LAPORAN TUGAS BESAR I
ALJABAR LINIER DAN GEOMETRI
SISTEM PERSAMAAN LINIER, DETERMINAN, DAN APLIKASINYA

Untuk memenuhi Tugas Besar Mata Kuliah IF 2123 Aljabar Linier dan Geometri

Disusun oleh :

Rinaldy Adin (13521134)
Enrique Alifio Ditya (13521142)
Rava Maulana Azzikri (13521149)

Dosen Pembimbing : Dr. Ir. Rinaldi Munir, M.T.

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

2022/2023

BAB I
Deskripsi Masalah

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer (khusus untuk SPL dengan n peubah dan n persamaan). Solusi dari sebuah SPL mungkin tidak ada, banyak (tidak berhingga), atau hanya satu (unik).

Penyelesaian SPL untuk sembarang jumlah peubah terkait erat dengan manipulasi dan pengoperasian terhadap suatu matriks. Di antara operasinya adalah mencari determinan dari suatu matriks, mencari balikannya jika matriks memiliki determinan bukan nol, serta pemberlakuan aturan Operasi Barisan Elementer untuk mendapatkan *Reduced Echelon Form* dari suatu matriks. Operasi-operasi tersebut juga dapat diperoleh dengan berbagai metode, misalnya metode Gauss dan Gauss-Jordan untuk mendapatkan matriks eselon baris dan eselon baris tereduksi, metode reduksi baris dan kofaktor untuk mencari determinan, serta metode reduksi baris dan Adjoin untuk mencari matriks balikan.

Berbagai aplikasi dalam kehidupan nyata dari metode-metode penyelesaian Sistem Persamaan Linier, antara lain Interpolasi Polinom, Interpolasi Bikubik, dan Regresi Linier Berganda. Pengaplikasian ini turutlah berguna terutama dalam pengaproksimasian suatu titik/nilai dari sejumlah sampel data yang dimiliki. Untuk Regresi Linier Berganda, misalnya, seringkali digunakan dalam algoritma pemodelan *Machine Learning*, serta Interpolasi Bikubik digunakan dalam pembesaran citra atau *image scaling*.

Pada Tugas Besar ini, dibuatlah library Aljabar Linier dalam Bahasa Java. Library tersebut berisi fungsi-fungsi seperti eliminasi Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung determinan, dan kaidah Cramer. Selanjutnya, dibuat program yang memanfaatkan *library* tersebut untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, seperti menyelesaikan persoalan interpolasi dan persoalan regresi.

BAB II

Teori Singkat

2.1 Sistem Persamaan Linier

Sistem Persamaan Linier dalam matematika merupakan perkumpulan dari satu atau lebih persamaan linier yang mengandung variabel yang sama. Teori Sistem Persamaan Linier merupakan landasan utama dalam Aljabar Linier, yakni suatu cabang matematika yang memiliki peran penting di dunia matematika modern, rekayasa, dan komputasi. Sistem persamaan non-linier seringkali dapat dihampiri dengan sistem linier, suatu teknik yang membantu ketika membuat model matematika atau simulasi komputer dari sistem yang relatif kompleks.

2.2 Determinan

Determinan dari sebuah matriks merupakan sebuah bilangan khusus yang digunakan untuk mendeskripsikan matriks tersebut. Determinan dari sebuah matriks dapat dihitung hanya jika matriks tersebut merupakan matriks persegi. Jika sebuah matriks memiliki determinan yang bukan 0, matriks tersebut memiliki balikan (*invertible*). Determinan dari sebuah matriks dapat digunakan untuk menyelesaikan sistem persamaan linear, menghasilkan matriks balikan dari sebuah matriks, dan menentukan persamaan karakteristik dari sebuah matriks. Untuk menghitung determinan dari sebuah matriks, dapat digunakan metode reduksi baris dan metode ekspansi kofaktor.

2.2.1 Metode Reduksi Baris

Determinan dari sebuah matriks dapat dihitung dengan melakukan operasi baris elementer hingga menghasilkan sebuah matriks segitiga. Determinan dari matriks segitiga dapat dihitung dengan mengalikan semua elemen-elemen pada diagonal utamanya. Terdapat tiga aturan dalam metode reduksi baris untuk menghitung determinan,

1. Jika sebuah baris dibagi dengan skalar k, nilai determinan dikalikan dengan k.
2. Jika dua buah baris ditukar posisinya, nilai determinan dikalikan dengan -1.
3. Jika sebuah baris ditambah k kali baris lainnya, nilai determinan tidak berubah.

2.2.2 Metode Ekspansi Kofaktor

Determinan dari sebuah matriks 2x2 dapat dihitung dengan,

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

Jika A merupakan sebuah matriks, minor a_{ij} (M_{ij}) didefinisikan sebagai submatrix yang dibentuk jika baris ke-i dan kolom ke-j dihilangkan dari A. Kofaktor dari elemen a_{ij} adalah $(-1)^{i+j}M_{ij}$. Determinan dari sebuah matriks dapat hitung dengan menjumlahkan kofaktor dari elemen-elemen pada sebuah kolom atau sebuah baris pada matriks tersebut. Dengan metode ini, determinan dari matriks NxN dapat dihitung dengan melakukan ekspansi kofaktor secara rekursif hingga mencapai matriks 2x2.

2.3 Matriks Balikan

Pada Aljabar Linier, suatu matriks persegi **A** dikatakan memiliki balikan jika terdapat matriks persegi **B** sehingga

$$\mathbf{AB} = \mathbf{BA} = \mathbf{I}_n$$

dengan **I** menyatakan matriks identitas. Suatu matriks memiliki balikan jika dan hanya jika merupakan matriks persegi dan bukanlah matriks singular, yakni memiliki determinan tidak sama dengan nol. Jika kondisi tersebut dipenuhi, maka matriks **B** dapat dikatakan Invers dari **A**, yakni $\mathbf{B} = \mathbf{A}^{-1}$.

2.4 Interpolasi Polinom

Matriks dapat digunakan untuk mengaproksimasi nilai sebuah fungsi jika hanya diketahui beberapa titik. Jika terdapat n buah titik yang diketahui, dapat ditentukan sebuah polinom P(x) berderajat n-1 yang melalui semua titik tersebut. Koefisien-koefisien dari polinom tersebut dapat ditentukan dengan membuat n buah persamaan dan mensubstitusikan n buah titik yang diketahui pada persamaan tersebut. Persoalan menentukan koefisien-koefisien dari polinom interpolasi menjadi persoalan menemukan solusi dari sebuah sistem persamaan linear yang dapat dihitung dengan menggunakan matriks. Jika koefisien-koefisien polinom interpolasi sudah diketahui, nilai sebuah fungsi dapat dihitung dengan menggunakan polinom interpolasi tersebut.

2.5 Interpolasi Bicubic

Interpolasi bicubic merupakan sebuah metode interpolasi yang dapat dilakukan pada sebuah data dua dimensi. Fungsi interpolasi dapat ditentukan dengan menentukan empat titik yang akan digunakan sebagai normalisasi dan dua belas titik di sekitarnya untuk membuat model fungsi,

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

Titik-titik yang diketahui disubtitusikan ke dalam model tersebut untuk menentukan koefisien-koefisien dari fungsi interpolasi. Dalam menentukan nilai-nilai koefisien, dapat digunakan metode penyelesaian sistem persamaan linear menggunakan matriks. Setelah nilai dari koefisien-koefisien fungsi interpolasi ditentukan, fungsi tersebut dapat digunakan untuk mengaproksimasi nilai dari sebuah titik yang berada pada daerah normalisasi.

2.6 Regresi Linier Berganda

Regresi Linier Berganda merupakan suatu metode regresi yang digunakan untuk mendeskripsikan relasi antara suatu variabel dependen dengan lebih dari satu variabel independen. Hal ini dilakukan dengan menyesuaikan suatu garis lurus pada sampel data yang diamati. Metode regresi memungkinkan untuk mendapatkan estimasi mengenai perilaku perubahan variabel dependen/terikat terhadap variabel independen/bebas. Regresi linier berganda dapat digunakan untuk mencari keterkaitan antara variabel dependen dan independen serta hampiran dari suatu nilai variabel dependen apabila diberikan suatu nilai variabel independen.

Pemodelan dari Regresi Linier Berganda menghasilkan persamaan mengikuti bentuk umum berikut.

$$\boxed{Y=\beta_0+\beta_1X_1+\beta_2X_2+\ \dots+\beta_iX_i}$$

Y : Dependent variable
β_0 : Intercept
β_i : Slope for X_i
X = Independent variable

Untuk mendapatkan nilai dari setiap βi dapat digunakan *Normal Estimation Equation for Multiple Linear Regression.*

$$
\begin{array}{rcl}
n\hat{\beta}_0 + \hat{\beta}_1 \sum_{i=1}^{n} x_{i1} + \hat{\beta}_2 \sum_{i=1}^{n} x_{i2} + \cdots + \hat{\beta}_k \sum_{i=1}^{n} x_{ik} & = & \sum_{i=1}^{n} y_i \\
\hat{\beta}_0 \sum_{i=1}^{n} x_{i1} + \hat{\beta}_1 \sum_{i=1}^{n} x_{i1}^2 + \hat{\beta}_2 \sum_{i=1}^{n} x_{i1}x_{i2} + \cdots + \hat{\beta}_k \sum_{i=1}^{n} x_{i1}x_{ik} & = & \sum_{i=1}^{n} x_{i1}y_i \\
\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots & & \\
\hat{\beta}_0 \sum_{i=1}^{n} x_{ik} + \hat{\beta}_1 \sum_{i=1}^{n} x_{ik}x_{i1} + \hat{\beta}_2 \sum_{i=1}^{n} x_{ik}x_{i2} + \cdots + \hat{\beta}_k \sum_{i=1}^{n} x_{ik}^2 & = & \sum_{i=1}^{n} x_{ik}y_i
\end{array}
$$

Sistem persamaan linier tersebut dapat diselesaikan dengan menggunakan metode eliminasi Gauss.

BAB III
Implementasi

3.1 Class Matrix

Attribute

Nama	Tipe	Deskripsi
`data`	`public double[][]`	Elemen-elemen dari matriks.
`nRows`	`protected int`	Jumlah baris.
`nCols`	`protected int`	Jumlah kolom.

Method

Nama	Tipe	Parameter	Deskripsi
`Matrix`	`public constructor`	`int nRows, int nCols`	Membuat matriks baru yang kosong.
`Matrix`	`public constructor`	`double[][] data`	Membuat matriks baru yang terisi dengan data.
`getNRows`	`public int`	`void`	Selektor jumlah baris.
`getNCols`	`public int`	`void`	Selektor jumlah kolom.
`display`	`public void`	`void`	Menampilkan matriks.
`getRowAsArray`	`public double[]`	`int R1`	Menghasilkan sebuah array yang berisi baris R1.
`getColAsArray`	`public double[]`	`int C1`	Menghasilkan sebuah array yang berisi kolom C1.
`getDiagonalAsArray`	`public double[]`	`void`	Menghasilkan sebuah array yang berisi elemen-elemen pada diagonal utama.
`multiplyRowByK`	`public void`	`int R1, double k`	Mengalikan baris R1 dengan k.

divideRowByK	public void	int R1, double k	Membagi baris R1 dengan k.
subtractRowByArray	public void	int R1, double[] arr	Mengurangkan baris R1 dengan elemen pada arr.
createIdentityMatrix	public static Matrix	int n	Menghasilkan matriks identitas berukuran nxn.
transpose	public static Matrix	Matrix matrix	Menghasilkan matriks yang sudah di-transpose.
copyMatrix	public static Matrix	Matrix matrix	Menghasilkan salinan dari sebuah matriks.
multiplyMatrix	public static Matrix	Matrix m1, Matrix m2	Menghasilkan matriks baru hasil perkalian dua buah matriks.
makeAugmented	public static Matrix	Matrix a, Matrix b	Menghasilkan matriks augmented dari matriks a dan b.
swapRow	public void	int R1, int R2	Menukar baris R1 dengan R2.

3.2 Class Util

Method

Nama	Tipe	Parameter	Deskripsi
indexOfVal	public static int	int[] arr, int val	Menghasilkan indeks dari val.
indexOfVal	public static int	double[] arr, double val	Menghasilkan indeks dari val.
parseElmt	public static double	String str	Fungsi antara untuk membaca file.
readFromFile	public static double[][]	String pathString	Membaca matriks dari sebuah file.
writeToFile	public static void	String fileName, String outputString	Menuliskan sebuah string ke dalam file.

3.3 Class LinearEquation

Method

Nama	Tipe	Parameter	Deskripsi
`gaussianElimination`	`public static Matrix`	`Matrix matrixIn`	Menghasilkan matriks augmented hasil eliminasi Gauss.
`solveRowEchelon`	`public static String[]`	`Matrix matrix`	Fungsi antara untuk menghasilkan solusi SPL hasil eliminasi Gauss.
`solveGauss`	`public static String[]`	`Matrix matrixIn`	Menghasilkan solusi SPL hasil eliminasi Gauss.
`gaussJordanElimination`	`public static Matrix`	`Matrix matrixIn`	Menghasilkan matriks augmented hasil eliminasi Gauss-Jordan.
`solveReducedRowEchelon`	`public static String[]`	`Matrix matrixIn`	Fungsi antara untuk menghasilkan solusi SPL hasil eliminasi Gauss-Jordan.
`solveGaussJordan`	`public static String[]`	`Matrix matrixIn`	Menghasilkan solusi SPL hasil eliminasi Gauss-Jordan.
`cramerRule`	`public static Matrix`	`Matrix matrixIn`	Menghasilkan matriks solusi hasil penggunaan aturan Cramer.
`solveLinearWithInverse`	`public static Matrix`	`Matrix coefficient, Matrix constant`	Menghasilkan solusi SPL menggunakan matriks balikan.

3.4 Class Determinant

Method

Nama	Tipe	Parameter	Deskripsi
`rowReduction`	`public static`	`Matrix`	Menghitung

	double	matrixIn	determinan menggunakan reduksi baris.
cofactor	public static double	Matrix matrixIn	Menghitung determinan menggunakan metode kofaktor.

3.5 Class Inverse

Method

Nama	Tipe	Parameter	Deskripsi
gaussJordanInverse	public static Matrix	Matrix matrix	Menghasilkan matriks balikan menggunakan metode Gauss-Jordan.
adjoinMethod	public static Matrix	Matrix matrix	Menghasilkan matriks balikan menggunakan matriks adjoin.

3.6 Class Interpolation

Method

Nama	Tipe	Parameter	Deskripsi
polynomialInterpolation	public static Matrix	Matrix points	Menghasilkan matriks koefisien dari polinomial interpolasi.
approximateFunction	public static double	Matrix points, double x	Mengaproksimasi nilai fungsi dari sebuah titik menggunakan polinomial interpolasi

3.7 Class BicubicInterpolation

Method

Nama	Tipe	Parameter	Deskripsi

generateVariablesMatrix	private static Matrix	void	Menghasilkan matrix X dalam persamaan y=Xa
interpolate	public static double	Matrix values, double x, double y	Mengaproksimasi nilai fungsi dari sebuah titik yang berada pada daerah normalisasi
scaleImage	public static void	String inputPath, String outputFileName	Melakukan perbesaran citra dengan skala 2.

3.8 Class LinearRegression

<u>Method</u>

Nama	Tipe	Parameter	Deskripsi
getMtxA	public static Matrix	Matrix x	Mendapatkan matriks koefisien dari Sistem Persamaan Linier *Normal Estimation Equation.*
getMtxH	public static Matrix	Matrix x, Matrix y	Mendapatkan matriks kolom dari konstanta Sistem Persamaan Linier *Normal Estimation Equation.*
multipleLinearReg	public static Matrix	Matrix x, Matrix y	Mendapatkan β_i dari Sistem Persamaan Linier *Normal Estimation Equation*.
writeMLREquation	public static String	Matrix x, Matrix y	Mengembalikan string dari persamaan Regresi Linier Berganda untuk dicetak pada program utama.
approxMLR	public static double	Matrix x, Matrix y,	Melakukan hampiran dari

		`Matrix k`	sembarang nilai variabel peubah.

BAB IV

Eksperimen

4.1 Solusi SPL Ax = b

a)
$$A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix} \qquad b = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$$
`X1 = Tidak ada solusi, X2 = Tidak ada solusi, X3 = Tidak ada solusi, X4 = Tidak ada solusi`
b)
$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix} \qquad b = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$$
`X1 = 3.0 + e, X2 = 2.0e, X3 = c, X4 = -1.0 + e, X5 = e`
c)

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

```
X1 = a, X2 = 1.0 - f, X3 = c, X4 = -2.0 - f, X5 = 1.0 + f, X6 = f
```

d1)

Hilbert Matrix, n = 6,

```
1 1/2 1/3 1/4 1/5 1/6 1
1/2 1/3 1/4 1/5 1/6 1/7 0
1/3 1/4 1/5 1/6 1/7 1/8 0
1/4 1/5 1/6 1/7 1/8 1/9 0
1/5 1/6 1/7 1/8 1/9 1/10 0
1/6 1/7 1/8 1/9 1/10 1/11 0
```

$$= \quad b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

```
X1 = 36.000000001176026, X2 = -630.0000000209296, X3 = 3360.0000001866633,
X4 = -7560.000000502329, X5 = 7560.000000563853, X6 = -2772.0000002246784
```

d2)

Hilbert Matrix, n = 10

```
1 1/2 1/3 1/4 1/5 1/6 1/7 1/8 1/9 1/10 1
1/2 1/3 1/4 1/5 1/6 1/7 1/8 1/9 1/10 1/11 0
1/3 1/4 1/5 1/6 1/7 1/8 1/9 1/10 1/11 1/12 0
1/4 1/5 1/6 1/7 1/8 1/9 1/10 1/11 1/12 1/13 0
1/5 1/6 1/7 1/8 1/9 1/10 1/11 1/12 1/13 1/14 0
1/6 1/7 1/8 1/9 1/10 1/11 1/12 1/13 1/14 1/15 0
1/7 1/8 1/9 1/10 1/11 1/12 1/13 1/14 1/15 1/16 0
1/8 1/9 1/10 1/11 1/12 1/13 1/14 1/15 1/16 1/17 0
1/9 1/10 1/11 1/12 1/13 1/14 1/15 1/16 1/17 1/18 0
1/10 1/11 1/12 1/13 1/14 1/15 1/16 1/17 1/18 1/19 0
```

$$= \quad b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

```
X1 = 100.0, X2 = -4949.8, X3 = 79195.67, X4 = -600560.53,
X5 = 2522331.25, X6 = -6305780.06, X7 = 9608745.47, X8 = -8750772.98,
X9 = 4375365.28, X10 = -923684.29
```

4.2 Solusi SPL matrix augmented

a)

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

```
X1 = -1.0 + d, X2 = 2.0c, X3 = c, X4 = d
```

b)

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

```
X1 = 0, X2 = 2.0, X3 = 1.0, X4 = 1.0
```

4.3 SPL berbentuk

a)

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

```
X1 = -0.22, X2 = 0.18, X3 = 0.71, X4 = -0.26
```

b)

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

```
X1 = Tidak ada solusi, X2 = Tidak ada solusi, X3 = Tidak ada solusi, X4 = Tidak ada solusi,
X5 = Tidak ada solusi, X6 = Tidak ada solusi, X7 = Tidak ada solusi, X8 = Tidak ada solusi, X9 = Tidak ada solusi
```

4.4 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:

Nilai	Taksiran
x = 0.2	`f(0.2) = 0.12999999999982742`
x = 0.55	`f(0.55) = 2.1375716209044953`
x = 0.85	`f(0.85) = -66.26963931520504`
x = 1.28	`f(1.28) = -3485.1449016082806`

Diperoleh persamaan,

```
f(x) = -0.18455901915660444x^6 + 10.276383989371126x^5 - 163.91566261121386x^4
+ 1220.8548906474787x^3 - 4346.313950916225x^2 + 7102.3991626818315x^1
```

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:

Tanggal	Tanggal (desimal)	Taksiran
16/07/2022	7,516	`f(7.516) = 53536.667968`
10/8/2022	8,322	`f(8.322) = 36341.929687`
10/9/2022	9,333	`f(9.333) = -3789846.093`
29/8/2022	8,935	`f(8.935) = 48647.976562`

Diperoleh persamaan,

```
f(x) = 7.187117988941859E12x^9 - 9.347057986137934E
12x^8 + 5.334238927153895E12x^7 - 1.756821693899884
E12x^6 + 3.6855316942098285E11x^5 - 5.1132198648868
91E10x^4 + 4.695835438374229E9x^3 - 2.7547622680787
98E8x^2 + 9372906.062802723x^1 - 140994.5597747811x
```

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.

Dengan n = 6, diperoleh taksiran

x	0	0.4	0.8	1.2	1.6	2
f(x)	`f(0.0) = 0.0`	`f(0.4) = 0.41888`	`f(0.8) = 0.5071`	`f(1.2) = 0.5609`	`f(1.6) = 0.5836`	`f(2.0) = 0.2863`

Diperoleh persamaan,

```
f(x) = 0.0x^4 + 1.8901012500000007x^3 - 2.796660937500
002x^2 + 1.9140781250000019x^1 - 0.4762402343750005x,
```

4.5 Studi kasus interpolasi bicubic

Dengan matriks input:

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

Tentukan nilai:

Nilai	Taksiran
f(0, 0)	`f(0.0, 0.0) = 161.0`
f(0.5, 0.5)	`f(0.25, 0.75) = 105.52758789062501`
f(0.25, 0.75)	`f(0.25, 0.75) = 105.52758789062501`
f(0.1, 0.9)	`f(0.1, 0.9) = 104.23194`

4.6 Studi Kasus Regresi Linier Berganda

Diberikan sekumpulan data sesuai pada tabel berikut ini.

Humidity (x_1)	Temp. (x_2)	Pressure (x_3)	Nitrous Oxide (y)
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

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.

```
Persamaan regresi:
f(X1, X2, X3) = -3.508 - 0.003 X1 + 0.001 X2 + 0.154 X3

Hasil taksiran Xk:
f(50.0, 76.0, 29.3) = 0.938434226221665
```

Diperoleh persamaan
$f(x_1, x_2, x_3) = -3.508 - 0.003\ x_1 + 0.001\ x_2 + 0.154\ x_3$

Dan diperoleh hasil
$f(50.0, 76.0, 29.3) = 0.938434226221665$

BAB V
Kesimpulan dan Saran

5.1 Kesimpulan

Pada tugas besar ini, kami membuat program yang dapat menyelesaikan berbagai masalah komputasi yang membutuhkan matriks, seperti persoalan interpolasi, regresi, dan perbesaran citre. Untuk menyelesaikan permasalahan tersebut, kami membuat operasi-operasi dasar pada matriks seperti perkalian suatu kolom dengan skalar, perkalian dua buah matriks, dan mentranspose matriks. Selain itu, kami juga membuat operasi-operasi lanjutan pada matriks seperti metode eliminasi Gauss-Jordan untuk mencari solusi sistem persamaan linear, metode reduksi baris untuk menghitung determinan, dan metode matriks adjoin untuk menghasilkan matriks balikan. Operasi-operasi tersebut digunakan untuk menyelesaikan persoalan interpolasi seperti interpolasi binomial dan interpolasi bicubic, menyelesaikan persoalan regresi seperti regresi linear berganda, dan menyelesaikan persoalan perbesaran citra yang merupakan aplikasi lanjutan dari interpolasi bicubic. Semua kode program ditulis menggunakan bahasa pemrograman Java.

5.2 Saran

Saran untuk kedepannya yaitu agar input/output yang menggunakan file disertakan dengan jelas, seperti lokasi file yang dijadikan input apakah sudah ditentukan atau harus diinput secara manual dan lokasi file output apakah harus di dalam sebuah folder khusus atau bebas dimana saja. Selain itu, pada spesifikasi untuk interpolasi bicubic dan regresi linear berganda sebaiknya disertakan contoh data dan contoh aproksimasi sehingga ketika proses pengujian program dapat lebih mudah karena tidak perlu mencari contoh data dari sumber lain.

Daftar Referensi

Anton, H., & Rorres, C. (2014). *Elementary Linear Algebra* (11th ed.). Wiley.

Munir, R. (2022). *IF2123 Aljabar Geometri - Semester I Tahun 2022/2023.* https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/algeo.htm

Lampiran

Link repository : https://github.com/Rinaldy-Adin/Algeo01-21134

Nama asisten : Insomnia

Jadwal demo : 4 Oktober 2022 (19.00 - 20.00)