LAPORAN TUGAS BESAR 1
IF2123 ALJABAR LINIER DAN GEOMETRI
SEMESTER I 2022/2023
SISTEM PERSAMAAN LINEAR, DETERMINAN, DAN APLIKASINYA

Disusun oleh:

(Kelompok 6 – Yato)

Eugene Yap Jin Quan	13521074
Tobias Natalio Sianipar	13521090
Akbar Maulana Ridho	13521093

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

DAFTAR ISI

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1. 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 ($x = A^{-1}b$), 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 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.

1.2. Spesifikasi Tugas

Berikut adalah spesifikasi dari Tugas Besar 1.

A. Buatlah pustaka dalam Bahasa Java untuk menemukan solusi SPL dengan metode eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n peubah dan n persamaan), menghitung determinan matriks dengan reduksi baris dan dengan ekspansi kofaktor, dan menghitung balikan matriks.
B. Gunakan pustaka di atas untuk membuat program penyelesaian berbagai persoalan dalam bentuk SPL, menyelesaikan persoalan interpolasi dan regresi linier, menghitung matriks balikan, menghitung determinan matriks dengan berbagai metode (reduksi baris dan ekspansi kofaktor). Spesifikasi program adalah sebagai berikut:
    a. Program dapat menerima masukan (*input*) baik dari *keyboard* maupun membaca masukan dari *file text*. Untuk SPL, masukan dari *keyboard* adalah *m*, *n*, koefisien a_{ij}, dan b_i. Masukan dari *file* berbentuk matriks *augmented* tanpa tanda kurung, setiap elemen matriks dipisah oleh spasi. Misalnya,
    ```
    3 4.5 2.8 10 12
    -3 7 8.3 11 -4
    0.5 -10 -9 12 0
    ```
    b. Untuk persoalan menghitung determinan dan matriks balikan, masukan dari *keyboard* adalah *n* dan koefisien a_{ij}. Masukan dari *file* berbentuk matriks, setiap elemen matriks dipisah oleh spasi. Misalnya,
    ```
    3 4.5 2.8
    -3 7 8.3
    0.5 -10 -9
    ```
    c. Untuk persoalan interpolasi, masukannya jika dari *keyboard* adalah n, (x_0, y_0), (x_1, y_1), ..., (x_n, y_n), dan nilai *x* yang akan ditaksir nilai fungsinya. Jika

masukannya dari *file*, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung. Misalnya jika titik-titik datanya adalah (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513), maka di dalam file text ditulis sebagai berikut:

```
8.0 2.0794
9.0 2.1972
9.5 2.2513
```

d. Untuk persoalan regresi, masukannya jika dari *keyboard* adalah *n* (jumlah peubah *x*), *m* (jumlah sampel), semua nilai-nilai $x_{1i}, x_{2i}, ..., x_{ni}$, nilai y_i, dan nilai-nilai x_k yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.

e. Untuk persoalan SPL, luaran (*output*) program adalah solusi SPL. Jika solusinya tunggal, tuliskan nilainya. Jika solusinya tidak ada, tuliskan solusi tidak ada, jika solusinya banyak, maka tuliskan solusinya dalam bentuk parametrik (misalnya $x_4 = -2$, $x_3 = 2s - t$, $x_2 = s$, dan $x_1 = t$).

f. Untuk persoalan determinan dan matriks balikan, maka luarannya sesuai dengan persoalan masing-masing

```
-3 7 8.3 11 -4
3 4.5 2.8 10 12
0.5 -10 -9 12 0
0.1 0.2
```

g. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan. Contoh luaran untuk interpolasi adalah $f(x) = -0.0064x^2 + 0.2266x + 0.6762, f(5) = \cdots$ dan untuk regresi adalah $f(x) = -9.5872 + 1.0732x_1, f(x_k) = \cdots$

h. Untuk persoalan interpolasi *bicubic*, masukan dari *file text* (.txt) yang berisi matriks berukuran 4x4 yang berisi nilai f(i,j) dengan i dan j adalah indeks matriks diikuti dengan nilai a dan b untuk mencari nilai f(a,b). Misalnya jika nilai dari f(-1,-1), f(-1,0), f(-1,1), f(-1,2), f(0,-1), f(0,0), f(0,1), f(0,2), f(1,-1), f(1,0), f(1,1), f(1,2), f(2,-1), f(2,0), f(2,1), f(2,2) berturut-turut adalah 1, 2, 3, 4, 5, 6, 7, 8, 9 ,10, 11, 12, 13, 14, 15 ,16 serta nilai a dan b yang dicari berturut-turut adalah 0.5 dan 0.5 maka isi *file text* ditulis sebagai berikut:

```
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
0.5 0.5
```

Luaran yang dihasilkan adalah nilai dari f(0.5,0.5). Jika masukannya adalah matriks 4 x 4, diikuti oleh nilai a dan b, maka luarannya adalah nilai f(a,b).

i. Luaran program harus dapat ditampilkan pada layar komputer dan dapat disimpan ke dalam file.

j. Bahasa program yang digunakan adalah Java.

k. Program tidak harus berbasis GUI, tetapi boleh menggunakan GUI.

l. Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan dirancang masing-masing.

BAB II

TEORI SINGKAT

2.1. Matriks dan Sistem Persamaan Linier

Sistem persamaan linier (SPL) adalah sebuah sistem yang terdiri atas beberapa persamaan linier yang terdiri atas beberapa variabel. Bentuk umum dari SPL adalah sebagai berikut.

$$
\begin{array}{c}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\
\vdots \qquad \vdots \qquad \ddots \quad \vdots \qquad \vdots \\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m
\end{array}
$$

Sebuah SPL hanya memiliki tiga kemungkinan solusi, yaitu tidak memiliki solusi, memiliki solusi tunggal, atau memiliki solusi banyak.

Sebuah SPL dapat direpresentasikan dalam bentuk matriks. Salah satu cara merepresentasikan SPL dengan matriks adalah dengan matriks augmentasi. Contoh, bentuk umum dari SPL yang telah dituliskan di atas dapat direpresentasikan sebagai matriks augmentasi berikut.

$$
\left[\begin{array}{cccc|c}
a_{11} & a_{12} & \dots & a_{1n} & b_1 \\
a_{21} & a_{22} & \dots & a_{2n} & b_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
a_{m1} & a_{m2} & \dots & a_{mn} & b_m
\end{array}\right]
$$

Sebuah matriks augmentasi yang merepresentasikan SPL dapat diselesaikan dengan menggunakan operasi baris elementer (OBE), sebagaimana penyelesaian pada SPL dilakukan dengan operasi aljabar. OBE tersebut terdiri atas tiga operasi, yaitu perkalian sebuah baris dengan konstanta tak nol, penukaran baris, dan penjumlahan sebuah baris dengan kelipatan konstan dari baris lainnya.

2.2. Metode Eliminasi Gauss dan Eliminasi Gauss-Jordan

Metode eliminasi Gauss dan eliminasi Gauss-Jordan adalah dua metode menyelesaikan sebuah SPL yang memanfaatkan OBE dan matriks eselon baris (eselon baris tereduksi pada metode eliminasi Gauss-Jordan). Matriks eselon baris adalah matriks yang berciri:

1. Bilangan tidak nol pertama pada baris yang tidak sepenuhnya nol adalah 1 (1 utama).
2. Baris yang seluruh elemennya bernilai nol diletakkan pada bagian bawah matriks.
3. Pada dua baris berurutan yang tidak seluruhnya bernilai nol, 1 utama pada baris yang lebih rendah terletak lebih jauh ke kanan daripada 1 utama pada baris yang lebih tinggi.

4. Khusus untuk matriks eselon baris tereduksi, setiap kolom yang memiliki 1 utama memiliki elemen lain yang bernilai nol.

Contoh matriks eselon baris adalah

$$\begin{bmatrix}1 & 2 & 7\\0 & 1 & 5\\0 & 0 & 1\end{bmatrix},\begin{bmatrix}1 & 2 & 0\\0 & 1 & 0\\0 & 0 & 0\end{bmatrix},\begin{bmatrix}0 & 1 & 2 & 6 & 0\\0 & 0 & 1 & -1 & 0\\0 & 0 & 0 & 0 & 1\end{bmatrix}$$

Contoh matriks eselon baris tereduksi adalah

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

Langkah penyelesaian SPL dengan metode eliminasi Gauss adalah melakukan OBE pada matriks SPL hingga terbentuk matriks eselon baris, kemudian melakukan substitusi mundur untuk mendapatkan solusi untuk masing-masing peubah. Untuk metode eliminasi Gauss-Jordan, proses OBE dilakukan hingga terbentuk matriks eselon baris tereduksi. Contoh penggunaan metode eliminasi Gauss atau eliminasi Gauss-Jordan adalah sebagai berikut.

Diketahui SPL berikut

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

$$4x + 4y - 3z = 3$$

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

Penyelesaian dengan Metode Eliminasi Gauss:

$$\left[\begin{array}{ccc|c}2 & 3 & -1 & 5\\4 & 4 & -3 & 3\\-2 & 3 & -1 & 1\end{array}\right] \sim\text{OBE}\sim \left[\begin{array}{ccc|c}1 & 3/2 & -1/2 & 5/2\\0 & 1 & 1/2 & 7/2\\0 & 0 & 1 & 3\end{array}\right]$$

$$z = 3; y = \left(\frac{7}{2} - \frac{1}{2}z\right) = 2; x = \left(\frac{5}{2} - \frac{3}{2}y + \frac{1}{2}z\right) = 1$$

$$\therefore x = 1, y = 2, z = 3$$

Penyelesaian dengan Metode Eliminasi Gauss-Jordan:

$$\left[\begin{array}{ccc|c}2 & 3 & -1 & 5\\4 & 4 & -3 & 3\\-2 & 3 & -1 & 1\end{array}\right] \sim\text{OBE}\sim \left[\begin{array}{ccc|c}1 & 0 & 0 & 1\\0 & 1 & 0 & 2\\0 & 0 & 1 & 3\end{array}\right]$$

$$\therefore x = 1, y = 2, z = 3$$

2.3. Matriks Balikan

Menurut definisi, sebuah matriks persegi A dikatakan mempunyai matriks balikan/invers jika terdapat matriks B yang berukuran sama dan memenuhi $AB = BA = I$, dengan I merupakan matriks identitas. Apabila terdapat B yang memenuhi, maka invers dari A adalah B, atau dapat dinotasikan sebagai A^{-1}. Apabila tidak terdapat B yang memenuhi, maka A disebut sebagai matriks singular (tidak mempunyai balikan).

Salah satu cara untuk menentukan matriks balikan adalah dengan matriks augmentasi dan OBE. Pertama, susun matriks augmentasi $[A|I]$. Setelah itu, lakukan beberapa langkah OBE hingga tercapai bentuk $[I|A^{-1}]$. Contoh pencarian matriks balikan adalah sebagai berikut.

$$[A|I]$$
$$\left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 2 & 5 & 3 & 0 & 1 & 0 \\ 1 & 0 & 8 & 0 & 0 & 1 \end{array}\right]$$
$$\sim_{\substack{R_2 \leftarrow R_2 - 2R_1 \\ R_3 \leftarrow R_3 - R_1}} \sim \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & -2 & 5 & -1 & 0 & 1 \end{array}\right]$$
$$\sim R_3 \leftarrow R_3 + 2R_2 \sim \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & 0 & -1 & -5 & 2 & 1 \end{array}\right]$$
$$\sim R_3 \leftarrow -R_3 \sim \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right]$$
$$\sim_{\substack{R_2 \leftarrow R_2 + 3R_3 \\ R_1 \leftarrow R_1 - 3R_3}} \sim \left[\begin{array}{ccc|ccc} 1 & 2 & 0 & -14 & 6 & 3 \\ 0 & 1 & 0 & 13 & -5 & -3 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right]$$
$$\sim R_1 \leftarrow R_1 - 2R_2 \sim \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & -40 & 16 & 9 \\ 0 & 1 & 0 & 13 & -5 & -3 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right]$$
$$\therefore A^{-1} = \begin{bmatrix} -40 & 16 & 9 \\ 13 & -5 & -3 \\ 5 & -2 & -1 \end{bmatrix}$$

Apabila sebuah matriks tidak mempunyai balikan, proses pencarian invers dengan OBE akan menghasilkan sebuah baris berisi nol pada sisi kiri matriks augmentasi.

$$[A|I]$$
$$\left[\begin{array}{ccc|ccc} 1 & 6 & 4 & 1 & 0 & 0 \\ 2 & 4 & -1 & 0 & 1 & 0 \\ -1 & 2 & 5 & 0 & 0 & 1 \end{array}\right]$$
$$\sim_{\substack{R_2 \leftarrow R_2 - 2R_1 \\ R_3 \leftarrow R_3 + R_1}} \sim \left[\begin{array}{ccc|ccc} 1 & 6 & 4 & 1 & 0 & 0 \\ 0 & -8 & -9 & -2 & 1 & 0 \\ 0 & 8 & 9 & 1 & 0 & 1 \end{array}\right]$$
$$\sim R_3 \leftarrow R_3 + R_2 \sim \left[\begin{array}{ccc|ccc} 1 & 6 & 4 & 1 & 0 & 0 \\ 0 & -8 & -9 & -2 & 1 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -1 & 1 & 1 \end{array}\right]$$
$$\therefore A \text{ adalah matriks singular}$$

Salah satu penggunaan dari matriks balikan adalah untuk menyelesaikan SPL. Sebuah SPL dapat dinyatakan sebagai persamaan perkalian matriks $Ax = b$, dengan

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, b = \begin{bmatrix} b_! \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Dengan menggunakan matriks balikan, solusi SPL, yaitu matriks peubah *x*, dapat diperoleh dari persamaan $x = A^{-1}b$, dengan syarat *A* mempunyai balikan. Apabila *A* tidak mempunyai balikan, SPL yang direpresentasikan oleh $Ax = b$ tidak mempunyai solusi tunggal. Khusus untuk SPL homogen, apabila *A* tidak mempunyai balikan, solusi SPL adalah solusi nontrivial.

2.4. Determinan

Determinan adalah nilai skalar yang diperoleh dari sebuah matriks persegi melalui sebuah fungsi. Determinan sebuah matriks dapat dituliskan sebagai berikut.

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

Terdapat berbagai metode untuk menghitung determinan dari sebuah matriks persegi. Matriks persegi sederhana, seperti matriks 2x2 dan matriks 3x3, determinan dapat dihitung dengan perhitungan yang mudah. Untuk matriks 1x1, determinan dari matriks adalah entri tunggal pada matriks.

$$A = [a] \Rightarrow \det(A) = a$$

Untuk matriks 2x2, determinan dapat dihitung sebagai berikut.

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \Rightarrow \det(A) = ad - bc$$

Untuk matriks 3x3, determinan dapat dihitung sebagai berikut.

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \Rightarrow \det(A) = (aei + bfg + cdh) - (ceg + afh + bdi)$$

Untuk matriks segitiga atas dan segitiga bawah, determinan dapat dihitung sebagai berikut.

$$A = \begin{bmatrix} a & e & h & j \\ 0 & b & f & i \\ 0 & 0 & c & g \\ 0 & 0 & 0 & d \end{bmatrix}; B = \begin{bmatrix} a & 0 & 0 & 0 \\ e & b & 0 & 0 \\ g & f & c & 0 \\ i & h & g & d \end{bmatrix} \Rightarrow \det(A) = \det(B) = abcd$$

Untuk matriks persegi yang lebih besar dari 3x3, salah satu cara untuk menghitung determinannya adalah dengan melakukan OBE hingga tercapai bentuk matriks segitiga. Berikut adalah teorema yang menyatakan hubungan antara OBE dan perhitungan determinan matriks untuk sebuah matriks persegi *A*.

1. Apabila *B* adalah matriks hasil operasi perkalian skalar *k* terhadap satu baris atau satu kolom dari *A*, maka $\det(B) = k\det(A)$.

2. Apabila B adalah matriks hasil penukaran antara sepasang baris atau sepasang kolom dari A, maka $\det(B) = -\det(A)$.

3. Apabila B adalah matriks hasil dari penjumlahan sebuah baris dengan kelipatan konstan dari baris lain, atau penjumlahan sebuah kolom dengan kelipatan konstan dari baris lain, maka $\det(B) = \det(A)$.

Nilai determinan dari sebuah matriks dapat digunakan untuk menentukan balikan dari matriks tersebut. Misal A adalah sebuah matriks persegi. Apabila $\det(A) \neq 0$, maka matriks A tidak singular dan memiliki balikan yang dinyatakan oleh

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

dengan *adj(A)* adalah matriks adjoin dari *A* yang akan dibahas pada subbab berikutnya.

2.5. Kofaktor dan Adjoin

Kofaktor dari entri a_{ij} didefinisikan sebagai perkalian $(-1)^{i+j}$ dengan minor entri a_{ij}, yaitu determinan dari submatriks yang diperoleh dengan menghapus baris ke-*i* dan menghapus kolom ke-*j* (disimbolkan sebagai M_{ij}). Secara umum, rumus dari kofaktor dinyatakan sebagai

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

Contoh perhitungan kofaktor adalah sebagai berikut.

$$A = \begin{bmatrix} 3 & 1 & -4 \\ 2 & 5 & 6 \\ 1 & 4 & 8 \end{bmatrix}$$

$$M_{11} = \begin{vmatrix} 3 & 1 & -4 \\ 2 & \mathbf{5} & \mathbf{6} \\ 1 & \mathbf{4} & \mathbf{8} \end{vmatrix} = \begin{vmatrix} 5 & 6 \\ 4 & 8 \end{vmatrix} = 16$$

$$C_{11} = (-1)^{1+1} M_{11} = 16$$

Nilai-nilai kofaktor dari sebuah matriks dapat digunakan untuk menghitung determinan. Metode ini disebut sebagai metode ekspansi kofaktor dan dirumuskan sebagai berikut.

$\det(A) = a_{1j}C_{1j} + a_{2j}C_{2j} + \cdots + a_{nj}C_{nj}$ [ekspansi kofaktor pada kolom ke-*j*]

$\det(A) = a_{i1}C_{i1} + a_{i2}C_{i2} + \cdots + a_{in}C_{in}$ [ekspansi kofaktor pada baris ke-*i*]

Pada metode ini, umumnya baris atau kolom yang mengandung jumlah entri bernilai nol terbanyak dipilih untuk menghemat perhitungan determinan dan kofaktor.

Penggunaan lain dari kofaktor adalah menyusun matriks kofaktor. Matriks kofaktor adalah matriks persegi yang terdiri atas kofaktor dari setiap entri yang ada di matriks persegi lain. Misal *A* adalah matriks berukuran nxn, dan C_{ij} adalah kofaktor entri a_{ij}; matriks kofaktor dari *A* adalah

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

Matriks kofaktor digunakan untuk menyusun matriks adjoin yang digunakan untuk menghitung invers dari suatu matriks. Matriks adjoin disusun dengan cara melakukan transpose terhadap matriks kofaktor.

$$A = \text{matriks n} \times \text{n};\ C(A) = \text{matriks kofaktor dari } A \Rightarrow \text{adj}(A) = \big(C(A)\big)^T$$

2.6. Kaidah Cramer

Kaidah Cramer adalah sebuah aturan yang digunakan untuk menentukan solusi dari SPL $Ax = b$ dari *n* banyak persamaan yang mengandung *n* buah peubah, dengan syarat matriks koefisien SPL (*A*) mempunyai balikan. Menurut Kaidah Cramer, solusi dari SPL yang memenuhi syarat di atas ditentukan oleh persamaan

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

dengan A_j adalah matriks yang diperoleh dengan menggantikan elemen-elemen kolom ke *j* dari matriks *A* dengan elemen pada matriks *b*.

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

Contohnya adalah sebagai berikut.

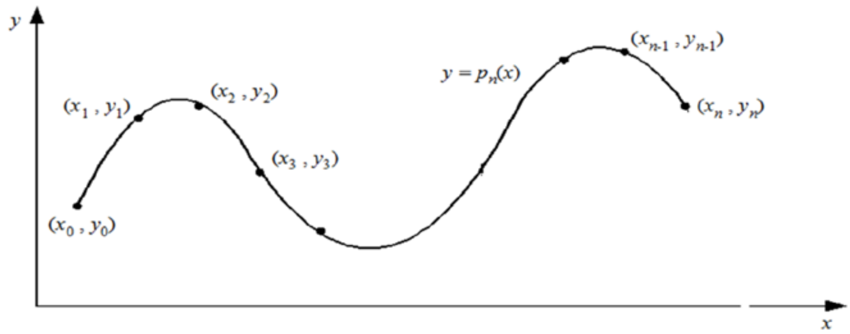
$$\begin{gathered} x_1 + 2x_3 = 6 \\ -3x_1 + 4x_2 + 6x_3 = 30 \\ -x_1 - 2x_2 + 3x_3 = 8 \end{gathered}$$

$$A = \begin{bmatrix} 1 & 0 & 2 \\ -3 & 4 & 6 \\ -1 & -2 & 3 \end{bmatrix}; b = \begin{bmatrix} 6 \\ 30 \\ 8 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} \mathbf{6} & 0 & 2 \\ \mathbf{30} & 4 & 6 \\ \mathbf{8} & -2 & 3 \end{bmatrix}; A_2 = \begin{bmatrix} 1 & \mathbf{6} & 2 \\ -3 & \mathbf{30} & 6 \\ -1 & \mathbf{8} & 3 \end{bmatrix}; A_3 = \begin{bmatrix} 1 & 0 & \mathbf{6} \\ -3 & 4 & \mathbf{30} \\ -1 & -2 & \mathbf{8} \end{bmatrix}$$

Maka solusinya adalah $x_1 = \frac{\det(A_1)}{\det(A)} = -\frac{10}{11}, x_2 = \frac{\det(A_2)}{\det(A)} = \frac{18}{11}, x_n = \frac{\det(A_n)}{\det(A)} = \frac{38}{11}$.

2.7. Interpolasi Polinom



Gambar 1 Interpolasi Polinom

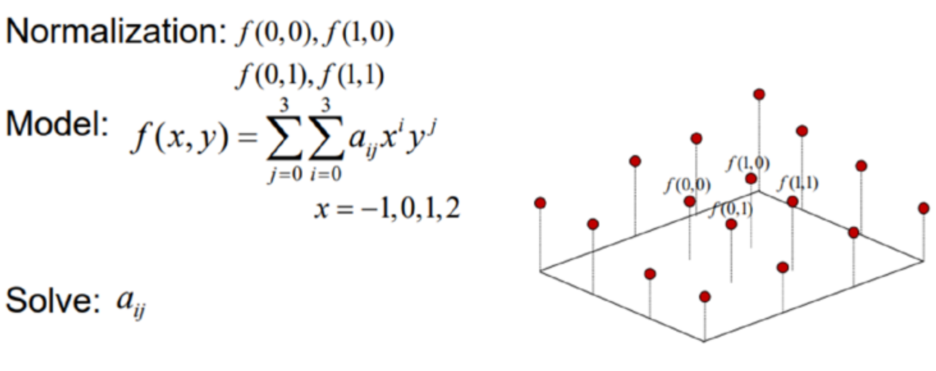
Interpolasi polinom adalah teknik interpolasi sekumpulan titik dengan menggunakan polinom. Hasil interpolasi adalah polinom berderajat n berbentuk $p_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$ yang melewati titik-titik $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ dan memenuhi $y_i = p_n(x_i)$ untuk $i = 0,1,2, \dots, n$. Nilai-nilai $a_0, a_1, \dots, a_n$ diperoleh dengan menyelesaikan SPL di bawah menggunakan metode eliminasi Gauss.

$$a_0 + a_1x_0 + a_2x_0^2 + \cdots + a_nx_0^n = y_0$$
$$a_0 + a_1x_1 + a_2x_1^2 + \cdots + a_nx_1^n = y_1$$
$$\dots$$
$$a_0 + a_1x_n + a_2x_n^2 + \cdots + a_nx_n^n = y_n$$

2.8. Interpolasi Bicubic

Interpolasi bikubik adalah teknik interpolasi pada data dua dimensi yang umumnya digunakan dalam pembesaran citra. Teknik ini merupakan pengembangan dari interpolasi linier dan kubik.

Misal diberikan sebuah matriks M. Persamaan interpolasi $f(x, y)$ dengan model seperti di bawah ditentukan dengan langkah-langkah berikut.



Gambar 2 Contoh Model Interpolasi Bicubic

Dengan melakukan substitusi nilai-nilai yang diketahui pada matriks 4x4 tersebut ke persamaan $f(x, y)$ akan menghasilkan sebuah persamaan matriks

$$y = Xa$$

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

Gambar 3 Persamaan Matriks untuk Contoh Interpolasi Bicubic

Elemen pada matriks *X* adalah koefisien a_{ij}, yang diperoleh dari persamaan $f(x, y)$ di atas. Contoh, elemen pada baris ke-4, kolom ke-10 adalah koefisien dari a_{12} dan diperoleh dari $2^1 \cdot (-1)^2 = 2$.

Vektor a dapat ditentukan nilainya menggunakan invers. Vektor a kemudian digunakan sebagai nilai variabel dalam $f(x, y)$ yang kemudian membentuk fungsi interpolasi bicubic sesuai model.

2.9. Regresi Linier Berganda

Regresi linier adalah salah satu metode untuk memprediksi nilai selain menggunakan metode interpolasi polinom. Rumus umum dari regresi linier adalah sebagai berikut.

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

Cara mendapatkan taksiran dari setiap koefisien regresi (β_i) adalah dengan menyelesaikan SPL berikut (*Normal Estimation Equation for Multiple Linear Regression*). Penyelesaian SPL untuk memperoleh setiap nilai b_i dapat dilakukan dengan eliminasi Gauss.

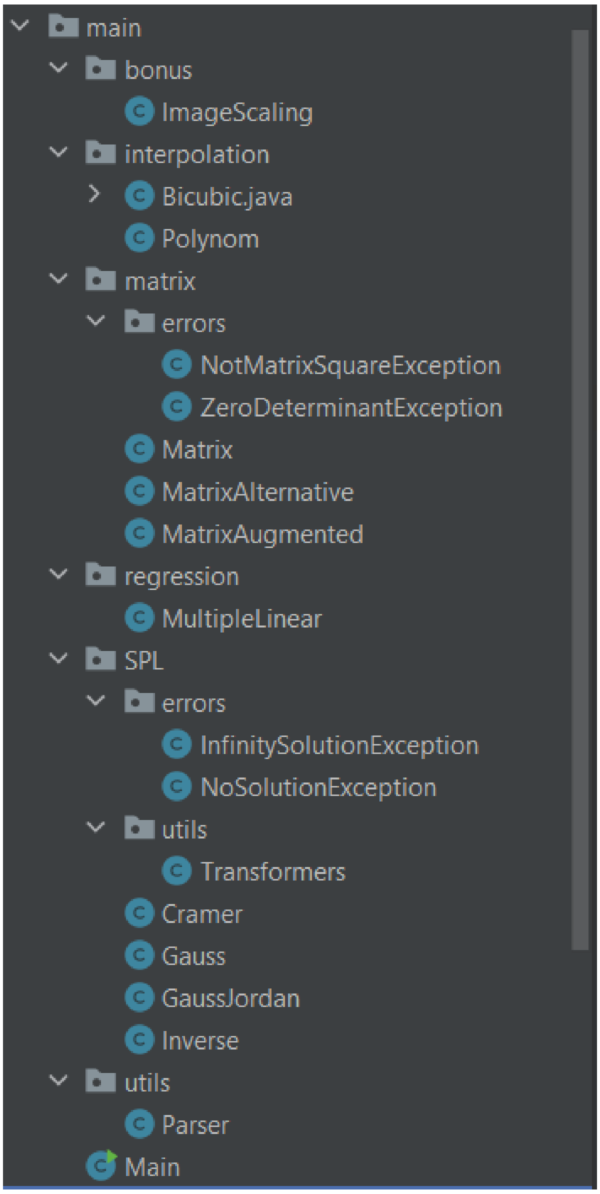
$$\begin{array}{ccccccccccc} nb_0 & + & b_1 \sum_{i=1}^{n} x_{1i} & + & b_2 \sum_{i=1}^{n} x_{2i} & + & \dots & + & b_k \sum_{i=1}^{n} x_{ki} & = & \sum_{i=1}^{n} y_i \\ b_0 \sum_{i=1}^{n} x_{1i} & + & b_1 \sum_{i=1}^{n} x_{1i}^2 & + & b_2 \sum_{i=1}^{n} x_{1i} x_{2i} & + & \dots & + & b_k \sum_{i=1}^{n}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

BAB III

IMPLEMENTASI

3.1. Project Structure

Berikut adalah project structure dari program ini.



Gambar 4 Struktur Proyek Program Pengolahan Matriks

Secara umum, project dibagi menjadi beberapa bagian, seperti: interpolasi, regresi, SPL, matriks, dan bonus. Folder bonus berisi implementasi soal bonus pembesar gambar, folder interpolation berisi implementasi untuk interpolasi menggunakan metode bicubic dan polinom. Folder matrix berisi class wrapper untuk metode matriks serta wrapper class untuk matriks teraugmentasi. Folder regression berisi implementasi regresi linear berganda. Folder SPL berisi implementasi empat metode penyelesaian SPL yang berbeda, yaitu Gauss, Gauss-Jordan, Invers, dan Cramer. Selain itu, terdapat file utils dan main program.

3.2. Matrix

Folder matrix secara umum terdiri atas tiga class, yaitu Matrix, MatrixAlternative, dan MatrixAugmented.

Class Matrix merupakan wrapper untuk array yang berisi konten matriks. Tipe data yang digunakan adalah 64 bit float atau double. Method operasi matriks antara lain: multiplyMatrix (perkalian matrix dengan matrix), multiplyCoef (perkalian matrix dengan koefisien), addMatrix (penjumlahan matrix dengan matrix), subMatrix (pengurangan matrix dengan matrix), addCoef (penjumlahan tiap elemen matrix dengan suatu koefisien). Setiap method ini tidak mengubah matriks asal sehingga mengembalikan matrix hasil operasi. Selain itu, terdapat operasi amtrix yang mengubah konten matrix, seperti swapRow (menukar posisi dua row), addRow (menambah suatu baris dengan kelipatan dari baris lain), multiplyRow (mengalikan nilai suatu baris dengan suatu pengali), deleteRows, deleteCols. Selain itu, terdapat juga method transpose untuk memperoleh transpos matriks.

Selain itu, terdapat method getInverse untuk menghitung invers dari sebuah matriks dengan metode adjoin. method ini membutuhkan method lain, yaitu getAdjoin untuk memperoleh matriks adjoin. Method getAdjoin sendiri membutuhkan method getCofactor. Terdapat pula method getDeterminant yang menghitung nilai determinan dari suatu matriks dengan metode minor-kofaktor. Method ini merupakan method rekursif yang bergantung pada fungsi getMinor (mencari nilai minor matriks).

Class MatrixAlternative berisi alternatif implementasi invers dan determinant matriks. Class ini menyediakan perhitungan invers matriks dengan menggunakan metode Gauss dan perhitungan determinan matriks dengan menggunakan OBE/matriks eselon.

Class MatrixAugmented berisi wrapper matriks teraugmentasi. Pada dasarnya, matriks teraugmentasi diibaratkan sebagai dua buah matriks, yakni matriks sisi kiri dan matriks augmentasi yang merupakan matriks sisi kanan. Pada dasarnya, fungsi ini digunakan untuk membantu proses eliminasi Gauss dengan menyediakan method seperti, swapRow, addRow, multiplyRow, dan deleteRow. Cara kerjanya sama seperti method pada class Matrix, hanya saja, method pada class ini memanggil method tersebut pada kedua matriks yang ia simpan, yakni matriks sisi kiri dan kanan.

3.3. Interpolation

Folder interpolation berisi dua class dengan dua metode implementasi yang berbeda, yakni class Polynom dan class Bicubic.

Class Polynom memiliki method yang cukup sederhana, yakni method solve yang menerima paramter array of point x, y, lalu membentuk persamaan ekspansi polinomial dan melakukan perhitungan konstanta polinom dengan menggunakan invers matriks. Selain itu, terdapat fungsi predict yang menerima list konstanta dan titik x prediksi dan melakukan prediksi nilai y berdasarkan hasil interpolasi. Terakhir, terdapat method buildEquation yang mengubah nilai konstanta menjadi persamaan berbentuk f(x) = ...

Class Bicubic terdiri atas class bicubic dan subclass CoefMatrix. CoefMatrix merupakan singleton yang menyimpan koefisien matriks untuk interpolasi bicubic yang mengikuti rumus berikut.

$$f(x,y) = \sum_{j=0}^{3}\sum_{i=0}^{3} a_{ij} x^i y^j$$
$$x = -1, 0, 1, 2$$

Gambar 5 Rumus Model Interpolasi Bicubic

Selain itu, class ini juga menyimpan hasil perhitungan invers matriksnya.

Class bicubic menerima sebuah matrix berukuran 4x4 sebagai titik patokan interpolasi. Konten matriks tersebut dikalikan dengan invers koefisien matriks sebelumnya sehingga diperoleh koefisien untuk interpolasi secara bicubic. Setelah itu, pengguna cukup memanggil method interpolate yang menerima titik x dan y apabila ingin melakukan interpolasi secara bicubic berdasarkan titik patokan sebelumnya.

3.4. Regression

Folder regression terdiri atas class MultipleLinear yang berisi implementasi regresi linear berganda. Class ini terdiri atas method normalEstSystem, yang berfungsi mengembalikan matriks SPL estimasi normal koefisien regresi, regCoefficients, yang berfungsi mengembalikan matriks yang berisi nilai-nilai koefisien regresi, predict, yang berfungsi mengembalikan nilai prediksi berdasarkan nilai koefisien regresi dan sejumlah peubah, dan method buildEquation, yang berfungsi mengembalikan String yang berisi fungsi regresi linier.

3.5. SPL

Folder SPL terdiri atas class Cramer, Gauss, GaussJordan, Inverse, serta folder utils dan errors. Class Cramer merupakan implementasi dari metode pemecahan masalah sistem persamaan linear dengan menggunakan metode Cramer. Class Gauss merupakan implementasi dari metode pemecahan masalah sistem persamaan linear dengan menggunakan metode Gauss. Sedangkan class GaussJordan merupakan implementasi pemecahan masalah sistem persamaan linear menggunakan metode Gauss Jordan yang memanfaatkan class Gauss. Selanjutnya folder utils berisi class Transformers yang dipakai oleh class – class SPL lainny untuk mengubah bentuk dan format dari Matrix serta metode output hasil dari sistem persamaan linear menggunakan metode – metode yang ada pada folder SPL. Kemudian folder errors merupakan folder yang menampung jenis – jenis class error yang di lontarkan oleh program – program yang ada pada folder SPL.

3.6. Main Program dan Parser

Class Main adalah class utama pada program dan berfungsi menggabungkan semua class pada project menjadi sebuah program utuh. Class Main terdiri atas method main, linearEquation, determinant, inverse, polynomialInterpolation, bicubicInterpolation, multipleRegression, imageScaling, fileOrInputChoice, fileOutputChoice, readFile, dan saveToFIle. Method main berfungsi sebagai prosedur utama program dan merupakan menu dari fitur-fitur yang tersedia pada program. Method linearEquation adalah submenu “Sistem Persamaan Linear”. Pada method ini, pengguna dapat memilih metode eliminasi Gauss, eliminasi Gauss-Jordan, matriks balikan, atau kaidah Cramer untuk menyelesaikan SPL. Method determinant adalah submenu “Determinan” yang berfungsi untuk menghitung determinan dengan metode minor-kofaktor atau metode matriks eselon. Method inverse adalah submenu “Matriks Balikan” untuk menghitung balikan matriks, dengan pilihan metode adalah metode adjoin dan metode Gauss.

Method polynomialInterpolation berfungsi untuk melakukan prediksi menggunakan interpolasi polinom. Method bicubicInterpolation berfungsi untuk melakukan prediksi menggunakan model interpolasi bicubic yang telah ditentukan. Method multipleRegression berfungsi untuk melakukan regresi linier berganda dan melakukan prediksi nilai. Method imageScaling berfungsi untuk mengakses menu dan fitur pembesaran gambar serta menerima *path* dari gambar yang in

Method selanjutnya adalah fileOrInputChoice yang berfungsi menanyakan apakah pengguna ingin memasukkan data, misal matriks atau titik-titik sampel pada regresi, dengan cara mengirimkan *file* teks atau menggunakan *input* manual pada program. Method fileOutputChoice berfungsi menanyakan kepada pengguna apakah hasil perhitungan ingin disimpan pada sebuah *file* teks. Method readFile berfungsi untuk menanyakan lokasi *file* teks pada komputer pengguna yang ingin digunakan sebagai data dalam perhitungan dan mengembalikan isi dari *file* tersebut. Method terakhir adalah saveToFile yang berfungsi untuk menyimpan hasil perhitungan ke dalam sebuah *file* teks.

Untuk membantu menjalankan fungsi utama, terdapat class Parser yang terdiri atas beberapa method. Method-method dalam class ini memiliki fungsi untuk membaca isi *file* teks, mengubah list string ke sebuah array double dua dimensi, mengubah array double dua dimensi menjadi list string, dan menulis dan menyimpan *file* teks pada komputer.

3.7. Bonus – Image Scaling

Method utama pada class ImageScaling adalah scale, yang menerima input gambar berupa BufferedImage, dan mengembalikan gambar hasil perbesaran dengan tipe BufferedImage. Secara umum, alur programnya adalah sebagai berikut.

1. Interpolasi pixel sisi ujung kiri, kanan, atas, bawah, menggunakan sampel 2x2 menjadi matriks 4x4 dengan metode bilinear interpolation.
2. Bila tinggi atau lebar gambar ganjil, interpolasi juga sisa ujungnya menggunakan sampel 2x2 menjadi matriks 2x4, 4x2, atau 2x2 dengan menggunakan metode bilinear interpolation
3. Selanjutnya, interpolasi pixel sisanya (pixel kedua hingga pixel kedua terakhir) dengan metode bicubic interpolation. Interpolasi ini menginterpolasi pixel 2x2 pusat menjadi pixel 4x4 berdasarkan pixel 4x4 pixel sekitar pixel pusat.

Proses scaling image ini menggunakan skema warna Red Green Blue Alpha (RGBA) sehingga interpolasi warna untuk setiap subpixel perlu dilakukan sebanyak empat kali karena terdapat empat channel yang berbeda.

Selain itu, proses interpolasi gambar menggunakan interpolasi bicubic merupakan proses yang *computationally intensive* sehingga perhitungannya dibuat paralel dengan memanfaatkan semua core processor yang dimiliki pengguna. Hal ini dicapai dengan menyimpan semua titik x, y yang harus diinterpolasi ke dalam sebuah Collection, lalu memanggil parallel stream dan melakukan pemrosesan untuk setiap titik tersebut.

BAB IV

EKSPERIMENTASI

4.1. Solusi SPL berbentuk $Ax = B$

a. Kasus pertama

Matriks asalnya adalah sebagai berikut.

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

Dengan menggunakan matriks invers, diperoleh solusi sebagai berikut.

```
SPL Case 1a

1.00x1 + 1.00x2 - 1.00x3 - 1.00x4 = 1.00
2.00x1 + 5.00x2 - 7.00x3 - 5.00x4 = -2.00
2.00x1 - 1.00x2 + 1.00x3 + 3.00x4 = 4.00
5.00x1 + 2.00x2 - 4.00x3 + 2.00x4 = 6.00
Tidak terdapat solusi SPL karena LHS 0 menghasilkan RHS tidak 0.
```

Gambar 6 Luaran SPL Kasus (Ax=b) - a

Ini berarti bahwa persamaan tidak memiliki solusi karena nilai dari determinannya adalah sama dengan nol.

b. Kasus kedua

Diketahui matriks berikut.

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

Dengan menggunakan metode eliminasi gauss, diperoleh solusi sebagai berikut.

```
SPL Case 1b

1.00x1 - 1.00x2 + 0.00x3 + 0.00x4 + 1.00x5 = 3.00
1.00x1 + 1.00x2 + 0.00x3 - 3.00x4 + 0.00x5 = 6.00
2.00x1 - 1.00x2 + 0.00x3 + 1.00x4 - 1.00x5 = 5.00
1.00x1 + 2.00x2 + 0.00x3 - 2.00x4 - 1.00x5 = -1.00
x1 = 3.0 + 0.9999999999999999(t5),
x2 = 0.0 + 2.0(t5),
x3 = (t3),
x4 = -1.0 + 1.0000000000000002(t5),
x5 = (t5)
```

Gambar 7 Luaran SPL Kasus (Ax=b) - b

Solusi yang dihasilkan berupa persamaan parametrik dua variabel karena terdapat satu kolom yang keseluruhannya bernilai nol dan kurangnya persamaan (terdapat 5 peubah tetapi hanya diberikan 4 persamaan).

c. Kasus ketiga

Diketahui matriks sebagaimana berikut.

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

Dengan menggunakan eliminasi Gauss, diperoleh persamaan parametrik dengan tiga peubah.

```
SPL Case 1c

0.00x1 + 1.00x2 + 0.00x3 + 0.00x4 + 1.00x5 + 0.00x6 = 2.00
0.00x1 + 0.00x2 + 0.00x3 + 1.00x4 + 1.00x5 + 0.00x6 = -1.00
0.00x1 + 1.00x2 + 0.00x3 + 0.00x4 + 0.00x5 + 1.00x6 = 1.00
x1 = (t1),
x2 = 1.0 + -1.0(t6),
x3 = (t3),
x4 = -2.0 + -1.0(t6),
x5 = 1.0 + 1.0(t6),
x6 = (t6)
```

Gambar 8 Luaran SPL Kasus (Ax=b) - c

d. Matriks Hilbert

Diketahui matriks Hilbert sebagaimana berikut.

$$A = \begin{bmatrix} 1 & \cdots & \frac{1}{n} \\ \vdots & \ddots & \vdots \\ \frac{1}{n} & \cdots & \frac{1}{2n+1} \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Untuk $n = 6$, dengan menggunakan metode invers, diperoleh hasil sebagaimana berikut.

```
1.00x1 + 0.50x2 + 0.33x3 + 0.25x4 + 0.20x5 + 0.17x6 = 1.00
0.50x1 + 0.33x2 + 0.25x3 + 0.20x4 + 0.17x5 + 0.14x6 = 0.00
0.33x1 + 0.25x2 + 0.20x3 + 0.17x4 + 0.14x5 + 0.13x6 = 0.00
0.25x1 + 0.20x2 + 0.17x3 + 0.14x4 + 0.13x5 + 0.11x6 = 0.00
0.20x1 + 0.17x2 + 0.14x3 + 0.13x4 + 0.11x5 + 0.10x6 = 0.00
0.17x1 + 0.14x2 + 0.13x3 + 0.11x4 + 0.10x5 + 0.09x6 = 0.00
x1 = 36.000000001864464,
x2 = -630.0000000423852,
x3 = 3360.000000275675,
x4 = -7560.0000007261,
x5 = 7560.000000808649,
x6 = -2772.000000320771
```

Gambar 9 Luaran Kasus Matriks Hilbert Orde 6

Sedangkan untuk $n = 10$, dengan menggunakan metode invers, diperoleh hasil sebagaimana berikut.

```
1.00x1 + 0.50x2 + 0.33x3 + 0.25x4 + 0.20x5 + 0.17x6 + 0.14x7 + 0.13x8 + 0.11x9 + 0.10x10 = 1.00
0.50x1 + 0.33x2 + 0.25x3 + 0.20x4 + 0.17x5 + 0.14x6 + 0.13x7 + 0.11x8 + 0.10x9 + 0.09x10 = 0.00
0.33x1 + 0.25x2 + 0.20x3 + 0.17x4 + 0.14x5 + 0.13x6 + 0.11x7 + 0.10x8 + 0.09x9 + 0.08x10 = 0.00
0.25x1 + 0.20x2 + 0.17x3 + 0.14x4 + 0.13x5 + 0.11x6 + 0.10x7 + 0.09x8 + 0.08x9 + 0.08x10 = 0.00
0.20x1 + 0.17x2 + 0.14x3 + 0.13x4 + 0.11x5 + 0.10x6 + 0.09x7 + 0.08x8 + 0.08x9 + 0.07x10 = 0.00
0.17x1 + 0.14x2 + 0.13x3 + 0.11x4 + 0.10x5 + 0.09x6 + 0.08x7 + 0.08x8 + 0.07x9 + 0.07x10 = 0.00
0.14x1 + 0.13x2 + 0.11x3 + 0.10x4 + 0.09x5 + 0.08x6 + 0.08x7 + 0.07x8 + 0.07x9 + 0.06x10 = 0.00
0.13x1 + 0.11x2 + 0.10x3 + 0.09x4 + 0.08x5 + 0.08x6 + 0.07x7 + 0.07x8 + 0.06x9 + 0.06x10 = 0.00
0.11x1 + 0.10x2 + 0.09x3 + 0.08x4 + 0.08x5 + 0.07x6 + 0.07x7 + 0.06x8 + 0.06x9 + 0.06x10 = 0.00
0.10x1 + 0.09x2 + 0.08x3 + 0.08x4 + 0.07x5 + 0.07x6 + 0.06x7 + 0.06x8 + 0.06x9 + 0.05x10 = 0.00
x1 = 99.99685558960846,
x2 = -4949.726911833874,
x3 = 79194.16138617636,
x4 = -600546.7691159137,
x5 = 2522265.5323173404,
x6 = -6305599.2273747735,
x7 = 9608448.59179239,
x8 = -8750485.979971208,
x9 = 4375214.57135724,
x10 = -923651.1480691661
```

Gambar 10 Luaran Kasus Matriks Hilbert Orde 10

4.2. Solusi SPL berbentuk Matriks *Augmented*

a. Kasus pertama

Diketahui matriks *augmented* berikut.

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

Dengan menggunakan metode Gauss, diperoleh hasil berikut.

```
SPL Case 2a

1.00x1 - 1.00x2 + 2.00x3 - 1.00x4 = -1.00
2.00x1 + 1.00x2 - 2.00x3 - 2.00x4 = -2.00
1.00x1 + 2.00x2 - 4.00x3 + 1.00x4 = 1.00
3.00x1 + 0.00x2 + 0.00x3 - 3.00x4 = -3.00
x1 = -1.0 + 1.0(t4),
x2 = 0.0 + 2.0(t3),
x3 = (t3),
x4 = (t4)
```

Gambar 11 Luaran SPL Kasus Augmented-a

Hasil merupakan persamaan parametrik dengan dua peubah.

b. Kasus kedua

Diketahui matriks *augmented* berikut.

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

Dengan menggunakan metode Gauss, diperoleh hasil berikut.

```
SPL Case 2b

2.00x1 + 0.00x2 + 8.00x3 + 0.00x4 = 8.00
0.00x1 + 1.00x2 + 0.00x3 + 4.00x4 = 6.00
4.00x1 + 0.00x2 + 6.00x3 + 0.00x4 = 6.00
0.00x1 - 2.00x2 + 0.00x3 + 3.00x4 = -1.00
2.00x1 + 0.00x2 - 4.00x3 + 0.00x4 = -4.00
0.00x1 + 1.00x2 + 0.00x3 - 2.00x4 = 0.00
x1 = 0.0,
x2 = 2.0,
x3 = 1.0,
x4 = 1.0
```

Gambar 12 Luaran SPL Kasus Augmented-b

Hasil merupakan persamaan parametrik dengan satu peubah. Selain itu, SPL ini memiliki kasus khusus mengingat SPL ini memiliki 4 peubah, sedangkan diberikan 6 persamaan, yang berari persamaan yang diberikan melebihi banyaknya persamaan yang diperlukan.

4.3. Solusi SPL berbentuk Persamaan

a. Kasus pertama

$$8x_1 + x_2 + 3x_3 + 2x_4 = 0$$

$$2x_1 + 9x_2 - x_3 - 2x_4 = 1$$

$$x_1 + 3x_2 + 3x_3 - x_4 = 2$$

$$x_1 + 6x_3 + 4x_4 = 3$$

Setelah mengubah persamaan tersebut ke dalam bentuk matriks *augmented* berbentuk 4x5, diperoleh hasil sebagaimana berikut.

```
SPL Case 3a

8.00x1 + 1.00x2 + 3.00x3 + 2.00x4 = 0.00
2.00x1 + 9.00x2 - 1.00x3 - 2.00x4 = 1.00
1.00x1 + 3.00x2 + 2.00x3 - 1.00x4 = 2.00
1.00x1 + 0.00x2 + 6.00x3 + 4.00x4 = 3.00
x1 = -0.2243243243243243,
x2 = 0.1824324324324243,
x3 = 0.7094594594594594,
x4 = -0.2581081081081081
```

Gambar 13 Luaran Kasus Persamaan SPL-a

Pada persamaan ini, persamaan memiliki solusi tunggal.

b. Kasus kedua

Diketahui SPL sebagaimana berikut.

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

Gambar 14 Kasus Pengujian Persamaan SPL-b

Setelah mengubah SPL tersebut ke dalam bentuk matriks *augmented* berukuran 9x10, diperoleh hasil sebagaimana berikut.

```
SPL Case 3b

0.00x1 + 0.00x2 + 0.00x3 + 0.00x4 + 0.00x5 + 0.00x6 + 1.00x7 + 1.00x8 + 1.00x9 = 13.00
0.00x1 + 0.00x2 + 0.00x3 + 1.00x4 + 1.00x5 + 1.00x6 + 0.00x7 + 0.00x8 + 0.00x9 = 15.00
1.00x1 + 1.00x2 + 1.00x3 + 0.00x4 + 0.00x5 + 0.00x6 + 0.00x7 + 0.00x8 + 0.00x9 = 8.00
0.00x1 + 0.00x2 + 0.04x3 + 0.00x4 + 0.04x5 + 0.75x6 + 0.04x7 + 0.75x8 + 0.61x9 = 14.79
0.00x1 + 0.25x2 + 0.91x3 + 0.25x4 + 0.91x5 + 0.25x6 + 0.91x7 + 0.25x8 + 0.00x9 = 14.31
0.61x1 + 0.75x2 + 0.04x3 + 0.75x4 + 0.04x5 + 0.00x6 + 0.04x7 + 0.00x8 + 0.61x9 = 3.81
0.00x1 + 0.00x2 + 1.00x3 + 0.00x4 + 0.00x5 + 1.00x6 + 0.00x7 + 0.00x8 + 1.00x9 = 18.00
0.00x1 + 1.00x2 + 0.00x3 + 0.00x4 + 1.00x5 + 0.00x6 + 0.00x7 + 1.00x8 + 0.00x9 = 12.00
1.00x1 + 0.00x2 + 0.00x3 + 1.00x4 + 0.00x5 + 0.00x6 + 1.00x7 + 0.00x8 + 0.00x9 = 6.00
0.04x1 + 0.75x2 + 0.61x3 + 0.00x4 + 0.04x5 + 0.75x6 + 0.00x7 + 0.00x8 + 0.04x9 = 10.51
0.91x1 + 0.25x2 + 0.00x3 + 0.25x4 + 0.91x5 + 0.25x6 + 0.00x7 + 0.25x8 + 0.91x9 = 16.13
0.04x1 + 0.00x2 + 0.00x3 + 0.75x4 + 0.04x5 + 0.00x6 + 0.61x7 + 0.75x8 + 0.04x9 = 7.04
x1 = 5.044925720592925,
x2 = -1.0526586613509838,
x3 = 4.007732940758059,
x4 = -1.0414994491466132,
x5 = 4.999403200908974,
x6 = 11.04209624823764,
x7 = 1.9965737285536882,
x8 = 8.05325546044201,
x9 = 2.950170811004301
```

Gambar 15 Luaran Kasus Persamaan SPL-b

Pada sistem persamaan ini, persamaan memiliki solusi tunggal.

4.4. Interpolasi Polinom

a. Kasus pertama

Diketahui data berikut.

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

Dengan melakukan *plotting* pada data tersebut, diperoleh persamaan polinom berikut.

```
CASE 4A
f(x) =  - 0.185 + 10.276x^1 - 163.916x^2 + 1220.855x^3 - 4346.314x^4 + 7102.399x^5 - 4212.435x^6
```

Gambar 16 Luaran Kasus Interpolasi Polinom-a (bagian 1)

Sementara itu, hasil prediksinya adalah

```
f(0.20) = 0.130000
f(0.55) = 2.137572
f(0.85) = -66.269639
f(1.28) = -3485.144902
```

Gambar 17 Luaran Kasus Interpolasi Polinom-a (bagian 2)

b. Kasus kedua

Diketahui data sebagaimana berikut.

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

Gambar 18 Data Kasus Pengujian Interpolasi Polinom-b

Setelah melakukan proses *plotting* dan penentuan nilai koefisien, diperoleh persamaan berikut.

$$\begin{aligned} f(x) = 7187700143017.766 - 9347732054580.110x^1 + 5334585430409.406x^2 \\ - 1756925483232.799x^3 + 368573133214.306x^4 \\ - 51134755924.227x^5 + 4696053591.060x^6 - 275488177.753x^7 \\ + 9373287.575x^8 - 140999.967x^9 \end{aligned}$$

Dengan hasil prediksi sebagaimana berikut.

```
f(7.52) = 53532
f(8.32) = 36315
f(9.17) = -664804
f(7.55) = 51952
```

Gambar 19 Luaran Kasus Interpolasi Polinom-b

Perhatikan bahwa prediksi saat $x = 9.17$ bernilai negatif. Hal ini tidak bisa disalahkan mengingat data yang tersedia hanya data pada bulan Juni hingga Agustus, sedangkan kita mencoba memprediksi nilai saat bulan September, yang berada di luar range data. Hal ini menunjukkan bahwa interpolasi hanya bisa digunakan untuk memprediksi data yang berada pada rentang data yang diketahui dan tidak bisa digunakan untuk memprediksi data di masa depan (data yang berada di luar rentang data yang diketahui).

c. Kasus ketiga

Diketahui fungsi berikut.

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

Bila menggunakan interpolasi derajat 5, diperoleh persamaan berikut.

```
Saat n = 5
f(x) = 0.000 - 0.031x^1 + 0.502x^2 - 0.241x^3 + 0.041x^4 - 0.002x^5
f(0.20) = 0.011991
f(0.55) = 0.098328
f(0.85) = 0.208825
f(1.28) = 0.380114
```

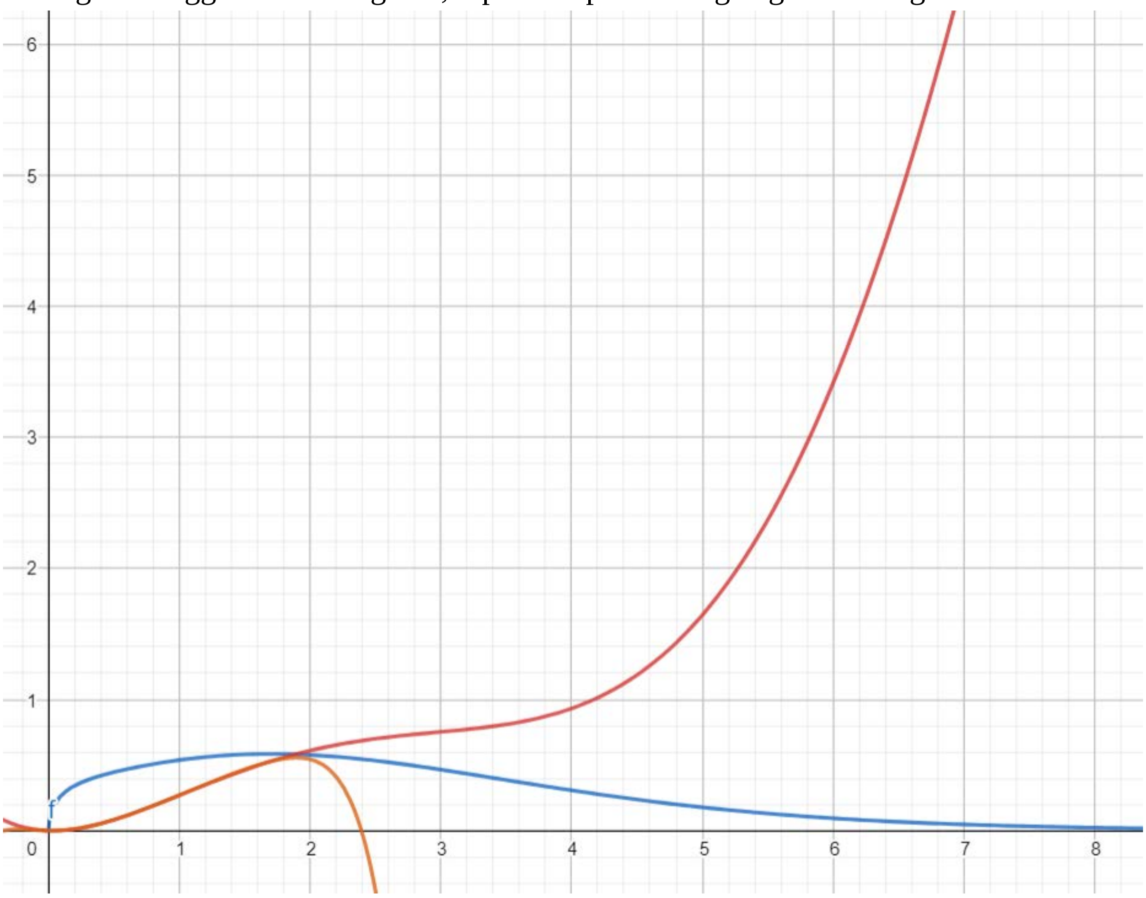
Gambar 20 Luaran Kasus Interpolasi Polinom-c (derajat 5)

Sedangkan bila menggunakan interpolasi derajat 10, diperoleh persamaan berikut.

```
Saat n = 10
f(x) = 0.000 - 0.010x^1 + 0.329x^2 + 0.374x^3 - 1.202x^4 + 1.590x^5 - 1.345x^6 + 0.750x^7 - 0.265x^8 + 0.054x^9 - 0.005x^10
f(0.20) = 0.012585
f(0.55) = 0.098253
f(0.85) = 0.208835
f(1.28) = 0.380105
```

Gambar 21 Luaran Kasus Interpolasi Polinom-b (derajat 10)

Dengan menggunakan Geogebra, diperoleh perbandingan grafik sebagaimana berikut.



Gambar 22 Grafik Perbandingan Interpolasi Polinom

Grafik berwarna biru adalah persamaan awal, grafik berwarna merah adalah interpolasi derajat 5, dan grafik berwarna oranye adalah interpolasi derajat 10.

4.5. Interpolasi Bicubic

Diketahui matriks berikut.

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

Setelah melakukan perkalian matriks dengan konstanta bicubic, diperoleh kontanta sebagai berikut.

```
0 = {double[1]@1036} [160.99999999999997]
1 = {double[1]@1037} [-7.666666666666705]
2 = {double[1]@1038} [-81.0]
3 = {double[1]@1039} [28.66666666666668]
4 = {double[1]@1040} [-63.66666666666664]
5 = {double[1]@1041} [-43.30555555555553]
6 = {double[1]@1042} [98.83333333333333]
7 = {double[1]@1043} [-35.02777777777778]
8 = {double[1]@1044} [-62.49999999999998]
9 = {double[1]@1045} [-33.833333333333336]
10 = {double[1]@1046} [108.25]
11 = {double[1]@1047} [-42.91666666666667]
12 = {double[1]@1048} [37.16666666666667]
13 = {double[1]@1049} [20.805555555555557]
14 = {double[1]@1050} [-72.08333333333334]
15 = {double[1]@1051} [29.277777777777775]
```

Gambar 23 Nilai Perhitungan Konstanta interpolasi

Dari atas ke bawah, nilai konstanta adalah konstanta a_{00}, a_{10}, a_{20}, a_{30}, a_{01}, ... a_{33}. Kemudian, hasil interpolasi pada beberapa titik adalah sebagaimana berikut.

```
Hasil prediksi f(0.00,0.00) adalah 161.000000
Hasil prediksi f(0.50,0.50) adalah 97.726562
Hasil prediksi f(0.25,0.75) adalah 82.502075
Hasil prediksi f(0.50,0.50) adalah 74.696119
```

Gambar 24 Luaran Hasil Prediksi Interpolasi Bicubic

4.6. Regresi Linier Berganda

Diketahui data berikut.

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.

Gambar 25 Data Sampel sebagai Kasus Uji Regresi Linier Berganda

Dengan menggunakan regresi linear berganda, diperoleh *Normal Estimation Equation* berikut.

```
SPL estimasi normal
20.000b0 + 863.100b1 + 1530.400b2 + 587.840b3 = 19.420
863.100b0 + 54876.890b1 + 67000.090b2 + 25283.395b3 = 779.477
1530.400b0 + 67000.090b1 + 117912.320b2 + 44976.867b3 = 1483.437
587.840b0 + 25283.395b1 + 44974.867b2 + 17278.509b3 = 571.122
```

Gambar 26 SPL Estimasi Normal untuk Kasus Uji Regresi Linier Berganda

Persamaan hasil regresi dan prediksi adalah sebagai berikut.

```
f(x) = -3.507778140862092 - 0.002624990745881746x1 + 7.989410472207176E-4x2 + 0.15415503019758064x3
f(50.000, 76.000, 29.300) = 0.9384342262217085
```

Gambar 27 Persamaan Regresi dan Hasil Prediksi f(x) berdasarkan x1, x2, dan x3

4.7. Performa Determinan dan Invers Matriks dengan Metode Minor-Kofaktor dan Eliminasi Gauss

Perhitungan determinan matriks menggunakan minor-kofaktor dan invers matriks dengan metode adjoin berjalan dengan lancar hingga kami harus melakukan invers matriks berukuran 16x16 untuk implementasi interpolasi bicubic. Saat itu, program kami memerlukan waktu puluhan detik untuk menghitung determinannya, lalu beberapa menit untuk menentukan invers matriks dengan metode adjoin. Akibat hal itu, akhirnya kami menghitung determinan dan menentukan invers matriks menggunakan metode Gauss-Jordan.

Setelah ditelusuri, kompeksitas algoritma untuk perhitungan determinan dengan minor-kofaktor dan invers matriks dengan metode adjoin adalah $O(n!)$. Sedangkan, menentukan determinan dan invers matriks menggunakan metode Gauss memiliki kompleksitas $O(n^3)$. Sehingga, bisa disimpulkan bahwa perhitungan secara iteratif menggunakan metode Gauss secara komputasi jauh lebih efisien daripada metode minor-kofaktor dan metode adjoin.

4.8. Bonus – Pembesaran Gambar

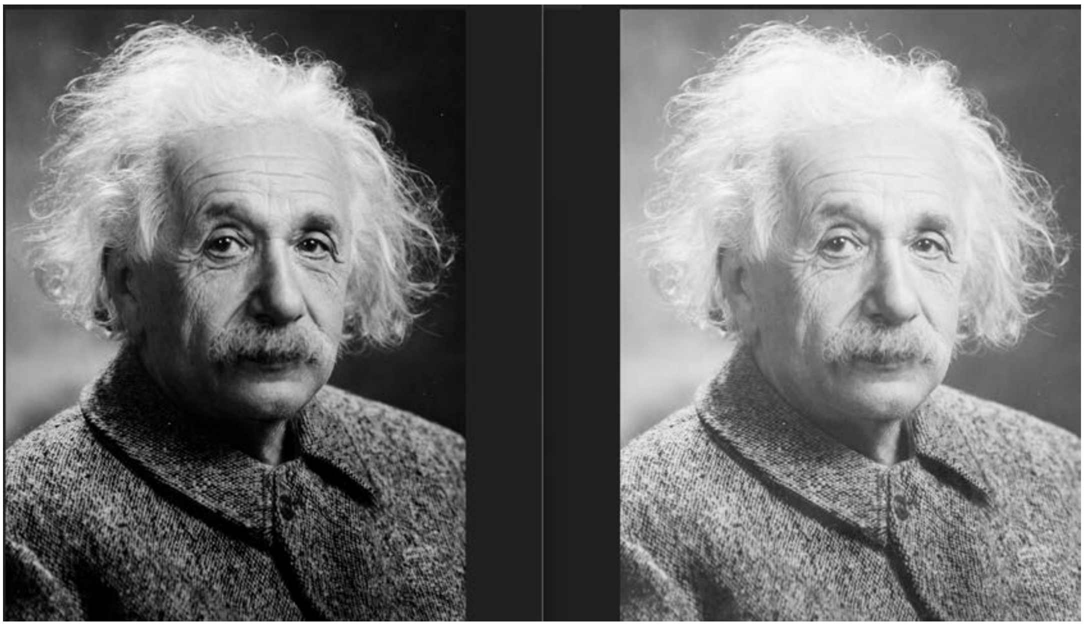
Implementasi interpolasi gambar menggunakan bicubic interpolation yang kami buat memberikan hasil yang bervariasi. Terkadang, hasil interpolasi memberikan hasil yang baik, terkadang pula memberikan hasil yang kurang memuaskan karena menghasilkan beberapa *artifacts*. Silakan perhatikan beberapa hasil interpolasi gambar berikut. Gambar bagian kiri adalah gambar asal, dan gambar kanan adalah gambar hasil interpolasi.

Gambar 28 Demonstrasi Pembesaran Citra pada Meme "Ano o-kata wa ancrit sugiru"

Gambar 29 Demonstrasi Pembesaran Citra pada Logo ITB

(Sumber: [https://id.wikipedia.org/wiki/Berkas:Logo_Institut_Teknologi_Bandung.png)](https://id.wikipedia.org/wiki/Berkas:Logo_Institut_Teknologi_Bandung.png)

Gambar 30 Demonstrasi Pembesaran Citra pada Gambar Kitagawa Marin



Gambar 31 Demonstrasi Pembesaran Citra pada Potret Albert Einstein

(Sumber: https://id.m.wikipedia.org/wiki/Berkas:Albert_Einstein_Head.jpg)

Berikut adalah tabel lengkap mengenai perbandingan dan performa untuk setiap percobaan.

Nama	Format	Ukuran awal	Ukuran akhir	Time (single core, dalam detik)	Time (eight core, dalam detik)
Ancrit	jpg	720x543	1440x1086	2.744	0.563
Logo ITB	png	512x512	1024x1024	2.073	0.505
Marin	png	1920x1080	3840x2160	16.216	2.390
Einstein	Jpg	3250x4333	6500x8666	95.412	12.936

Catatan: setiap gambar hasil interpolasi disimpan dalam format png.

Implementasi awal program adalah menggunakan *sequential processing* saat melakukan proses interpolasi menggunakan interpolasi bicubic. Namun, seiring dengan meningkatnya ukuran gambar yang diinterpolasi, waktu yang diperlukan juga meningkat dengan pesat sehingga perlu adanya optimasi. Oleh karena itu, kami melakukan implementasi *parallel processing* untuk operasi interpolasi bicubic menggunakan *parallel stream*. Hasilnya, pada gambar dengan ukuran cukup besar kecepatan pemrosesan meningkat beberapa kali lipat, berbanding lurus dengan jumlah core yang dimiliki sistem.

Selain itu, program mulai bermasalah saat kami mencoba untuk menginterpolasi gambar dengan ukuran kecil. Gambar Ancrit, misalnya, muncul artifacts pada outline text “Ano o-kata wa ancrit sugiru" sehingga terlihat banyak pixel kotak-kotak. Hal yang sama terjadi saat kami mencoba menginterpolasi logo ITB yang memilki format png. Garis yang membentuk logo gajah, outline text, serta bagian lingkaran bermunculan banyak artifacts berwarna merah. Berdasarkan hipotesis kami, artifacts tersebut muncul saat program mencoba menginterpolasi nilai yang berbatasan dengan pixel yang memiliki transparansi tinggi serta mencoba menginterpolasi bagian gambar yang berupa garis dengan lebar hanya beberapa pixel.

Pada gambar Marin, tidak terlihat masalah dalam proses interpolasinya. Sedangkan, pada gambar Einstein, brighness gambar hasil interpolasi malah memiliki kecerahan yang lebih tinggi daripada gambar awal. Sayangnya, kami belum mengetahui penyebab bug ini. Namun, hal tersebut bisa saja disebabkan karena perbedaan format warna pada input gambar mengingat implementasi kami hanya menggunakan 4 channel RGBA.

Bisa disimpulkan bahwa implementasi pembesar gambar kami berjalan dengan baik apabila masukan gambar memiliki resolusi yang cukup besar, seperti 1080p, agar tidak timbul artifacts yang tidak diinginkan. Selain itu, algoritma ini bermasalah ketika mencoba menginterpolasi bagian yang berupa garis dengan lebar kecil.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil pengerjaan kami, kami menyimpulkan bahwa kami cukup berhasil membuat program dalam bahasa Java yang mampu mengkalkulasi matriks, menyelesaikan SPL dengan berbagai metode, memroses data, melakukan prediksi dan interpolasi terhadap data, serta melakukan perbesaran citra. Dengan ini kami juga menyimpulkan bahwa algoritma dari perhitungan yang melibatkan matriks, baik penyelesaian SPL, determinan, maupun interpolasi, dapat diterjemahkan menjadi sebuah program komputer.

5.2. Saran

Sebagai manusia, kami menyadari bahwa hasil pengerjaan kami tidak sempurna, baik terkait optimisasi performa program, estetika tampilan program, kerapian laporan ini, ataupun hal yang lain. Maka dari itu, kami sangat terbuka untuk menerima masukan, perbaikan, dan pengembangan lebih lanjut terhadap karya kami.

5.3. Refleksi

Kami bersyukur dapat menyeselaikan tugas besar (tubes) ini dengan tepat waktu dan juga bisa menyelesaikan soal bonus. Selain itu, tubes ini telah menjadi sarana bagi kami untuk mengeksplor dunia pemrograman dan memperdalam konsep yang telah diajarkan. Terlebih lagi, proses pembuatan modul penyelesaian SPL dengan metode Gauss yang juga harus menangani persamaan parametrik sangat menantang.

Tidak banyak hal yang ingin kami kembangkan lebih jauh terkait program utama selain optimasi performa. Bagi kami, hal yang menarik untuk dikembangkan lebih lanjut adalah soal bonus, seperti menangani lebih banyak *color space* agar warna tidak hilang, mengatur skala pembesaran sehingga tidak hanya 2x saja, serta optimasi performa interpolasi agar bisa lebih efisien secara kecepatan dan penggunaan memori.

DAFTAR REFERENSI

Anton, H., & Rorres, C. (2013). *Elementary Linear Algebra, 11th edition*. John Wiley & Sons.

JetBrains s.r.o. (2022). *Compile and build applications with intellij IDEA: Intellij idea*. IntelliJ IDEA Help. Diakses October 1, 2022, dari https://www.jetbrains.com/help/idea/compiling-applications.html

Munir, R. (2020). *Determinan (bagian 1)*. IF2123 Aljabar Geometri - Semester I Tahun 2022/2023. Diakses September 29, 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-08-Determinan-bagian1.pdf

Munir, R. (2020). *Determinan (bagian 2)*. IF2123 Aljabar Geometri - Semester I Tahun 2022/2023. Diakses September 30, 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-09-Determinan-bagian2.pdf

Munir, R. (2020). *Matriks Eselon*. IF2123 Aljabar Geometri - Semester I Tahun 2022/2023. Diakses September 27, 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-02-Matriks-Eselon.pdf

Munir, R. (2020). *Review Matriks*. IF2123 Aljabar Geometri - Semester I Tahun 2022/2023. Diakses September 27, 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-01-Review-Matriks.pdf

Munir, R. (2020). *Sistem Persamaan Linier (SPL) - Pokok bahasan: Metode Eliminasi Gauss-Jordan*. IF2123 Aljabar Geometri - Semester I Tahun 2022/2023. Diakses September 29, 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-05-Sistem-Persamaan-Linier-2.pdf

Munir, R. (2020). *Sistem Persamaan Linier (SPL) - Pokok bahasan: Tiga kemungkinan solusi SPL*. IF2123 Aljabar Geometri - Semester I Tahun 2022/2023. Diakses September 28, 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-04-Tiga-Kemungkinan-Solusi-SPL.pdf

Munir, R. (2020). *Sistem Persamaan Linier (SPL)*. IF2123 Aljabar Geometri - Semester I Tahun 2022/2023. Diakses September 28, 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-03-Sistem-Persamaan-Linier.pdf

Munir, R. (2022). *Tugas Besar 1 IF 2123 Aljabar Linier dan Geometri, Sistem Persamaan Linier, Determinan, dan Aplikasinya, Semester I Tahun 2022/2023*. IF2123 Aljabar Geometri - Semester I Tahun 2022/2023. Diakses September 25, 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2022-2023/Tubes1-Algeo-2022.pdf

Rowe, D. B. (2018). *Bilinear, bicubic, and in between spline interpolation - marquette*. Mathematical and Statistical Sciences // Marquette University. Diakses October 1, 2022, dari https://www.mssc.mu.edu/~daniel/pubs/RoweTalkMSCS_BiCubic.pdf

LAMPIRAN

Repository GitHub Pengerjaan Tugas Besar 1: https://github.com/haiakbar/Algeo01-21074