LAPORAN TUGAS BESAR 1 IF2123 ALJABAR LINIER DAN GEOMETRI

SISTEM PERSAMAAN LINIER, DETERMINAN, DAN APLIKASINYA

Kelompok 11 – BukanBjorka

Rizky Abdillah Rasyid – 13521109

Saddam Annais Shaquille – 13521121

Hanif Muhammad Zhafran – 13521157

Semester 1 Tahun 2022/2023

BAB 1
DESKRIPSI MASALAH

A. Buatlah pustaka dalam Bahasa Java untuk menemukan solusi SPL dengan metode eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n peubah dan n persamaan), menghitung determinan matriks dengan reduksi baris dan dengan ekspansi kofaktor, dan menghitung balikan matriks.

B. Gunakan pustaka di atas untuk membuat program penyelesaian berbagai persoalan dalam bentuk SPL, menyelesaikan persoalan interpolasi dan regresi linier, menghitung matriks balikan, menghitung determinan matriks dengan berbagai metode (reduksi baris dan ekspansi kofaktor).

BAB 2
TEORI SINGKAT

A. Eliminasi Gauss

Eliminasi Gauss adalah suatu metode untuk menemukan solusi sistem persamaan linier dengan cara merepresentasikannya dalam bentuk matriks kemudian melakukan Operasi Baris Elementer (OBE). OBE dilakukan sampai terbentuk matriks eselon baris sehingga dapat dilakukan *back-subtitution* dari nilai variable yang telah ada untuk mendapatkan nilai dari setiap variabelnya.

Misalkan suatu persamaan linear seperti berikut:

$$2x + 5y + 3z = 1$$
$$3x + 4y + 2z = -3$$
$$x + 3y + z = 2$$

Untuk mencari nilai dari tiap variabel, persamaan tersebut perlu diubah ke dalam bentuk matriks terlebih dahulu. Matriks yang terbentuk adalah sebagai berikut:

$$\left[\begin{array}{ccc|c} 2 & 5 & 3 & 1 \\ 3 & 4 & 2 & -3 \\ 1 & 3 & 1 & 2 \end{array}\right]$$

Dengan melakukan Operasi Baris Elementer, dapat diperoleh matriks eselon baris, yaitu matriks yang memiliki 1 utama pada setiap baris, kecuali baris yang seluruhnya nol.

$$\left[\begin{array}{ccc|c} 1 & 3 & 1 & 2 \\ 0 & 1 & -7 & 9 \\ 0 & 0 & 1 & -1 \end{array}\right]$$

Setelah diperoleh matriks eselon baris, matriks tersebut dapat diubah ke dalam pesamaan linear kembali menjadi seperti berikut:

$$x + 3y + z = 2$$
$$y - 7z = 9$$
$$z = -1$$

Dengan melakukan *back-subtitution* pada persamaan baru, kita dapat menemukan nilai dari tiap variabel yaitu $x = -3, y = 2 \text{ dan } z = -1.$

B. Eliminasi Gauss-Jordan

Eliminasi Gauss-Jordan merupakan pengembangan dari metode eliminasi Gauss. Eliminasi ini sama seperti eliminasi Gauss yaitu prosedur pemecahan sistem persamaan linear dengan mengubahnya menjadi bentuk matriks. Perbedaannya adalah metode ini membentuk eselon baris tereduksi dengan Operasi Baris Elementer.

$$\left[\begin{array}{ccc|c} 2 & 5 & 3 & 1 \\ 3 & 4 & 2 & -3 \\ 1 & 3 & 1 & 2 \end{array}\right]$$

Dengan melakukan Operasi Baris Elementer, dapat diperoleh matriks eselon baris tereduksi, yaitu matriks yang memiliki 1 utama pada setiap baris, kecuali baris yang seluruhnya nol dan juga memiliki angka 0 di atas dan di bawah 1 utama pada tiap baris.

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & -1 \end{array}\right]$$

Kita dapat menemukan nilai dari tiap variabel yaitu x=–3, y=2 dan z=-1.

C. Determinan

Determinan adalah nilai skalar yang terdapat pada matriks persegi. Misalkan X adalah suatu matriks persegi, determinan dari suatu matriks dilambangkan dengan |X| atau det (X). Tugas besar ini hanya akan menggunakan dua metode untuk mencari nilai suatu determinan, yaitu metode operasi baris dan ekspansi kofaktor.

Metode operasi baris menggunakan operasi baris sehingga terbentuk matriks segitiga bawah. Kemudian, determinannya dapat dicari dengan cara mengalikan semua elemen diagonalnya.

Misal matriks M yaitu

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Dengan operasi baris, akan diperoleh bentuk matriks

$$\begin{bmatrix} x & p & q \\ 0 & y & r \\ 0 & 0 & z \end{bmatrix}$$

Sehingga det(M) adalah x*y*z

Metode ekspansi kofaktor adalah suatu metode dengan cara mengalikan suatu entri dengan minor dari matriksnya.

Misal kita memiliki matriks M yaitu

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Dengan ekspansi kofaktor, kita akan memanfaatkan baris pertama sehingga det(A) adalah $a * Ma - b * Mb + c * Mc$

D. Matriks Balikan

Matriks A yang memiliki ukuran n x n dapat memiliki matriks balikan yaitu A^{-1} yaitu matriks yang memenuhi A A^{-1} = I atau A^{-1}A = I dengan I adalah matriks identitas. Tugas besar ini akan membahas matriks balikan menggunakan adjoinnya.

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

Misal matriks A adalah

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

Dapat dicari $adj(A)$ yaitu

$$\begin{bmatrix} 12 & 6 & -16 \\ 4 & 2 & 16 \\ 12 & -10 & 16 \end{bmatrix}$$

Dapat dicari determinan dari A yaitu

$$det(A) = \ 64$$

Sehingga dapat dicari matriks balikannya yaitu

$$A^{-1} = \frac{1}{64}\begin{bmatrix} 12 & 6 & -16 \\ 4 & 2 & 16 \\ 12 & -10 & 16 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{12}{64} & \frac{6}{64} & -\frac{16}{64} \\ \frac{4}{64} & \frac{2}{64} & \frac{16}{64} \\ \frac{12}{64} & -\frac{10}{64} & \frac{16}{64} \end{bmatrix}$$

E. Matriks Kofaktor

Matriks Kofaktor adalah matriks berisi nilai kofaktor yang diperoleh dari perkalian minor dan penandanya. Kofaktor dari sebuah elemen baris ke-i dan kolom ke-j dari matriks A akan bisa dikenali melalui lambangnya, yaitu Cij. Elemen tiap matriks minor entri aij dilambangkan sebagai Mij sehingga dapat dinyatakan sebagai berikut:

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

Misal matriks A adalah

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

Maka

$$M_{11} = \begin{bmatrix} 2 & -4 \\ 5 & 3 \end{bmatrix} = 26 \qquad C_{11} = (-1)^{1+1} \cdot\ 26 = 26$$
$$M_{12} = \begin{bmatrix} 2 & -4 \\ 1 & 3 \end{bmatrix} = 10 \qquad C_{12} = (-1)^{1+2} \cdot 10 = \ -10$$

dst ...

Sehingga terbentuk matriks kofaktornya yaitu

$$\begin{bmatrix} 26 & -10 & 8 \\ 14 & 5 & -13 \\ 10 & 10 & 10 \end{bmatrix}$$

F. Matriks Adjoin

Matrix adjoin adalah transpose dari matrix kofaktor.
Misalkan Ac adalah matrix kofaktor dari A maka

$$Adj(A) \ = \ Ac^T$$

G. Kaidah Cramer

Kaidah Cramer merupakan cara untuk menyelesaikan sistem persamaan linear multivariabel. Jika Ax = b adalah sistem persamaan linear yang terdiri dari n persamaan linier dengan n variable dan memiliki $det(A) \ = \ 0$, maka sistem persamaan lienar tersebut memiliki solusi unik yang memiliki persamaan sebagai berikut:

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

Aj merupakan matriks yang diperoleh dari mengganti kolom ke – j dari matriks A dengan matriks

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ . \\ b_n \end{bmatrix}$$

H. Interpolasi Polinom

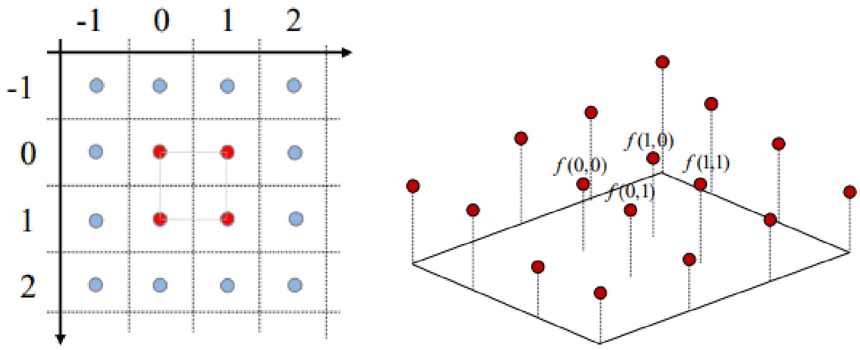
Interpolasi polinom adalah suatu metode untuk memprediksi nilai pada suatu titik yang datanya tidak tersedia dengan mengasumsikan pola data yang diorediksi mengikuti persamaan polinomial. Persamaan polinomial yang terbentuk selanjutnya digunakan untuk memprediksi data yang tidak diketahui. Polinom interpolasi derajat n yang menginterpolasi titik-titik $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ adalah berbentuk $P_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$. Dengan mensubtitusi titik-titik tersebut akan didapat sistem persamaan linear sebagai berikut:

$$
\begin{aligned}
P_n(x_0) &= a_0 + a_1x_0 + a_2{x_0}^2 + \ldots + a_n{x_0}^n = y_0 \\
P_n(x_1) &= a_0 + a_1x_1 + a_2{x_1}^2 + \ldots + a_n{x_1}^n = y_1 \\
P_n(x_2) &= a_0 + a_1x_2 + a_2{x_2}^2 + \ldots + a_n{x_2}^n = y_2 \\
&\vdots \\
P_n(x_n) &= a_0 + a_1x_n + a_2{x_n}^2 + \ldots + a_n{x_n}^n = y_n
\end{aligned}
$$

Dari sistem persamaan linear tersebut dapat diperoleh koefisien dari $a_0, a_1, \dots\ a_n$ yang nilainya dapat disubtitusi ke persamaan $P_n(x)$. Kemudian, $P_n(x)$ dapat digunakan untuk memprediksi nilai baru yang datanya tidak tersedia.

I. Interpolasi Bicubic

Interpolasi bicubic adalah suatu teknik interpolasi data 2D dengan mengambil matriks berukuran 4 x 4 sebagai acuan.



Indeks dari matriks acuan dimulai dari -1 dan diakhiri oleh 2. Nilai dari titik-titik yang akan diprediksi merupakan titik yang berada di dalam daerah berbentuk kotak dengan tiap titik sudutnya adalah f(0, 0), f(1, 1), f(1, 0), dan f(0, 1). Persamaan linier f(x, y) yang digunakan untuk menginterpolasi titik-titik tersebut adalah sebagai berikut:

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

a_{ij} merupakan koefisien a untuk index i dan j. Sistem Persamaan Linier dari persamaan f(x, y) diatas dapat diselesaikan untuk memperoleh seluruh koefisien a. Kemudian, untuk mendapatkan nilai dari suatu titik, substitusikan absis dan ordinat titik ke dalam f(x, y). Interval dari nilai x dan y adalah [0, 1].

J. Regresi Linier Berganda

Regresi linier merupakan model sederhana dengan pendekatan garis linier dengan meminimalkan jumalh kuadrat residual pada data. Model yang terbentuk akan menghasilkan nilai konstanta dan nilai gradien kurva. Regresi linier berganda adalah teknik statistika yang digunakan untuk menganalisis hubungan antara satu variabel dependen dan banyak variabel independen. Metode ini juga digunakan dalam memprediksi nilai dengan variabel independen atau peubah yang banyak. Persamaan umum regresi linier berganda :

$$y_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + \cdots + b_k x_{ki} + \epsilon_i$$

Untuk mendapatkan nilai dari setiap b_i dapat menggunaakan *Normal Estimation Equation for Multiple Linear Regression*.

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

Persamaan ini dapat membantu untuk menentukan nilai $b_0, b_1, b_2, \ldots, b_k$ dengan menggunakan metode dalam emncari solusi Sistem persamaan linier

BAB 3
IMPLEMENTASI

1. Class Matrix

Atribut

Nama	Tipe Data	Parameter	Deskripsi
`row`	`protected int`	-	Menyimpan data jumlah baris pada matrix
`col`	`protected int`	-	Menyampan data jumlah kolom pada matrix
`data`	`public double[][]`	-	Menyimpan data setiap elemen matrix

Method

Nama	Tipe Data	Parameter	Deskripsi
`Matrix`	`Public Matrix`	`Int row` `Int col`	Sebagai Konstruktor untuk membuat matrix kosong

`setELMT`	`Public void`	`Int i`
		`Int j`
		`Double val`
			Sebagai selector elemen matrix

`getELMT`	`public double`	`Int i` `Int j`	Mengambil nilai elemen matrik pada baris ke-i dan kolom ke-j
`getData`	`Public double[][]`	-	Mengambil nilai atribut data
`isMatrixIdxValid`	`Public boolean`	`Int i` `Int j`	Mengembalikan nilai true jika indeks matrix valid
`getLastIdxRow`	`Public int`	-	Mengembalikan nilai indeks baris terakhir
`getLastIdxCol`	`Public int`	-	Mengembalikan nilai indeks kolom terakhir
`getElmtDiagonal`	`Public double`	`Int i`	Mengembalikan elemen matrix dengan indeks (i,i)
`readMatrix`	`Public void`	-	Input elemen matrix melalui terminal
`readMatrixPeubah`	`Public void`	-	Input elemen matrix peubah
`readMatrixHasil`	`Public void`	-	Input elemen matrix hasil
`displayMatrix`	`Public void`	-	Menampilkan matrix pada terminal
`matrixToString`	`Public string`	-	Mengubah tipe data matrix menjadi string
`countElmt`	`Public int`	-	Mengrmbalikan banyak elemen pada matrix
`isSquare`	`Public boolean`	-	Mengembalikan true jika matrix merupakan matrix persegi, vice versa

copyMatrix	Public Matrix	-	Menduplikat matrix
subMatrix	Public Matrix	Int rowDel Int colDel	Membentuk submatrix dengan konfigurasi (m.length-1, m[].length-1)
determinanKof	Public double	-	Mengembalikan nilai determinan matrix yang ditentukan dengan metode ekspansi kofaktor
determinanOBE	Public double	-	Mengembalikan nilai determinan matrix yang ditentukan dengan operasi baris elementer
multiplyByConst	Public void	Double f	Mengalikan setiap elemen matrix dengan suatu konstanta
transposeMatrix	Public void	-	Melakukan operasi transpose pada matrix
inverseMatrix	Public Matrix	-	Mengembalikan inverse dari Matrix
rowZero	Public boolean	Int row	Mengembalikan true jika semua elemen pada indeks baris row adalah 0, vice versa
satuUtamaIdx	Public int	Int row	Mengembalikan indeks satu utama pada baris indeks row
switchCol	Public Matrix	Matrix mCol Int colIdx	Switch suatu kolom dengan matrix tertentu yang memiliki ukuran (row, 1)
isParametricSolution	Public boolean	-	Mengembalikan true jika hasil OBE memiliki solusi SPL parametrik, vice versa
isNoSolution	Public boolean	-	Mengembalikan true jika hasil OBE tidak memiliki solusi SPL, vice versa
isUniqueSolution	Public boolean	-	Mengembalikan true jika hasil OBE memiliki hasil SPL unik, vice versa
multiplyMatrix	Public Matrix	Matrix y	Mengembalikan hasil perkalian matrix dengan matrix y
concatCol	Public Matrix	Matrix m	Mengembalikan matrix dari hasil konkatenasi matrix dengan matrix m
driverDeterminan	Public static void	-	Driver unutk menjalankan fitur determinan matrix
driverInverse	Public static void	-	Driver untuk menjalankan fitur Inverse matrix

2. Class SPL

Method

Nama	Tipe Data	Parameter	Deskripsi
driverSPL	Public static void	-	Driver untuk menjalankan fitur solusi Sistem Persamaan Linier
InverseSPL	Public static Matrix	Matrix augm	Mengembalikan matrix solusi SPL dengan metode matrix balikan
displaySPL	Public static string	Matrix m-sol	Mengembalikan string yang merupakan solusi dari sistem persamaan linier
cramer	Public static Matrix	Matrix augm	Mengembalikan matrix hasil solusi Sistem persaamaan linier dengan metode cramer
makeSatuUtama	Public static void	Matrix augm	Melakukan OBE pada matrix dan memunculkan satu utama di setiap baris
gauss	Public static Matrix	Matrix augm	Mengembalikan matrix hasil eliminasi gauss
gaussJordan	Public static Matrix	Matrix augm	Mengembalikan matrix hasil eliminasi gauss-jordan
SolFormatting	Public static Matrix	Matrix m_sol	Mengembalikan nilai matrix yang merupakan solusi yang sudah diformat

3. Class InterpolasiPolinom

Method

Nama	Tipe Data	Parameter	Deskripsi
interpolPolinom	Public static String	Matrix koordinat Double val	Mengembalikan hasil persamaan interpolasi polinom dan hasil teksirannya dalam tipe data string
driverInterpolPolinom	Public static void	-	Driver untuk menjalankan fitur interpolasi polinom

4. Class Bicubic

Method

Nama	Tipe Data	Parameter	Deskripsi
bicubicInterpolationKoef	Public static Matrix	Matrix m	Mengambalikan matrix koefisien dari interpolasi bicubic

bicubicInterpolationVal	Public static Double	Matrix a_koef Double a Double b	Mengembalikan value dari hasil taksiran interpolasi bicubic
driverBicubic	Public static void	-	Driver untuk menjalankan fotru Interpolasi Bicubic

5. Class ImageResize
Method

Nama	Tipe Data	Paremeter	Deskripsi
imageResize	Public static BufferedImage	BufferedImage img	Mengembalikan image hasil resize gambar dengan mengaplikasikan interpolasi bicubic. Gambar diperbesar menjadi 2 kali lipat ukuran semula
driverImageResize	Public static void	-	Driver untuk menjalankan fitur resize

6. Class Regresi
Method

Nama	Tipe Data	Parameter	Deskripsi
getCoefRegresi	Public static double[]	Double[][]	Mengembalikan list yang merupakan koefisien dari persamaan hasil regresi linier
MultiRegresi	Public static void	-	Driver untuk menjalankan fitur regresi linier

7. Class App
Method

Nama	Tipe Data	Parameter	Deskripsi
main	Public static void	Strings[] args	Driver utama, kompilasi dari setiap driver fitur

8. Class IOFile
Method

Nama	Tipe Data	Parameter	Deskripsi
getRow	Public static int	String fileName	Mengembalikan panjang baris dari matrix pada file

getCol	Public static int	String fileName	Mengembalikan panjang kolom dari matrix pada file
readFileMat	Public static Matrix	String fileName	Mengembalikan matrix dari pembacaan input file
readBcb	Public static Matrix	String fileName	Mengembalikan matrix dari pembacaan input file khusus untuk bicubic
coorBcb	Public static double[]	String fileName	Mengembalikan nilai koordniat unuk mencari taksiran interpolasi bicubic
createEmptyFile	Public static void	String fileName	Membuat file kosong pada directory /test/
writeMatrix	Public static void	String filename Matrix data	Menuliskan matrix ke dalam file
writeString	Public static void	String filename String s	Menuliskan string s ke dalam file
isFileExist	Public statis boolean	String fileName	Mengembalikan nilai true jika file terdapat pada folder test
readImage	Public static BufferedImage	String fileName	Mengemabalikan image bertipe BufferedImage hasil pembacaan citra
writeImageResize	Public static void	BufferedImage result String filename String fileExtension	Menuliskan dan menyimpan citra ke dalam file citra

9. Class UI

Atribute

Nama	Tipe Data	Parameter	Deskripsi
sc	Public static Scanner	-	Membaca input

Method

Nama	Tipe Data	Paraameter	Dekripsi
printMainMenu	Public static void	-	Menampilkan menu pilihan fitur
Pilih	Public static int	Int n	Mengembalikan hasil pilihan user

simpan	Public static void	String output	Menyimpan string pada file

BAB 4
EKSPERIMEN

1. Solusi Sistem Persamaan Linier Ax = B
   a. Menggunakan metode Gauss (file : input-1a.txt)

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

```
1.0000 1.0000 -1.0000 -1.0000 1.0000
0.0000 1.0000 -1.6667 -1.0000 -1.3333
0.0000 0.0000 1.0000 -1.0000 1.0000
0.0000 0.0000 0.0000 0.0000 1.0000

Hasil SPL:
SPL tidak memiliki solusi.
```

   b. Menggunakan metode Gauss-Jordan (file : input-1b.txt)

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

```
Hasil SPL:
x_1 = 1.0000a_0 + 3.0000
x_2 = 2.0000a_0
x_4 = 1.0000a_0 - 1.0000
x_5 = a_0
```

   c. Menggunakan metode Gauss-Jordan (file : input-1c.txt)

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

```
0.0000 1.0000 0.0000 0.0000 0.0000 1.0000 1.0000
0.0000 0.0000 0.0000 1.0000 0.0000 1.0000 -2.0000
0.0000 0.0000 0.0000 0.0000 1.0000 -1.0000 1.0000

Hasil SPL:
x_2 = - 1.0000a_0 + 1.0000
x_4 = - 1.0000a_0 - 2.0000
x_5 = 1.0000a_0 + 1.0000
x_6 = a_0
```

d. Menggunakan metode Gauss-Jordan (file : input-1d-6.txt)

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

```
1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 36.0000
0.0000 1.0000 0.0000 0.0000 0.0000 0.0000 -630.0000
0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 3360.0000
0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 -7560.0000
0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 7560.0000
0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 -2772.0000

Hasil SPL:
x_0 = 36.0000
x_1 = -630.0000
x_2 = 3360.0000
x_3 = -7560.0000
x_4 = 7560.0000
x_5 = -2772.0000
```

e. Menggunakan metode Gauss-Jordan (file : input-1d-10.txt)

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

```
1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 99.9904
0.0000 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 -4949.1596
0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 79181.9853
0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 -600435.4050
0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 2521731.7416
0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000 -6304125.9094
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 9606023.1512
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 -8748135.3476
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 4373977.4702
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 -923378.5099

Hasil SPL:
x_0 = 99.9904
x_1 = -4949.1596
x_2 = 79181.9853
x_3 = -600435.4050
x_4 = 2521731.7416
x_5 = -6304125.9094
x_6 = 9606023.1512
x_7 = -8748135.3476
x_8 = 4373977.4702
x_9 = -923378.5099
```

2. Sistem Persamaan Linier Berbentuk matrix augmented
   a. Menggunakan metode Gauss-Jordan

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

```
Hasil SPL:
x_1 = 1.0000a_1 - 1.0000
x_2 = 2.0000a_0
x_3 = a_0
x_4 = a_1
```

   b. Menggunakan metode Gauss-Jordan

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

```
Hasil SPL:
x_0 = 0.0000
x_1 = 2.0000
x_2 = 1.0000
x_3 = 1.0000
```

3. Sistem Persamaan Linier
    a. Menggunakan metode Gauss-Jordan

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

```
1.0000 0.0000 0.0000 0.0000 -0.2243
0.0000 1.0000 0.0000 0.0000 0.1824
0.0000 0.0000 1.0000 0.0000 0.7095
0.0000 0.0000 0.0000 1.0000 -0.2581

Hasil SPL:
x_0 = -0.2243
x_1 = 0.1824
x_2 = 0.7095
x_3 = -0.2581
```

    b. Menggunakan metode Gauss-Jordan

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

```
Masukkan nama file
>> input-3b

Hasil SPL:
SPL tidak memiliki solusi.
```

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

x	0.4	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

Lakukan pengujian pada nilai-nilai default berikut:

$$x = 0.2 \rightarrow f(x) = 0.12999999999998382$$

```
f(x) = - -0.1845590191298605 + 10.276383988581529x^1 - 163.91566260202262x^2
+ 1220.8548905938487x^3 - 4346.3139507523465x^4 + 7102.399162436538x^5 - 4212.434531756722x^6

f(0.2) = - -0.1845590191298605 + 10.276383988581529*0.2 - 163.91566260202262*0.04000000000000001
+ 1220.8548905938487*0.008000000000000002 - 4346.3139507523465*0.0016000000000000003 +
7102.399162436538*3.2000000000000001E-4 - 4212.434531756722*6.400000000000002E-5

f(0.2) = 0.1299999999998382
```

$$x = 0.55 \rightarrow f(x) = 2.1375716208394806$$

```
f(x) = - -0.1845590191298605 + 10.276383988581529x^1 - 163.91566260202262x^2 + 1220.8548905938487x^3
- 4346.3139507523465x^4 + 7102.399162436538x^5 - 4212.434531756722x^6

f(0.55) = - -0.1845590191298605 + 10.276383988581529*0.55 - 163.91566260202262*0.30250000000000005
+ 1220.8548905938487*0.16637500000000005 - 4346.3139507523465*0.09150625000000003
+ 7102.399162436538*0.05032843750000002 - 4212.434531756722*0.027680640625000013

f(0.55) = 2.1375716208394806
```

$$x = 0.85 \rightarrow f(x) = -66.26963931319551$$

```
f(x) = -0.1845590191298605 + 10.276383988581529x^1 - 163.91566260202262x^2 + 1220.8548905938487x^3
- 4346.3139507523465x^4 + 7102.399162436538x^5 - 4212.434531756722x^6

f(0.85) = -0.1845590191298605 + 10.276383988581529*0.85 - 163.91566260202262*0.7224999999999999
+ 1220.8548905938487*0.6141249999999999 - 4346.3139507523465*0.5220062499999999
+ 7102.399162436538*0.44370531249999995 - 4212.434531756722*0.37714951562499993

f(0.85) = -66.26963931319551
```

$$x = 1.28 \rightarrow f(x) = -3485.144901500389$$

```
f(x) =  -0.1845590191298605 + 10.276383988581529x^1 - 163.91566260202262x^2 + 1220.8548905938487x^3
- 4346.3139507523465x^4 + 7102.399162436538x^5 - 4212.434531756722x^6

f(1.28) =  -0.1845590191298605 + 10.276383988581529*1.28 - 163.91566260202262*1.6384
+ 1220.8548905938487*2.0971520000000003 - 4346.3139507523465*2.68435456
+ 7102.399162436538*3.4359738368000006 - 4212.434531756722*4.398046511104001

f(1.28) = -3485.144901500389
```

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

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

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

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

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

1) 16/07/2022 $\rightarrow 7 + \left(\frac{16}{31}\right) = 0.516129$

```
f(x) = 7.187066071658637E12 - 9.346993079172963E12x^1 + 5.334203055240283E12x^2 - 1.756810186
3613738E12x^3 + 3.685508071755316E11x^4 - 5.1131876760132576E10x^5 + 4.695806315428787E9x^6 -
 2.7547453942066926E8x^7 + 9372849.23910132x^8 - 140993.71224863594x^9
f(0.516129) = 7.187066071658637E12 - 9.346993079172963E12*0.516129 + 5.334203055240283E12*0.2
6638914464099994 - 1.7568101863613738E12*0.13749116283441465 + 3.685508071755316E11*0.0709631
763825636 - 5.1131876760132576E10*0.03662615326315616 + 4.695806315428787E9*0.018903819857559
522 - 2.7547453942066926E8*0.00975680963926234 + 9372849.23910132*0.005035772402302831 - 1409
93.71224863594*0.0025991081742281577
f(0.516129) = 3.566606694260665E12
```

2) 10/08/2022 $\rightarrow 8 + \left(\frac{10}{31}\right) = 8.322581$

```
f(x) = 7.187066071658637E12 - 9.346993079172963E12x^1 + 5.334203055240283E12x^2 - 1.756810186
3613738E12x^3 + 3.685508071755316E11x^4 - 5.1131876760132576E10x^5 + 4.695806315428787E9x^6 -
 2.7547453942066926E8x^7 + 9372849.23910132x^8 - 140993.71224863594x^9
f(8.322581) = 7.187066071658637E12 - 9.346993079172963E12*8.322581 + 5.334203055240283E12*69.
26535450156099 - 1.7568101863613738E12*576.466523332956 + 3.685508071755316E11*4797.689334226
916 - 5.1131876760132576E10*39929.15809693957 + 4.695806315428787E9*332313.6525235855 - 2.754
7453942066926E8*2765707.290533394 + 9372849.23910132*2.3017822947754707E7 - 140993.7122486359
4*1.915676959263473E8
f(8.322581) = 36315.90234375
```

3) 05/09/2022 $\rightarrow 9 + \left(\frac{5}{30}\right) = 0.322580$

```
f(x) = 7.187066071658637E12 - 9.346993079172963E12x^1 + 5.334203055240283E12x^2 - 1.756810186
3613738E12x^3 + 3.685508071755316E11x^4 - 5.1131876760132576E10x^5 + 4.695806315428787E9x^6 -
 2.7547453942066926E8x^7 + 9372849.23910132x^8 - 140993.71224863594x^9
f(0.32258) = 7.187066071658637E12 - 9.346993079172963E12*0.32258 + 5.334203055240283E12*0.104
05785639999998 - 1.7568101863613738E12*0.03356698331751199 + 3.685508071755316E11*0.010828037
478563018 - 5.1131876760132576E10*0.003492908329834858 + 4.695806315428787E9*0.00112674236903
81284 - 2.7547453942066926E8*3.6346455340431947E-4 + 9372849.23910132*1.1724639563716536E-4 -
 140993.71224863594*3.78213423046368E-5
f(0.32258) = 4.671825236418928E12
```

4) 28/02/2022 $\rightarrow 2 + \left(\frac{28}{28}\right) = 3$

```
f(x) = 7.187066071658637E12 - 9.346993079172963E12x^1 + 5.334203055240283E12x^2 - 1.756810186
3613738E12x^3 + 3.685508071755316E11x^4 - 5.1131876760132576E10x^5 + 4.695806315428787E9x^6 -
 2.7547453942066926E8x^7 + 9372849.23910132x^8 - 140993.71224863594x^9
f(3.0) = 7.187066071658637E12 - 9.346993079172963E12*3.0 + 5.334203055240283E12*9.0 - 1.75681
01863613738E12*27.0 + 3.685508071755316E11*81.0 - 5.1131876760132576E10*243.0 + 4.69580631542
8787E9*729.0 - 2.7547453942066926E8*2187.0 + 9372849.23910132*6561.0 - 140993.71224863594*196
83.0
f(3.0) = 2.710869890518106E10
```

c. Sederhanakan fungsi

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

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

```
f(x) = 0.0 + 2.0352567500000065x^1 - 3.5526791666666973x^2 + 3.2371100260417136x^3 - 1.421263
0208333623x^4 + 0.2362556966145896x^5
```

5. Studi kasus Interpolasi Bicubic

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

Tentukan nilai:

f(0,0) = ?

f(0.5, 0.5) = ?

f(0.25, 0.75) = ?

f(0.1, 0.9) = ?

a. $f(0,0) = 161$ (file : input-5a.txt)

```
Masukkan nama file
>> input-5a

Nilai f(0.00, 0.00) = 161.0000
```

b. $f(0.5, 0.5) = 98.9922$ (file : input-5b.txt)

```
Masukkan nama file
>> input-5b

Nilai f(0.50, 0.50) = 98.9922
```

c. $f(0.25, 0.75) =$ 82.8866 (file : input-5c.txt)

```
Masukkan nama file
>> input-5c

Nilai f(0.25, 0.75) = 82.8866
```

d. $f(0.1, 0.9) = 74.7582$ (file : input-5d.txt)

```
Masukkan nama file
>> input-5d

Nilai f(0.10, 0.90) = 74.7582
```

6. Studi kasus Regresi Linier
Diberikan sekumpulan data sesuai pada tabel berikut ini

Table 12.1: Data for Example 12.1

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

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

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

```
Bentuk Normal Estimation Equation untuk Regresi Linier Barganda :

20.0000 863.1000 1530.4000 587.8400 19.4200
863.1000 54876.8900 67000.0900 25283.3950 779.4770
1530.4000 67000.0900 117912.3200 44976.8670 1483.4370
587.8400 25283.3950 44976.8670 17278.5086 571.1219

Hasil persamaan Regresi Linier Berganda :

y = -3.507778  -0.002625 x1 + 0.000799 x2 + 0.154155 x3

Menaksir nilai dari fungsi Reegresi Linier
Masukkan 3 peubah yang akan ditaksir nilai fungsinya
>> 50 76 29.3
Nilai taksirannya adalah 0.955461
```

7. **Pembesaran Citra**
   Pembesaran citra menjadi 2 kali dari ukuran awal. Dilakukan dengan citra ber-format .jpg, .png, .bmp.

BAB 5
KESIMPULAN, SARAN, DAN REFLEKSI

A. KESIMPULAN

Hasil program yang kami buat dapat digunakan untuk :

1. Menetukan solusi Sistem Persamaan Linier mengunakan metode eliminasi Gauss, eliminasi Gauss-Jordan untuk matriks augmented non persegi dan persegi.
2. Menetukan solusi Sistem Persamaan Linier menggunakan metode matriks balikan dan kaidah cramer khusus untuk matriks persegi.
3. Menghitung determinan matriks dengan menggunakan metode reduksi baris dan ekspansi kofaktor.
4. Menentukan matriks balikan.
5. Menentukan persamaan dan taksiran dari interpolasi polinom, interpolasi bicubic, dan regresi linier berganda.
6. Membesarkan ukuran citra dengan aplikasi interpolasi bicubic.

B. SARAN

Penulis menyadari bahwa dalam pengerjaan dan pembuatan program dalam tugas besar ini dapat dikembangkan lebih baik lagi. Hal-hal yang dapat dikembangkan menjadi lebih baik lagi adalah sebagai berikut:

1. Dalam pembuatan suatu spesifikasi program, lebih baik untuk melakukan dekomposisi persoalan yang mendalam dan menyeluruh sebelum memulai membuatnya dalam bentuk *code*. Hal ini dilakukan supaya program yang dibuat menjadi lebih mudah dalam proses *debugging* dan pembacaan serta meningkatkan reusability dari prosedur atau fungsi yang dibuat.
2. Setiap fungsi atau prosedur lebih baik dituliskan beserta dengan komen singkat untuk menjelaskan apa kegunaan dari fungsi atau prosedur tersebut. Selain itu, penulisan langkah-langkah singkat apa yang dilakukan fungsi atau prosedur juga sangat diperlukan. Hal tersebut berguna untuk memudahkan dalam hal debugging dan juga pembacaan.

C. REFLEKSI

Penulis menemukan berbagai pelajaran berharga dalam pembuatan tugas besar ini. Dari sisi nonteknis penulis belajar untuk bekerja sama dalam menyelesaikan pekerjaan sehingga pekerjaan tersebut dapat selesai dengan cepat. Pembagian tugas masing-masing individu juga penting supaya jelas siapa yang menegrejakan suatu bagian dan juga dapat mempertanggungjawabkan bagian tersebut. Selain itu, penulis juga menyadari pentingnya untuk tidak menunda-nunda suatu pekerjaan. Dari sisi teknis, penulis mendapat pelajaran berharga yaitu dapat memahami dan menggunakan bahasa pemrograman baru yaitu java. Pelajaran mengenai c*ontrol-flow* dari pembuatan suatu program juga didapat dalam menggunakan git dan github.

REFERENSI

Munir, R. (2022, September 13). *Aljabar Geometri.* Retrieved from Homepage Rinaldi Munir: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-09-Determinan-bagian2.pdf

Profematika. (2019, March 23). *Eliminasi Gauss Jordan beserta contoh Penerapannya*. Retrieved from Profematika: https://www.profematika.com/eliminasi-gauss-jordan-beserta-contoh-penerapannya/

Rosidi, M. (2019, December 23). *Chapter 12 Pemodelan Data: Regresi Linier*. Retrieved from Matode Numerik Menggunakan R untuk Teknik Lingkungan : https://bookdown.org/moh_rosidi2610/Metode_Numerik/

Rowe, D. B. (2018, February 15). *Bilinear, Bicubic, and In Between Spline Interpolation.* Retrieved from Marquette University, Mathematical and Statistical Sciences Department Website: https://www.mssc.mu.edu/~daniel/pubs/RoweTalkMSCS_BiCubic.pdf

Wapole, R. E., Myers, R. H., Myers, S. L., & Ye, K. (2012). *Probability & Staatistics for Engineers & Scientist* (9th ed.). Boston: Pearson Eduation.

LAMPIRAN

Github : https://github.com/rizkyrsyd28/Algeo01-21109