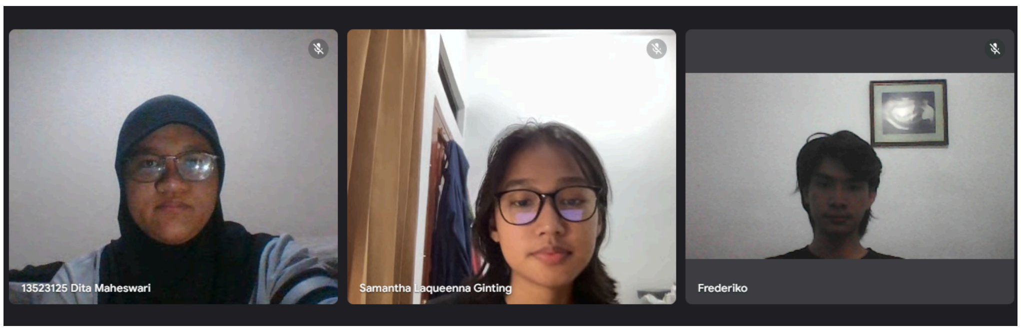
TUGAS BESAR
IF2123 ALJABAR LINIER DAN GEOMETRI



Dipersiapkan oleh:

Kelompok 38

13523125 - Dita Maheswari
13523138 - Samantha Laqueenna Ginting
13523147 - Frederiko Eldad Mugiyono

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
JL. GANESA 10, BANDUNG 40132

2024

DAFTAR ISI

BAB 1 DESKRIPSI MASALAH

Matriks adalah kumpulan bilangan real atau kompleks yang disusun dalam baris dan kolom. Informasi di bidang sains dan matematika seringkali memanfaatkan matriks. Dengan menggunakan matriks, bilangan-bilangan dapat dengan mudah dioperasikan sehingga mendapat solusi yang diinginkan.

Sistem Persamaan Linier (SPL) adalah kumpulan persamaan linier yang saling terkait. Berbagai metode dapat digunakan untuk mencari solusi SPL. Metode-metode tersebut seperti Metode Eliminasi Gauss, Metode Eliminasi Gauss-Jordan, dan berbagai metode lainnya yang juga memanfaatkan matriks dalam implementasinya.

Dalam Tugas Besar 1 mata kuliah Aljabar Linier dan Geometri ini, penulis membuat sebuah pustaka dalam bahasa Java untuk menyelesaikan perihal seputar matriks, seperti Sistem Persamaan Linear (SPL), menghitung determinan, menentukan matriks balikan, menentukan interpolasi polinomial, menghitung regresi linear berganda, menghitung regresi kuadratik berganda, dan menentukan *bicubic spline interpolation*. Pustaka tersebut akan kami gunakan dalam sebuah kalkulator matriks. Kalkulator matriks tersebut dapat menyelesaikan berbagai persoalan matriks dalam bentuk SPL, interpolasi, dan regresi.

Spesifikasi program adalah sebagai berikut:

1. Program dapat menerima masukan (*input*) baik dari *keyboard* maupun membaca masukan dari *file text*. Untuk SPL, masukan dari *keyboard* adalah m, n, koefisien a_{ij} , dan b_i. Masukan dari *file* berbentuk matriks *augmented* tanpa tanda kurung, setiap elemen matriks dipisah oleh spasi. Misalnya,

   3 4.5 2.8 10 12

   -3 7 8.3 11 -4

   0.5 -10 -9 12 0

2. Untuk persoalan menghitung determinan dan matriks balikan, masukan dari *keyboard* adalah n dan koefisien a_{ij}. Masukan dari *file* berbentuk matriks, setiap elemen matriks dipisah oleh spasi. Misalnya,

   3 4.5 2.8

   -3 7 8.3

   0.5 -10 -9

   Luaran (*output*) disesuaikan dengan persoalan (determinan atau invers) dan penghitungan balikan/invers dilakukan dengan metode matriks balikan dan adjoin.

3. Untuk persoalan invers, metode yang digunakan ada 2 yaitu menggunakan OBE dan Matriks Adjoin
4. Untuk persoalan interpolasi, masukannya jika dari *keyboard* adalah n, (x_0, y_0), (x_1, y_1), …, (x_n, y_n), dan nilai x yang akan ditaksir nilai fungsinya. Jika masukannya dari *file*, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung. Masukan kemudian dilanjutkan dengan satu buah baris berisi satu buah nilai x yang akan ditaksir menggunakan fungsi interpolasi yang telah didefinisikan. Misalnya jika titik-titik datanya adalah (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513) dan akan mencari nilai y saat x = 8.3, maka di dalam *file text* ditulis sebagai berikut:

   8.0 2.0794

   9.0 2.1972

9.5 2.2513

8.3

5. Untuk persoalan regresi, masukannya jika dari *keyboard* adalah n (jumlah peubah x), m (jumlah sampel), semua nilai-nilai x_{1i}, x_{2i}, …, x_{ni}, nilai y_i, dan nilai-nilai x_k yang akan ditaksir nilai fungsinya. Jika masukannya dari *file*, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.
6. Untuk persoalan SPL, luaran program adalah solusi SPL. Jika solusinya tunggal, tuliskan nilainya. Jika solusinya tidak ada, tuliskan solusi tidak ada, jika solusinya banyak, maka tuliskan solusinya dalam bentuk parametrik (misalnya $x_4 = -2$, $x_3 = 2s - t$, $x_2 = s$, dan $x_1 = t$).
7. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan. Contoh luaran untuk interpolasi adalah

   $$f(x) = -0.0064x^2 + 0.2266x + 0.6762, \quad f(5) = \ldots$$

   dan untuk regresi adalah

   $$f(x) = -9.5872 + 1.0732x_1, \quad f(x_k) = \ldots$$

   untuk kasus regresi kuadratik, variabel boleh menggunakan x_1, x_2, dan lain-lain tetapi perlu dijelaskan variabel tersebut merepresentasikan apa. Contoh

   x_1 = X

   .

   x_3 = X²

   .

   x_5 = XY

   [Persamaan dan Solusi]

8. Untuk persoalan *bicubic spline interpolation*, masukan dari *file text* (.txt) yang berisi matriks berukuran 4 x 4 yang berisi konfigurasi nilai fungsi dan turunan berarah disekitarnya, diikuti dengan nilai a dan b untuk mencari nilai $f(a, b)$.

   Misalnya jika nilai dari $f(0, 0)$, $f(1, 0)$, $f(0, 1)$, $f(1, 1)$, $f_x(0, 0)$, $f_x(1, 0)$, $f_x(0, 1)$, $f_x(1, 1)$, $f_y(0, 0)$, $f_y(1, 0)$, $f_y(0, 1)$, $f_y(1, 1)$, $f_{xy}(0, 0)$, $f_{xy}(1, 0)$, $f_{xy}(0, 1)$, $f_{xy}(1, 1)$ berturut-turut adalah 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 serta nilai a dan b yang dicari berturut-turut adalah 0.5 dan 0.5 maka isi *file text* ditulis sebagai berikut:

   1 2 3 4

   5 6 7 8

   9 10 11 12

   13 14 15 16

   0.5 0.5

   Luaran yang dihasilkan adalah nilai dari $f(0.5, 0.5)$.

9. Luaran program harus dapat ditampilkan pada layar komputer dan dapat disimpan ke dalam *file*.
10. Bahasa program yang digunakan adalah Java. Anda bebas untuk menggunakan versi java apapun dengan catatan di atas java versi 8 (8/9/11/15/17/19/20/21).

11. Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan dirancang masing-masing. Misalnya, menu:

    MENU

    1. Sistem Persamaaan Linier
    2. Determinan
    3. Matriks balikan
    4. Interpolasi Polinom
    5. Interpolasi Bicubic Spline
    6. Regresi linier dan kuadratik berganda
    7. Interpolasi Gambar (Bonus)
    8. Keluar

    Untuk pilihan menu nomor 1 ada sub-menu lagi yaitu pilihan metode:

    1. Metode eliminasi Gauss
    2. Metode eliminasi Gauss-Jordan
    3. Metode matriks balikan
    4. Kaidah Cramer

BAB 2 TEORI SINGKAT

2.1. Sistem Persamaan Linier

Persamaan linier adalah persamaan yang memiliki variabel berpangkat satu. Sistem persamaan linier adalah dua atau lebih persamaan linier yang saling terkait.

2.1.1. Metode Eliminasi Gauss

Metode Eliminasi Gauss adalah salah satu prosedur sistematis untuk menyelesaikan Sistem Persamaan Linier. Metode ini mengubah bentuk matriks *augmented* menjadi matriks eselon. Matriks eselon (*row echelon form*) adalah matriks yang mempunyai *leading one* atau satu utama pada setiap barisnya, kecuali baris yang seluruhnya nol. Untuk mendapatkan bentuk matriks eselon, Operasi Baris Elementer (OBE) yang meliputi pertukaran baris, mengalikan baris dengan koefisien, dan menjumlahkan atau mengurangkan satu baris dengan baris lainnya, diperlukan.

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

Gambar 2.1.1. Mengubah Matriks *Augmented* Menjadi Matriks Eselon dengan OBE

Setelah melakukan OBE, matriks yang didapat adalah matriks *augmented* akhir. Matriks *augmented* kemudian diselesaikan menggunakan teknik penyulihan mundur untuk menyelesaikan Sistem Persamaan Linier.

2.1.2. Metode Eliminasi Gauss-Jordan

Metode Eliminasi Gauss-Jordan mirip dengan Metode Eliminasi Gauss. Namun, metode ini mengubah bentuk matriks *augmented* menjadi matriks eselon tereduksi. Matriks eselon tereduksi adalah matriks yang mempunyai *leading one* pada setiap barisnya, dan di atas serta di bawah *leading one* semua nilainya bernilai nol. Untuk mendapatkan bentuk matriks eselon tereduksi, Operasi Baris Elementer (OBE) kembali digunakan.

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

Gambar 2.1.2. Mengubah Matriks *Augmented* Menjadi Matriks Eselon dengan OBE

Setelah melakukan OBE, matriks yang didapat adalah matriks augmented akhir. Matriks ini kemudian diselesaikan menggunakan teknik penyulihan mundur untuk menyelesaikan Sistem Persamaan Linier.

2.1.3. Metode Matriks Balikan

Matriks Balikan dapat dihitung menggunakan Metode Eliminasi Gauss-Jordan. Misalkan sebuah matriks persegi dengan determinan $\neq 0$, diberi nama matriks A. Balikan atau *inverse* matriks A adalah A^{-1} sedemikian sehingga $AA^{-1} = I$ (dengan I adalah matriks identitas yang seukuran dengan A). Untuk mendapat matriks balikan, dibuat matriks augmented dari A dan I untuk mencapai matriks augmented I dan A^{-1} dengan menggunakan Metode Eliminasi Gauss-Jordan.

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

Gambar 2.1.3. Mengubah Matriks Augmented A dan I Menjadi Matriks Augmented I dan A^{-1}

Solusi Sistem Persamaan Linier didapat dengan mengalikan matriks A^{-1} dengan matriks b, dimana b adalah matriks konstan dari Sistem Persamaan Linier, atau $x = A^{-1} \cdot b$.

2.1.4. Kaidah Cramer

Kaidah Cramer berbunyi jika sebuah Sistem Persamaan Linier $Ax = b$, dengan A adalah matriks persegi dan determinannya tidak 0, maka sistem ini mempunyai solusi yang unik sebagai berikut.

$$x_1 = \frac{\det(A_1)}{\det(A)}, \qquad x_2 = \frac{\det(A_2)}{\det(A)}, \ldots, \qquad x_n = \frac{\det(A_n)}{\det(A)}$$

Gambar 2.1.4. Solusi Kaidah Cramer

Dalam hal ini, A_1, A_2, hingga A_n (A_j) adalah matriks yang diperoleh dengan mengganti kolom ke-j dari A menjadi matriks b, dimana b adalah matriks konstan.

2.2. Determinan

Determinan adalah sebuah nilai skalar (bilangan real) dari suatu matriks persegi.

2.2.1. Metode Reduksi Baris

Determinan sebuah matriks persegi dapat dihitung dengan mereduksi matriks menggunakan Operasi Baris Elementer (OBE) hingga tercapai matriks segitiga bawah atau atas. Metode ini yang dinamakan Metode Reduksi Baris.

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

Gambar 2.2.1.1. Mengubah Matriks Persegi Menjadi Matriks Segitiga Bawah

Kemudian, determinan dapat dihitung dengan formula sebagai berikut,

$$\det(A) = \frac{(-1)^p \, a'_{11} \; a'_{22} \ldots a'_{nn}}{k_1 k_2 \ldots k_m}$$

Gambar 2.2.1.2. Formula Metode Reduksi Baris

Dengan p adalah banyaknya operasi pertukaran baris di dalam OBE dan k adalah perkalian baris-baris matriks dengan konstanta-konstanta tertentu.

2.2.2. Metode Ekspansi Kofaktor

Determinan dari suatu matriks persegi dapat dicari dengan mencari determinan dari minor-minornya, atau yang disebut dengan kofaktor. Matriks yang merupakan determinan dari minor-minornya disebut dengan matriks kofaktor.

Misalkan sebuah matriks A yang merupakan matriks persegi sebagai berikut,

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

Gambar 2.2.2.1. Matriks Persegi A

M_{ij} atau minor entri a_{ij} adalah determinan submatriks yang elemen-elemennya tidak berada pada baris i dan kolom j. Sehingga kofaktor dari entri a_{ij} adalah,

$$C_{ij} = (-1)^{i+j} M_{ij}$$

Gambar 2.2.2.2. Formula Kofaktor Entri a_{ij}

Kofaktor C_{ij} berkoresponden dengan minor entri M_{ij} dan hanya berbeda tanda positif atau negatif. Berikut merupakan tanda positif dan negatif untuk kofaktor C_{ij} untuk matriks persegi berukuran $n \times n$,

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

Gambar 2.2.2.3. Tanda Positif dan Negatif Kofaktor C_{ij}

Maka, determinan dapat dihitung dengan menjumlahkan hasil perkalian dari a_{ij} dan C_{ij} pada satu baris atau kolom.

2.3. Balikan (*Inverse*)

Jika A adalah sebuah matriks persegi yang berukuran sama dengan matriks B, sedemikian rupa sehingga $A \cdot B = B \cdot A = I$, maka B dapat dikatakan *inverse* dari A (I adalah matriks identitas yang seukuran dengan A).

2.3.1. Metode Operasi Baris Elementer

Balikan matriks dapat dihitung menggunakan Operasi Baris Elementer (OBE) dan Metode Eliminasi Gauss-Jordan. Untuk mendapat matriks balikan, dibuat matriks augmented dari A dan I untuk mencapai matriks augmented I dan A^{-1} dengan menggunakan OBE dan Metode Eliminasi Gauss-Jordan.

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

Gambar 2.3.1. Mengubah Matriks Augmented A dan I Menjadi Matriks Augmented I dan A^{-1}

2.3.2. Metode Matriks Adjoin

Balikan matriks dapat dicari dengan membagi matriks adjoinnya dengan determinannya, atau dengan formula sebagai berikut,

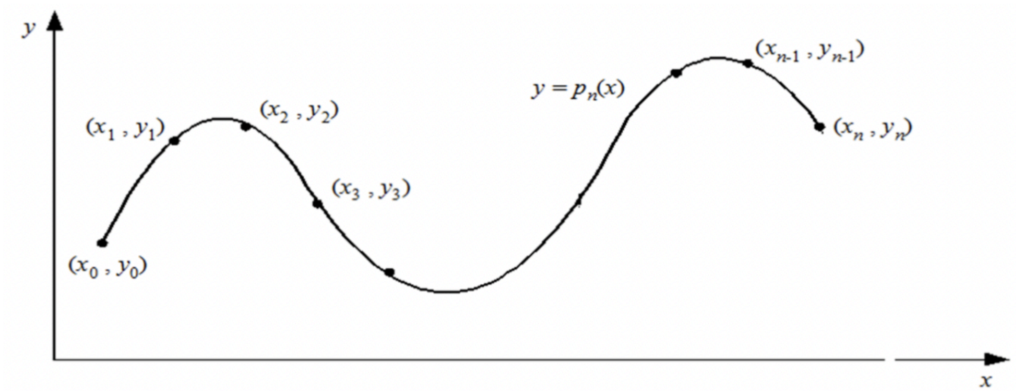
$$A^{-1} = \frac{1}{\det(A)} \mathrm{adj}(A)$$

Gambar 2.3.2. Formula Balikan Matriks dengan Metode Matriks Adjoin

Matriks adjoin adalah hasil transpose dari matriks kofaktornya.

2.4. Interpolasi Polinomial

Interpolasi Polinomial adalah sebuah teknik untuk mencari sebuah kurva (dalam bentuk persamaan polinomial) yang menginterpolasi atau melewati semua titik-titik yang diberikan. Interpolasi dilakukan dengan membentuk persamaan polinomial $p_n(x)$ dari $n + 1$, dimana n adalah buah titik yang berbeda. Sehingga $y_i = p_n(x_i)$ untuk $i = 0, 1, 2, ..., n$.



Gambar 2.4. Kurva Titik-Titik yang Diinterpolasi Secara Polinomial

Setelah persamaan polinom ditemukan, $p_n(x)$ dapat digunakan untuk menghitung perkiraan nilai y di sembarang titik dalam selang $[x_0, x_n]$. Hal ini dapat dilakukan dengan memanfaatkan matriks dan Metode Eliminasi Gauss.

2.5. Regresi Berganda

2.5.1. Regresi Linier Berganda

Regresi Linier Berganda adalah sebuah metode statistik yang digunakan untuk memprediksi hubungan linier antara satu variabel dengan variabel-variabel lainnya. Persamaan umum dari regresi linier yang bisa digunakan untuk regresi linier berganda sebagai berikut,

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki} + \epsilon_i$$

Gambar 2.5.1.1. Persamaan Umum Regresi Linier

Kemudian, untuk mendapat nilai dari setiap β_i dapat digunakan *Normal Estimation Equation for Multiple Linear Regression* sebagai berikut,

$$
\begin{array}{lllll}
nb_0 + b_1 \sum_{i=1}^{n} x_{1i} & + b_2 \sum_{i=1}^{n} x_{2i} & + \ \cdots & + b_k \sum_{i=1}^{n} x_{ki} & = \sum_{i=1}^{n} y_i \\
b_0 \sum_{i=1}^{n} x_{1i} + b_1 \sum_{i=1}^{n} x_{1i}^2 & + b_2 \sum_{i=1}^{n} x_{1i} x_{2i} & + \ \cdots & + b_k \sum_{i=1}^{n} x_{1i} x_{ki} & = \sum_{i=1}^{n} x_{1i} y_i \\
\vdots \qquad \vdots & \vdots & & \vdots & \vdots \\
b_0 \sum_{i=1}^{n} x_{ki} + b_1 \sum_{i=1}^{n} x_{ki} x_{1i} & + b_2 \sum_{i=1}^{n} x_{ki} x_{2i} & + \ \cdots & + b_k \sum_{i=1}^{n} x_{ki}^2 & = \sum_{i=1}^{n} x_{ki} y_i
\end{array}
$$

Gambar 2.5.1.2. *Normal Estimation Equation for Multiple Linear Regression*

Model regresi dijadikan Sistem Persamaan Linier yang dapat diselesaikan dengan menggunakan Metode Eliminasi Gauss.

2.5.2. Regresi Kuadratik Berganda

Terdapat tiga bentuk persamaan dari Regresi Kuadratik Berganda,

a. Variabel linier, yang adalah variabel dengan derajat satu,
b. Variabel kuadrat, yang merupakan variabel dengan derajat dua, dan
c. Variabel interaksi, yang merupakan dua variabel dengan derajat satu yang dikalikan satu sama lain.

Setiap n-peubah, jumlah variabel linier, kuadrat, dan interaksi akan berbeda-beda, dengan contoh sebagai berikut,

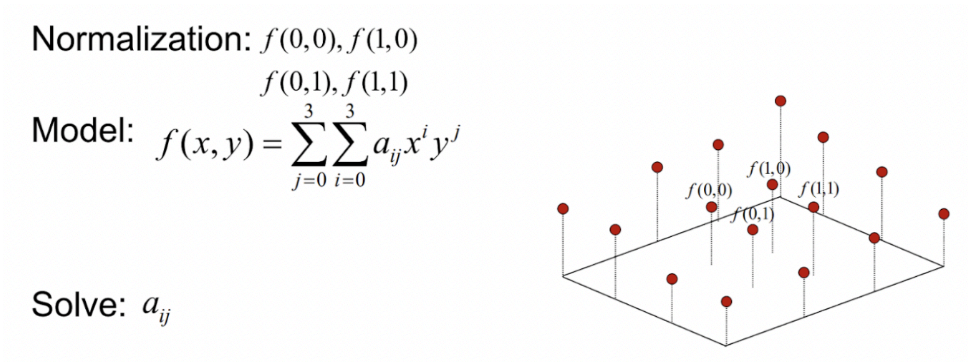
$$
\begin{pmatrix}
N & \sum u_i & \sum v_i & \sum u_i^2 & \sum u_i v_i & \sum v_i^2 \\
\sum u_i & \sum u_i^2 & \sum u_i v_i & \sum u_i^3 & \sum u_i^2 v_i & \sum u_i v_i^2 \\
\sum v_i & \sum u_i v_i & \sum v_i^2 & \sum u_i^2 v_i & \sum u_i v_i^2 & \sum v_i^3 \\
\sum u_i^2 & \sum u_i^3 & \sum u_i^2 v_i & \sum u_i^4 & \sum u_i^3 v_i & \sum u_i^2 v_i^2 \\
\sum u_i v_i & \sum u_i^2 v_i & \sum u_i v_i^2 & \sum u_i^3 v_i & \sum u_i^2 v_i^2 & \sum u_i v_i^3 \\
\sum v_i^2 & \sum u_i v_i^2 & \sum v_i^3 & \sum u_i^2 v_i^2 & \sum u_i v_i^3 & \sum v_i^4
\end{pmatrix}
\begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix}
=
\begin{pmatrix} \sum y_i \\ \sum y_i u_i \\ \sum y_i v_i \\ \sum y_i u_i^2 \\ \sum y_i u_i v_i \\ \sum y_i v_i^2 \end{pmatrix}
$$

Gambar 2.5.2.1. Contoh Regresi Kuadratik Dua Variabel Peubah

N merupakan suatu konstan. Model regresi ini kemudian dapat dijadikan Sistem Persamaan Linier yang bisa diselesaikan menggunakan Metode Eliminasi Gauss.

2.6. *Bicubic Spline Interpolation*

Mirip dengan Interpolasi Polinomial, *Bicubic Spline Interpolation* merupakan sebuah teknik untuk mencari sebuah kurva yang menginterpolasi titik-titik yang diberikan di ruang dua dimensi. Dalam pemrosesan interpolasi ini, digunakan 16 buah titik, 4 titik referensi utama di bagian pusat, dan 12 titik di sekitarnya sebagai aproksimasi turunan dari keempat titik referensi untuk membangun permukaan bikubik. Bentuk pemodelannya sebagai berikut,



Gambar 2.6.1. Pemodelan *Bicubic Spline Interpolation*

Persamaan polinomial yang digunakan adalah sebagai berikut,

$$f(x,y)=\sum_{i=0}^{3}\sum_{j=0}^{3}a_{ij}x^i y^j$$

$$f_x(x,y)=\sum_{j=0}^{3}\sum_{i=1}^{3}a_{ij}ix^{i-1}y^j$$

$$f_y(x,y)=\sum_{j=1}^{3}\sum_{i=0}^{3}a_{ij}jx^i y^{j-1}$$

$$f_{xy}(x,y)=\sum_{j=0}^{3}\sum_{i=0}^{3}a_{ij}ijx^{i-1}y^{j-1}$$

Gambar 2.6.2. Persamaan Polinomial *Bicubic Spline Interpolation*

BAB 3 IMPLEMENTASI PUSTAKA

3.1 Class Matrix

Class ini berisi tipe data matriks yang digunakan dalam berbagai fitur di program ini

3.1.1 Atribut

row	Integer yang menyimpan jumlah baris dalam matriks
col	Integer yang menyimpan jumlah kolom dalam matriks
Matrix	Array bertipe double yang digunakan untuk menyimpan elemen matriks

3.1.2 Metode

`public class Matrix {`	Untuk men-*define* atribut matriks
`public Matrix(int row, int col){`	Konstruktor untuk membuat sebuah matriks dengan jumlah baris (row) dan kolom (col) tertentu
`public void set(int row, int col, double val){`	Konstruktor untuk menyimpan matriks berukuran row x col menggunakan metode set untuk mengubah elemen pada row dan col yang ditentukan
`public double elmt(int row, int col){`	Mengembalikan elemen matriks yang merupakan array 2D dan menerima dua parameter, yaitu row dan col
`public void readMatrix(Scanner in){`	Mengisi matriks dari input user dengan membaca nilai double untuk setiap elemen matriks sesuai dengan ukuran baris dan kolom yang ada
`public void printMatrix(){`	Mencetak elemen-elemen dari sebuah matriks ke layar dengan format tertentu
`public Matrix copyMatrix(Matrix m){`	Membuat salinan dari objek salinan matriks m dengan ukuran dan elemen yang sama serta mengembalikannya sebagai objek matriks baru
`public void swapRow(int r1, int r2){`	Menukarkan dua baris pada matriks, yaitu r1 dan r2 saat menggunakan metode Operasi Baris Elementer (OBE)

`public void swapCol(int c1, int c2){`	Menukarkan dua kolom pada matriks, yaitu c1 dan c2 saat menggunakan metode Operasi Baris Elementer (OBE)
`public void multRowByK(int r, double k){`	Mengalikan salah satu baris pada matriks dengan konstanta k bertipe double saat menggunakan metode Operasi Baris Elementer (OBE)
`public void addRow(int r1, int r2, double k){`	Menjumlahkan salah satu baris dengan satu baris lainnya yang sudah dikalikan dengan konstanta bertipe double saat menggunakan metode Operasi Baris Elementer (OBE)
`public static Matrix multMatrix(Matrix m1, Matrix m2){`	Mengalikan dua buah matriks m1 dan m2 dan mengembailkan hasilnya dalam bentuk objek matriks baru
`public static Matrix transposeMatrix(Matrix m){`	Melakukan transposisi pada sebuah matriks m , yaitu merubah baris menjadi kolom dan kolom menjadi baris
`public boolean isZeroRow(int r){`	Mengecek apakah semua elemen dalam sebuah matriks bernilai nol
`public boolean isSquare() {`	Mengecek apakah sebuah matriks memiliki jumlah baris dan kolom yang sama

3.2 Class ReadWrite

Class ini berfungsi untuk membaca dan menulis data dari input pengguna atau dari file

3.2.1 Atribut

Tidak ada

3.2.2 Metode

`public class ReadWrite {`	Membaca dan menulis data dari dan ke file teks serta memproses input dari pengguna
`public static int fileOrKeys(Scanner in) {`	Meminta pengguna memilih cara memasukkan input untuk matriks, apakah melalui keyboard atau file
`public static Matrix txtRead(Scanner in) {`	Membaca matriks dari file teks dan mengembalikannya sebagai objek Matrix
`public static void txtWrite(Scanner in, String str){`	Menulis sebuah string ke dalam file teks atau menambah teks dalam file jika sudah ada atau

	membuat file baru jika belum ada
`public static void doubleToFile(double value, Scanner in) {`	Menerima dua parameter, yaitu sebuah nilai double bernama value dan objek Scanner bernama in yang digunakan untuk membaca input dari pengguna
`public static void arrToFile(double[] solusi, Scanner in) {`	Menerima array bertipe double kemudian memberikan pilihan kepada pengguna melalui input apakah mereka ingin menulis hasil array tersebut ke dalam file .txt
`public static void matrixToFile(Matrix solusi, Scanner in) {`	Meminta pengguna apakah mereka ingin menyimpan hasil matriks ke dalam sebuah file .txt atau tidak

3.3 Class Gauss

Class ini berfungsi untuk menyelesaikan SPL menggunakan metode eliminasi Gauss

3.3.1 Atribut

Tidak ada

3.3.2 Metode

`public class Gauss {`	Metode penyelesaian SPL menggunakan eliminasi Gauss
`public static double[] gauss(Matrix m, Scanner in){`	Menerima input berupa matriks m dan mencari solusi dari SPL dalam matriks augmented
`public static Matrix mgauss(Matrix m, Scanner in){`	Mengubah matriks menjadi matriks eselon baris
`public static double[] msolution(Matrix m){`	Menyelesaikan SPL dalam bentuk matriks augmented menggunakan metode substitusi balik
`public static void call() {`	Mengatur alur kerja untuk memasukkan atau membaca matriks, melakukan eliminasi Gauss pada matriks, dan kemudian menuliskan hasilnya ke file .txt

3.4 Class GaussJordan

Class ini berfungsi untuk menyelesaikan SPL menggunakan metode eliminasi Gauss-Jordan

3.4.1 Atribut

Tidak ada

3.4.2 Metode

`public class GaussJordan {`	Metode penyelesaian SPL menggunakan eliminasi Gauss Jordan
`public static double[] gaussJordan(Matrix m, Scanner in){`	Menerima input berupa matriks m dan mencari solusi dari SPL pada matriks augmented
`public static Matrix mgaussJordan(Matrix m, Scanner in){`	Mengubah matriks augmented menjadi matriks baris eselon tereduksi
`public static void call() {`	Mengatur alur kerja untuk memasukkan atau membaca matriks, melakukan eliminasi Gauss Jordan pada matriks, dan kemudian menuliskan hasilnya ke file .txt

3.5 Class Cramer

Class ini berfungsi untuk menyelesaikan SPL menggunakan kaidah Cramer

3.5.1 Atribut

Tidak ada

3.5.2 Metode

`public class Cramer {`	Menyelesaikan SPL menggunakan kaidah Cramer
`public static double[] cramer(Matrix m, Scanner in){`	Menyelesaikan SPL menggunakan kaidah Cramer dan mencetak solusi setiap variabel
`public static void call() {`	Mengatur alur program untuk memasukkan matriks secara manual atau melalui file, menyelesaikan SPL menggunakan kaidah Cramer, dan hasilnya disimpan dalam file .txt

3.6 Class Determinan

Class ini berfungsi untuk menentukan nilai determinan suatu matriks

3.6.1 Atribut

Tidak ada

3.6.2 Metode

`public class Determinant {`	Mencari nilai determinan dari suatu matriks
`public static double det(Matrix m, Scanner in){`	Menghitung determinan matriks persegi dengan memanfaatkan transformasi ke bentuk eselon baris dan mengalikan elemen diagonal utama
`public static void call() {`	Mengatur alur program untuk memasukkan matriks secara manual atau melalui file, memvalidasi serta menghitung determinan, dan hasilnya disimpan dalam file .txt

3.7 Class DeterminanKofaktor

Class ini berfungsi untuk menentukan nilai determinan suatu matriks melalui ekspansi kofaktor

3.7.1 Atribut

Tidak ada

3.7.2 Metode

`public class DeterminanKofaktor {`	Mencari nilai determinan melalui ekspansi kofaktor dari suatu matriks
`public static double determinan(Matrix matrix, int n) {`	Menghitung determinan matriks dengan cara rekursif menggunakan ekspansi kofaktor
`public static void getSubmatrix(Matrix matrix, Matrix submatrix, int row, int col, int n) {`	Mengambil submatriks dengan mengecualikan baris dan kolom tertentu dari matriks asal
`public static void call() {`	Mengatur alur program untuk memasukkan matriks secara manual atau melalui file, memvalidasi serta menghitung determinan jika matriks persegi, dan hasilnya disimpan dalam file .txt

3.8 Class InverseSPL

Class ini berfungsi untuk menyelesaikan SPL dengan metode balikan dari suatu matriks

3.8.1 Atribut

Tidak ada

3.8.2 Metode

`public class InverseSPL {`	Menyelesaikan SPL dengan balikan dari suatu matriks
`public static Matrix inverseSPL(Matrix m, Scanner in) {`	Menyelesaikan SPL dengan metode invers dengan syarat matriks persegi dan determinan tidak nol
`public static void call() {`	Mengatur alur program untuk memasukkan matriks secara manual atau melalui file, menyelesaikan SPL dengan metode balikan matriks, dan hasilnya disimpan dalam file .txt

3.9 Class InverseOBE

Class ini berfungsi untuk menentukan balikan matriks menggunakan metode OBE

3.9.1 Atribut

Tidak ada

3.9.2 Metode

`public class InverseOBE {`	Menentukan balikan matriks dengan menerapkan OBE pada matriks augmented
`public static Matrix inverseOBE(Matrix m, Scanner in){`	Menentukan balikan sebuah matriks menggunakan metode eliminasi Gauss-Jordan dengan OBE
`public static void call() {`	Mengatur alur program untuk memasukkan matriks secara manual atau melalui file, menentukan balikan matriks melalui metode eliminasi Gauss-Jordan dengan OBE, dan hasilnya disimpan dalam file .txt

3.10 Class InverseAdj

Class ini berfungsi untuk menentukan balikan matriks menggunakan metode adjoin pada matriks

3.10.1 Atribut

Tidak ada

3.10.2 Metode

`public class InverseAdj {`	Menentukan balikan matriks menggunakan metode adjoin pada matriks
`public static double determinant(Matrix matrix, int n) {`	Menghitung determinan matriks dengan cara rekursif menggunakan ekspansi kofaktor
`public static void getSubmatrix(Matrix matrix, Matrix submatrix, int row, int col, int n) {`	Mengambil submatriks dengan mengecualikan baris dan kolom tertentu dari matriks asal
`public static Matrix cofactor(Matrix matrix, int n) {`	Menentukan matriks kofaktor dari sebuah matriks persegi
`public static Matrix transpose(Matrix matrix, int n) {`	Melakukan transposisi pada matriks, yaitu baris menjadi kolom dan kolom menjadi baris
`public static Matrix adjoint(Matrix matrix, int n) {`	Menentukan adjoin dari sebuah matriks dengan cara menentukan matriks kofaktor dan men-*transpose* matriks kofaktor
`public static Matrix inversAdj(Matrix m, Scanner in){`	Menentukan balikan matriks menggunakan metode adjoin
`public static void call() {`	Mengatur alur program untuk memasukkan matriks secara manual atau melalui file, menentukan balikan matriks melalui metode adjoin, dan hasilnya disimpan dalam file .txt

3.11 Class Interpolation

Class ini berfungsi untuk memperkirakan nilai dengan menggunakan polinomial untuk mencocokkan beberapa titik data yang diketahui

3.11.1 Atribut

Tidak ada

3.11.2 Metode

`public class Interpolation {`	Menentukan nilai dari hasil interpolasi polinomial

`public static double findX0(Matrix m){`	Menemukan dan mengembalikan nilai terkecil dari elemen-elemen kolom 1 pada matriks
`public static double findX0c1(Matrix m){`	Mencari elemen terkecil di kolom pertama pada matriks
`public static double findXn(Matrix m){`	Menemukan dan mengembalikan nilai terbesar dari elemen-elemen di kolom 1 suatu matriks
`public static double findXnc1(Matrix m){`	Mencari elemen terbesar dari kolom 1 pada matriks
`public static boolean arePointsUnique(Matrix m){`	Memeriksa tidak ada dua titik yang sama dalam data yang diberikan
`public static double[] interpolMatrix(Matrix m){`	Implementasi dari metode eliminasi Gauss untuk menyelesaikan SPL
`public static double[] polynomialInterpolation(double[] x, double[] y) {`	Menemukan polinomial yang melewati serangkaian titik data yang diberikan
`public static void interpolF(Scanner in){`	Metode yang melakukan interpolasi polinomial berdasarkan input dari pengguna, baik manual maupun dari file, memvalidasi input, dan menyimpan hasil interpolasi ke dalam file .txt

3.12 Class RegLinier

Class ini berfungsi untuk menentukan hasil dari regresi linear berganda

3.12.1 Atribut

Tidak ada

3.12.2 Metode

`public class RegLinier {`	Menentukan hasil nilai dari regresi linear berganda
`public static double[] regLin(Matrix m, Scanner in) {`	Implementasi metode regresi linear berganda melalui persamaan normal dan diselesaikan menggunakan matriks augmented dan metode eliminasi Gauss untuk mendapatkan koefisien regresi
`public static void f(double[] solusi, Scanner in, double[] taksiran) {`	Menghitung hasil regresi linear berganda dan hasil taksiran dan menyimpan hasilnya ke file .txt

`public static void call() {`	Mengatur alur program untuk memasukkan matriks secara manual atau melalui file, menentukan hasil regresi linier dan taksirannya, serta hasilnya disimpan dalam file .txt

3.13 Class RegQuadratic

Class ini berfungsi untuk menentukan hasil dari regresi linear berganda

3.13.1 Atribut

Tidak ada

3.13.2 Metode

`public class RegQuadratic{`	Menentukan hasil nilai dari regresi kuadratik berganda
`public static double[] regQuad(Matrix m, Scanner in) {`	Melakukan regresi kuadratik berganda melalui persamaan normal, memperluas matriks variabel independen untuk memasukkan elemen linear, kuadratik, dan variabel interaksi, kemudian diselesaikan dengan membentuk matriks augmented dan menggunakan metode eliminasi Gauss
`public static void f(double[] solusi, Scanner in, double[] taksiran) {`	Menghitung hasil regresi kuadratik berganda dan hasil taksiran dan menyimpan hasilnya ke file .txt
`public static void call() {`	Mengatur alur program untuk memasukkan matriks secara manual atau melalui file, menentukan hasil regresi kuadratik dan taksirannya, serta hasilnya disimpan dalam file .txt

3.14 Class BicSplineInterpolation

Class ini berfungsi untuk menentukan nilai dari *Bicubic Spline Interpolation*

3.14.1 Atribut

Tidak ada

3.14.2 Metode

`public class BicSplineInterpolation {`	Menentukan nilai dari *Bicubic Spline Interpolation*, yaitu metode interpolasi yang digunakan untuk mengaproksimasi fungsi di antara titik-titik data yang diketahui.
`public static Matrix xMatrix(){`	Menghasilkan sebuah matriks berukuran 16x16 yang berisi nilai-nilai interpolasi *bicubic spline*
`public static Matrix readF(Scanner sc, Matrix m){`	Membaca elemen-elemen matriks m dan menyimpannya ke matriks baru
`public static double countResult(Matrix m,double a,double b){`	Menghitung nilai interpolasi *bicubic spline* dari suatu matriks m pada titik (a,b)
`public static void BicInterpol(){`	Melakukan interpolasi bicubic berdasarkan input yang dimasukkan pengguna, baik langsung maupun dari file .txt, kemudian menyimpan hasilnya ke dalam file .txt

BAB 4 EKSPERIMEN

4.1. Sistem Persamaan Linier

<table>
<tr><th>Masukan (Soal)</th><th>Metode</th><th>Keluaran</th></tr>
<tr><td rowspan="4">spl1a.txt
1 1 -1 -1 1
2 5 -7 -5 -2
2 -1 1 3 4
5 2 -4 2 6</td><td>Gauss</td><td>Matriks sebelum Eliminasi Gauss:
1.0 1.0 -1.0 -1.0 1.0
2.0 5.0 -7.0 -5.0 -2.0
2.0 -1.0 1.0 3.0 4.0
5.0 2.0 -4.0 2.0 6.0

Matriks setelah Eliminasi Gauss:
1.0 1.0 -1.0 -1.0 1.0
0.0 1.0 -1.6666666666666665 -1.0 -1.3333333333333333
0.0 0.0 1.0 -1.0 1.0
0.0 0.0 0.0 0.0 1.0</td></tr>
<tr><td>Gauss-Jordan</td><td>Matriks sebelum Eliminasi Gauss-Jordan:
1.0 1.0 -1.0 -1.0 1.0
2.0 5.0 -7.0 -5.0 -2.0
2.0 -1.0 1.0 3.0 4.0
5.0 2.0 -4.0 2.0 6.0

Matriks setelah Eliminasi Gauss-Jordan:
1.0 0.0 0.0 0.6666666666666665 1.0
0.0 1.0 0.0 -2.6666666666666665 3.0
0.0 0.0 1.0 -1.0 2.0
0.0 0.0 0.0 0.0 1.0

Tidak dapat mencari solusi SPL.</td></tr>
<tr><td>Cramer</td><td>Matriks sebelum operasi Cramer:
1.0 1.0 -1.0 -1.0 1.0
2.0 5.0 -7.0 -5.0 -2.0
2.0 -1.0 1.0 3.0 4.0
5.0 2.0 -4.0 2.0 6.0

Tidak dapat mencari solusi kaidah cramer karena determinan matriks = 0.</td></tr>
<tr><td>Balikan (Invers)</td><td>Matriks sebelum operasi Inverse untuk mencari SPL:
1.0 1.0 -1.0 -1.0 1.0
2.0 5.0 -7.0 -5.0 -2.0
2.0 -1.0 1.0 3.0 4.0
5.0 2.0 -4.0 2.0 6.0

Tidak dapat mencari solusi SPL karena matriks tidak dapat di-inverse.</td></tr>
</table>

spl1b.txt 1 -1 0 0 1 3 1 1 0 -3 0 6 2 -1 0 1 -1 5 -1 2 0 -2 -1 -1	Gauss	Matriks sebelum Eliminasi Gauss: 1.0 -1.0 0.0 0.0 1.0 3.0 1.0 1.0 0.0 -3.0 0.0 6.0 2.0 -1.0 0.0 1.0 -1.0 5.0 -1.0 2.0 0.0 -2.0 -1.0 -1.0 Matriks setelah Eliminasi Gauss: 1.0 -1.0 0.0 0.0 1.0 3.0 0.0 1.0 0.0 -1.5 -0.5 1.5 0.0 0.0 0.0 2.5 -2.5 -2.5 0.0 0.0 0.0 1.0 -1.0 -1.0 Hasil perhitungan menggunakan metode Gauss : x1 : NaN x2 : NaN x3 : NaN x4 : -1.0 x5 : 0.0
	Gauss-Jordan	Matriks sebelum Eliminasi Gauss-Jordan: 1.0 -1.0 0.0 0.0 1.0 3.0 1.0 1.0 0.0 -3.0 0.0 6.0 2.0 -1.0 0.0 1.0 -1.0 5.0 -1.0 2.0 0.0 -2.0 -1.0 -1.0 Matriks setelah Eliminasi Gauss-Jordan: 1.0 0.0 0.0 0.0 -1.0 3.0 0.0 1.0 0.0 0.0 -2.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 -1.0 Hasil perhitungan menggunakan metode Gauss-Jordan: x1 : NaN x2 : NaN x3 : NaN x4 : -1.0 x5 : 0.0
	Cramer	Matriks sebelum operasi Cramer: 1.0 -1.0 0.0 0.0 1.0 3.0 1.0 1.0 0.0 -3.0 0.0 6.0 2.0 -1.0 0.0 1.0 -1.0 5.0 -1.0 2.0 0.0 -2.0 -1.0 -1.0 Tidak dapat mencari solusi kaidah cramer karena determinan matriks = 0.
	Balikan (Invers)	Matriks sebelum operasi Inverse untuk mencari SPL: 1.0 -1.0 0.0 0.0 1.0 3.0 1.0 1.0 0.0 -3.0 0.0 6.0 2.0 -1.0 0.0 1.0 -1.0 5.0 -1.0 2.0 0.0 -2.0 -1.0 -1.0 Tidak dapat mencari solusi SPL karena matriks bukan matriks persegi.

<table>
<tr>
<td rowspan="4">spl1c.txt
0 1 0 0 1 0 2
0 0 0 1 1 0 -1
0 1 0 0 0 1 1</td>
<td>Gauss</td>
<td>Matriks sebelum Eliminasi Gauss:
0.0 1.0 0.0 0.0 1.0 0.0 2.0
0.0 0.0 0.0 1.0 1.0 0.0 -1.0
0.0 1.0 0.0 0.0 0.0 1.0 1.0

Matriks setelah Eliminasi Gauss:
0.0 1.0 0.0 0.0 1.0 0.0 2.0
0.0 1.0 0.0 0.0 0.0 1.0 1.0
0.0 0.0 0.0 1.0 1.0 0.0 -1.0

Hasil perhitungan menggunakan metode Gauss :
x1 : NaN
x2 : NaN
x3 : -Infinity
x4 : 0.0
x5 : 0.0
x6 : 0.0</td>
</tr>
<tr>
<td>Gauss-Jordan</td>
<td>Matriks sebelum Eliminasi Gauss-Jordan:
0.0 1.0 0.0 0.0 1.0 0.0 2.0
0.0 0.0 0.0 1.0 1.0 0.0 -1.0
0.0 1.0 0.0 0.0 0.0 1.0 1.0

Matriks setelah Eliminasi Gauss-Jordan:
0.0 0.0 0.0 0.0 1.0 -1.0 1.0
0.0 1.0 0.0 0.0 0.0 1.0 1.0
0.0 0.0 0.0 1.0 1.0 0.0 -1.0

Hasil perhitungan menggunakan metode Gauss-Jordan:
x1 : NaN
x2 : NaN
x3 : -Infinity
x4 : 0.0
x5 : 0.0
x6 : 0.0</td>
</tr>
<tr>
<td>Cramer</td>
<td>Matriks sebelum operasi Cramer:
0.0 1.0 0.0 0.0 1.0 0.0 2.0
0.0 0.0 0.0 1.0 1.0 0.0 -1.0
0.0 1.0 0.0 0.0 0.0 1.0 1.0

Tidak dapat mencari solusi kaidah cramer karena determinan matriks = 0.</td>
</tr>
<tr>
<td>Balikan (Invers)</td>
<td>Matriks sebelum operasi Inverse untuk mencari SPL:
0.0 1.0 0.0 0.0 1.0 0.0 2.0
0.0 0.0 0.0 1.0 1.0 0.0 -1.0
0.0 1.0 0.0 0.0 0.0 1.0 1.0

Tidak dapat mencari solusi SPL karena matriks bukan matriks persegi.</td>
</tr>
<tr>
<td>spl2a.txt
1 -1 2 -1 -1
2 1 -2 -2 -2
-1 2 -4 1 1
3 0 0 -3 -3</td>
<td>Gauss</td>
<td>Matriks sebelum Eliminasi Gauss:
1.0 -1.0 2.0 -1.0 -1.0
2.0 1.0 -2.0 -2.0 -2.0
-1.0 2.0 -4.0 1.0 1.0
3.0 0.0 0.0 -3.0 -3.0

Matriks setelah Eliminasi Gauss:
1.0 -1.0 2.0 -1.0 -1.0
0.0 1.0 -2.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0

Hasil perhitungan menggunakan metode Gauss :
x1 : -1.0 * variabel bebas 2.0 * variabel bebas -1.0 * variabel bebas
x2 : -2.0 * variabel bebas
x3 : variabel bebas
x4 : variabel bebas</td>
</tr>
</table>

	Gauss-Jordan	Matriks sebelum Eliminasi Gauss-Jordan: 1.0 -1.0 2.0 -1.0 -1.0 2.0 1.0 -2.0 -2.0 -2.0 -1.0 2.0 -4.0 1.0 1.0 3.0 0.0 0.0 -3.0 -3.0 Matriks setelah Eliminasi Gauss-Jordan: 1.0 0.0 0.0 -1.0 -1.0 0.0 1.0 -2.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 Hasil perhitungan menggunakan metode Gauss-Jordan: x1 : NaN x2 : NaN x3 : NaN x4 : NaN
	Cramer	Matriks sebelum operasi Cramer: 1.0 -1.0 2.0 -1.0 -1.0 2.0 1.0 -2.0 -2.0 -2.0 -1.0 2.0 -4.0 1.0 1.0 3.0 0.0 0.0 -3.0 -3.0 Tidak dapat mencari solusi kaidah cramer karena determinan matriks = 0.
	Balikan (Invers)	Matriks sebelum operasi Inverse untuk mencari SPL: 1.0 -1.0 2.0 -1.0 -1.0 2.0 1.0 -2.0 -2.0 -2.0 -1.0 2.0 -4.0 1.0 1.0 3.0 0.0 0.0 -3.0 -3.0 Tidak dapat mencari solusi SPL karena matriks tidak dapat di-inverse.
spl3a.txt 8 1 3 2 0 2 9 -1 2 1 1 3 2 -1 2 1 0 6 4 3	Gauss	Matriks sebelum Eliminasi Gauss: 8.0 1.0 3.0 2.0 0.0 2.0 9.0 -1.0 2.0 1.0 1.0 3.0 2.0 -1.0 2.0 1.0 0.0 6.0 4.0 3.0 Matriks setelah Eliminasi Gauss: 1.0 0.125 0.375 0.25 0.0 0.0 1.0 -0.19999999999999998 0.17142857142857143 0.11428571428571428 0.0 0.0 1.0 -0.7922077922077921 0.7597402597402597 0.0 0.0 0.0 1.0 -0.15110759493670883 Hasil perhitungan menggunakan metode Gauss : x1 : -0.23575949367088606 x2 : 0.26819620253164556 x3 : 0.6400316455696202 x4 : -0.15110759493670883
	Gauss-Jordan	Matriks sebelum Eliminasi Gauss-Jordan: 8.0 1.0 3.0 2.0 0.0 2.0 9.0 -1.0 2.0 1.0 1.0 3.0 2.0 -1.0 2.0 1.0 0.0 6.0 4.0 3.0 Matriks setelah Eliminasi Gauss-Jordan: 1.0 0.0 0.0 0.0 -0.23575949367088608 0.0 1.0 0.0 0.0 0.26819620253164556 0.0 0.0 1.0 0.0 0.6400316455696202 0.0 0.0 0.0 1.0 -0.15110759493670883 Hasil perhitungan menggunakan metode Gauss-Jordan: x1 : -0.23575949367088608 x2 : 0.26819620253164556 x3 : 0.6400316455696202 x4 : -0.15110759493670883

	Cramer	Matriks sebelum operasi Cramer: 8.0 1.0 3.0 2.0 0.0 2.0 9.0 -1.0 2.0 1.0 1.0 3.0 2.0 -1.0 2.0 1.0 0.0 6.0 4.0 3.0 Solusi SPL: x1 : 0.23575949367088594 x2 : 0.26819620253164567 x3 : 0.6400316455696203 x4 : -0.15110759493670883
	Balikan (Invers)	Matriks sebelum operasi Inverse untuk mencari SPL: 8.0 1.0 3.0 2.0 0.0 2.0 9.0 -1.0 2.0 1.0 1.0 3.0 2.0 -1.0 2.0 1.0 0.0 6.0 4.0 3.0 x1 = 0.0 x2 = 0.0 x3 = 0.0 x4 = 0.0

4.2. Determinan

Masukan (Soal)	Metode	Keluaran
det.txt U × test > det.txt 1 2 1 3 2 1 0 1 3 3 2 4	Reduksi Baris	Matriks sebelum operasi determinan: 2.0 1.0 3.0 1.0 0.0 1.0 3.0 2.0 4.0 Determinan : 1.0
	Kofaktor	Matriks sebelum operasi determinan kofaktor: 2.0 1.0 3.0 1.0 0.0 1.0 3.0 2.0 4.0 Determinan matriks adalah: 1.0

4.3. Balikan (Inverse)

Masukan (Soal)	Metode	Keluaran

det.txt U × test > det.txt 1 2 1 3 2 1 0 1 3 3 2 4	OBE	Matriks sebelum operasi invers: 2.0 1.0 3.0 1.0 0.0 1.0 3.0 2.0 4.0 Matriks setelah operasi invers dengan OBE: -2.0 2.0 1.0 -1.0 -1.0 1.0 2.0 -1.0 -1.0
	Adjoin	Matriks sebelum operasi Invers Adjoint: 2.0 1.0 3.0 1.0 0.0 1.0 3.0 2.0 4.0 Matriks setelah operasi Invers Adjoint: -2.0 2.0 1.0 -1.0 -1.0 1.0 2.0 -1.0 -1.0

4.4. Interpolasi Polinomial

Masukan (Soal)	Keluaran
polInterA.txt 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 0.2	Hasil perhitungan interpolasi polinomial : f(x) = -0.022976562500000193 + 0.24000000000000324x^1 + 0.197 39583333331565x^2 + 4.381217610927024E-14x^3 + 0.026041666666 612957x^4 + 3.180326372625392E-14x^5 - 7.265660392006549E-15x ^6 Dengan taksiran untuk f(0.2) adalah 0.03296093750000002
polInterB.txt 6.567 12624 7 21807 7.258 38391 7.451 54517 7.548 51952 7.839 28228 8.161 35764 8.484 20813 8.709 12408 8.322	Hasil perhitungan interpolasi polinomial : f(x) = -5.31681771597685E12 + 5.437840121163588E12x^1 - 2.428 215356171536E12x^2 + 6.183196810767151E11x^3 - 9.820121907008 473E10x^4 + 9.960770740614372E9x^5 - 6.301372713954141E8x^6 + 2.273101053168845E7x^7 - 357974.69529317954x^8 Dengan taksiran untuk f(8.322) adalah 37489.1767578125

polInterC.txt 0 0 0.2 0.34277 0.4 0.41888 0.6 0.46843 0.8 0.50716 1 0.53788 1.2 0.56092 1.4 0.57619 1.6 0.58368 1.8 0.58367 2 0.57665 1.2	Hasil perhitungan interpolasi polinomial : f(x) = 0.0 + 3.881760396828323x^1 - 18.814652966312767x^2 + 57.33050433364476x^3 - 111.53709394373062x^4 + 143.34300763327067x^5 - 123.20037887205449x^6 + 70.034806187446x^7 - 25.257155000296056x^8 + 5.230481452385642x^9 - 0.4733992211814385x^10 Dengan taksiran untuk f(1.2) adalah 0.5609200000000683

4.5. Regresi Berganda

Masukan (Soal)	Metode	Keluaran
linreg.txt test > linreg.txt 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 86.8 29.03 0.82 54.9 70.9 29.37 0.95	Linear	Matriks sebelum regresi linier: 72.4 76.3 29.18 0.9 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 1.0 12.9 67.4 29.39 1.1 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 29.6 1.07 23.2 76.8 29.38 1.07 47.4 86.6 29.35 0.94 31.5 76.9 29.63 1.1 10.6 86.3 29.56 1.1 11.2 86.0 29.48 1.1 73.3 76.3 29.4 0.91 75.4 77.9 29.28 0.87 96.6 78.7 29.29 0.78 107.4 86.8 29.03 0.82 54.9 70.9 29.37 0.95 Masukkan taksiran (x1, x2, ...): x_1: 50 76 29.30 x_2: x_3: Hasil perhitungan regresi linear berganda : f(x) = 1.6531569669532473 - 54.6028998695634*x_1 + 189.6487923097071*x_2 - 136.33928851604017*x_3 Dengan taksiran untuk f((50.0, 76.0, 29.3) adalah 7690.075225506545

	Kuadratik	Matriks sebelum regresi kuadratik: 72.4 76.3 29.18 0.9 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 1.0 12.9 67.4 29.39 1.1 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 29.6 1.07 23.2 76.8 29.38 1.07 47.4 86.6 29.35 0.94 31.5 76.9 29.63 1.1 10.6 86.3 29.56 1.1 11.2 86.0 29.48 1.1 73.3 76.3 29.4 0.91 75.4 77.9 29.28 0.87 96.6 78.7 29.29 0.78 107.4 86.8 29.03 0.82 54.9 70.9 29.37 0.95 Hasil perhitungan regresi kuadratik berganda : f(x) = 427.4412831515074 + 3975.135673433588*x_1 - 938.05970 81084821*x_2 - 3452.8027285011976*x_3 + 183.9522175779307 5*x _1^2 - 2736.798631262599*x_2^2 + 2362.6380760085453*x_3^2 + 3141.429933145087*x_1*x_2 - 10649.797995043693*x_1*x_3 + 768 6.86655190132*x_2*x_3 Dengan taksiran untuk f((50.0, 76.0, 29.3) adalah 159751.499 82431158

4.6. Bicubic Spline Interpolation

Masukan (Soal)	Keluaran
bspline1.txt 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 0.5 0.5	Nama file yang dipilih: bspline1.txt f(0.5,0.5) = 8.125

BAB 5 KESIMPULAN

5.1 Kesimpulan

Dalam proyek Tugas Besar 1 IF2123 Aljabar Linier dan Geometri ini, kami telah mengimplementasikan objek-objek matriks beserta sejumlah fungsi pengolahan matriks diantaranya metode eliminasi Gauss, metode eliminasi Gauss-Jordan, perhitungan determinan matriks, penentuan matriks balikan, dan fungsi-fungsi lainnya. Implementasi ini berhasil diintegrasikan sebuah proyek berbasis Java yang mampu menyelesaikan berbagai permasalahan matriks yang direpresentasikan dalam bentuk Sistem Persamaan Linier. Hasil penyelesaian tersebut dapat berupa tiga jenis solusi, yaitu solusi banyak, solusi uni, dan ketiadaan solusi.

Dalam tugas besar ini, tingkat efisiensi dalam menyelesaikan masalah Sistem Persamaan Linier (SPL) bergantung dari jenis matriks yang diberikan. Jika menggunakan metode kaidah Cramer atau metode matriks balikan, maka metode tersebut hanya dapat digunakan pada matriks persegi. Jika kita ingin menyelesaikan permasalahan SPL pada jenis matriks lain, maka kita dapat menggunakan metode eliminasi Gauss dan eliminasi Gauss-Jordan. Kedua metode tersebut jauh lebih cepat dan jauh lebih efisien dibandingkan menggunakan kaidah Cramer jika ingin menyelesaikan permasalahan SPL. Selanjutnya, metode penyelesaian SPL ini dapat digunakan sebagai dasar untuk menangani permasalahan lainnya seperti interpolasi polinom, regresi linier berganda, regresi kuadratik berganda, dan *bicubic spline interpolation*.

Semua solusi di atas disatukan dalam program utama berbasis Java yang mencakup :

1. Penyelesaian Sistem Persamaan Linier (SPL), yang menggunakan metode eliminasi Gauss, eliminasi Guass-Jordan, kaidah Cramer, dan metode matriks balikan
2. Penyelesaian determinan matriks yang menggunakan metode Operasi Baris Elementer(OBE) dan metode kofaktor
3. Penyelesaian matriks balikan suatu matriks dengan metode reduksi baris dan metode matriks adjoin
4. Penyelesaian interpolasi polinomial
5. Penyelesaian regresi linier berganda
6. Penyelesaian regresi kuadratik berganda
7. Penyelesaian *bicubic spline interpolation*

5.2 Saran

Bagi kelompok kami, Tugas Besar 1 IF2123 Aljabar Linier dan Geometri Semester I Tahun 2024/2025 ini, yang juga merupakan Tugas Besar pertama kami di Teknik Informatika ITB, adalah sebuah pengalaman belajar yang cukup berharga. Dengan berakhirnya perjuangan kami untuk tugas ini, kami ingin membagikan saran-saran kepada pembaca yang mungkin akan atau sedang menghadapi tugas serupa.

1. Pemahaman Materi dan Bahasa Pemrograman

   Tugas Besar mata kuliah pemrograman mengharuskan penulisan program dalam bahasa tertentu. Namun, pembelajaran mata kuliah di kelas umumnya hanya mempelajari sintaks dan logika dasar bahasa pemrograman. Kelompok kami menyarankan pembaca untuk meluangkan waktu dalam mempelajari bahasa pemrograman yang akan digunakan. Eksplorasi juga dapat dilakukan seiring berjalannya pengerjaan tugas. Carilah banyak dokumentasi dan tutorial bahasa pemrograman di internet.

Pembaca juga harus memahami materi yang sedang diterapkan. Pembaca dapat merangkum teori dasar lebih dahulu agar materi dapat lebih dipahami. Selain itu, untuk dapat menerjemahkan teori menjadi algoritma bahasa pemrograman, pembaca dapat membuat *flowchart*.

2. Kerja Sama Kelompok dan Manajemen Waktu

    Pengerjaan tugas akan lebih mudah dilalui jika kelompok telah membuat strategi terlebih dahulu. Strategi yang dimaksud dapat berupa pembagian tugas yang adil. Masing-masing anggota kelompok juga harus berkontribusi aktif selama pengerjaan tugas. Tidak hanya itu, untuk mempermudah kolaborasi baik secara sinkron maupun asinkron, kelompok juga disarankan untuk memanfaatkan *software* kolaborasi seperti Github dan VSCode, Google Docs, serta Zoom.

    Terakhir, manajemen waktu merupakan aspek yang sangat penting dalam pengerjaan suatu tugas mandiri maupun berkelompok. Setiap anggota kelompok harus dapat membagi waktu dengan baik dan mempunyai kedisiplinan. Jika anggota kelompok sedang terkendala waktu dikarenakan kesibukan lainnya, anggota tersebut wajib mengabari kelompoknya agar kelompok dapat saling membantu.

5.3 Refleksi

Tugas Besar 1 IF2123 Aljabar Linier dan Geometri di Semester I Tahun 2024/2025 merupakan perjuangan pertama yang kelompok kami hadapi di jurusan Teknik Informatika ITB. Selayaknya sebuah perjuangan, pengerjaan tugas ini tidak datang tanpa hambatan. Melalui tugas ini, kami mempelajari materi lebih dalam dari apa yang telah diajarkan di kelas. Kami belajar bahwa matriks merupakan salah satu struktur data yang dapat dimanfaatkan untuk memecahkan berbagai masalah matematika dan teknologi. Khusus di tugas besar ini, kami memanfaatkan matriks untuk menyelesaikan masalah-masalah Sistem Persamaan Linier.

Dalam perjuangan kelompok kami selama beberapa minggu ini, kami merasakan kegembiraan dan kepuasan tersendiri dalam mempelajari matriks, dan mempelajari sebuah bahasa pemrograman baru, yaitu Java. Walaupun banyak rintangan, dengan kerja sama tim yang baik dan kontribusi aktif setiap anggota kelompok, kami berhasil menyelesaikan Tugas Besar ini.

LAMPIRAN

1. Video presentasi program
   Link: Video Demo
2. Repository
   Link : Release v1

DAFTAR PUSTAKA

Anton, H., & Rorres, C. (2010). *Elementary Linear Algebra with Applications* (9th ed.). John Wiley & Sons. Diakses dari https://s2pnd-matematika.fkip.unpatti.ac.id/wp-content/uploads/2019/03/Elementary-Linear-Algebra-with-Applications-Anton-Rorres.pdf.

Munir, Rinaldi. (2024). Slide Bahan Kuliah IF2123 Aljabar Geometri - Semester I 2024/2025. Diakses dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2024-2025/algeo24-25.htm#SlideKuliah

Rowe, D. B. (2001). *Bi-Cubic Interpolation*. Diakses dari https://www.mssc.mu.edu/~daniel/pubs/RoweTalkMSCS_BiCubic.pdf.