.binus.ac.id/2021/08/12/memahami-analisis-regresi-linear-berganda/#:~:text=Regresi%20linear%20berganda%20merupakan%20model,dependen%20(Ghozali%2C%202018)

REPOSITORY

https://github.com/JeffreyChow19/Algeo01-21046.git