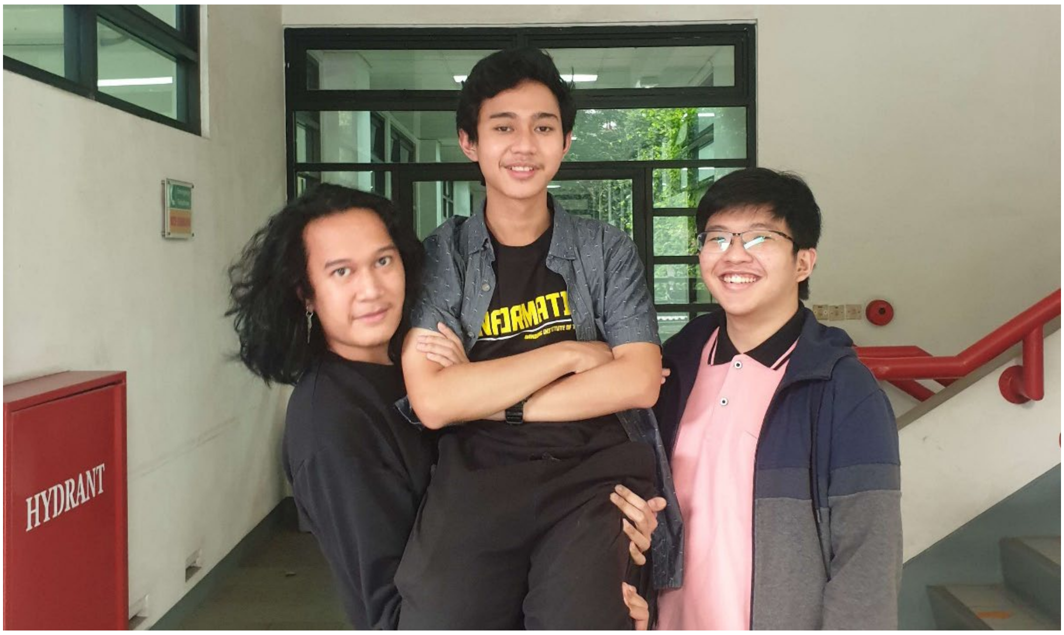
LAPORAN TUGAS BESAR 1
IF2123-K02 ALJABAR LINIER DAN GEOMETRI
SEMESTER I 2022-2023

Sistem Persamaan Linier, Determinan, dan Aplikasinya

Disusun oleh:

Kelompok 12 (KANI)

Ammar Rasyad Chaeroel	13521136
Edia Zaki Naufal Ilman	13521141
Bintang Dwi Marthen	13521144



PROGRAM STUDI
TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO
DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG 2022

1. DESKRIPSI MASALAH

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

Di dalam tugas besar 1 ini, mahasiswa diminta untuk membuat satu atau lebih *library* aljabar linier dalam Bahasa Java. Library tersebut berisi fungsi-fungsi seperti eliminasi Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung determinan, kaidah Cramer (kaidah Cramer khusus untuk SPL dengan *n* peubah dan *n* persamaan). Selanjutnya, *library* tersebut digunakan di dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi.

2. TEORI SINGKAT

Metode eliminasi Gauss dan metode eliminasi Gauss-Jordan merupakan beberapa cara menyelesaikan SPL dengan menggunakan matriks. Kedua metode tersebut dapat digunakan untuk menyelesaikan SPL yang besar karena dapat diimplementasikan pada program komputer sehingga komputasinya dilakukan oleh komputer. Metode eliminasi Gauss secara sekilas merupakan proses perubahan matriks *augmented* menjadi matriks eselon baris melalui penerapan Operasi Baris Elementer (OBE) yang kemudian dilanjutkan dengan penyulihan mundur. Di lain sisi, metode eliminasi Gauss-Jordan merupakan proses perubahan matriks *augmented* menjadi matriks eselon baris tereduksi. Metode eliminasi Gauss-Jordan terdiri atas dua fase: fase maju (perubahan menjadi matriks eselon baris (sama dengan metode eliminasi Gauss)) dan fase mundur (merubah matriks eselon baris yang telah didapatkan menjadi matriks eselon baris tereduksi).

$$\begin{bmatrix} 1 & * & * & * \\ 0 & 1 & * & * \\ 0 & 0 & 1 & * \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & * & * & * \\ 0 & 1 & * & * \\ 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & * & * & * \\ 0 & 1 & * & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & * & * & * & * & * & * & * & * \\ 0 & 0 & 0 & 1 & * & * & * & * & * & * \\ 0 & 0 & 0 & 0 & 1 & * & * & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 1 & * & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & * \end{bmatrix}$$

Gambar 2.1 Matriks Eselon Baris

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & * \\ 0 & 1 & 0 & * \\ 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & * & * \\ 0 & 1 & * & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & * & 0 & 0 & 0 & * & * & 0 & * \\ 0 & 0 & 0 & 1 & 0 & 0 & * & * & 0 & * \\ 0 & 0 & 0 & 0 & 1 & 0 & * & * & 0 & * \\ 0 & 0 & 0 & 0 & 0 & 1 & * & * & 0 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & * \end{bmatrix}$$

Gambar 2.2 Matriks Eselon Baris Tereduksi

Berikut merupakan contoh penyelesaian SPL empat variabel menggunakan metode Gauss dan metode Gauss-Jordan:

$$2x_1 - x_2 + 3x_3 + 4x_4 = 9$$
$$x_1 - 2x_3 + 7x_4 = 11$$
$$3x_1 - 3x_2 + x_3 + 5x_4 = 8$$
$$2x_1 + x_2 + 4x_3 + 4x_4 = 10$$

$$\begin{bmatrix} 2 & -1 & 3 & 4 & 9 \\ 1 & 0 & -2 & 7 & 11 \\ 3 & -3 & 1 & 5 & 8 \\ 2 & 1 & 4 & 4 & 10 \end{bmatrix} \xrightarrow{R2 \Leftrightarrow R1} \begin{bmatrix} 1 & 0 & -2 & 7 & 11 \\ 2 & -1 & 3 & 4 & 9 \\ 3 & -3 & 1 & 5 & 8 \\ 2 & 1 & 4 & 4 & 10 \end{bmatrix} \xrightarrow{\substack{R2-2R1 \\ R3-3R1 \\ R4-2R1}}$$

$$\begin{bmatrix} 1 & 0 & -2 & 7 & 11 \\ 0 & -1 & 7 & -10 & -13 \\ 0 & -3 & 7 & -16 & -25 \\ 0 & 1 & 8 & -10 & -12 \end{bmatrix} \xrightarrow{R2\times(-1)} \begin{bmatrix} 1 & 0 & -2 & 7 & 11 \\ 0 & 1 & -7 & 10 & 13 \\ 0 & -3 & 7 & -16 & -25 \\ 0 & 1 & 8 & -10 & -12 \end{bmatrix} \xrightarrow{\substack{R3+3R2 \\ R4-R2}}$$

$$\begin{bmatrix} 1 & 0 & -2 & 7 & 11 \\ 0 & 1 & -7 & 10 & 13 \\ 0 & 0 & -14 & 14 & 14 \\ 0 & 0 & 15 & -20 & -25 \end{bmatrix} \xrightarrow{R3/_{-14}} \begin{bmatrix} 1 & 0 & -2 & 7 & 11 \\ 0 & 1 & -7 & 10 & 13 \\ 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 15 & -20 & -25 \end{bmatrix} \xrightarrow{R4-15R3}$$

$$\begin{bmatrix} 1 & 0 & -2 & 7 & 11 \\ 0 & 1 & -7 & 10 & 13 \\ 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & -5 & -10 \end{bmatrix} \xrightarrow{R4/_{-5}} \begin{bmatrix} 1 & 0 & -2 & 7 & 11 \\ 0 & 1 & -7 & 10 & 13 \\ 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

Setelah mendapatkan matriks eselon baris, dapat dilakukan penyulihan mundur untuk mendapatkan nilai dari x_1, x_2, x_3, dan x_4.

$$x_4 = 2$$
$$x_3 - x_4 = -1 \rightarrow x_3 - 2 = -1 \rightarrow x_3 = 1$$
$$x_2 - 7x_3 + 10x_4 = 13 \rightarrow x_2 - 7 \times 1 + 10 \times 2 = 13 \rightarrow x_2 = 0$$
$$x_1 - 2x_3 + 7x_4 = 11 \rightarrow x_1 - 2 \times 1 + 7 \times 2 = 11 \rightarrow x_1 = -1$$

Untuk menggunakan metode eliminasi Gauss-Jordan, dapat dilanjutkan fase mundurnya karena fase maju dari metode eliminasi Gauss-Jordan adalah metode eliminasi Gauss.

$$\begin{bmatrix} 1 & 0 & -2 & 7 & 11 \\ 0 & 1 & -7 & 10 & 13 \\ 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix} \xrightarrow{\substack{R3+R4 \\ R2-10R4 \\ R1-7R4}} \begin{bmatrix} 1 & 0 & -2 & 0 & -3 \\ 0 & 1 & -7 & 0 & -7 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix} \xrightarrow{\substack{R2+7R3 \\ R1+2R3}}$$

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

Dari metode Gauss-Jordan didapatkan nilai dari x_1 adalah -1, nilai dari x_2 adalah 0, nilai dari x_3 adalah 1, dan nilai dari x_4 adalah 2. Melalui metode eliminasi Gauss maupun Gauss-Jordan, didapatkan nilai yang sama.

Matriks memiliki suatu properti kuantitatif yang disebut dengan determinan. Determinan dapat digunakan untuk menentukan matriks balikan dari suatu matriks. Ide utama dari penentuan determinan suatu matriks adalah ekspansi kofaktor. Matriks kofaktor dari matriks M merupakan matriks yang untuk setiap elemen pada baris *i* dan kolom *j*, merupakan minor $M_{i,j}$ (determinan dari sub-matriks dengan baris *i* dan kolom *j* dihapus) dari matriks M dikalikan dengan $(-1)^{i+j}$. Setelah didapatkan matriks kofaktor maka determinan dari suatu matriks dengan *m* baris dan *n* baris adalah

$$det(A) = a_{i,j}C_{i,j} + a_{i+1,j}C_{i+1,j} + \cdots + a_{n,j}C_{n,j}$$
$$det(A) = a_{i,j}C_{i,j} + a_{i,j+1}C_{i,j+1} + \cdots + a_{i,m}C_{i,m}$$

dengan $a_{i,j}$ merupakan elemen matriks pada baris *i* dan kolom *j* dan $C_{i,j}$ adalah elemen dari matriks kofaktor pada baris *i* dan kolom *j*.

Sebelum melihat contoh ekspansi kofaktor untuk matriks berukuran 3×3 atau lebih, dapat dibuktikan konsep dari ekspansi kofaktor untuk matriks berukuran 2×2 terlebih dahulu. Perlu diperhatiakan untuk suatu matriks berukuran 1×1, maka determinan dari matriks tersebut adalah elemen satu-satunya dari matriks tersebut. Berikut merupakan contoh aplikasi konsep ekspansi kofaktor untuk pencarian determinan matriks 2×2:

$$Terdapat\ matriks\ M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$
$$Matriks\ kofaktor\ dari\ matriks\ M\ adalah\ \begin{bmatrix} d & -c \\ -b & a \end{bmatrix}$$
$$Maka, det(M) = a \times d + b \times (-c) = ad - bc$$

dapat diperhatikan bahwa hasil dari determinan matriks 2×2 tersebut sama dengan rumus determinan matriks 2×2 pada umumnya.

Berikut merupakan contoh pencarian determinan dari matriks 3×3 dengan menggunakan ekspansi kofaktor:

$$M = \begin{bmatrix} 3 & 0 & 0 \\ 0 & -1 & 3 \\ 0 & -3 & -1 \end{bmatrix}$$
$$cof(M) = \begin{bmatrix} \begin{vmatrix} -1 & 3 \\ -3 & -1 \end{vmatrix} & \begin{vmatrix} 0 & 3 \\ 0 & -1 \end{vmatrix} & \begin{vmatrix} 0 & -1 \\ 0 & -3 \end{vmatrix} \\ \begin{vmatrix} 0 & 0 \\ -3 & -1 \end{vmatrix} & \begin{vmatrix} 3 & 0 \\ 0 & -1 \end{vmatrix} & \begin{vmatrix} 3 & 0 \\ 0 & -3 \end{vmatrix} \\ \begin{vmatrix} 0 & 0 \\ -1 & 3 \end{vmatrix} & \begin{vmatrix} 3 & 0 \\ 0 & 3 \end{vmatrix} & \begin{vmatrix} 3 & 0 \\ 0 & -1 \end{vmatrix} \end{bmatrix} = \begin{bmatrix} 10 & 0 & 0 \\ 0 & -3 & 9 \\ 0 & 9 & -3 \end{bmatrix}$$
$$\det(M) = a_{1,1}C_{1,1} + a_{1,2}C_{1,2} + a_{1,3}C_{1,3} = 3 \times 10 + 0 \times 0 + 0 \times 0 = 30$$

Salah satu aplikasi dari determinan selain menentukan matriks balikan (yang akan dibahas kemudian) adalah menyelesaikan SPL *n* peubah dengan *n* persamaan. Penyelesaian SPL dengan konsep determinan disebut dengan kaidah Cramer. Secara singkat, penyelesaian SPL $Ax = b$ dengan kaidah Cramer adalah sebagai berikut:

$$x_n = \frac{\det(A_n)}{\det(A)}$$

dengan A_n adalah matriks A yang kolom ke-n ditukarkan dengan elemen-elemen matriks b.

Selain determinan, terdapat salah satu properti aljabar dari matriks yaitu balikan. Dalam aritmetika, untuk setiap bilangan bukan nol x maka akan ada suatu bilangan x^{-1} yang ketika dikalikan dengan x akan menghasilkan satu. Begitu pula dengan matriks, akan tetapi hasil kali dari suatu matriks dengan matriks balikannya akan menghasilkan matriks identitas ($AA^{-1}=I$ dan $A^{-1}A=I$).

Salah satu cara menentukan matriks balikan adalah menggunakan matriks kofaktor dan determinan yang telah dibahas sebelumnya. Rumus untuk menentukan matriks balikan dari suatu matriks A:

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

Melalui persamaan tersebut, didapati bahwa ketika determinan dari matriks A adalah nol maka matriks A tidak memiliki balikan (*non-invertible*). Selain itu, transpose dari matriks kofaktor disebut dengan matriks adjoin.

Selain menggunakan determinan, balikan dari suatu matriks dapat ditemukan dengan menggunakan OBE dan matriks *augmented*. Untuk mencari matriks balikan menggunakan metode ini, maka dari matriks *augmented* $[A \mid I]$ harus diubah menjadi $[I \mid A^{-1}]$. Berikut merupakan contoh untuk mencari suatu matriks balikan dari matriks 3×3 menggunakan metode ini:

$$\begin{bmatrix} 1 & 2 & 3 & 1 & 0 & 0 \\ 2 & 5 & 3 & 0 & 1 & 0 \\ 1 & 0 & 8 & 0 & 0 & 1 \end{bmatrix} \xrightarrow[R3-R1]{R2-2R1} \begin{bmatrix} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & -2 & 5 & -1 & 0 & 1 \end{bmatrix} \xrightarrow[R1-2R2]{R3+2R2}$$

$$\begin{bmatrix} 1 & 0 & 9 & 5 & -2 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & 0 & -1 & -5 & 2 & 1 \end{bmatrix} \xrightarrow{R3\times(-1)} \begin{bmatrix} 1 & 0 & 9 & 5 & -2 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{bmatrix} \xrightarrow[R2+3R3]{R1-9R3}$$

$$\begin{bmatrix} 1 & 0 & 0 & -40 & 16 & 9 \\ 0 & 1 & 0 & 13 & -5 & -3 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{bmatrix}$$

$$sehingga\ matriks\ balikan\ dari\ matriks \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 3 \\ 1 & 0 & 8 \end{bmatrix} adalah \begin{bmatrix} -40 & 16 & 9 \\ 13 & -5 & -3 \\ 5 & -2 & -1 \end{bmatrix}.$$

Salah satu implementasi dari SPL adalah interpolasi polinomial. Interpolasi polinomial merupakan suatu metode untuk menginterpolasikan suatu fungsi polinomial dari pola data yang ada. Suatu fungsi polinomial memiliki bentuk standar $f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$. Perlu diperhatikan kembali bahwa untuk sistem persamaan dengan n peubah, dibutuhkan setidaknya n persamaan untuk mendapatkan solusi eksak. Berikut merupakan contoh interpolasi polinomial:

Terdapat suatu polinomial pangkat tiga yang melalui titik-titik berikut: (1,3), (2,-2), (3,-5), dan (4,0). Tentukan fungsi dari polinomial tersebut!

$$Bentuk\ umum\ dari\ polinomial\ pangkat\ tiga\ adalah$$

$$p(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

$$Bentuk\ persamaan\ tersebut\ dapat\ diubah\ menjadi\ bentuk\ matriks$$
$$augmented\ sesuai\ dengan\ bentuk\ umum\ dan\ koordinat\ yang\ telah\ ada$$

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & y_1 \\ 1 & x_2 & x_2^2 & x_2^3 & y_2 \\ 1 & x_3 & x_3^2 & x_3^3 & y_3 \\ 1 & x_4 & x_4^2 & x_4^3 & y_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 3 \\ 1 & 2 & 4 & 8 & -2 \\ 1 & 3 & 9 & 27 & -5 \\ 1 & 4 & 16 & 64 & 0 \end{bmatrix}$$

$$Kemudian\ diselesaikan\ dengan\ eliminasi\ Gauss - Jordan$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & -5 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$Didapatkan\ a_0 = 4, a_1 = 3, a_2 = -5, dan\ a_4 = 1\ sehingga$$

$$p(x) = 4 + 3x - 5x^2 + x^3$$

Interpolasi polinomial merupakan interpolasi untuk data 1D. Akan tetapi, terdapat suatu jenis interpolasi untuk data 2D yaitu *bicubic interpolation*. *Bicubic interpolation* umumnya digunakan untuk pembesaran citra (peningkatan resolusi). Dalam *bicubic interpolation*, diperlukan nilai-nilai dari titik yang bersebelahan dari titik-titik *normalization*. Sehingga suatu persamaan 4 titik akan menjadi persamaan 16 titik saat menggunakan *bicubic interpolation*. Berikut merupakan contoh penerapan *bicubic interpolation*:

Terdapat suatu pemodelan untuk interpolasi persamaan *f(x,y)* sebagai berikut:

Normalization : *f(0,0), f(1,0),f(0,1),* dan *f(1,1)*
Model : $f(x,y) = \sum_{j=0}^{3} \sum_{i=0}^{3} a_{ij} x^i y^j$; *x = -1,0,1,2*
Solve : a_{ij}

$$Sistem\ persamaan\ yang\ terbentuk\ adalah\ y = Xa$$

$$\begin{bmatrix} f(-1,-1) \\ f(0,-1) \\ f(1,-10 \\ f(2,-1) \\ f(-1,0) \\ f(0,0) \\ f(1,0) \\ f(2,0) \\ f(-1,1) \\ f(0,1) \\ f(1,1) \\ f(2,1) \\ f(-1,2) \\ f(0,2) \\ f(1,2) \\ f(2,2) \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 2 & 4 & 8 & -1 & -2 & -4 & -8 & 1 & 2 & 4 & 8 & -1 & -2 & -4 & -8 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 \\ 1 & -1 & 1 & -1 & 2 & -2 & 2 & -2 & 4 & -4 & 4 & -4 & 8 & -8 & 8 & -8 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 8 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 1 & 2 & 4 & 8 & 2 & 4 & 8 & 16 & 4 & 8 & 16 & 32 & 8 & 16 & 32 & 64 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30} \\ a_{01} \\ a_{11} \\ a_{21} \\ a_{31} \\ a_{02} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{03} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}$$

$$Untuk\ mendapatkan\ matriks\ a\ dapat\ digunakan\ pendekatan\ SPL.$$

Selain interpolasi polinomial, regresi linier berganda juga dapat digunakan untuk memprediksikan suatu nilai. Pada metode ini, diperlukan setidaknya persamaan yang memiliki (x_1, x_2, x_3,…,x_k, y). Metode regresi linier berganda menggunakan *Normal Estimation Equation for Multiple Linear Regression*:

$$nb_0 + b_1 \sum_{i=1}^{n} x_{1i} + b_2 \sum_{i=1}^{n} x_{2i} + \cdots + b_k \sum_{i=1}^{n} x_{ki} = \sum_{i=1}^{n} y_i$$

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

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

.
.
.

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

Setelah SPL tersebut diselesaikan, maka didapatkan nilai-nilai dari b_0, b_1, b_2, ..., b_k. Dengan adanya nilai untuk setiap *b* maka dapat dilakukan penyulihan ke persamaan umum dari regresi linier untuk mendapatkan nilai yang ingin diprediksi. Berikut merupakan persamaan umum dari regresi linier:

$$y_i = b_0 + b_1x_{1i} + b_2x_{2i} + \cdots + b_kx_{ki} + \epsilon_i$$

3. IMPLEMENTASI PUSTAKA DAN PROGRAM DALAM JAVA

1. Class DoubleMatrix

Nama	**Tipe**	**Parameter**	**Deskripsi**
DoubleMatrix	public	List<List<Double>> x	*Constructor* DoubleMatrix dari suatu list of list
DoubleMatrix	public	int row, int col	*Constructor* DoubleMatrix dengan baris sebanyak row dan kolom sebanyak col
DoubleMatrix	public	int row	*Constructor* DoubleMatrix dengan dimensi row x row
DoubleMatrix	public	int row, int col, List<List<Double>> copyList	*Constructor* DoubleMatrix dengan baris sebanyak row, kolom sebanyak col, dan isinya merupakan list of list copyList
getLHS	public DoubleMatrix	-	Memotong DoubleMatrix menjadi sebuah DoubleMatrix berukuran row x row dari kiri

2. Class IntegerMatrix

Nama	**Tipe**	**Parameter**	**Deskripsi**
IntegerMatrix	public	List<List<Integer>> x	*Constructor* IntegerMatrix dari suatu list of list
IntegerMatrix	public	int row, int col, List<List<Integer>> copyList	*Constructor* IntegerMatrix dengan baris sebanyak row, kolom sebanyak col, dan isinya merupakan list of list copyList

3. Interface Ioperators
   Sebuah lambda untuk mempermudah proses eliminasi Gauss dan eliminasi Gauss-Jordan

4. Class Matrix

Atribut	**Deskripsi**
row	Jumlah baris matriks
col	Jumlah kolom matriks
matrix	List of List yang berisi elemen dari matriks

Nama	**Tipe**	**Parameter**	**Deskripsi**
Matrix	public	int row, int col, List<List<T>> x	*Constructor* matriks dengan dimensi row x col dan isinya dari list of list x
Matrix	public	int row, int col	*Constructor* matriks dengan dimenxi row x col
Matrix	public	List<List<T>> x	*Constructor* matriks dari suatu list of list x
Matrix	public	int size	*Constructor* matriks *square* dengan ukuran size
printMatrix	public static <E> void	Matrix <E> m	Mencetak matriks ke layar
getMatrix	public List<List<T>>	-	*Getter* dari List of List sebuah matriks
getRow	public int	-	*Getter* dari jumlah baris sebuah matriks
getCol	public int	-	*Getter* dari jumlah kolom sebuah matriks
getConstants	public double[]	-	Mendapatkan kolom terakhir dari sebuah matriks
getLHS	public Matrix<T>	-	Memotong matriks menjadi sebuah square matriks berukuran row dari kiri
getElement	public T	int i, int j	Mendapatkan elemen matriks pada baris i, kolom j
getRowElements	public T[]	int row	Mendapatkan elemen matriks pada baris row
getColElements	public T[]	int col	Mendapatkan elemen matriks pada kolom col
setElement	public void	int row, int col, T element	Mengubah nilai dari elemen matriks pada baris i, kolom j menjadi element
setRowElements	public void	int row, ArrayList<T> element	Mengubah nilai dari sebuah baris matriks sesuai dengan ArrayList element

getIdentityMatrix	public static <T extends Number>	int size	Mengembalikan sebuah matriks identitas dengan ukuran size
convertToInteger	public static IntegerMatrix	Matrix<? extends Number> m	Mengkonversi sebuah matriks menjadi IntegerMatrix
convertToDouble	public static DoubleMatrix	Matrix<? extends Number> m	Mengkonversi sebuah matriks menjadi DoubleMarix

5. Class MatrixFileOperator

Nama	**Tipe**	**Parameter**	**Deskripsi**
createDMFromFile	public static DoubleMatrix	String filename	Mengkonstruksi sebuah DoubleMatrix dari file
createIMFromFile	public static IntegerMatrix	String filename	Mengkonstruksi sebuah IntegerMatrix dari file
writeMatrixToFile	public static void	String filename, DoubleMatrix matrix	Mencetak isi sebuah DoubleMatrix ke file

6. Class MatrixOperators

Nama	**Tipe**	**Parameter**	**Deskripsi**
getInstance	public static MatrixOperators	-	Singleton dari MatrixOperators
addMatrix	public DoubleMatrix	DoubleMatrix m1, DoubleMatrix m2	Pertambahan dua matriks (m1 + m2)
subtractMatrix	public DoubleMatrix	DoubleMatrix m1, DoubleMatrix m2	Pengurangan dua matriks (m1 – m2)
multiplyMatrixByConst	public DoubleMatrix	DoubleMatrix m1, double multiplier	Perkalian matriks dengan sebuah konstanta
multiplyMatrixByMatrix	public DoubleMatrix	DoubleMatrix m1, DoubleMatrix m2	Perkalian dua matriks (m1 * m2)
divideMatrix	public DoubleMatrix	DoubleMatrix m1, DoubleMatrix m2	Pembagian dua matriks (m1 / m2)
cofactor	public DoubleMatrix	DoubleMatrix m	Mengembalikan matriks kofaktor dari matriks m
determinant	public double	Matrix<? extends Number> m, int mode	Mencari determinan dari suatu matriks m dengan metode reduksi baris atau ekspansi kofaktor
determinant	public double	Matrix<? extends Number> m	Mencari determinan dari suatu matriks m dengan ekspansi kofaktor
adjugate	public DoubleMatrix	Matrix<? extends Number> m	Mengembalikan matriks adjoin/adjugate dari matriks m
transpose	public DoubleMatrix	Matrix<? extends Number> m	Mengembalikan matriks m yang telah ditranspose
inverse	public DoubleMatrix	Matrix<? extends Number> m, int method	Mengembalikan matriks balikan dari matriks m dengan metode matriks adjoin atau Gauss-Jordan
inverse	public DoubleMatrix	Matrix<? extends Number> m	Mengembalikan matriks balikan dari matriks m dengan metode matriks adjoin

removeRowCols	private DoubleMatrix	DoubleMatrix m, int row, int col	Mengembalikan matriks m dengan baris row dan kolom col telah dihapus
cramer	public double[]	Matrix<? extends Number> m	Mengembalikan hasil SPL dari suatu matriks dengan kaidah Cramer
doesInvereExist	private boolean	DoubleMatrix m	Mengecek apakah matriks m memiliki balikan
gauss	public DoubleMatrix	Matrix<? extends Number> m	Melakukan eliminasi gauss terhadap suatu *augmented matrix*
gauss	private DoubleMatrix	Matrix<Double> result, double[] constants, int offset	Melakukan eliminasi gauss terhadap suatu *augmented matrix*
gaussJordan	public DoubleMatrix	Matrix<? extends Number> m	Melakukan eliminasi Gauss-Jordan terhadap suatu *augmented matrix*
rowApply	private ArrayList <Double>	Double[] row 1, Double[] row 2, Ioperators<Double> operator	Melakukan OBE antara dua baris (misal R1 – R2)
rowApply	private ArrayList <Double>	Double[] row, ArrayList<Double> list, Ioperators<Double> operator	Melakukan OBE antara dua baris (misal R1 – R2)
rowApply	private ArrayList <Double>	Double[] row 1, double constant, Ioperators<Double> operator	Mengali sebuah baris dengan konstanta
swapRow	private DoubleMatrix	DoubleMatrix m, int row	Melakukan pertukaran dua baris pada matriks
swapCol	private void	Matrix<Double> matrix, int col, int pivot	Melakukan pertukaran dua kolom pada matriks

7. Class MatrixType

Nama	**Tipe**	**Parameter**	**Deskripsi**
getMatrixType	static MatrixType (enum)	Matrix<Double> m	Mendapatkan tipe solusi dari SPL dengan matriks *augmented* m (parametrik, unik, atau tidak ada solusi)

8. Class ParametricSolver

Nama	**Tipe**	**Parameter**	**Deskripsi**
solve	public static String[]	Matrix<? extends Number> matrix	Menyelesaikan suatu SPL dari augmented matriks dengan solusi parametrik
makeVariable	private static char	int j, int[] idx, int count	Membuat sebuah variabel bagi solusi parametrik

9. Class Driver

Nama	**Tipe**	**Parameter**	**Deskripsi**

driverSPL	protected static void	-	Fungsi berisi jalannya sub-program penyelesaian SPL
driverDeterminan	protected static void	-	Fungsi berisi jalannya sub-program penyelesaian determinan
driverBalikan	protected static void	-	Fungsi berisi jalannya sub-program penyelesaian inverse matriks
driverPolinomial	protected static void	-	Fungsi berisi jalannya sub-program penyelesaian Interpolasi Polinomial
driverBicubic	protected static void	-	Fungsi berisi jalannya sub-program penyelesaian *Bicubic Interpolation*
driverRegresi	protected static void	-	Fungsi berisi jalannya sub-program penyelesaian RegresiLinier
driverBonus	protected static void	-	Fungsi berisi jalannya sub-program *image upscaling*

10. Class IOHandler

Nama	**Tipe**	**Parameter**	**Deskripsi**
opsi	protected static int	int awal, int akhir	Mendapatkan input dari pilihan menu dengan batas bawah awal dan batas atas akhir
inputFile	protected static boolean	-	Menanyakan user apakah input dari file
inputDoubleMatrix	protected static DoubleMatrix	int row, int col	Input matriks berukuran row x col dari keyboard
fileDoubleMatrix	protected static DoubleMatrix	-	Input matriks dari file
fileOutput	protected static boolean	-	Menanyakan user apakah output akan disimpan ke file
outputFile	protected static String	-	Mendapatkan nama file yang akan digunakan sebagai file output

11. Class Menu

Nama	**Tipe**	**Parameter**	**Deskripsi**
identitas	protected static void	-	Berisikan print CLI mengenai identitas anggota kelompok
menuUtama	protected static void	-	Berisikan print CLI mengenai opsi-opsi pada menu utama (main menu)
menuSPL	protected static void	-	Berisikan print CLI mengenai opsi-opsi pada menu subprogram SPL
menuDeterminan	protected static void	-	Berisikan print CLI mengenai opsi-opsi pada menu subprogram Determinan
menuBalikan	protected static void	-	Berisikan print CLI mengenai opsi-opsi pada menu subprogram Inverse Matriks
menuInput	protected static void	-	Berisikan print CLI mengenai opsi-opsi pada menu input matriks
menuOutput	protected static void	-	Berisikan print CLI mengenai opsi-opsi apakah hasil dioutput ke file
menuBicubic	protected static void	-	Berisikan print CLI mengenai opsi-opsi pada menu subprogram *Bicubic Interpolation*

12. Class InterpolasiPolinomial

Nama	**Tipe**	**Parameter**	**Deskripsi**
getEstimate	public static double	double[] coefficient, double x	Mentaksir nilai dari f(x) bila telah diketahui fungsi f(x)

getCoefficient	public static double[]	DoubleMatrix m	Mendapatkan koefisien dari setiap x pada fungsi f(x)
printPolinom	public static void	double[] b	Mencetak fungsi f(x)

13. Class RegresiLinier

Nama	**Tipe**	**Parameter**	**Deskripsi**
getSolution	public static double[]	DoubleMatrix m	Mendapatkan solusi regresi linier berganda dari matriks *augmented* m
cetakFungsi	public static void	double[] a	Mencetak fungsi hasil regresi
getEstimate	public static double	double []a, double[] peubah	Menaksir nilai f(x) yang fungsi f(x) nya telah diketahui

14. Class BicubicInterpolation

Nama	**Tipe**	**Parameter**	**Deskripsi**
matrixInterpolation	public static double	double x, double y, DoubleMatrix m	Melakukan interpolasi pada titik-titik dan matriks yang sudah ditentukan
getCoeffMatrix	public static DoubleMatrix	-	Mengembalikan matriks yang berisi koefisien berukuran 16 x 16
getFunctionMatrix	public static DoubleMatrix	DoubleMatrix masukan	Mengembalikan matriks yang merupakan matriks fungsi 16 x 1
findValues	public static DoubleMatrix	DoubleMatrix m, DoubleMatrix n	Mengembalikan nilai variabel-variabel a dengan mengalikan matriks invers koefisien (m) dengan matriks fungsi (n)
interpolation	public static float	double x, double y, DoubleMatrix m	Menghitung hasil interpolasi pada titik x dan y

15. Class ImageUpscale

Nama	**Tipe**	**Parameter**	**Deskripsi**
getNewRGB	public static int[]	BufferedImage src	Mendapatkan nilai array data RGB baru hasil interpolasi bicubic
getRGBFunc	private static DoubleMatrix	int X, int Y, BufferedImage src	Mengambil dan membuat sampel matriks fungsi dari data RGB citra
CLAMP	private static int	int rgb, int min, int max	Memastikan nilai rgb tidak keluar batas min dan max

4. EKSPERIMEN

1. Solusi SPL Ax = B

SPL	**Hasil**	**Metode**
$A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$	Tidak ada solusi	Eliminasi gauss

$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$	x1 = 3.0 + 1.0 b x2 = 2.0 b x3 = a x4 = -1.0 + 1.0 b x5 = b	Eliminasi Gauss
$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$	x1 = a x2 = 1.0 - 1.0 c x3 = b x4 = -2.0 - 1.0 c x5 = 1.0 + 1.0 c x6 = c	Eliminasi Gauss
n=6 $H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n+1} \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$	x1 = 35.99999999847885 x2 = -629.9999999566953 x3 = 3359.9999997089526 x4 = -7559.9999992480525 x5 = 7559.99999917505 x6 = -2771.9999996766783	Kaidah Cramer
n=10 $H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n+1} \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$	x1 = 99.99035437041893 x2 = -4949.159598718361 x3 = 79181.98528596727 x4 = -600435.4050103612 x5 = 2521731.7415962443 x6 = -6304125.909429202 x7 = 9606023.151161699 x8 = -8748135.347572321 x9 = 4373977.470228117 x10 = -923378.5098546401	Eliminasi Gauss-Jordan

2. SPL berbentuk Matriks Augmented

SPL	Hasil	Metode
$\begin{bmatrix} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{bmatrix}.$	x1 = -1.0 + 1.0 b x2 = 2.0 a x3 = a x4 = b	Eliminasi Gauss
$\begin{bmatrix} 2 & 0 & 8 & 0 & 8 \\ 0 & 1 & 0 & 4 & 6 \\ -4 & 0 & 6 & 0 & 6 \\ 0 & -2 & 0 & 3 & -1 \\ 2 & 0 & -4 & 0 & -4 \\ 0 & 1 & 0 & -2 & 0 \end{bmatrix}.$	x1 = 0.0 x2 = 2.0 x3 = 1.0 x4 = 1.0	Eliminasi Gauss

3. SPL dalam bentuk persamaan

SPL	Hasil	Metode
$8x_1 + x_2 + 3x_3 + 2x_4 = 0$ $2x_1 + 9x_2 - x_3 - 2x_4 = 1$ $x_1 + 3x_2 + 2x_3 - x_4 = 2$ $x_1 + 6x_3 + 4x_4 = 3$	x1 = -0.22432432432432434 x2 = 0.18243243243243243 x3 = 0.7094594594594594 x4 = -0.25810810810810814	Matriks Balikan

<table>
<tr>
<td>$$\begin{aligned} x_7 + x_8 + x_9 &= 13.00 \\ x_4 + x_5 + x_6 &= 15.00 \\ x_1 + x_2 + x_3 &= 8.00 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_6 + x_8) + 0.61396x_9 &= 14.79 \\ 0.91421(x_3 + x_5 + x_7) + 0.25(x_2 + x_4 + x_6 + x_8) &= 14.31 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_2 + x_4) + 0.61396x_1 &= 3.81 \\ x_3 + x_6 + x_9 &= 18.00 \\ x_2 + x_5 + x_8 &= 12.00 \\ x_1 + x_4 + x_7 &= 6.00 \\ 0.04289(x_1 + x_5 + x_9) + 0.75(x_2 + x_6) + 0.61396x_3 &= 10.51 \\ 0.91421(x_1 + x_5 + x_9) + 0.25(x_2 + x_4 + x_6 + x_8) &= 16.13 \\ 0.04289(x_1 + x_5 + x_9) + 0.75(x_4 + x_8) + 0.61396x_7 &= 7.04 \end{aligned}$$</td>
<td>Tidak ada solusi</td>
<td>Eliminasi Gauss</td>
</tr>
</table>

4. Studi Kasus Interpolasi

<table>
<tr><th>Data</th><th>Hasil Interpolasi</th><th>Hasil Taksiran</th></tr>
<tr>
<td><table><tr><td>x</td><td>0.4</td><td>0.7</td><td>0.11</td><td>0.14</td><td>0.17</td><td>0.2</td><td>0.23</td></tr><tr><td>f(x)</td><td>0.043</td><td>0.005</td><td>0. 058</td><td>0.072</td><td>0.1</td><td>0.13</td><td>0.147</td></tr></table></td>
<td>f(x) = -4212.434531756722 x^6 + 7102.399162436538 x^5- 4346.3139507523465 x^4 + 1220.8548905938487 x^3 - 163.91566260202129 x^2 + 10.276383988580168 x - 0.18455901912967704</td>
<td>f (0.2) = 0.13
f (0.55) = 2.137572
f (0.85) = -66.269639
f (1.28) =</td>
</tr>
<tr>
<td><table><tr><th>Tanggal</th><th>Tanggal (desimal)</th><th>Jumlah Kasus Baru</th></tr><tr><td>17/06/2022</td><td>6,567</td><td>12.624</td></tr><tr><td>30/06/2022</td><td>7</td><td>21.807</td></tr><tr><td>08/07/2022</td><td>7,258</td><td>38.391</td></tr><tr><td>14/07/2022</td><td>7,451</td><td>54.517</td></tr><tr><td>17/07/2022</td><td>7,548</td><td>51.952</td></tr><tr><td>26/07/2022</td><td>7,839</td><td>28.228</td></tr><tr><td>05/08/2022</td><td>8,161</td><td>35.764</td></tr><tr><td>15/08/2022</td><td>8,484</td><td>20.813</td></tr><tr><td>22/08/2022</td><td>8,709</td><td>12.408</td></tr><tr><td>31/08/2022</td><td>9</td><td>10.534</td></tr></table></td>
<td>f(x) = -140993.71224863594 x^9 + 9372849.23910132 x^8- 2.7547453942066944E8 x^7 + 4.695806315428793E9 x^6- 5.113187676013281E10 x^5 + 3.68550807175535E11 x^4- 1.7568101863613564E12 x^3 + 5.334203055240578E12 x^2- 9.346993079173438E12 x + 7.187066071661201E12</td>
<td>16/07/2022 = 53566.808594
10/08/2022 = 3631.722656
05/09/2022 = -667646.218750
07/04/2022 = -1708343623.869331</td>
</tr>
<tr>
<td>$$f(x) = \frac{x^2 + \sqrt{x}}{e^x + x}$$ disederhanakan 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</td>
<td>f(x) = 0.2362556966145896 x^5 - 1.4212630208333623 x^4 + 3.23711002604171 3 x^3 - 3.5526791666666973 x^2 + 2.0352567500000065 x</td>
<td>-</td>
</tr>
</table>

5. Studi Kasus Interpolasi Bicubic

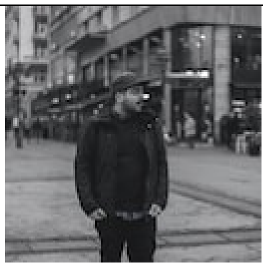
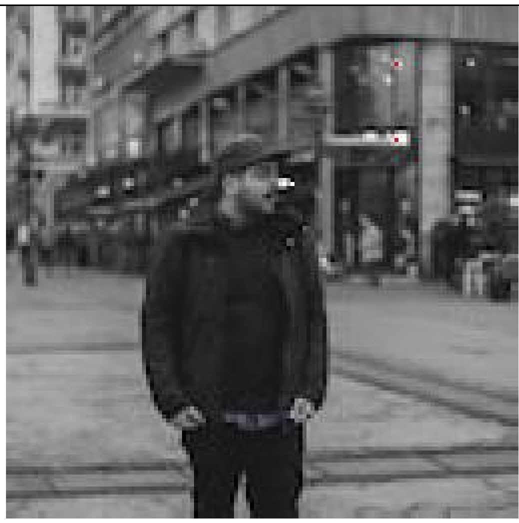
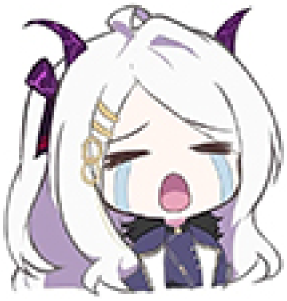
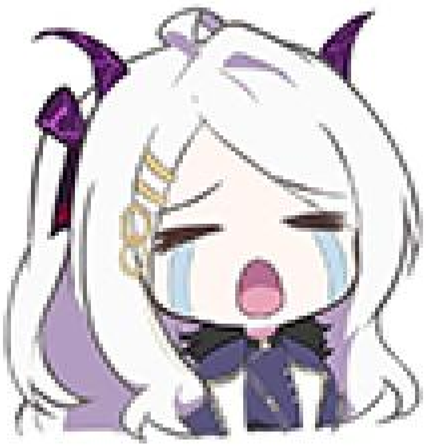
<table>
<tr><th>Matriks</th><th>Hasil Interpolasi</th></tr>
<tr>
<td>$$\begin{bmatrix} 153 & 59 & 210 & 96 \\ 125 & 161 & 72 & 81 \\ 98 & 101 & 42 & 12 \\ 21 & 51 & 0 & 16 \end{bmatrix}$$</td>
<td>f (0,0) = 161
f (0.5,0.5) = 97.73
f (0.25,0.75) = 105.51
f (0.1,0.9) = 104.23</td>
</tr>
</table>

6. Studi Kasus Regresi Linier Berganda

<table>
<tr><th>Data</th><th>Hasil</th></tr>
<tr>
<td>Table 12.1: Data for Example 12.1<table><tr><th>Nitrous Oxide, y</th><th>Humidity, x_1</th><th>Temp., x_2</th><th>Pressure, x_3</th><th>Nitrous Oxide, y</th><th>Humidity, x_1</th><th>Temp., x_2</th><th>Pressure, x_3</th></tr><tr><td>0.90</td><td>72.4</td><td>76.3</td><td>29.18</td><td>1.07</td><td>23.2</td><td>76.8</td><td>29.38</td></tr><tr><td>0.91</td><td>41.6</td><td>70.3</td><td>29.35</td><td>0.94</td><td>47.4</td><td>86.6</td><td>29.35</td></tr><tr><td>0.96</td><td>34.3</td><td>77.1</td><td>29.24</td><td>1.10</td><td>31.5</td><td>76.9</td><td>29.63</td></tr><tr><td>0.89</td><td>35.1</td><td>68.0</td><td>29.27</td><td>1.10</td><td>10.6</td><td>86.3</td><td>29.56</td></tr><tr><td>1.00</td><td>10.7</td><td>79.0</td><td>29.78</td><td>1.10</td><td>11.2</td><td>86.0</td><td>29.48</td></tr><tr><td>1.10</td><td>12.9</td><td>67.4</td><td>29.39</td><td>0.91</td><td>73.3</td><td>76.3</td><td>29.40</td></tr><tr><td>1.15</td><td>8.3</td><td>66.8</td><td>29.69</td><td>0.87</td><td>75.4</td><td>77.9</td><td>29.28</td></tr><tr><td>1.03</td><td>20.1</td><td>76.9</td><td>29.48</td><td>0.78</td><td>96.6</td><td>78.7</td><td>29.29</td></tr><tr><td>0.77</td><td>72.2</td><td>77.7</td><td>29.09</td><td>0.82</td><td>107.4</td><td>86.8</td><td>29.03</td></tr><tr><td>1.07</td><td>24.0</td><td>67.7</td><td>29.60</td><td>0.95</td><td>54.9</td><td>70.9</td><td>29.37</td></tr></table>Source: Charles T. Hare, "Light-Duty Diesel Emission Correction Factors for Ambient Conditions," EPA-600/2-77-116. U.S. Environmental Protection Agency.</td>
<td>y = -3.5077781408835103 - 0.002624990745878327 x^1 + 7.989410472218274E-4 x^2 + 0.15415503019830143 x^3</td>
</tr>
</table>

	Ketika *Humidity* bernilai 50%, temperatur 76°F, dan tekanan udara bernilai 29.30 y = 0.9384342262216645

7. Studi Kasus *Image Upscaling* dengan *Bicubic Interpolation*

Citra Awal	**Citra Olahan**
 128 x 128 px	 256 x 256 px
 128 x 128 px	 256 x 256 px

5. KESIMPULAN, SARAN, DAN REFLEKSI

<u>Kesimpulan</u>

1. Penyelesaian sistem persamaan linier dengan matriks dapat dilakukan dengan pendekatan eliminasi Gauss, eliminasi Gauss-Jordan, matriks balikan, dan kaidah Cramer
2. Mencari determinan matriks dapat dilakukan dengan pendekatan reduksi baris dan ekspansi kofaktor
3. Mencari balikan dari suatu matriks dapat dilakukan dengan pendekatan determinan-matriks adjoin dan eliminasi Gauss-Jordan
4. Matriks dapat digunakan untuk menyelesaikan beberapa persoalan seperti interpolasi polinomial, *bicubic interpolation*, dan regresi linier berganda

5. *Bicubic interpolation* dapat digunakan untuk melakukan *upscaling* sebuah citra visual

Saran

1. *Test case* dapat disertai dengan jawaban sehingga tidak menerka-menerka dan membandingkan dengan teman ketika melakukan pengetesan
2. Diberikan besar galat yang diizinkan, mengingat *floating point* pada komputer sangat rentan kehilangan presisi

Refleksi

1. Membuat desain program terlebih dahulu supaya alur program lebih jelas dan elegan
2. Membuat driver untuk setiap sub-program supaya program *Main* lebih elegan
3. Melakukan pengetesan sedini mungkin supaya proses *debugging* dapat segera dilaksanakan
4. Melakukan pengawasan terhadap pekerjaan setiap anggota kelompok
5. Melakukan operasi dengan *floating point* sangat rentan terjadi galat
6. Mempertahankan komunikasi yang jelas dan sering melakukan konfirmasi ulang akan informasi yang ditangkap

6. Referensi

1. Elementary Linear Algebra 11^{th} Edition oleh Howard Anton dan Chris Rorres
2. https://www.mssc.mu.edu/~daniel/pubs/RoweTalkMSCS_BiCubic.pdf
3. https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2022-2023/Tubes1-Algeo-2022.pdf

7. Lampiran

1. Uji coba program untuk SPL Ax = B

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic
6. Regresi Linier Berganda
7. Keluar
MASUKKAN ANGKA ANTARA 1 HINGGA 7
> 1
TENTUKAN JENIS INPUT
1. Input Keyboard
2. Input File
MASUKKAN ANGKA ANTARA 1 HINGGA 2
> 2
MASUKKAN NAMA FILE
> 1a.txt
SISTEM PERSAMAAN LINIER AKAN DISELESAIKAN DENGAN:
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
MASUKKAN ANGKA ANTARA 1 HINGGA 4
HASIL DARI SPL TERSEBUT DENGAN METODE GAUSS
Tidak ada solusi.
MASUKKAN NAMA FILE
> 1d-6.txt
SISTEM PERSAMAAN LINIER AKAN DISELESAIKAN DENGAN:
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
MASUKKAN ANGKA ANTARA 1 HINGGA 4
> 4
HASIL DARI SPL TERSEBUT DENGAN KAIDAH CRAMER
x1 = 35.999936
x2 = -629.998676
x3 = 3359.992833
x4 = -7559.984017
x5 = 7559.984117
x6 = -2771.994170
```

```
MASUKKAN NAMA FILE
> 1d-10.txt
SISTEM PERSAMAAN LINIER AKAN DISELESAIKAN DENGAN:
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
MASUKKAN ANGKA ANTARA 1 HINGGA 4
> 2
HASIL DARI SPL TERSEBUT DENGAN METODE GAUSS-JORDAN
x1 = 99.990354
x2 = -4949.159599
x3 = 79181.985286
x4 = -600435.405010
x5 = 2521731.741596
x6 = -6304125.909429
x7 = 9606023.151162
x8 = -8748135.347572
x9 = 4373977.470228
x10 = -923378.509855
```

2. Uji coba program untuk SPL dalam *augmented matrix*

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic
6. Regresi Linier Berganda
7. Keluar
MASUKKAN ANGKA ANTARA 1 HINGGA 7
> 1
TENTUKAN JENIS INPUT
1. Input Keyboard
2. Input File
MASUKKAN ANGKA ANTARA 1 HINGGA 2
> 2
MASUKKAN NAMA FILE
> 2b.txt
SISTEM PERSAMAAN LINIER AKAN DISELESAIKAN DENGAN:
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
MASUKKAN ANGKA ANTARA 1 HINGGA 4
> 1
HASIL DARI SPL TERSEBUT DENGAN METODE GAUSS
[1.0, 0.0, 4.0, 0.0, 4.0]
[0.0, 1.0, 0.0, 4.0, 6.0]
[0.0, 0.0, 1.0, 0.0, 1.0]
[0.0, 0.0, 0.0, 1.0, 1.0]
x1 = 0.000000
x2 = 2.000000
x3 = 1.000000
x4 = 1.000000
```

3. Uji coba program untuk SPL dalam persamaan

```
SISTEM PERSAMAAN LINIER AKAN DISELESAIKAN DENGAN:
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
MASUKKAN ANGKA ANTARA 1 HINGGA 4
> 3
HASIL DARI SPL TERSEBUT DENGAN METODE MATRIKS BALIKAN
x1 = -0.224324
x2 = 0.182432
x3 = 0.709459
x4 = -0.258108
```

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic
6. Regresi Linier Berganda
7. Keluar
MASUKKAN ANGKA ANTARA 1 HINGGA 7
> 1
TENTUKAN JENIS INPUT
1. Input Keyboard
2. Input File
MASUKKAN ANGKA ANTARA 1 HINGGA 2
> 2
MASUKKAN NAMA FILE
> 3b.txt
SISTEM PERSAMAAN LINIER AKAN DISELESAIKAN DENGAN:
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
MASUKKAN ANGKA ANTARA 1 HINGGA 4
> 1
HASIL DARI SPL TERSEBUT DENGAN METODE GAUSS
Tidak ada solusi.
```

4. Uji coba program untuk interpolasi polinomial

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic
6. Regresi Linier Berganda
7. Keluar
MASUKKAN ANGKA ANTARA 1 HINGGA 7
> 4
TENTUKAN JENIS INPUT
1. Input Keyboard
2. Input File
MASUKKAN ANGKA ANTARA 1 HINGGA 2
> 2
MASUKKAN NAMA FILE
> polinom_a.txt
Persamaan polinomial yang didapatkan dari interpolasi:
f(x) = -4212.434532 x^6 + 7102.399162 x^5-4346.313951 x^4 + 1220.854891 x^3-163.915663 x^2 + 10.276384 x^1-0.184559
Apakah ingin mengtaksir suatu nilai?
1. Ya
2. Tidak
MASUKKAN ANGKA ANTARA 1 HINGGA 2
> 1
Masukkan nilai dari x yang ingin ditaksir
> 0.2
Hasil estimasi dari f(0.200000): 0.130000
```

```
Masukkan nilai dari x yang ingin ditaksir
> 0.55
Hasil estimasi dari f(0.550000): 2.137572
```

```
Masukkan nilai dari x yang ingin ditaksir
> 0.85
Hasil estimasi dari f(0.850000): -66.269639
```

```
Masukkan nilai dari x yang ingin ditaksir
> 1.28
Hasil estimasi dari f(1.280000): -3485.144902
```

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic
6. Regresi Linier Berganda
7. Keluar
MASUKKAN ANGKA ANTARA 1 HINGGA 7
> 4
TENTUKAN JENIS INPUT
1. Input Keyboard
2. Input File
MASUKKAN ANGKA ANTARA 1 HINGGA 2
> 2
MASUKKAN NAMA FILE
> polinom_b.txt
Persamaan polinomial yang didapatkan dari interpolasi:
f(x) = -140993.712249 x^9 + 9372849.239101 x^8-275474539.420669 x^7 + 4695806315.428793 x^6-51131876760.132810 x^5 + 368550807175.535000 x^4-1756810186361.356400 x^3 + 5334203055240.578000 x^2-9346993079173.438000 x^1 + 7187066071
661.201000
Apakah ingin mengtaksir suatu nilai?
1. Ya
2. Tidak
MASUKKAN ANGKA ANTARA 1 HINGGA 2
> 1
Masukkan nilai dari x yang ingin ditaksir
> 9.167
Hasil estimasi dari f(9.167000): -667646.218750
```

```
Masukkan nilai dari x yang ingin ditaksir
> 4.233
Hasil estimasi dari f(4.233000): -1708343623.869331
```

```
Masukkan nilai dari x yang ingin ditaksir
> 8.323
Hasil estimasi dari f(8.323000): 36331.722656
```

```
Masukkan nilai dari x yang ingin ditaksir
> 7.516
Hasil estimasi dari f(7.516000): 53566.808594
```

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic
6. Regresi Linier Berganda
7. Keluar
MASUKKAN ANGKA ANTARA 1 HINGGA 7
> 4
TENTUKAN JENIS INPUT
1. Input Keyboard
2. Input File
MASUKKAN ANGKA ANTARA 1 HINGGA 2
> 2
MASUKKAN NAMA FILE
> polinom_c.txt
Persamaan polinomial yang didapatkan dari interpolasi:
f(x) = 0.236256 x^5-1.421263 x^4 + 3.237110 x^3-3.552679 x^2 + 2.035257 x^1
```

5. Uji coba program untuk *bicubic interpolation*

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic
6. Regresi Linier Berganda
7. Keluar
MASUKKAN ANGKA ANTARA 1 HINGGA 7
> 5
Apakah ingin memperbesar citra(1) atau interpolasi menggunakan matrix(2)?
MASUKKAN ANGKA ANTARA 1 HINGGA 2
> 2
MASUKKAN MATRIKS 4x4!
TENTUKAN JENIS INPUT
1. Input Keyboard
2. Input File
MASUKKAN ANGKA ANTARA 1 HINGGA 2
> 2
MASUKKAN NAMA FILE
> bicubic.txt
MASUKKAN X DAN Y UNTUK DIINTERPOLASI
> 0 0
Berikut hasil interpolasi: 161.0
```

```
MASUKKAN X DAN Y UNTUK DIINTERPOLASI
> 0.5 0.5
Berikut hasil interpolasi: 97.73
```

```
MASUKKAN X DAN Y UNTUK DIINTERPOLASI
> 0.25 0.75
Berikut hasil interpolasi: 105.51
```

```
MASUKKAN X DAN Y UNTUK DIINTERPOLASI
> 0.1 0.9
Berikut hasil interpolasi: 104.23
```

6. Uji coba program untuk regresi linier berganda

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic
6. Regresi Linier Berganda
7. Keluar
MASUKKAN ANGKA ANTARA 1 HINGGA 7
> 6
TENTUKAN JENIS INPUT
1. Input Keyboard
2. Input File
MASUKKAN ANGKA ANTARA 1 HINGGA 2
> 2
MASUKKAN NAMA FILE
> regresi.txt

Diperoleh Matriks SPL sebagai berikut:
[20.0, 863.0999999999999, 1530.4000000000003, 587.8399999999999, 19.42]
[863.0999999999999, 54876.89, 67000.09, 25283.395, 779.4769999999999]
[1530.4000000000003, 67000.09, 117912.32000000002, 44976.866999999984, 1483.4369999999997]
[587.8399999999999, 25283.395, 44976.866999999984, 17278.508600000005, 571.1219000000001]

Persamaan yang didapatkan:
y = -3.507778 - 0.002625 x^1 + 0.000799 x^2 + 0.154155 x^3
Apakah ingin mengtaksir suatu nilai?
1. Ya
2. Tidak
MASUKKAN ANGKA ANTARA 1 HINGGA 2
> 1
Masukkan nilai dari 3 peubahnya
> 50 76 29.3
Hasil estimasi atau taksirannya: 0.938434
```

7. Uji coba program untuk *image upscale* dengan *bicubic interpolation*

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic
6. Regresi Linier Berganda
7. Keluar
MASUKKAN ANGKA ANTARA 1 HINGGA 7
> 5
MENU INTERPOLASI BICUBIC
1. Memperbesar citra
2. Interpolasi matriks
MASUKKAN ANGKA ANTARA 1 HINGGA 2
> 1
MASUKKAN NAMA FILE
> test.jpg
Sedang mengupscale...
Citra berhasil diperbesar!
```

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic
6. Regresi Linier Berganda
7. Keluar
MASUKKAN ANGKA ANTARA 1 HINGGA 7
> 5
MENU INTERPOLASI BICUBIC
1. Memperbesar citra
2. Interpolasi matriks
MASUKKAN ANGKA ANTARA 1 HINGGA 2
> 1
MASUKKAN NAMA FILE
> hina.jpg
Sedang mengupscale...
Citra berhasil diperbesar!
```

Link Repo GitHub
https://github.com/ammarasyad/Algeo01-21136