Tugas Besar 1 IF2123 Aljabar Linier dan Geometri
Sistem Persamaan Linier, Determinan, dan Aplikasinya
Semester I Tahun 2023/2024

Laporan Tugas Besar 1

Disusun untuk memenuhi tugas mata kuliah Aljabar Linear dan Geometri pada Semester 1 (satu) Tahun Akademik 2023/2024.

Oleh

Hugo Sabam Augusto Sianturi	**13522129**
Muhammad Zaki	**13522136**
Andi Marihot Sitorus	**13522138**

Kelompok pinjamdulu100

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2023

BAB I

Deskripsi Masalah

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

BAB 2

Teori Singkat

2.1. Sistem Persamaan Linear

2.1.1 Metode Eliminasi Gauss

Metode Gauss ini ditemukan oleh Carl Friedrich Gauss (1777-1855),metode merupakan salah satu cara untuk menyelesaikan permasalahan SPL. Metode ini membutuhkan suatu matrix augmented yang dibentuk dari SPL. Lalu dilakukan operasi baris elementer untuk mengubah matrix itu menjadi matriks eselon baris. Setelah itu dilakukan teknik penyulihan mundur untuk mendapatkan solusi.

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

2.1.2 Metode Eliminasi Gauss - Jordan

Metode Eliminasi Gauss-Jordan ini merupakan variasi lanjutan dan pengembangan dari metode sebelumnya yaitu Metode Eliminasi Gauss. Metode ini menerapkan OBE(Operasi Baris Elementer) pada matriks augmented dan akhirnya akan menghasilkan matriks eselon baris tereduksi.

Dalam metode ini, tidak diperlukan kembali substitusi secara mundur untuk memperoleh nilai karena variabel bisa langsung dianalisis dan diperoleh dari matriks augmented akhir, tentu saja dengan syarat apabila solusinya unik/satu,

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

S

2.1.3 Metode Matriks Balikan

Metode penyelesaian SPL dengan menggunakan matriks balikan sangat berguna untuk menyelesaikan sejumlah SPL Ax = b, akan tetapi metode hanya dapat digunakan pada matrix persegi dan ketika determinan dari matriks tersebut != 0. Solusi SPL didapatkan dengan x = A^-1 b

$$\mathbf{x} = A^{-1}\,\mathbf{b} = \begin{bmatrix} -40 & 16 & 9 \\ 13 & -5 & -3 \\ 5 & -2 & -1 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}$$

2.1.4 Metode Kaidah Cramer

Jika Ax = b adalah SPL yang terdiri dari n persamaan linier dengan n peubah (variable) sedemikian sehingga det(A) 0, maka SPL tersebut memiliki solusi yang unik yaitu, dimana kita menyulihkan matrix “b” ke kolom dari matrix A.

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

2.2 Determinan

2.2.1 Metode Reduksi Baris

Untuk mencari determinan suatu matriks dengan metode Reduksi Baris,diperlukan OBE pada matriks tersebut hingga diperoleh matriks segitiga(bisa segitiga atas atau bawah)

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

Apabila sudah didapatkan suatu matriks dengan bentuk segitiga atas atau bawah, maka bisa dirumuskan bahwa **det(A) = $(-1)^p$ a'_{11} a'_{22} …. a'_{nn}**

dengan p menyatakan banyaknya operasi pertukaran di dalam OBE.

Tak hanya berhenti disini, apabila selama reduksi baris ada OBE yang menggunakan operasi perkalian baris-baris matriks dengan k1,k2,...,km

maka det(A) =

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

2.2.2 Metode Ekspansi Kofaktor

Misalkan ada sebuah Matriks persegi A

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

M_{ij} = minor entri a_{ij}
= determinan upa-matriks (*submatrix*) yang elemen-elemennya tidak berada pada baris *i* dan kolom *j*

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

Untuk menghitung minor entri kita dapat melakukan langkah seperti contoh berikut

$$A = \begin{bmatrix} 6 & -3 & 1 \\ 2 & 2 & -4 \\ 1 & 5 & 3 \end{bmatrix} \qquad M_{11} = \begin{vmatrix} 2 & -4 \\ 5 & 3 \end{vmatrix} = (2)(3) - (-4)(5) = 26$$

Lakukan untuk setiap kolom / baris. Lalu kita dapat menghitung determinan menggunakan salah satu dari kolom / baris

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

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

2.3 Balikan Matrix

2.3.1 Metode Matrix Adjoint

Misalkan A adalah matriks n x n dan C_{ij} adalah kofaktor entri a_{ij}. Maka matriks kofaktor dari A adalah

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

Adjoin dari matriks A adalah transpose dari matriks kofaktor A. Balikan matriks A dapat dihitung dengan menggunakan rumus

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

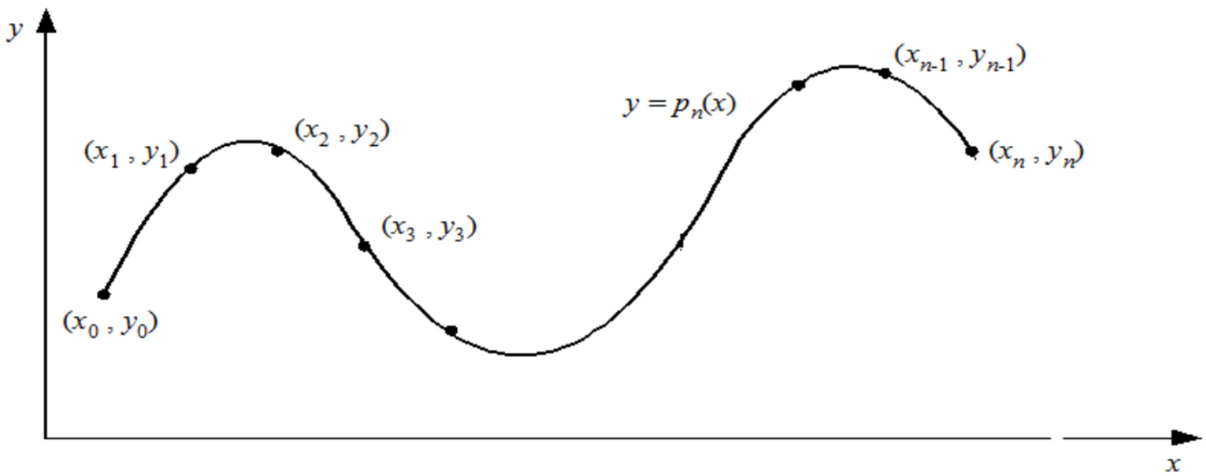
2.3.2 Metode Matrix Balikan

Misalkan A adalah matriks persegi berukuran n x n. Balikan (inverse) matriks A adalah A–1 sedemikian sehingga $AA^{-1} = A^{-1}A = I$.Sehingga kita dapat menggunakan metode eliminasi Gauss-Jordan untuk mendapatkan matrix balikan dari A dengan cara berikut

$$[A|I] \overset{\text{G-J}}{\sim} [I|A^{-1}]$$

2.4 Interpolasi Polinomial

Interpolasi polinomial merupakan teknik interpolasi dengan mengasumsikan pola data yang kita miliki mengikuti pola polinomial. Metode dilakukan dengan cara membentuk terlebih dahulu persamaan polinomial $P_n(x)$ dari n + 1 buah titik berbeda,sehingga $y_i = P_n(x_i)$ untuk i = 0,1,2,...,n. Lalu, kita dapat menghitung perkiraan nilai y pada x = a, yaitu $y = P_n(a)$.



2.5 Regresi Linear

Regresi linear berganda merupakan model regresi yang melibatkan lebih dari satu variabel independen. Analisis regresi linear berganda dilakukan untuk mengetahui arah dan seberapa besar pengaruh variabel independen terhadap variabel dependen (Ghozali, 2018).Terdapat persamaan umum dari regresi linear yang bisa digunakan untuk regresi linear berganda, yaitu:

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

Cara mendapatkan nilai β dari persamaan di atas, bisa menggunakan *Normal Estimation for Multiple Linear Regression*

2.6 Bicubic Spline Interpolations

Bicubic spline interpolation adalah metode interpolasi yang digunakan untuk mengaproksimasi fungsi di antara titik-titik data yang diketahui

BAB 3

Implementasi pustaka dan program dalam Java

3.1 Folder iomatrix

3.1.1 InputMatrix.java

Method	Deskripsi
Public static Matrix inputInterpolasi(Scanner scanner)	Menerima inputan untuk metode Interpolasi Polinomial
public static Matrix inputHilbert(Scanner scanner)	Menerima Scanner dari main, Mengembalikan dan menghasilkan nilai matriks dengan pola matriks Hilbert
public static String inputRegresiLinierFile(Scanner scanner)	Menerima Scanner, lalu mengembalikan matriks yang sudah diolah menjadi regresi dengan input FILE, lalu print solusi dan save mengembalikan string solusi agar bisa di save
public static String inputRegresiLinierKeyboard(Scanner scanner)	Menerima Scanner, lalu mengembalikan matriks yang sudah diolah menjadi regresi dengan input KEYBOARD, lalu print solusi dan save mengembalikan string solusi agar bisa di save
Public static Matrix inputMatrixKeyboard(Scanner scanner)	Menerima scanner, membuat dan mengembalikan matrix dengan nilai elemen sesuai input keyboardnya
public static Matrix inputFileMatrix(Scanner scanner)	Menerima scanner, membuat dan mengembalikan matrix dengan nilai elemen

	sesuai filenya, termasuk dalam pengolahan file unik di kasus Interpolasi,Bicubic,dan Regresi

3.1.2 OutputMatrix.java

Method	Deskripsi
public static void tulisMatrix(Matrix M)	Metode ini digunakan untuk memprint matrix
public static String tulisSolusiGaussJordan(String[] ans, Matrix m)	Metode ini digunakan untuk memprint solusi spl yang dihasilkan oleh gauss jordan ketika terjadi parametrik
public static void tulisSolusi(Matrix x)	Metode ini digunakan untuk menuliskan hasil atau solusi dari SPL
public static void createFile()	Metode ini digunakan untuk membuat sebuah file yang nantinya digunakan untuk menyimpan hasil
public static String matrixToString(Matrix M)	Metode ini digunakan untuk mengubah matriks ke string
public static void pathMaker()	Metode ini digunakan untuk membuat sekaligus mencari path
public static void TuliskeTxt(String s)	Metode ini digunakan menulis string ke dalam sebuah file
public static void tulisSolusiInterpolasiP(Matrix m)	Metode ini digunakan untuk menghasilkan output dari solusi ketika menggunakan metode interpolasi polinomial

3.2 Folder matrix

3.2.1 MatrixBalikan.java

Method	Deskripsi
public static Matrix GaussJordan(Matrix m)	Metode ini akan menghasilkan invers dari matrix yang diinput dengan metode gauss jordan

3.2.2 InterpolasiPolinomial,java

Method	Deskripsi
public static Matrix Interpolasi(Matrix m, boolean isFile,Scanner scanner)	Metode ini digunakan untuk mencari hasil dari interpolasi polinomial
public static Matrix createAugmentedMatrix(Matrix dataXY)	Metode ini akan membuat matrix augmented dari input yang didapat

3.2.3 SPL.java

Method	Deskripsi
public static Matrix inverseMatrix(Matrix m)	Mencari penyelesaian SPL menggunakan metode matrix balikan
public static Matrix Cramer(Matrix m)	Mencari penyelesaian SPL menggunakan metode atau kaidah cramer
public static void OBEBarisPlusMinusFaseMaju(Matrix m, int i1, int j1, int i2)	Menerima matrix m dan mengolah baris tersebut dengan OBE fase maju(digunakan untuk membuat matriks eselon pada Gauss &

	Gauss Jordan)
public static Matrix createMatriksEselon(Matrix m)	Digunakan untuk membuat matriks menjadi matriks eselon
public static boolean cekKolom0(Matrix m, int col)	Digunakan untuk men-check apakah kolom tersebut bernilai 0 atau tidak
public static void createEselonTereduksi(Matrix m)	Prosedur menerima matriks dan mengolahnya menjadi matriks eselon tereduksi dengan prekondisi input matriks eselon.
public static String[] solutionGaussJordan(Matrix m)	Digunakan untuk mencari solusi dari SPL menggunakan metode gauss jordan
public static String[] solutionGauss(Matrix m)	Digunakan untuk mencari solusi dari SPL menggunakan metode gauss

3.2.4 Tools.java

Method	Deskripsi
public static void pause()	Digunakan untuk pause atau menunggu ada input untuk melanjutkan program dan juga clear screen
public static boolean checkSquare(Matrix m)	Digunakan untuk check apakah sebuah matrix itu kotak atau tidak
public static Matrix extractB(Matrix m)	Digunakan untuk mengambil ‘b’ dari matrix untuk diolah lebih lanjut
public static Matrix matrixWithoutB(Matrix m)	Digunakan untuk mengambil matrix tanpa ada ‘b’ didalamnya

public static Matrix konversiFloattoDouble(Matrix m)	Digunakan untuk konversi dari float ke double
public static void copyMatrix(Matrix ref,Matrix dest)	Digunakan untuk copy matrix dari reference ke destination
public static Matrix multiplyMatrix(Matrix matrix,Matrix b)	Digunakan untuk mengalikan matrix
public static void kirimOpsiMethod()	Digunakan untuk mengirim opsi metode untuk SPL
public static void kirimOpsiSimpan()	Digunakan untuk mengirim opsi simpan apakah mau disimpan pada file atau tidak
public static boolean cekNoSolution(Matrix m)	Digunakan untuk Cek apabila ditemukan pola tak ada solusi (di baris i semua 0 kecuali di kolom akhir
public static void buat1Utama(Matrix m, int row)	Digunakan untuk Menerima 1 baris dalam matrix, dan tentukan 1 utama serta dibagi dalam kolom "row tsb
public static void swapBaris(Matrix m, int i1, int i2)	Prosedur menukar nilai baris i1 dengan i2, i2 menjadi i1
public static boolean Same1Utama(Matrix m)	Mengembalikan nilai TRUE apabila ditemukan 1 utama yang lebih dari 1 dalam 1 kolom(berarti tidak ada solusi)
public static boolean cekSolusiBanyak(Matrix m)	Return true apabila ditemukan 1 baris berisi 0 semua

public static int getBarisSama(Matrix m, int row)	Mengembalikan baris ke berapa yang isi elemen sama persis dengan baris "row" Bila tidak ditemukan, return baris yang sama dengan row
public static void delBarisSama(Matrix m)	baris yang diketahui sama, salah satu nya dibuat jadi 0
public static boolean cekBarisUnik(Matrix m, int row)	mengembalikan nilai true apabila baris itu hasil unik / koefisien 1
public static double getElmtDiagonal(Matrix m , int i)	Mengembalikan nilai elemen diagonal dari matriks m
public static Matrix DividebyCons(Matrix matrix, double x)	Membagi matrix dengan constanta
public static Matrix Transpose(Matrix matrix)	Membuat matrix menjadi transpose

3.2.5 ImageUpscale.java

Method	Deskripsi
public static String pathmaker(String dir)	Membuat path ke folder test
public static Matrix matrixpojok(int height, int width, Buffered image)	Fungsi untuk memperoleh matriks 4x4 dari setiap pixel bagian pojok
public static Matrix matrixpingggir(int height, int width, Buffered image)	Fungsi yang menghasilkan Matrix 4x4 untuk pixel bagian pinggir
public static Matrix matrixtengah(int height, int width, Buffered image)	Fungsi yang menghasilkan Matrix 4x4 untuk pixel bagian tengah

public static Matrix Warna(Matrix matrix, String s)	Fungsi untuk memperoleh matriks 4x4 berisi nilai warna pixel sesuai dengan nama warna yang di input
Public static int bicubicInterpolate(Buffered image, double x, double y)	Fungsi untuk memperoleh nilai pixel hasil interpolasi sesuai dengan posisi pixel pada gambar.
public static void ImageUps(BufferedImage inputImage, int s, String dir)	Procedur untuk melakukan proses image scaling.

3.2.6 DeterminanOBE.java

Method	Deskripsi
public static double determinanOBE(Matrix m)	Menerima Matrix lalu mengolah menggunakan OBE dan menghasilkan matriks segitiga atas dan mengembalikan nilai determinannya.

3.2.7 BicubicSpline.java

Method	Deskripsi
Public static void fungsid(int bariskoef, Matrix koef)	Procedure untuk mengisi bagian matrix X 4 baris pertama
Public static void fungsidx(int bariskoef, Matrix koef)	Procedure untuk mengisi bagian matrix X baris 4 hingga 7
Public static void fungsidy(int bariskoef, Matrix koef)	Procedure untuk mengisi bagian matrix X baris 8 hingga 11

Public static void fungsidxy(int bariskoef, Matrix koef)	Procedure untuk mengisi bagian matrix X baris 12 hingga 15
Public static matrix bikinKoef(Matrix koef)	Fungsi yang menghasilkan matrix X
Public static double fspline(double x, double y, Matrix aij)	Fungsi untuk mengimplementasikan interpolasi bicubic sehingga menghasilkan value yang telah diinterpolasi
Public static void koeff(Matrix mat, double x, double y)	Prosedur untuk menuliskan hasil interpolasi bicubic spline menggunakan koefisien X
Public static void fungsid(int bariskoef, Matrix koef)	Procedure untuk mengisi bagian matrix D 4 baris pertama
Public static void fungsidx(int bariskoef, Matrix koef)	Procedure untuk mengisi bagian matrix D baris 4 hingga 7
Public static void fungsidy(int bariskoef, Matrix koef)	Procedure untuk mengisi bagian matrix D baris 8 hingga 11
Public static void fungsidxy(int bariskoef, Matrix koef)	Procedure untuk mengisi bagian matrix D baris 12 hingga 15
Public static matrix bikinKoefD(Matrix koef)	Fungsi yang menghasilkan matrix D
Public static double spline (Matrix matrix, double x, double y)	Fungsi yang menghasilkan value yang telah diinterpolasikan dengan koefisien $X^{-1}D$

3.2.8 RegresiLinierBerganda.java

Method	Deskripsi
public static Matrix regresiGanda(Matrix m)	Prekondisi : Menerima matriks yang sudah berbentuk data regresi linier lalu

	Mengembalikan Matrix eEquation(matriks yang sudah diolah dengan Rumus Normal Equation)
public static double sumKolom(Matrix m,int col)	Menghasilkan sum dan menjumlahkan seluruh baris dalam col tersebut
public static Float[] tulisSolusiRegresi(String[] ans, Matrix m)	Menerima array String dari persamaan jawaban dan menuliskan serta operasi nilai variabel ke dalam array double dan mengembalikannya(digunakan dalam inputRegresiLinierFile dan inputRegresiLinierKeyboard

3.2.9 Kofaktor.java

Method	Deskripsi
public static double hitungDeterminan(Matrix matrix)	Fungsi yang menghasilkan determinan dari matrix dengan cara ekspansi kofaktor
public static Matrix getSubMatrix(Matrix matrix, int row, int col)	Fungsi yang menghasilkan submatrix tanpa baris dan kolom tertentu
Public static Matrix MatrixKofaktor(Matrix matrix)	Fungsi yang menghasilkan matrix kofaktor dari matriks input
Public static Matrix InverseKofaktor(Matrix matrix)	Fungsi yang menghasilkan matrix inverse dari matriks input menggunakan matriks adjoin yang didapat dari matrix kofaktor
Public static void TulisInverse(Matrix matrix)	Prosedur untuk menuliskan matriks inverse
Public static void TulisDetKofaktor(Matrix	Prosedur untuk menuliskan determinan hasil

matrix)	kofaktor

3.2.10 Matrix.java (disini digunakan Class Matrix)

public Matrix(int row,int col)	Constructor class Matrix
public double getElmt(int i,int j)	Getter untuk class Matrix -Mengambil nilai elemen di row ke-i dan col ke-j
public int getRowEff()	Getter untuk class Matrix -Mengambil jumlah baris di Matrix tersebut
public int getColEff()	Getter untuk class Matrix -Mengambil jumlah kolom di Matrix tersebut
public int setElmt(int i,int j,double x)	Setter untuk class Matrix -Mengganti elemen matrix di row ke-i dan col ke-j dengan double x

BAB 4

EKSPERIMEN

4.1 Temukan solusi SPL Ax = b, berikut:

a. Test case 1

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

Gauss:

```
masukkan nama file txt:  TC1a
Tidak ada solusi!
```

Gauss-Jordan:

```
masukkan nama file txt:  TC1a
Tidak ada solusi!
```

Matrix Balikan:

```
masukkan nama file txt:  TC1a
Matrix tidak valid / atau tidak memiliki solusi unik
```

Kaidah Cramer:

```
Determinan matriks adalah nol, tidak ada solusi atau solusi tidak unik.
```

b. Test Case 2

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

Gauss:

```
=============SOLUSI===============
X1 = 1.0R5 + 3.0
X2 = 2.0R5
X3 = R3 - isi bebas
X4 = 1.0R5 -1.0
X5 = R5 - isi bebas
====================================
```

Gauss-Jordan :

```
=============SOLUSI===============
X1 = 1.0R5 + 3.0
X2 = 2.0R5
X3 = R3 - isi bebas
X4 = 1.0R5 -1.0
X5 = R5 - isi bebas
====================================
```

Matrix Balikan :

```
masukkan nama file txt:  TC1b
Matrix tidak valid / atau tidak memiliki solusi unik
```

Kaidah Cramer:

```
masukkan nama file txt:  TC1b
0.0
Determinan matriks adalah nol, tidak ada solusi atau solusi tidak unik.
```

c. Test case 3

Gauss :

```
=============SOLUSI================
X1 = R1 - isi bebas
X2 = - 1.0R6 + 1.0
X3 = R3 - isi bebas
X4 = R4 - isi bebas
X5 = 1.0R6 + 1.0
X6 = R6 - isi bebas
===================================
```

Gauss-Jordan:

```
=============SOLUSI================
X1 = R1 - isi bebas
X2 = - 1.0R6 + 1.0
X3 = R3 - isi bebas
X4 = R4 - isi bebas
X5 = 1.0R6 + 1.0
X6 = R6 - isi bebas
===================================
```

Matrix Balikan :

```
masukkan nama file txt:  TC1c
Matrix tidak valid / atau tidak memiliki solusi unik
```

Kaidah Cramer :

```
0.0
Determinan matriks adalah nol, tidak ada solusi atau solusi tidak unik.
```

4.2 SPL berbentuk matriks *augmented*

a.Test Case 1

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

Gauss:

```
=============SOLUSI===============
X1 = 1.0R4 -1.0
X2 = 2.0R3
X3 = R3 - isi bebas
X4 = R4 - isi bebas
==================================
```

Gauss-Jordan :

```
=============SOLUSI===============
X1 = 1.0R4 -1.0
X2 = 2.0R3
X3 = R3 - isi bebas
X4 = R4 - isi bebas
==================================
```

Matriks Balikan

```
Matrix tidak valid / atau tidak memiliki solusi unik
```

Metode Cramer

```
Determinan matriks adalah nol, tidak ada solusi atau solusi tidak unik
.
```

B. Test Case 2

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

Gauss :

```
=============SOLUSI=========
X1 = R1 - isi bebas
X2 = 2.0
X3 = 1.0
X4 = 1.0
```

Gauss-Jordan:

```
=============SOLUSI==============
X1 = R1 - isi bebas
X2 = 2.0
X3 = 1.0
X4 = 1.0
```

Matriks Balikan:

```
Matrix tidak valid / atau tidak memiliki solusi unik
```

Metode Cramer:

```
Matrix tidak valid untuk metode ini
```

4.3 SPL Berbentuk

a. Test case 1

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

Gauss :

```
=============SOLUSI================
X1 = -0.22432432432432453
X2 = 0.18243243243243235
X3 = 0.7094594594594594
X4 = -0.2581081081081081
====================================
```

Gauss - Jordan :

```
=============SOLUSI================
X1 = -0.22432432432432453
X2 = 0.18243243243243235
X3 = 0.7094594594594594
X4 = -0.2581081081081081
====================================
```

Matrix Balikan :

```
Solusi Sistem Persamaan Linear:
x1 = -0.224324
x2 = 0.182432
x3 = 0.709459
x4 = -0.258108
```

Kaidah Cramer:

```
Solusi Sistem Persamaan Linear:
x1 = -0.224324
x2 = 0.182432
x3 = 0.709459
x4 = -0.258108
```

b. Test case 2

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

Gauss :

```
masukkan nama file txt:  TC3b
Tidak ada solusi!
```

Gauss - Jordan :

```
Tidak ada solusi!
```

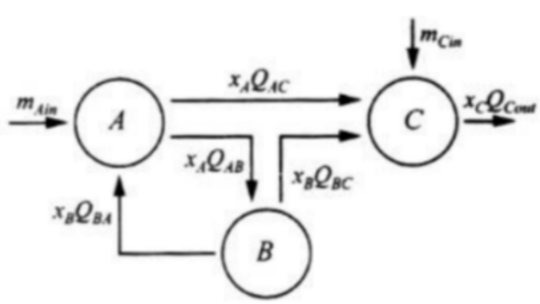
Matrix Balikan :

```
Matrix tidak valid / atau tidak memiliki solusi unik
```

Kaidah Cramer :

```
Matrix tidak valid untuk metode ini
```

4.4 Studi Kasus Sistem Reaktor dengan SPL



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.

Gauss:

```
=============SOLUSI===============
X1 = 14.444444444444446
X2 = 7.222222222222223
X3 = 10.0
```

Gauss Jordan:

```
X1 = 14.444444444444446
X2 = 7.222222222222223
X3 = 10.0
```

Matriks Balikan:

```
Solusi Sistem Persamaan Linear:
x1 = 14.444444
x2 = 7.222222
x3 = 10.000000
```

Metode Cramer:

```
Solusi Sistem Persamaan Linear
x1 = 14.444444
x2 = 7.222222
x3 = 10.000000
```

4.5 Studi Kasus Interpolasi

a. Test Case 1

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

i. X = 0.2

```
P6(x) = -0.023 + 0.240x^1 + 0.197x^2 + -0.000x^3 + 0.026x^4 + -0.000x^5 + -0.000x^6
Hasilnya adalah: 0.032960937500003805
```

ii. X = 0.55

```
P6(x) = -0.023 + 0.240x^1 + 0.197x^2 + -0.000x^3 + 0.026x^4 + -0.000x^5 + -0.000x^6
Hasilnya adalah: 0.17111865234366566
```

iii. X =0.85

```
P6(x) = -0.023 + 0.240x^1 + 0.197x^2 + -0.000x^3 + 0.026x^4 + -0.000x^5 + -0.000x^6
Hasilnya adalah: 0.337235839843685
```

iv. X = 1.25

```
P6(x) = -0.023 + 0.240x^1 + 0.197x^2 + -0.000x^3 + 0.026x^4 + -0.000x^5 + -0.000x^6
Hasilnya adalah: 0.6775418375011235
```

b. Test Case 2

i. X = 7.156

```
P9(x) = 43592644.759 + -5496138.312x^1 + -2402974.292x^2 + 225877.884x^3 + -20195.714x^4 + -2432.422x^5 + 650.170x^6 + -
85.831x^7 + 6.088x^8 + -0.321x^9
Hasilnya adalah: -1.2202789027974717E8
```

ii. X = 8.323

```
P9(x) = 43592644.759 + -5496138.312x^1 + -2402974.292x^2 + 225877.884x^3 + -20195.714x^4 + -2432.422x^5 + 650.170x^6 + -
85.831x^7 + 6.088x^8 + -0.321x^9
Hasilnya adalah: -1.7517320482895213E8
```

iii. X = 9.167

```
P9(x) = 43592644.759 + -5496138.312x^1 + -2402974.292x^2 + 225877.884x^3 + -20195.714x^4 + -2432.422x^5 + 650.170x^6 + -
85.831x^7 + 6.088x^8 + -0.321x^9
Hasilnya adalah: -2.5917019953769353E8
```

iv. X = 13

```
P9(x) = 43592644.759 + -5496138.312x^1 + -2402974.292x^2 + 225877.884x^3 + -20195.714x^4 + -2432.422x^5 + 650.170x^6 + -
85.831x^7 + 6.088x^8 + -0.321x^9
Hasilnya adalah: -2.106411814975124E9
```

c. Test case 3

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

Hasil simplifikasi rumusnya adalah :

```
P5(x) = 0.000 + 2.035x^1 + -3.553x^2 + 3.237x^3 + -1.421x^4 + 0.236x^5
```

4.6 Studi Kasus Regresi Linear Ganda

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.

Didapatkan persamaan dan hasil :

```
Persamaan nya adalah :
y = -3.5077782-0.0026249907 x1  + 7.9894107E-4 x2  + 0.15415503 x3
X1 = 50.0
X2 = 76.0
X3 = 29.3
Dan Hampirannya adalah y =  0.9384343
```

4.7 Studi Kasus Interpolasi Bicubic Spline

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

A. Test case 1

f(0,0) = ….

```
Masukkan nilai x :
0
Masukkan nilai y :
0
HASIL : 21.0
```

B. Test case 2

f(0.5,0.5) = ….

```
Masukkan nilai x :
0.5
Masukkan nilai y :
0.5
HASIL : 87.796875
```

C. Test case 3

f(0.25,0.75) = ….

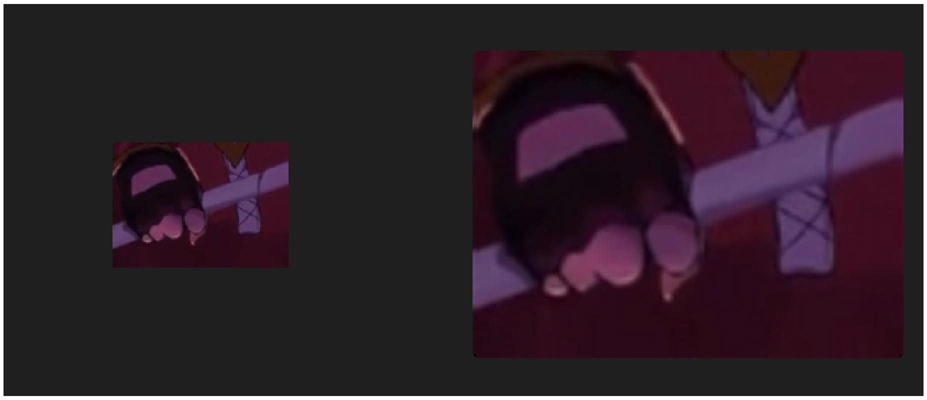
```
Masukkan nilai x :
0.25
Masukkan nilai y :
0.75
HASIL : 82.148193359375
```

D. Test case 4

f(1,0.9) = ….

```
Masukkan nilai x :
1
Masukkan nilai y :
0.9
HASIL : 148.88600000000014
```

4.8 Perbesaran Citra Gambar (Image Upscale)



BAB 5

KESIMPULAN, SARAN, KOMENTAR, DAN REFLEKSI

5.1 Kesimpulan

Terdapat berbagai cara ataupun metode untuk menyelesaikan suatu permasalahan SPL, mencari sebuah determinan, menentukan invers dari sebuah matriks ,menyelesaikan masalah interpolasi baik interpolasi polinomial ataupun bicubic spline interpolation. Dalam menyelesaikan masalah SPL, ada empat metode yang dapat kami gunakan yaitu metode gauss, gauss jordan, matriks balikan,dan juga kaidah cramer.

Pada tugas besar ini kami membuat sebuah implementasi ataupun pengaplikasian dari metode ataupun cara yang telah disebutkan di atas. Program yang kami tulis ini diimplementasikan dengan bahasa Java. Program kami diharapkan dapat membantu untuk menyelesaikan berbagai permasalahan yang melibatkan matriks, contohnya adalah menyelesaikan beberapa *problem-set* pada studi kasus menggunakan metode yang sudah disebutkan hingga cara untuk memperbesar gambar menggunakan metode bicubic spline interpolation.

5.2 Saran

Tugas Besar 1 IF2123 Aljabar Linier dan Geometri Semester I Tahun 2023/2024 merupakan salah satu tugas yang memberikan banyak pelajaran bagi penulis. Berdasarkan pengalaman penulis dalam mengerjakan tugas ini, berikut adalah saran untuk pembaca yang ingin melakukan atau mengerjakan hal yang serupa:

Spesifikasi program meminta penulis untuk menulis program dalam bahasa Java, bahasa yang belum pernah disentuh oleh sebagian besar mahasiswa IF2123 Semester I Tahun 2023/2024. Penulis merekomendasikan menyediakan waktu yang cukup untuk mempelajari dan mengerjakan tugas ini karena fitur yang harus dimuat sangat banyak dan rinci.

Keefektifan dalam kerja sama tim merupakan hal yang penting dalam mengerjakan tugas ini. Untuk pengerjaan secara sinkron, tugas ini sangat terbantu oleh penggunaan aplikasi kolaborasi, seperti github untuk pembuatan program, dan kolaborasi langsung pada google docs untuk pembuatan laporan. Selain itu, dalam pembuatan source code aplikasi, seringkali terjadi konflik antara dua atau lebih anggota. Oleh karena itu, pemakaian aplikasi pengelola versi seperti Github sangat disarankan untuk memudahkan pengelolaan pekerjaan secara asinkron.

5.3 Komentar

Menurut kelompok kami tugas besar ini cukup challenging dikarenakan menggunakan baru dikenali yang didalamnya syntax dan library yang sangat beragam dan berbeda dari bahasa yang sebelumnya pernah kami gunakan. Pada permasalahan bicubic spline interpolation dan juga bonus menurut kelompok kami sebaiknya diberikan penjelesan yang lebih banyak karena kurangnya referensi yang ada pada saat ini.

5.4 Refleksi

Tugas Besar 1 IF2123 Aljabar Linier dan Geometri Semester I Tahun 2023/2024 merupakan salah satu tugas besar pertama yang penulis dapatkan pada semester ini. Proses pengerjaan tugas ini tentunya melalui berbagai rintangan. Dengan tugas ini, penulis merasakan apresiasi yang besar terhadap penggunaan matriks serta implementasinya dalam kehidupan, baik secara matematis ataupun dalam bidang teknologi. Keberasaan matriks serta atribut-atribut yang dimilikinya memberikan dampak tersendiri dalam menyelesaikan persoalan, terutama persoalan sistem persamaan linier.

Kegunaan matriks juga dapat dijumpai pada sebagian besar bidang Informatika, contohnya dalam pengolahan citra gambar (image processing). Pada tugas ini, penulis diberi kesempatan untuk mengimplementasikan pengolahan citra pada matriks dan timbul rasa apresiasi yang tinggi dan kebanggaan dalam keberhasilan untuk memproduksi sebuah citra baru dari sekumpulan angka-angka berbentuk matriks. Dalam mengerjakan tugas ini, penulis juga merasakan euforia berhasil mempelajari bahasa pemrograman baru (Java) dan mengimplementasikannya dengan kurun waktu di bawah tiga minggu.

Rintangan yang dihadapi untuk mempelajari dan menekuni matriks selama beberapa minggu ini dilengkapi dengan sinergi dari setiap anggota kelompok yang saling membantu dan mengerjakan tugasnya dengan baik.

REFERENSI:

Ghozali, Imam. 2018. Aplikasi Analisis Multivariate dengan Program IBM SPSS 25. Badan Penerbit Universitas Diponegoro: Semarang.

Informatika.stei.itb.ac.id. (2023). Diakses terakhir tanggal 5 Oktober, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/algeo23-24.htm#SlideKuliah

Repo github : https://github.com/mzaki9/Algeo01-22129