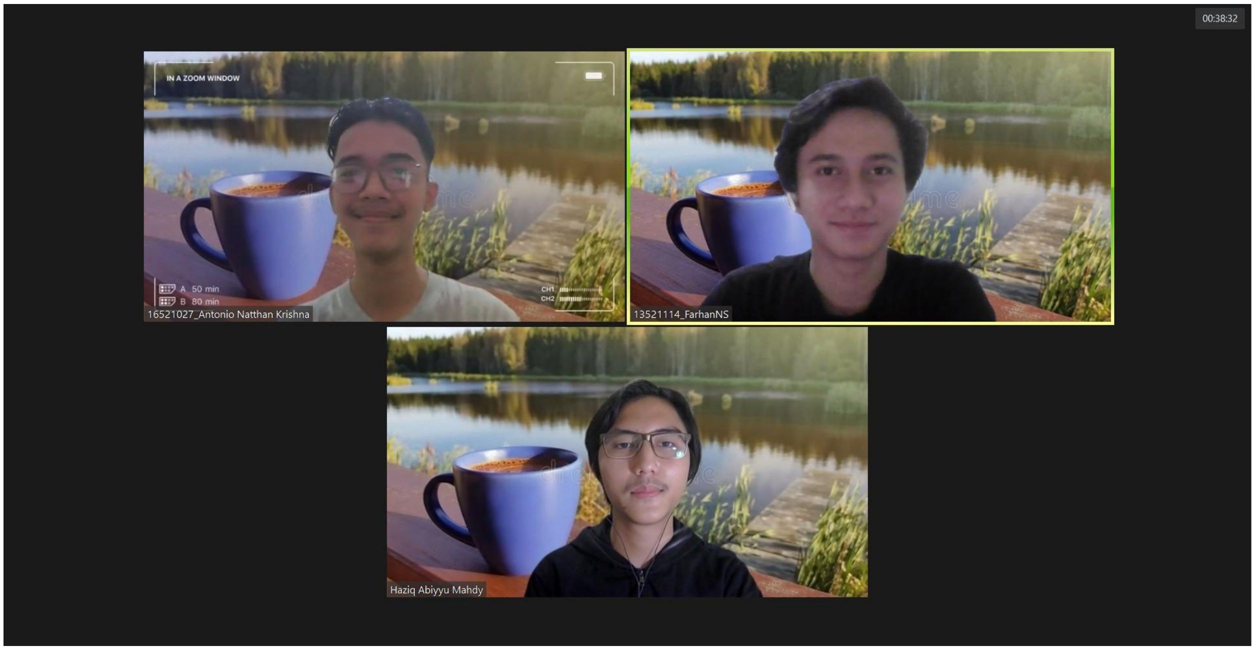
KALKULATOR MATRIKS, REGRESI LINEAR, INTERPOLASI, DAN BICUBIC INTERPOLATION.

Diajukan sebagai pemenuhan tugas besar I.



Oleh:

Kelompok 44 (*Coffee Lake*)

1. 13521114 - Farhan Nabil Suryono
2. 13521162 - Antonio Natthan Krishna
3. 13521170 - Haziq Abiyyu Mahdy

Dosen Pengampu : Dr. Ir. Rinaldi Munir, MT.

IF2123 - Aljabar Linier dan Geometri

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2022

BAB 1
DESKRIPSI MASALAH

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

Di dalam Tugas Besar 1 ini, kami diminta membuat satu atau lebih library aljabar linear 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, kami menggunakan library tersebut di dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi.

BAB 2
TEORI SINGKAT

1. Eliminasi Gauss

Sebelum memahami metode eliminasi Gauss, kita perlu memahami suatu konsep yang bernama matriks eselon baris atau *row echelon form*. Matriks eselon baris adalah matriks yang memiliki 1 utama di setiap barisnya kecuali pada baris yang seluruhnya bernilai 0. Berikut adalah contoh dari matriks eselon baris.

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

Gambar 2.1 Contoh Matriks Eselon Baris

Matriks eselon baris memiliki beberapa sifat yaitu:

1. Jika sebuah baris pada matriks tidak seluruhnya bernilai 0, maka bilangan tidak nol pertama di dalam baris tersebut adalah 1 yang disebut 1 utama.
2. Jika ada baris yang seluruhnya bernilai 0, maka semua baris itu dikumpulkan di bagian bawah baris.
3. Pada 2 baris yang berurutan dimana tidak semuanya bernilai 0, maka 1 utama pada baris yang lebih rendah berada di kolom setidaknya 1 kolom di sebelah kanan 1 utama pada baris yang lebih tinggi.

Selanjutnya, konsep yang perlu dipahami untuk memahami eliminasi gauss ini adalah operasi baris elementer. Operasi baris elementer (OBE) terdiri atas 3 operasi yaitu menukar 2 baris pada suatu matriks, membagi semua nilai pada suatu baris pada matriks dengan suatu konstanta bukan 0, dan menambah nilai suatu baris dengan kelipatan dari baris lainnya.

Lalu, langkah-langkah untuk mencari solusi dari Sistem Persamaan Linear dengan metode Gauss adalah:

1. SPL dinyatakan dalam bentuk matriks *augmented*
2. Pada matriks yang telah terbentuk, terapkan operasi baris elementer sehingga matriks terbentuk menjadi matriks eselon baris
3. Cari solusi dari sistem persamaan linear dengan menggunakan *backward substitution*

Berikut adalah contoh penerapan metode eliminasi Gauss untuk mencari solusi sistem persamaan linear.

2x + 3y - z = 5

4x + 4y - 3z = 3

-2x + 3y - 3z = 1

Penyelesaian:

$$\begin{bmatrix} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{bmatrix} \xrightarrow{\frac{R_1}{2}} \begin{bmatrix} 1 & \frac{3}{2} & -\frac{1}{2} & \frac{5}{2} \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{bmatrix} \xrightarrow[R_3+2R_1]{R_2-4R_1}$$

$$\begin{bmatrix} 1 & \frac{3}{2} & -\frac{1}{2} & \frac{5}{2} \\ 0 & -2 & -1 & -7 \\ 0 & 6 & -2 & 6 \end{bmatrix} \xrightarrow{\frac{R_2}{-2}} \begin{bmatrix} 1 & \frac{3}{2} & -\frac{1}{2} & \frac{5}{2} \\ 0 & 1 & \frac{1}{2} & \frac{7}{2} \\ 0 & 6 & -2 & 6 \end{bmatrix} \xrightarrow{R_3-6R_2}$$

$$\begin{bmatrix} 1 & \frac{3}{2} & -\frac{1}{2} & \frac{5}{2} \\ 0 & 1 & \frac{1}{2} & \frac{7}{2} \\ 0 & 0 & -5 & -15 \end{bmatrix} \xrightarrow{\frac{R_3}{-5}} \begin{bmatrix} 1 & \frac{3}{2} & -\frac{1}{2} & \frac{5}{2} \\ 0 & 1 & \frac{1}{2} & \frac{7}{2} \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

Dari matriks hasil OBE, diperoleh beberapa persamaan linear sebagai berikut:

$x + \frac{3}{2}y - \frac{1}{2}z = \frac{5}{2}$

$y + \frac{1}{2}z = \frac{7}{2}$

$z = 3$

Menggunakan *backward substitution*, didapat solusi dari persamaan linear ini adalah:

z = 3

y = 2

x = 1

2. **Eliminasi Gauss Jordan**

Selain metode eliminasi Gauss, kita bisa menggunakan metode eliminasi Gauss-Jordan untuk mencari solusi dari Sistem Persamaan Linear. Kedua metode ini cukup mirip karena sama-sama memanipulasi matriks *augmented* dari sistem persamaan linear dengan OBE. Beda metode ini dengan metode Gauss adalah matriks pada metode ini berakhir dalam bentuk matriks eselon baris tereduksi atau *reduced row echelon form* sehingga tidak memerlukan *backward substitution*. Hal yang membedakan antara matriks eselon baris tereduksi dengan matriks eselon baris adalah adanya sifat tambahan pada matriks eselon baris tereduksi yaitu setiap kolom yang memiliki 1 utama memiliki nilai 0 pada setiap baris lainnya. Berikut beberapa contoh matriks eselon baris tereduksi.

$$\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix} \quad \begin{bmatrix}1 & 0 & 0 & 4\\0 & 1 & 0 & 7\\0 & 0 & 1 & -1\end{bmatrix} \quad \begin{bmatrix}0 & 1 & -2 & 0 & 1\\0 & 0 & 0 & 1 & 3\\0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0\end{bmatrix} \quad \begin{bmatrix}0 & 0\\0 & 0\end{bmatrix}$$

Gambar 2.2 Contoh Matriks Eselon Baris Tereduksi

Langkah untuk mencari solusi dari Sistem Persamaan Linear dengan metode Gauss-Jordan terdiri atas 2 fase yaitu:

1. Fase maju atau fase eliminasi Gauss untuk membuat nilai 0 di bawah 1 utama
2. Fase mundur untuk membuat nilai-nilai di atas 1 utama menjadi 0

Berikut adalah contoh penggunaan metode Gauss-Jordan untuk menyelesaikan sistem persamaan linear yang terdapat pada contoh metode Gauss:

$$\begin{bmatrix} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{bmatrix} \xrightarrow{OBE} \begin{bmatrix} 1 & \frac{3}{2} & -\frac{1}{2} & \frac{5}{2} \\ 0 & 1 & \frac{1}{2} & \frac{7}{2} \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & \frac{3}{2} & -\frac{1}{2} & \frac{5}{2} \\ 0 & 1 & \frac{1}{2} & \frac{7}{2} \\ 0 & 0 & 1 & 3 \end{bmatrix} \xrightarrow{R_1 - \frac{3}{2}R_2} \begin{bmatrix} 1 & 0 & -\frac{5}{4} & -\frac{11}{4} \\ 0 & 1 & \frac{1}{2} & \frac{7}{2} \\ 0 & 0 & 1 & 3 \end{bmatrix} \xrightarrow[R_2 - \frac{1}{2}R_3]{R_1 + \frac{5}{4}R_3}$$

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

Dari matriks hasil OBE, diperoleh beberapa persamaan linear sebagai berikut:

x = 1

y = 2

z = 3

3. Determinan dengan Reduksi Baris

Salah satu metode untuk memperoleh determinan adalah dengan menggunakan reduksi baris. Pada suatu matriks dapat dilakukan operasi baris elementer sedemikian rupa sehingga matriks tersebut menjadi matriks segitiga atas atau segitiga bawah. Determinan matriks segitiga dapat diperoleh dengan mengalikan seluruh elemen pada diagonal utama matriks.

Terdapat sifat-sifat determinan yang perlu diketahui ketika melakukan operasi baris elementer pada matriks.

1. Jika suatu baris pada matriks ditambahkan dengan kelipatan baris lainnya, maka determinan matriks yang diperoleh akan sama dengan determinan matriks awal.
2. Jika suatu baris pada matriks dipertukarkan dengan baris lainnya, maka determinan matriks yang diperoleh akan sama dengan determinan matriks awal dikalikan dengan -1.
3. Jika suatu baris pada matriks dikalikan dengan suatu skalar k, maka determinan matriks yang diperoleh akan sama dengan determinan matriks awal dikalikan dengan k.

Misalkan A adalah suatu matriks berukuran m×m. Pada matriks A, dilakukan operasi baris elementer sedemikian rupa sehingga menjadi matriks R. Misalkan pada baris matriks dilakukan perkalian dengan skalar k_1, k_2, ..., k_n serta dilakukan pertukaran baris sebanyak p kali, maka dengan memanfaatkan sifat-sifat di atas, kita dapat memperoleh determinan matriks A sebagai berikut.

$$\det(R) = (-1)^p\, k_1 k_2 \ldots k_n \det(A)$$

$$r_{11} r_{12} \ldots r_{mm} = (-1)^p\, k_1 k_2 \ldots k_n\, det(A)$$

$$\det(A) = \frac{(-1)^p\, r_{11} r_{12} \ldots r_{mm}}{k_1 k_2 \ldots k_n}$$

4. Determinan dengan Ekspansi Kofaktor

Selain dengan reduksi baris, determinan matriks juga dapat dihitung dengan metode ekspansi kofaktor. Pada suatu matriks A dengan ukuran m×m, didefinisikan minor entri M_{ij} sebagai determinan submatriks A dengan baris i dan kolom j dari matriks A dihilangkan.

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

Kofaktor C_{ij} dapat diperoleh dengan rumus berikut.

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

Kita dapat menghitung determinan matriks A dengan memilih salah satu baris atau kolom, kemudian menggunakan rumus berikut.
Secara baris;

$$\det(A) = \ a_{n1}C_{n1} + a_{n2}C_{n2} + \ldots + a_{nn}C_{nn}$$

Secara kolom;

$$\det(A) = \ a_{1n}C_{1n} + a_{2n}C_{2n} + \ldots + a_{nn}C_{nn}$$

Dengan n adalah sembarang baris atau kolom yang dipilih.

5. **Metode Cramer**

Metode Cramer merupakan salah satu cara singkat untuk memecahkan Sistem Persamaan Linear (SPL) hanya untuk variabel tertentu di dalamnya tanpa harus memecahkan seluruh variabel dalam persamaan tersebut. Misalkan kita mempunyai SPL sebagai berikut,

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} j \\ k \\ l \end{bmatrix}$$

Daripada melakukan metode Gauss atau Gauss-Jordan untuk menyelesaikan persamaannya, kita dapat menggunakan cara Cramer ini. Langkah pertama, cari determinan matriks SPL,

$$\det(a) = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}$$

Misalkan kita hanya memerlukan nilai dari variabel y. Kita akan mengganti kolom variabel y, pada matriks, dengan hasil SPL, sesuai dengan penggambaran berikut,

$$\begin{bmatrix} a & j & c \\ d & k & f \\ g & l & i \end{bmatrix}$$

Selanjutnya, kita juga akan mencari determinan dari matriks buatan kita,

$$\det(a_y) = \begin{vmatrix} a & j & c \\ d & k & f \\ g & l & i \end{vmatrix}$$

Nilai y pada metode Cramer didefinisikan sebagai pembagian nilai determinan matriks buatan dengan nilai determinan matriks SPL,

$$y = \frac{\det(a_y)}{\det(a)}$$

Perlu diingat, bahwa metode ini efektif jika kita hanya mencari salah satu dari variabel variabel yang ada. Tentunya akan memakan banyak waktu jika kita memakai metode ini untuk seluruh variabel yang ada pada SPL. Pada kasus tersebut, lebih baik untuk mempertimbangkan metode lain dalam penyelesaiannya.

6. **Invers Matriks**

Matriks Invers merupakan matriks yang apabila matriks tersebut dikalikan dengan matriks sekawannya akan menghasilkan matriks identitas.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} j & k & l \\ m & n & o \\ p & q & r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Dalam kasus ini, dapat dikatakan bahwa,

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = \begin{bmatrix} j & k & l \\ m & n & o \\ p & q & r \end{bmatrix}^{-1} \; atau \; \begin{bmatrix} j & k & l \\ m & n & o \\ p & q & r \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^{-1}$$

Misalkan kita mempunyai matriks,

$$\begin{bmatrix} a & j & c \\ d & k & f \\ g & l & i \end{bmatrix}$$

Kita akan mengubah bentuk matriks ini menjadi,

$$\left[\begin{array}{ccc|ccc} a & b & c & 1 & 0 & 0 \\ d & e & f & 0 & 1 & 0 \\ g & h & i & 0 & 0 & 1 \end{array}\right]$$

Dengan menerapkan OBE menggunakan metode Gauss-Jordan, sehingga mendapatkan sebuah matriks berbentuk,

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & j & k & l \\ 0 & 1 & 0 & m & n & o \\ 0 & 0 & 1 & p & q & r \end{array}\right]$$

Dengan itu, maka didapatlah sebuah matriks invers yang bernilai,

$$\begin{bmatrix} j & k & l \\ m & n & o \\ p & q & r \end{bmatrix}$$

7. Interpolasi Polinom

Persoalan interpolasi polinom adalah sebagai berikut: Diberikan n+1 buah titik berbeda, $(x_0\ , y_0)$, $(x_1\ , y_1)$, ..., $(x_n\ , y_n)$. Tentukan polinom $p_n(x)$ yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga $y_i = p_n(x_i)$ untuk i = 0, 1, 2, …, n.

Setelah polinom interpolasi $p_n(x)$ ditemukan, $p_n(x)$ dapat digunakan untuk menghitung perkiraan nilai y di sembarang titik di dalam selang $[x_0\ , x_n]$. Polinom interpolasi derajat n yang menginterpolasi titik-titik $(x_0\ , y_0)$, $(x_1\ , y_1)$, ..., $(x_n\ , y_n)$. adalah berbentuk $p_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$. Jika hanya ada dua titik, $(x_0\ , y_{0)}$ dan $(x_1\ , y_1)$, maka polinom yang menginterpolasi kedua titik tersebut adalah $p_1(x) = a_0 + a_1x$ yaitu berupa persamaan garis lurus. Jika tersedia tiga titik, $(x_0\ , y_0)$, $(x_1\ , y_1)$, dan $(x_2\ , y_2)$, maka polinom yang menginterpolasi ketiga titik tersebut adalah $p_2(x) = a_0 + a_1x + a_2x2$ atau persaman kuadrat dan kurvanya berupa parabola. Jika tersedia empat titik, (x_0 , y_0), (x_1 , y_1), (x_2 , y_2), dan (x_3 , y_3), polinom yang menginterpolasi keempat titik tersebut adalah $p_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3$, demikian seterusnya. Dengan cara yang sama kita dapat membuat polinom interpolasi berderajat n untuk n yang lebih tinggi asalkan tersedia (n+1) buah titik data. Dengan menyulihkan (x_i , y_i) ke dalam persamaan polinom $p_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$ untuk i = 0, 1, 2, …, n, akan diperoleh n buah sistem persamaan lanjar dalam $a_0 , a_1 , a_2 , \ldots, a_n.$

$$\begin{array}{c}
a_0 + a_1x_0 + a_2x_0^{\,2} + \ldots + a_n\, x_0^{\,n} = y_0 \\
a_0 + a_1x_1 + a_2x_1^{\,2} + \ldots + a_n\, x_1^{\,n} = y_1 \\
\ldots \qquad\qquad \ldots \\
a_0 + a_1x_n + a_2x_n^{\,2} + \ldots + a_n\, x_n^{\,n} = y_n
\end{array}$$

Solusi sistem persamaan lanjar ini, yaitu nilai a_0 , a_1 , …, a_n , diperoleh dengan menggunakan metode eliminasi Gauss.

8. Bicubic Interpolation

Bicubic interpolation merupakan teknik interpolasi pada data 2D umumnya digunakan dalam pembesaran citra yang merupakan pengembangan dari interpolasi linear dan cubic yang telah dipelajari pada kuliah metode numerik di aljabar geometri.

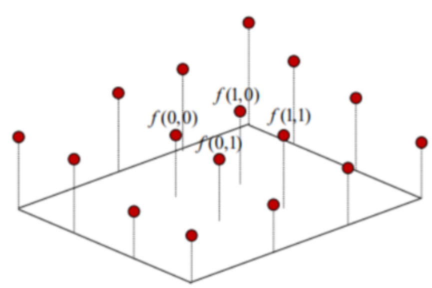
Diberikan sebuah matrix awal, misal M, kita akan mencari persamaan interpolasi f(x,y) dengan pemodelan sebagai berikut:

Normalization: $f(0,0), f(1,0)$
$f(0,1), f(1,1)$

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}



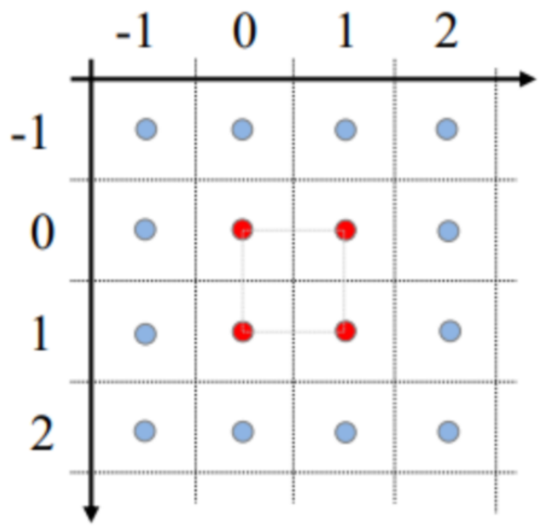
Melakukan substitusi nilai-nilai diketahui pada matriks 4 x 4 tersebut ke persamaan f(x,y) akan menghasilkan sebuah matriks persamaan:

$$y = Xa$$

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

Elemen pada matrix X adalah koefisien a_{ij} yang diperoleh dari persamaan f(x,y) di atas. Sebagai contoh, elemen pada baris 4 kolom ke 10 adalah koefisien dari a_{12} dan diperoleh dari $2^1*(-1)^2 = 2$, sesuai persamaan $x^i * y^j$.

Vektor a dapat dicari dari persamaan tersebut (menggunakan inverse), lalu vektor a digunakan sebagai nilai variabel dalam f(x,y). Sehingga terbentuk fungsi interpolasi bicubic sesuai model. Tugas Anda adalah menentukan persamaan f(x,y) lalu melakukan interpolasi berdasarkan f(a,b) dari masukan matriks 4 x 4. Nilai masukan a dan b dalam rentang [0,1] (Referensi gambar di bawah, nilai untuk diinterpolasi dalam kotak merah).



9. Regresi Linear Berganda

Regresi Linear (akan dipelajari lebih lanjut di Probabilitas dan Statistika) merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Meskipun sudah ada rumus jadi untuk menghitung regresi linear sederhana, terdapat rumus umum dari regresi linear yang bisa digunakan untuk regresi linear berganda, yaitu,

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

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

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

Sistem persamaan linear tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

Bab 3
Implementasi Pustaka dan Program

Class Matrix

a. Attributes

Nama	Tipe	Deskripsi
matrix	double[][]	Array of array of double sebagai struktur data pembangun matriks
row	int	Jumlah baris pada matriks
column	int	Jumlah baris pada kolom
MARK	final int	Nilai default pada matriks yang menandakan bahwa elemen tersebut belum terisi

b. Methods

Nama	Tipe	Parameter	Deskripsi
createMatrix	Public Constructor	Int row Int column	Membuat matriks dengan ukuran row x column lalu mengisi semua elemennya dengan MARK
readMatrixFromFile	Public Constructor	String filePath	Membuat matriks berdasarkan masukan berupa file
readMatrixPointFromKeyboard	Public Constructor	-	Membuat matriks of point dengan masukan banyak jumlah titik dan absis serta ordinat setiap titiknya
readMatrixPointFromFile	Public Constructor	String filePath	Membuat matriks of point berdasarkan masukan berupa file
getElement	Public Double	Int numRow Int numCol	Mengembalikan elemen matriks pada baris ke numRow dan kolom ke numCol. Indeks dimulai dari 1
getRow	Public Double[]	Int numRow	Mengembalikan array berupa sebaris matrix yaitu pada baris numRow
getColumn	Public Double[]	Int numCol	Mengembalikan array berupa sekolom matrix yaitu pada kolom

			numCol
getRowLength	Public Int	-	Mengembalikan jumlah baris pada matriks
getColumnLength	Public Int	-	Mengembalikan jumlah kolom pada matriks
setElement	Public Void	Double val Int numRow Int numCol	Mengubah nilai matriks pada baris numRow dan kolom numCol menjadi val
getMatrixPoint	Public Double[][]	-	Mengembalikan matriks Point
getCountPoint	Public Int	-	Mengembalikan jumlah Point
getMatrix	Public Double[][]	-	Mengembalikan matriks
writeMatrix	Public Void	-	Meminta masukan pilihan metode penulisan dan memanggil fungsi penulisan matriks
writeTerminal	Public Void	-	Menuliskan elemen elemen matriks pada terminal
writeFile	Public Void	-	Meminta masukan path folder tujuan dan namafile lalu menuliskan matriks pada file tersebut
makeZero	Public Double	Double Element Double leadElement	Mengembalikan nilai konstanta agar dapat menjadi 0 untuk OBE
isSquare	Public Boolean	-	Mengembalikan nilai true bila matriks berbentuk persegi (jumlah baris sama dengan jumlah kolom)
countElement	Public Int	-	Mengembalikan jumlah elemen pada matriks
swapRow	Public Void	Int a Int b	Menukar baris a dengan baris b pada matriks untuk proses OBE
transpose	Public	-	Melakukan transpose pada

	Void		matriks
OBE	Public Void	Int i Int idxrow Double zero	Melakukan operasi OBE yaitu pengurangan baris idxrow dengan baris i yang telah dikalikan nilai zero pada matriks.
divideRow	Public Void	Int i Double val	Membagi seluruh elemen pada baris i matriks dengan nilai val
roundMatElmt	Public Void	Int sc	Mengembalikan matriks yang telah dibulatkan sebanyak sc angka di belakang koma

Class Point

a. Attributes

Nama	Tipe	Deskripsi
x	double	Nilai absis Point
y	double	Nilai ordinat Point

b. Methods

Nama	Tipe	Parameter	Deskripsi
readPoint	Public Constructor	-	Membuat sebuah Point dengan meminta masukan berupa absis dan ordinat titik
setAbsis	Public Void	Double val	Mengubah nilai absis pada Point dengan val
setOrdinat	Public Void	Double val	Mengubah nilai ordinat pada Point dengan val
getPointAbsis	Public Double	-	Mengembalikan nilai absis pada Point
getPointOrdinat	Public Double	-	Mengembalikan nilai ordinat pada Point
writePoint	Public Void	-	Menuliskan Point pada terminal
isOrigin	Public Boolean	-	Mengembalikan nilai true jika titik berada di titik (0,0)

isOnSbX	Public Boolean	-	Mengembalikan nilai true jika titik berada di sumbu x
isOnSbY	Public Boolean	-	Mengembalikan nilai true jika titik berada di sumbu y
Kuadran	Public Int	-	Mengembalikan kuadran posiis titik
plusDelta	Public Void	Double deltaX Double deltaY	Menambah masing-masing nilai x dan y pada point dengan deltaX dan deltaY
distancefromOrigin	Public Double	-	Mengembalikan jarak titik dari titik origin
moveToSbX	Public Void	-	Mengubah nilai y titik menjadi 0
moveToSbY	Public Void	-	Mengubah nilai x titik menjadi 0
mirror	Public Void	Boolean toSbX	Merefleksikan point dengan sumbu x bila toSbX true dan dengan sumbu y bila toSbX false
PutartoOrigin	Public Void	Double sudutDerajat	Memutar titik sebesar sudut dengan titik origin sebagai acuan

Class Operations

a. Attributes

Class ini tidak memiliki attribute

b. Methods

Nama	Tipe	Parameter	Deskripsi
copyMatrix	Public Static Matrix	Matrix mat	Mengembalikan matriks hasil copy dari matrix mat
multiplyMatrix	Public Static Matrix	Matrix m1 Matrix m2	Mengembalikan matriks hasil perkalian m1 dengan m2

Class Bicubic

a. Attributes

Class ini tidak memiliki attribute

b. Methods

Nama	Tipe	Parameter	Deskripsi
generateInverse MatrixX	Public Static Matrix	-	Mengembalikan invers matriks dari matriks X untuk proses interpolasi bikubik
convMatrix	Public Static Matrix	Matrix m	Mengembalikan matriks hasil perubahan dari 4x4 menjadi 16x1
getBicubicCof	Public Static Matrix	Matrix invX Matrix m	Mengembalikan matriks koefisien persamaan bikubik yaitu hasil perkalian inverse matrix X dengan matrix input yang telah dikonversi
calcBicubic	Public Static Double	Matrix m Double x Double y	Mengembalikan hasil interpolasi bikubik pada titik (x, y) pada matriks 4x4 m

Class Cramer

a. Attributes

Nama	Tipe	Deskripsi
noSol	Static Final double[]	Array berisi mark yang akan dikembalikan oleh method cramerSPL() jika suatu sistem persamaan linear tidak memiliki solusi unik.

b. Methods

Nama	Tipe	Parameter	Deskripsi
cramerSPL	Public Static Double[]	Matrix augmentedMatrix, int var	Mengembalikan solusi sistem persamaan linear berupa array [x_1, x_2, …, x_{var}] dari input berupa matriks augmented dan jumlah variabel. Sistem persamaan diselesaikan dengan dengan kaidah Cramer.
displaySPLCramer	Public	double[] sol	Menampilkan solusi sistem

Result	Static Void		persamaan linear ke layar. Jika determinan matriks koefisien = 0, maka akan menampilkan pesan kesalahan di layar.

Class Determinant

a. Attributes

Nama	Tipe	Deskripsi
UNDEF_DET	Static Final Double	Merupakan nilai mark yang akan dikembalikan oleh method detExCof() jika suatu matriks tidak memiliki determinan (bukan matriks persegi).

b. Methods

Nama	Tipe	Parameter	Deskripsi
detExCof	Public Static Double	Matrix m, int size	Mengembalikan determinan suatu matriks dengan metode ekspansi kofaktor.
minor	Public Static Matrix	Matrix m, int brs, int klm, int size	Mengembalikan submatriks berupa minor matriks ke-(brs, klm).
cofactor	Public Static Double	Matrix m, int brs, int klm, int size	Mengembalikan kofaktor matriks ke-(brs,klm)

Class detRedux

a. Attributes

Class ini tidak memiliki attribute

b. Methods

Nama	Tipe	Parameter	Deskripsi
detByReduction	Public static double	Matrix inMatrix	Mengembalikan nilai determinan menggunakan metode reduksi baris.
detReduxMatrix	Public Static Matrix	Matrix inMatrix	Mengembalikan matriks hasil reduksi baris

Class Gauss

a. Attributes

Class ini tidak memiliki attribute

b. Methods

Nama	Tipe	Parameter	Deskripsi
matrixGauss	Public Static Matrix	Matrix mat	Mengembalikan matriks hasil OBE sesuai dengan metode Gauss
matrixGaussJordan	Public Static Matrix	Matrix mat	Mengembalikan matriks hasil OBE sesuai dengan metode Gauss-Jordan
indexNotZeroInCol	Public Static Int	Matrix m Int numCol Int rowMin	Mengembalikan index bukan 0 teratas pada kolom numCol mulai dari baris rowMin hingga baris terbawah
solveSPL	Public Static Void	Matrix m	Memproses dan melakukan output solusi SPL berdasarkan masukan matrix hasil Gauss atau Gauss-Jordan
writeSPLFile	Public Static Void	FileWriter fileWriter Matrix m String filePath	Memproses dan melakukan output solusi SPL lalu menuliskannya dalam bentuk file pada filePath
inverseGauss	Public Static Matrix	Matrix m	Mengembalikan invers dari matriks dengan metode Gauss-Jordan
identityMatrix	Public Static Matrix	Matrix m	Membuat matriks identitas yang seukuran dengan Matrix m

Class Interpolasi_polinom

a. Attributes

Class ini tidak memiliki attribute.

b. Methods

Nama	Tipe	Parameter	Deskripsi
interpolasi	Public Static	Matrix givenData	Menerima parameter berupa titik-titik yang akan diinterpolasi

	Matrix		(dalam tipe Matrix), kemudian mengembalikan matriks yang berisi koefisien polinom interpolasi.
displayPolinom	Public Static Void	Matrix solution	Menerima parameter berupa matriks koefisien polinom yang diperoleh dari method interpolasi() dan menampilkan fungsi polinom di layar.
displayValueOfFx	Public Static Void	Matrix solution, Matrix findValueOf	Menerima parameter berupa matriks koefisien polinom yang diperoleh dari method interpolasi() dan matriks berisi nilai x yang akan diaproksimasi, lalu menampilkan di layar hasil aproksimasi f(x).

Class Inverse

a. Attributes

Nama	Tipe	Deskripsi
noInv	Static Final Matrix	Nilai konstanta yang akan dikembalikan oleh method inverse() jika suatu matriks tidak memiliki balikan.
noSol	Static Final double[]	Array berisi mark yang akan dikembalikan oleh method inverseSPL() jika suatu sistem persamaan linear tidak memiliki solusi unik.

b. Methods

Nama	Tipe	Parameter	Deskripsi
inverse	Public Static Matrix	Matrix m, int size	Mengembalikan balikan dari matriks m dengan menggunakan adjoin.
inverseSPL	Public Static Matrix	Matrix augmentedMatrix, int var	Mengembalikan solusi sistem persamaan linear berupa array [x_1, x_2, …, x_{var}] dari input berupa matriks augmented dan jumlah variabel. Sistem persamaan diselesaikan dengan dengan metode balikan.

displayInverse	Public Static Void	Matrix inv	Menampilkan hasil balikan matriks ke layar. Jika matriks tidak memiliki balikan, maka akan menampilkan pesan kesalahan di layar.
displayInvSPLResult	Public Static Void	double[] sol	Menampilkan solusi sistem persamaan linear ke layar. Jika matriks tidak memiliki balikan, maka akan menampilkan pesan kesalahan di layar.

Class Regression

a. Attributes

Class ini tidak memiliki attribute

b. Methods

Nama	Tipe	Parameter	Deskripsi
evaluateRegressionEquation	Public static double[]	Matrix matrixDataSample, Matrix matrixDataPredict	Mengembalikan konstanta konstanta persamaan regresi di dalam array.
evaluateRegressionValue	Public static double[]	Double[] regressionEquation, Matrix matrixDataPredict	Mengembalikan nilai taksiran berdasarkan hasil persamaan regresi.

Class ScaleImage

a. Attributes

Class ini tidak memiliki attribute

b. Methods

Nama	Tipe	Parameter	Deskripsi
readImage	Public Static BufferedImage	-	Membaca file gambar dan mengembalikan gambar tersebut dalam bentuk BufferedImage
canvasScaledImage	Public Static Matrix	Matrix mimg	Membuat matrix baru sebagai kanvas untuk matrix data gambar yang telah

			discale 2 kali
scaleImage	Public Static Void	BufferedImage mimg	Membuat gambar hasil perbesar 2 kali dan mempertahankan kualitas dengan proses bicubic interpolation
addPadding	Public Static Matrix	Matrix m	Menambah layer tambahan di matrix sebelum melakukan interpolasi bikubik untuk interpolasi di bagian pinggir

Bab 4
Eksperimen

1. Solusi SPL Ax = B

No	Matrix	Solusi
a)	$A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$	`SPL tidak memiliki solusi`
b)	$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, B = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$	Solusi SPL: x1 = 3.0 + b x2 = 2.0b x3 = a x4 = -1.0 + b x5 = b
c)	$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$	Solusi SPL: x1 = a x2 = 1.0 - c x3 = b x4 = -2.0 - c x5 = 1.0 + c x6 = c
d)	Matriks Hilbert untuk n = 6 dan n = 10	n = 6

		Solusi SPL: x1 = 36.0 x2 = -630.0 x3 = 3360.0 x4 = -7560.0 x5 = 7560.0 x6 = -2772.0 n = 10 Solusi SPL: x1 = 100.0 x2 = -4949.68 x3 = 79193.21 x4 = -600538.14 x5 = 2522224.26 x6 = -6305485.55 x7 = 9608261.78 x8 = -8750305.21 x9 = 4375119.57 x10 = -923630.23

2. SPL berbentuk matriks augmented

No	Matriks	Solusi
a)	$A = \begin{bmatrix} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{bmatrix}$	Solusi SPL: x1 = -1.0 + b x2 = 2.0a x3 = a x4 = b

b)	$A = \begin{bmatrix} 2 & 0 & 8 & 0 & 8 \\ 0 & 1 & 0 & 4 & 6 \\ -4 & 0 & 6 & 0 & 6 \\ 0 & -2 & 0 & 3 & -1 \\ 2 & 0 & -4 & 0 & -4 \\ 0 & 1 & 0 & -2 & 0 \end{bmatrix}$	Solusi SPL: x1 = 0.0 x2 = 2.0 x3 = 1.0 x4 = 1.0

3. SPL berbentuk persamaan

No	Persamaan	Solusi
a)	$A = \begin{bmatrix} 8 & 1 & 3 & 2 \\ 2 & 9 & -1 & 2 \\ 1 & 3 & 2 & -1 \\ 1 & 0 & 6 & 4 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix}$	Solusi SPL: x1 = -0.24 x2 = 0.27 x3 = 0.64 x4 = -0.15

b)	$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.04289 & 0 & 0.04289 & 0.75 & 0.04289 & 0.75 & 0.61396 \\ 0 & 0.25 & 0.91421 & 0.25 & 0.91421 & 0.25 & 0.91421 & 0.25 & 0 \\ 0.61396 & 0.75 & 0.04289 & 0.75 & 0.04289 & 0 & 0.04289 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0.04289 & 0.75 & 0.61396 & 0 & 0.04289 & 0.75 & 0 & 0 & 0.04289 \\ 0.91421 & 0.25 & 0 & 0.25 & 0.91421 & 0.25 & 0 & 0.25 & 0.91421 \\ 0.04289 & 0 & 0 & 0.75 & 0.04289 & 0 & 0.61396 & 0.75 & 0.04289 \end{bmatrix}, B = \begin{bmatrix} 13 \\ 15 \\ 8 \\ 14.79 \\ 14.31 \\ 3.81 \\ 18 \\ 12 \\ 6 \\ 10.51 \\ 16.13 \\ 7.04 \end{bmatrix}$
Solusi	SPL tidak memiliki solusi

4. Studi kasus Interpolasi Polinom

Kasus 1

Data	x	0.4	0.7	0.11	0.14	0.17	0.2	0.23

<table>
<tr><td></td><td>

f(x)	0.043	0.005	0.058	0.072	0.1	0.13	0.147

</td></tr>
<tr><td>Hasil studi kasus</td><td>

```
---------------------------------
HASIL PERHITUNGAN
1. Fungsi Polinomial
f(x) = -0.18456 + (10.27638)x - (163.91566)x^2 + (1220.85489)x^3
- (4346.31395)x^4 + (7102.39916)x^5 - (4212.43453)x^6
2. Hasil Taksiran berdasarkan Fungsi Polinomial
f(0.2) = 0.13
f(0.55) = 2.13757
f(0.85) = -66.26964
f(1.28) = -3485.1449
---------------------------------
```

</td></tr>
</table>

Kasus II

<table>
<tr><td>Data</td><td>

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

</td></tr>
</table>

Hasil studi kasus

```
----------------------------------
HASIL PERHITUNGAN
1. Fungsi Polinomial
f(x) = 7.187066071652806E12 - (9.346993079163879E12)x + (5.334203055235448E12
)x^2 - (1.7568101863598044E12)x^3
+ (3.6855080717524054E11)x^4 - (5.113187676009309E10)x^5 + (4.69580631542635E
9)x^6 - (2.7547453942044E8)x^7 + (9372849.23881)x^8
- (140993.71196)x^9
2. Hasil Taksiran berdasarkan Fungsi Polinomial
f(7.516) = 73007.91992
f(8.323) = 85548.82422
f(9.167) = -549093.6875
----------------------------------
```

Kasus III

Data

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

Diambil 6 titik dalam selang [0, 2] dengan jarak antar titik sebesar 0.4 satuan

x	f(x)
0.0	0.0
0.4	0.418884230141255
0.8	0.5071579685304317
1.2	0.5609246748146806
1.6	0.5836856612868684
2.0	0.5766515297517221

Hasil studi kasus

```
---------------------------------
HASIL PERHITUNGAN
1. Fungsi Polinomial
f(x) = + (2.03526)x - (3.55268)x^2 + (3.23711)x^3
- (1.42126)x^4 + (0.23626)x^5
2. Hasil Taksiran berdasarkan Fungsi Polinomial
f(0.1) = 0.1711
---------------------------------
```

5. Studi kasus Interpolasi Bicubic

Matriks

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

Hasil

```
HASIL PERHITUNGAN
1. Hasil Taksiran Menggunakan Interpolasi Bicubic
f(0.000000, 0.000000) = 161.0
f(0.500000, 0.500000) = 97.72656281531255
f(0.250000, 0.750000) = 105.51477075468749
f(0.100000, 0.900000) = 104.22911861910521
```

6. Studi kasus Regresi Linear Berganda

Kasus

Table 12.1: Data for Example 12.1

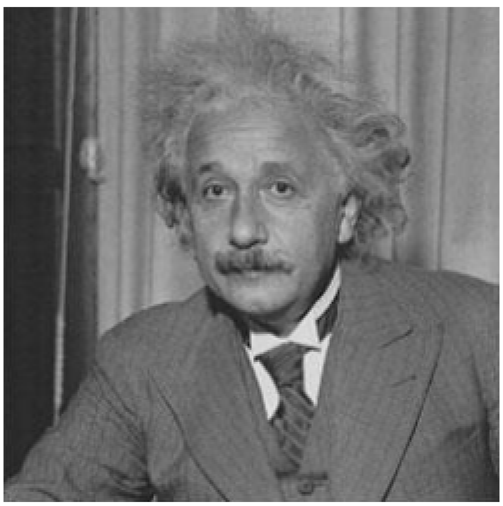
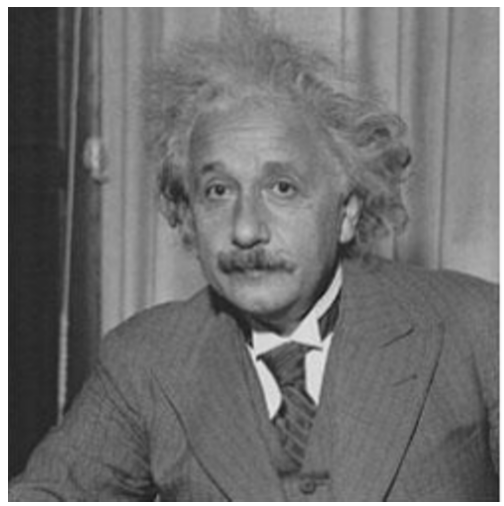
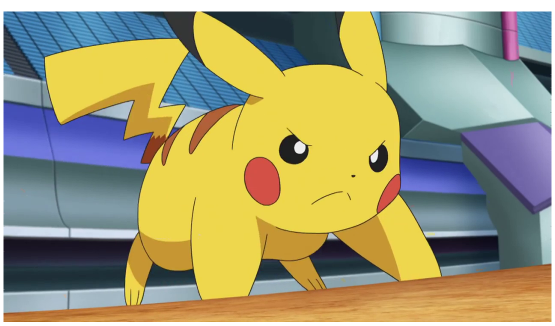
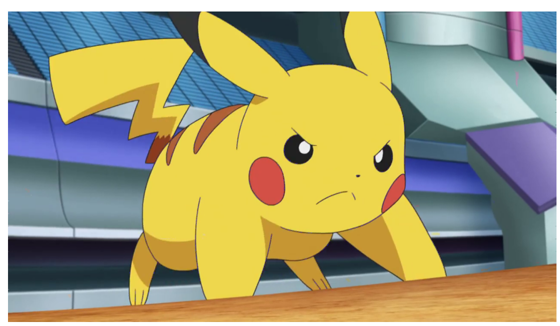
Nitrous Oxide, y	Humidity, x_1	Temp., x_2	Pressure, x_3	Nitrous Oxide, y	Humidity, x_1	Temp., x_2	Pressure, x_3
0.90	72.4	76.3	29.18	1.07	23.2	76.8	29.38
0.91	41.6	70.3	29.35	0.94	47.4	86.6	29.35
0.96	34.3	77.1	29.24	1.10	31.5	76.9	29.63
0.89	35.1	68.0	29.27	1.10	10.6	86.3	29.56
1.00	10.7	79.0	29.78	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.

Hasil

`----------------------------------`
 `HASIL PERHITUNGAN`
 `1. Persamaan Regresi`
 `y = -3.507778141545714 -0.0026249907453333733*(x1) +7.989410528578544E-4*(x2) +0.15415503025496946*(x3)`
 `2. Hasil Taksiran Berdasarkan Persamaan Regresi`
 ` f(50.0, 76.0, 29.3) = 0.9384342276754021`
 `----------------------------------`

7. Studi kasus Image Scaling

Gambar dan Ukuran Sebelum Dilakukan Scaling Image	Gambar dan Ukuran Setelah Dilakukan Scaling Image
 249x247 pixels	 498x494 pixels
 1280x720 pixels	 2560x1440 pixels

Bab 5
Kesimpulan, Saran, dan Refleksi

Proyek ini sangat membantu kami dalam mengeksplorasi properti operasi-operasi matriks dan secara langsung mengimplementasikannya pada program. Proyek kalkulator matriks ini ditulis dengan menggunakan bahasa Java yang mana bahasa tersebut merupakan bahasa yang baru kami pelajari selama 2 minggu terakhir secara mandiri. Oleh karena itu, kami tentunya sangat bangga terhadap diri kami karena telah menyelesaikan seluruh spesifikasi tugas yang diberikan dengan sangat baik, termasuk spesifikasi bonus.

Seluruh fitur dalam program ini sudah berjalan dengan baik. Namun, kami masih sadar bahwa terdapat banyak kekurangan dari proyek ini. Salah satu kekurangan dari program kami adalah kurangnya *prompt* yang diberikan pada pengguna. Meskipun *prompt* yang diberikan sudah cukup baik, namun kami berharap pengguna sudah mengetahui alur kerja program kami terlebih dahulu sebelum menggunakan program kami.

Kami menyarankan agar di penugasan proyek ini di tahun mendatang, spesifikasi program lebih dibuat dengan lebih rinci. Masih terdapat bagian-bagian dalam spesifikasi penugasan yang memiliki kalimat yang ambigu. Selain itu, sumber pembuatan file JAR yang dilampirkan pada spesifikasi laporan tidak terlalu membantu kami dalam penyelesaian proyek ini sehingga kami perlu lagi melakukan eksplorasi mandiri terkait hal tersebut.

Sebagai tambahan, sebaiknya diberikan solusi pada bagian *test-case* yang diberikan pada spesifikasi program. Hal itu bertujuan supaya kami dapat memverifikasi kebenaran program yang kami buat dengan terpusat. Hal ini juga bertujuan untuk mengurangi miskonsepsi terkait *test-case* yang diberikan dan membuat kami lebih percaya diri atas kebenaran program kami.

Akhir kata, sekali lagi kami persembahkan proyek ini yang telah susah payah kami kerjakan selama *weekend* 3 minggu terakhir ini. Semoga dengan hilangnya *weekend* kami, eksplorasi yang telah kami lakukan dapat berguna di kehidupan kami mendatang. Terima kasih kepada Farhan, Haziq dan Nate beserta Bapak Rinaldi dan para asisten Aljabar Linier dan Geometri Tahun 2022 yang telah memberikan kami pengalaman tubes yang lumayan menyenangkan.

REFERENSI

https://www.analyzemath.com/linear-algebra/determinants/find-determinant-using-row-reduction.html
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-03-Sistem-Persamaan-Linier.pdf
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-05-Sistem-Persamaan-Linier-2.pdf
https://people.math.carleton.ca/~kcheung/math/notes/MATH1107/wk07/07_computing_via_row_reduction.html
https://www.baeldung.com/java-create-jar
https://www.purplemath.com/modules/cramers.htm
https://www.mssc.mu.edu/~daniel/pubs/RoweTalkMSCS_BiCubic.pdf
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-08-Determinan-bagian1.pdf
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-09-Determinan-bagian2.pdf

LINK REPOSITORY

https://github.com/Altair1618/Algeo01-21114