IF2123 Aljabar Linear dan Geometri

IMPLEMENTASI KALKULATOR MATRIKS, SISTEM PERSAMAAN LINEAR, DAN APLIKASINYA MENGGUNAKAN BAHASA PEMROGRAMAN JAVA

Laporan Tugas Besar 1

DIsusun untuk memenuhi tugas mata kuliah Aljabar Linear dan Geometri
pada Semester 1 (satu) Tahun Akademik 2023/2024

Oleh

Rizqika Mulia Pratama	**13522126**
Shabrina Maharani	**13522134**
Auralea Alvinia Syaikha	**13522148**

Kelompok Mandala Jaya

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2023

DAFTAR ISI

BAB 1 DESKRIPSI MASALAH

Matriks adalah susunan angka-angka yang diatur dalam baris dan kolom. Matriks memiliki banyak kegunaan dalam berbagai bidang, seperti matematika, sains, rekayasa, dan ekonomi. Matriks dapat digunakan untuk merepresentasikan objek atau proses matematika, seperti persamaan linear dan transformasi linear.

Sistem persamaan linier (SPL) adalah sekumpulan persamaan linier yang saling terkait. SPL banyak ditemukan dalam bidang sains dan rekayasa. SPL dapat diselesaikan dengan berbagai metode, seperti metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer. Solusi SPL bisa saja satu, banyak, atau bahkan tidak ada sama sekali.

Dalam Tugas Besar 1 Aljabar Linier dan Geometri, penulis membuat pustaka aljabar linier dalam bahasa Java. Pustaka tersebut berfungsi untuk menyelesaikan SPL, menentukan matriks balikan, menghitung determinan, menyelesaikan interpolasi polinom, menghitung regresi linier berganda, dan menyelesaikan interpolasi bikubik. Pustaka tersebut kemudian digunakan dalam program Kalkulator Matriks by Mandala Jaya. Program tersebut dapat menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, interpolasi, dan regresi.

Spesifikasi program yang dibuat adalah sebagai berikut:

1. 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

2. 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

   Luaran (*output*) disesuaikan dengan persoalan (determinan atau invers) dan penghitungan balikan/invers dilakukan dengan metode matriks balikan dan adjoin.

1. 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. Masukan kemudian dilanjutkan dengan satu buah baris berisi satu buah nilai x yang akan ditaksir menggunakan fungsi interpolasi yang telah didefinisikan. Misalnya jika titik-titik datanya

adalah (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513) dan akan mencari nilai y saat x = 8.3, maka di dalam *file text* ditulis sebagai berikut:

8.0 2.0794

9.0 2.1972

9.5 2.2513

8.3

2. 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.
3. Untuk persoalan SPL, luaran 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 $x_1 = t$).
4. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan. Contoh luaran untuk interpolasi adalah

$$f(x) = -0.0064x^2 + 0.2266x + 0.6762, \quad f(5) = \ldots$$

dan untuk regresi adalah

$$f(x) = -9.5872 + 1.0732x_1, \quad f(x_k) = \ldots$$

5. Untuk persoalan *bicubic spline interpolation*, masukan dari *file text* (.txt) yang berisi matriks berukuran 4 x 4 yang berisi konfigurasi nilai fungsi dan turunan berarah disekitarnya, diikuti dengan nilai a dan b untuk mencari nilai $f(a, b)$.

   Misalnya jika nilai dari $f(0, 0)$, $f(1, 0)$, $f(0, 1)$, $f(1, 1)$, $f_x(0, 0)$, $f_x(1, 0)$, $f_x(0, 1)$, $f_x(1, 1)$, $f_y(0, 0)$, $f_y(1, 0)$, $f_y(0, 1)$, $f_y(1, 1)$, $f_{xy}(0, 0)$, $f_{xy}(1, 0)$, $f_{xy}(0, 1)$, $f_{xy}(1, 1)$ 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 $f(0.5, 0.5)$.
6. Luaran program dapat ditampilkan pada layar komputer dan dapat disimpan ke dalam *file*.
7. Bahasa program yang digunakan adalah Java.
8. Program berbasis *text-based*.

9. Program dibuat dengan pilihan menu. Dengan urutan menu:

   MENU
   1. Sistem Persamaaan Linier
   2. Determinan
   3. Matriks balikan
   4. Interpolasi Polinom
   5. Interpolasi Bicubic Spline
   6. Regresi linier berganda
   7. Keluar

   Untuk pilihan menu nomor 1 ada sub-menu lagi yaitu pilihan metode:
   1. Metode eliminasi Gauss
   2. Metode eliminasi Gauss-Jordan
   3. Metode matriks balikan
   4. Kaidah Cramer

   Begitu juga untuk pilihan menu nomor 2 dan 3.

BAB 2 TEORI SINGKAT

Dalam implementasi pustaka Tugas Besar 1 ini terdapat lima fungsi besar, yaitu penyelesaian SPL, interpolasi polinom, interpolasi bicubic, regresi linier berganda, dan image scaling. Berikut merupakan penjelasan tentang setiap bagian.

2.1 Sistem Persamaan Linear

Sistem persamaan linear adalah kumpulan persamaan linear yang saling terkait. Persamaan linear adalah persamaan yang memiliki variabel dengan pangkat satu. Sistem persamaan linear dapat terdiri dari satu variabel, dua variabel, atau lebih.

2.1.1 Metode Eliminasi Gauss

Metode eliminasi Gauss adalah teknik matematika yang digunakan untuk menyelesaikan sistem persamaan linear dengan mengubah matriks *augmented* menjadi bentuk eselon, di mana setiap baris memiliki leading one (satu elemen terkemuka) di kolom tertentu dan semua elemen di bawah leading one adalah nol. Langkah-langkahnya melibatkan operasi baris seperti pertukaran baris, penggantian baris, dan pembagian baris untuk mencapai bentuk eselon ini.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_n \end{bmatrix} \sim\text{OBE}\sim \begin{bmatrix} 1 & * & * & \dots & * & * \\ 0 & 1 & * & \dots & * & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \vdots & 1 & * \end{bmatrix}$$

2.1.2 Metode Eliminasi Gauss-Jordan

Metode eliminasi Gauss-Jordan adalah teknik matematika yang digunakan untuk menemukan solusi sistem persamaan linear dan mengubah matriks augmented menjadi bentuk eselon tereduksi, di mana setiap baris memiliki satu elemen terkemuka (leading one) di kolom tertentu, dengan semua elemen di atas dan di bawah leading one menjadi nol. Ini dilakukan melalui serangkaian operasi elemen baris, termasuk pertukaran baris, penggantian baris, dan pembagian baris.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_m \end{bmatrix} \sim\text{OBE}\sim \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & * \\ 0 & 1 & 0 & \dots & 0 & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \vdots & 1 & * \end{bmatrix}$$

2.1.3 Metode Matriks Balikan

Metode penentuan SPL dengan menggunakan matriks balikan hanya dapat digunakan pada matriks persegi dan juga ketika determinan ≠ 0. Ide dasarnya adalah dengan mengalikan matriks koefisien dari sistem persamaan dengan matriks inversnya (x = A ^-1 b), sehingga kita mendapatkan matriks solusi yang berisi nilai-nilai variabel yang dicari.

2.1.4 Kaidah Cramer

Kaidah Cramer adalah metode khusus untuk menyelesaikan sistem persamaan linear dengan variabel sebanyak yang sama dengan jumlah persamaan. Menurut Kaidah Cramer, jika Ax = b adalah SPL yang terdiri dari n persamaan linier dengan n peubah (variable) sedemikian sehingga det(A) ≠ 0, maka SPL tersebut memiliki solusi yang unik, yaitu:

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

dimana Ai adalah matriks yang diperoleh dengan mengganti entri pada kolom ke-i dari A dengan entri dari matriks b.

2.2 Determinan

Determinan adalah sebuah abstraksi yang melambangkan suatu nilai yang bisa didapatkan dari sebuah matriks persegi. Nilai determinan digunakan untuk mengukur sifat-sifat geometris dan aljabar matriks tersebut. Determinan dari suatu matriks persegi A umumnya dilambangkan dengan det(A). Proses perhitungan determinan melibatkan aturan dan metode khusus, seperti metode ekspansi kofaktor atau metode reduksi baris, tergantung pada ukuran dan sifat matriks yang bersangkutan.

2.2.1 Metode Reduksi Baris

Metode reduksi baris adalah salah satu teknik yang digunakan dalam aljabar linear untuk menyederhanakan dan mempermudah pemecahan sistem persamaan linear. Determinan matriks A dapat diperoleh dengan melakukan OBE pada matriks A sampai diperoleh matriks segitiga (segitiga bawah atau atas).

$$[A] \overset{\text{OBE}}{\sim} [\text{matriks segitiga bawah}]$$

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \overset{\text{OBE}}{\sim} \begin{bmatrix} a'_{11} & a'_{12} & \dots & a'_{1n} \\ 0 & a'_{22} & \dots & a'_{2n} \\ \vdots & \vdots & \dots & a'_{3n} \\ 0 & 0 & 0 & a'_{nn} \end{bmatrix}$$

$$\text{maka } \det(A) = (-1)^p \, a'_{11} a'_{22} \dots a'_{nn}$$

p menyatakan banyaknya operasi pertukaran baris di dalam OBE

2.2.2 Metode Ekspansi Kofaktor

Terdapat sebuah matriks persegi A seperti berikut:

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

Definisikan notasi Mij sebagai minor dari entri aij, yaitu determinan dari sub-matriks yang elemen-elemennya adalah elemen matriks A yang tidak berada pada baris i dan kolom j. Lalu didefinisikan juga Cij sebagai kofaktor dari entri aij, dengan rumus Cij sebagai berikut:

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

Maka, determinan dari matriks A dapat ditentukan dengan salah satu dari persamaan berikut.

$$\begin{aligned} \det(A) &= a_{11}C_{11} + a_{12}C_{12} + \dots + a_{1n}C_{1n} \\ \det(A) &= a_{21}C_{21} + a_{22}C_{22} + \dots + a_{2n}C_{2n} \\ &\vdots \\ \det(A) &= a_{n1}C_{n1} + a_{n2}C_{n2} + \dots + a_{nn}C_{nn} \end{aligned}$$

Secara baris

$$\begin{aligned} \det(A) &= a_{11}C_{11} + a_{21}C_{21} + \dots + a_{n1}C_{n1} \\ \det(A) &= a_{12}C_{12} + a_{22}C_{22} + \dots + a_{n2}C_{n2} \\ &\vdots \\ \det(A) &= a_{1n}C_{1n} + a_{2n}C_{2n} + \dots + a_{nn}C_{nn} \end{aligned}$$

Secara kolom

2.3 Balikan Matriks

Balikan matriks, juga dikenal sebagai matriks invers, adalah matriks yang ketika dikalikan dengan matriks asalnya menghasilkan matriks identitas. Matriks balikan hanya ada jika determinan matriks asalnya tidak nol. Proses menemukan matriks balikan melibatkan teknik-teknik seperti eliminasi Gauss-Jordan atau menggunakan metode matriks Adjoin.

2.3.1 Metode Matriks Adjoin

Metode matriks adjoin adalah teknik dalam aljabar matriks yang digunakan untuk menemukan matriks adjoin dari matriks persegi. Matriks adjoin adalah matriks yang diperoleh dengan menukar elemen-elemen matriks dengan kofaktor masing-masing dan kemudian melakukan tranposisi. Balikan matriks A dapat dihitung dengan menggunakan rumus:

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

2.3.2 Metode Eliminasi Gauss Jordan

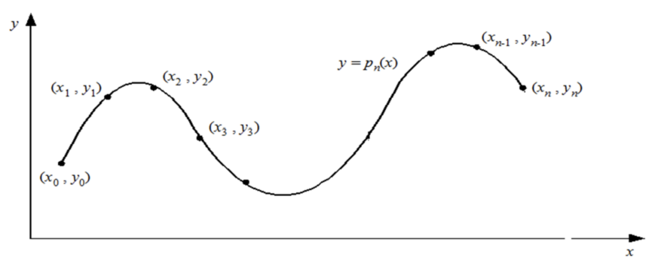
Metode eliminasi Gauss-Jordan dapat digunakan untuk menghitung matriks balikan. Misalkan A adalah matriks persegi berukuran n x n. Balikan (inverse) matriks A adalah A^-1 sedemikian sehingga AA^-1 = A^-1 A = I. Untuk matriks A yang berukuran n x n, matriks balikannya, yaitu A^-1, dicari dengan cara berikut:

$$[A|I] \overset{\text{G-J}}{\sim} [I|A^{-1}]$$

Yang dalam hal ini I adalah matriks identitas berukuran n x n.

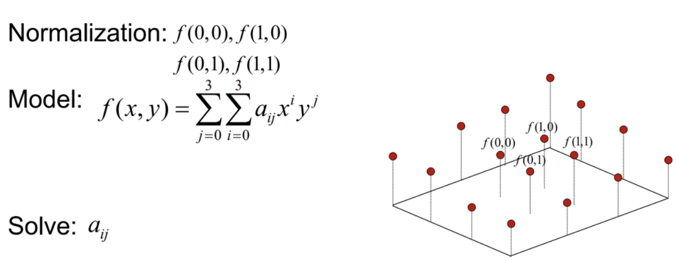
2.4 Interpolasi Polinom

Interpolasi polinom merupakan teknik interpolasi dengan mengasumsikan pola data yang kita miliki mengikuti pola polinomial baik berderajat satu (linier) maupun berderajat tinggi. Interpolasi dengan metode ini dilakukan dengan terlebih dahulu membentuk persamaan polinomial Pn (x) dari n+1 buah titik berbeda, (x0 , y0), (x1 , y1), …, (xn , yn) sehingga yi = Pn (xi) untuk i = 0, 1, 2, …, n. Sehingga, kita dapat menggunakan persamaan polinomial tersebut untuk menghitung perkiraan nilai y di x sembarang.



2.5 Interpolasi Bicubic Spline

Metode interpolasi bikubik spline adalah pendekatan dalam analisis numerik yang digunakan untuk mengaproksimasi fungsi berkelanjutan yang melewati sejumlah titik data. Metode ini melibatkan konsep spline dan pembentukan serangkaian polinomial kubik di dalam setiap sel segi empat dari data yang diberikan. Dalam penggunaan interpolasi bicubic spline, 16 titik digunakan, yang terdiri dari 4 titik referensi utama di pusat sel, serta 12 titik tambahan di sekitarnya yang digunakan sebagai estimasi turunan dari keempat titik referensi untuk membangun permukaan bikubik.



Persamaan polinomial yang digunakan adalah sebagai berikut.

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

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

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

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

2.6 Regresi Linear Berganda

Regresi linear berganda matriks adalah teknik statistik yang digunakan untuk memodelkan hubungan antara variabel dependen dan satu atau lebih variabel independen dengan menggunakan matriks. Hal ini dilakukan dengan memprediksi persamaan yang berlaku menggunakan data-data yang ada hingga membentuk sebuah persamaan linear. Terdapat persamaan 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:

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

BAB 3 IMPLEMENTASI PUSTAKA

3.1 Class matrix

Class ini berisi tipe data matriks yang digunakan dalam berbagai aplikasi program.

3.1.1 Atribut

CAPACITY	integer yang menyimpan jumlah elemen yang dapat ditampung oleh satu baris atau kolom matriks
nCol	integer yang menyimpan jumlah kolom dalam matrik
nRow	integer yang menyimpan jumlah baris dalam matriks
Matrix	array 2D bertipe double yang digunakan untuk menyimpan elemen matriks.

3.1.2 Metode

Implementasi Metode	Garis Besar Program
`public void readMatrix(int m, int n)`	Metode untuk membaca matriks dari terminal dengan parameter m berisikan jumlah baris dan parameter n berisikan jumlah kolom.
`public void readFileMatrix(String filename)`	Metode untuk membaca matriks dari sebuah file yang berada di folder test, parameter filename berisikan nama file.
`public void readFileMatrixBolong(String filename, int nKosong)`	Metode untuk membaca matriks dari sebuah file yang berada di folder test, yang memiliki parameter filename berisikan nama file dan parameter nKosong berisikan banyak elemen matriks pada baris terbawah yang kosong. Metode ini secara khusus digunakan untuk membaca matriks dari file yang akan diolah dengan interpolasi dan regresi.
`public void writeMatrixFile(matrix m)`	Metode untuk menuliskan matriks di sebuah file yang berada di folder test, nama

	file akan bergantung kepada input pengguna pada saat fungsi ini dijalankan
`public boolean isAllZero()`	Metode untuk mengecek apakah semua elemen dalam matriks bernilai nol
`public void writeMatrix()`	Metode untuk menuliskan isi matriks pada terminal.

3.2 Class matrixOperation

Class ini berisikan metode operasi biner maupun uner yang dilakukan pada matriks.

3.2.1 Atribut

Class ini tidak memiliki atribut.

3.2.2 Metode

Implementasi Kode	Garis Besar Program
`public static boolean isEqual(matrix m1, matrix m2)`	Metode untuk mengecek apakah dimensi matriks m1 dan m2 sama
`public static matrix cloneMatrix(matrix mIn)`	Metode untuk menduplikasi matriks
`public static matrix multiplyMatrix(matrix m1, matrix m2)`	Metode untuk mengalikan dua buah matriks (m1 dan m2)
`public static matrix multiplyMatrixBik(matrix m1, matrix m2)`	Metode untuk mengalikan dua buah matriks untuk interpolasi bikubik
`public static matrix transpose(matrix mIn)`	Metode untuk melakukan transpose matrix
`public static matrix rowSwap(matrix mIn, int b1, int b2)`	Metode untuk menukar baris matriks
`public static matrix slice(matrix mIn, int i, int j)`	Metode untuk mengambil elemen yang barisnya bukan i dan kolomnya bukan j
`public static matrix sliceLastRow(matrix mIn)`	Metode untuk membuang baris terakhir

`public static matrix sliceLastCol(matrix mIn)`	Metode untuk membuang kolom terakhir
`public static matrix takeLastRow(matrix mIn)`	Metode untuk mengambil baris terakhir
`public static matrix takeLastCol(matrix mIn)`	Metode untuk mengambil kolom terakhir.
`static void tidyUp(matrix mIn)`	Metode untuk mengatur nilai-nilai dalam matriks sehingga nilai-nilainya mendekati 0 atau 1 dalam batasan toleransi yang sangat kecil.
`public static matrix compactzero(matrix mIn)`	Metode untuk mengkompakkan matriks dengan menggeser semua elemen nol ke bagian bawah matriks.
`public static matrix rowXConst(matrix mIn, int baris, double konstanta)`	Metode untuk mengalikan semua elemen dalam satu baris tertentu dari matriks dengan sebuah konstanta.
`public static matrix minKaliBaris(matrix mIn, int barisTujuan, int barisPengurang, double konstanta)`	Metode untuk mengurangkan dan mengalikan satu baris tertentu dari matriks.
`public static matrix concatCol(matrix m1, matrix m2)`	Metode untuk menyatukan m1 dan m2 secara kolom.
`public static matrix cramerSwap(matrix m2, matrix m1, int col)`	Metode yang akan mengembalikan matrix untuk menyelesaikan sistem persamaan linear dengan kaidah cramer.

3.3 Class SPL

3.3.1 Atribut

Class ini tidak memiliki atribut

3.3.2 Konstruktor

Implementasi Kode	Garis Besar Program
`public SPL()`	Konstruktor untuk SPL

3.3.3 Metode

`public static void solveSPL(matrix mIn)`	Metode untuk menuliskan solusi SPL di terminal sesuai dengan salah satu dari tiga jenis solusi matriks tersebut
`public static void solveSPLFile(matrix mIn)`	Metode untuk menuliskan solusi SPL di file sesuai dengan salah satu dari tiga jenis solusi matriks tersebut
`public static void solveWithInverse(matrix mIn)`	Metode untuk implementasi solusi inverse
`public static void solveWithInverseFile(matrix mIn)`	Metode untuk membuat file dari solusi inverse
`public static void solveCramer(matrix mIn)`	Metode untuk implementasi solusi kaidah Cramer
`public static void solveCramerFile(matrix mIn)`	Metode untuk membuat file dari solusi kaidah cramer
`public static int checkSPL(matrix mIn)`	Metode untuk mengecek jenis solusi matriks, dengan prekondisi bahwa matriks mIn adalah matriks gauss/gauss-jordan SPL. 0 = Matriks kosong (semua 0) 1 = Solusi unik 2 = Solusi banyak 3 = Solusi tidak ada
`static matrix removerow0 (matrix mIn)`	Metode untuk menghapus baris yang elemennya 0 semua
`static boolean solvable(matrix mIn)`	Metode untuk mengecek apakah matriks punya solusi atau tidak
`public static int cariSatu(matrix mIn, int row)`	Metode untuk mencari 1 pertama di suatu baris
`public static void solusiKosong(matrix mIn)`	Metode untuk mencetak solusi kosong
`public static void solusiKosongFile(matrix mIn, String filename)`	Metode untuk menulis solusi kosong di file
`public static void solusiUnik(matrix mIn)`	Metode untuk mencari solusi unik dari persamaan gauss
`public static void solusiUnikFile(matrix mIn,`	Metode untuk menulis file solusi unik

String filename)	dari persamaan gauss
public static void solusiBanyak(matrix mIn)	Metode untuk mencari solusi banyak dari persamaan gauss
public static void solusiBanyakFile(matrix mIn, String filename)	Metode untuk menulis file solusi banyak dari persamaan gauss
public static void solusiNone()	Metode untuk menampilkan tidak ada solusi
public static void solusiNoneFile(matrix mIn, String filename)	Metode untuk menulis file tidak ada solusi
public static double recursion(int toplimit, int bottomlimit, int row, int varCol, double arrayHasil[], String arrayString[], matrix mIn)	Metode untuk melakukan perhitungan rekursif yang terkait dengan manipulasi matriks dan sistem persamaan linear
public static matrix *cramerSwap*(matrix m2, matrix m1, int col)	Metode yang akan mengembalikan matrix untuk menyelesaikan sistem persamaan linear dengan kaidah cramer.
public static void kaidahCramer(matrix m)	Metode yang digunakan untuk mengimplementasikan kaidah cramer.
public static matrix matrixCof(matrix mIn)	Metode yang digunakan untuk mendapatkan matriks kofaktor tiap elemen dari baris dan kolom.
public static matrix gaussJordan(matrix mIn)	Metode yang digunakan untuk mengimplementasikan langkah langkah dari Gauss Jordan.
public static matrix gauss(matrix mIn)	Metode yang digunakan untuk mengimplementasikan langkah-langkah dari metode Gauss.

3.4 Class InterpolasiPolinom

3.4.1 Atribut

Class ini tidak memiliki atribut

3.4.2 Metode

Implementasi Kode	Garis Besar Program
public static matrix inputKeyboard()	Metode untuk menerima input dari keyboard
public static matrix x (matrix inputMatrix)	Metode untuk membuat matriks tanpa kolom pertama
public static matrix fx (matrix inputMatrix)	Metode untuk membuat matriks hasil perhitungan
public static double a (matrix inputMatrix)	Metode untuk mengekstrak nilai a dari matriks inputMatriks
public static matrix xi (matrix x)	Metode untuk mengembalikkan matriks xi dari matriks x
public static matrix ai (matrix xi, matrix fx)	Metode untuk membuat matriks ai dengan metode gauss dari matriks augmented xi | fx
public static double fa (matrix ai, double a)	Metode yang mengembalikan nilai polinom jika disubstitusikan x = a. Kembalian fungsi bernilai valid jika a berada dalam range data interpolasi.
public static String fxString(matrix ai)	Metode untuk menghasilkan string berupa penjabaran polinom hasil interpolasi
public static void IPFile(matrix ai, double a)	Metode yang menuliskan hasil interpolasi (penjabaran fungsi beserta substitusinya) ke dalam file .txt.

3.5 Class bicubicInterpolation

3.5.1 Atribut

Class ini tidak memiliki atribut

3.5.2 Metode

Implementasi Kode	Garis Besar Program
public static double getA(matrix mInput)	Metode untuk mengambil nilai a yang merupakan bagian dari matriks mInput

`public static double getB(matrix mInput)`	Metode untuk mengambil nilai b yang merupakan bagian dari matriks mInput
`public static matrix m4x4(matrix mInput)`	Metode yang mengembalikkan matriks berukuran 4x4 yang merupakan bagian dari matriks mInput
`public static matrix m16x1(matrix m4x4)`	Metode yang mengembalikkan matriks berukuran 16x1 yang dibuat dari matriks 4x4 (m4x4)
`public static matrix m16x16()`	Metode yang mengembalikkan matriks berukuran 16x16 (matriks X dalam bikubik interpolation)
`public static matrix mAij(matrix m16x1)`	Metode yang mengembalikkan matriks aij yang dibuat dengan cara mengalikan invers matriks m16x16 dengan matriks m16x1
`public static double getFab(matrix mAij, double a, double b)`	Metode untuk mengembalikkan nilai fungsi interpolasi jika disubtitusikan x=a dan y=b.
`public static void bicubicInterFile(matrix mAij, double a, double b)`	Metode yang menuliskan hasil interpolasi bikubik (hasil getFab) ke dalam file .txt

3.6 Class RegresiLinearBerganda

3.6.1 Atribut

Class ini tidak memiliki atribut

3.6.2 Metode

Implementasi Kode	Representasi Program
`public static matrix mtxfromkeyboard()`	Metode untuk mengambil data dari keyboard yang akan dijadikan sebagai elemen dari mtxInput.
`public static double ROW(matrix m, int s, int rowReg, int colReg)`	Metode untuk merubah matriks dari matriks vertikal menjadi matriks horizontal.
`public static matrix FUNCTION(matrix m)`	Metode yang mengembalikkan matriks yang merepresentasikan persamaan hasil

	metode Normal Estimation Equation for Multiple Linear Regression.
`public static void SOLUTION(matrix m)`	Metode yang digunakan untuk menghasilkan persamaan regresi linear dan nilai hampiran dari suatu nilai.
`public static void RLBFile(matrix m)`	Metode yang digunakan untuk menyimpan solusi dari persoalan regresi linear berganda ke sebuah file .txt.
`public static matrix gaussJordanReg(matrix mIn)`	Metode untuk melakukan gauss-jordan khusus untuk persamaan regresi linear berganda.

3.7 Class determinan

3.7.1 Atribut

Class ini tidak memiliki atribut

3.7.2 Metode

Implementasi Kode	Garis Besar Program
`public static double determinanKofaktor(matrix mIn)`	Metode untuk menghitung determinan dengan cara ekspansi kofaktor dari sebuah matriks yang diinialisasi sebagai mIn.
`public static double detOBE(matrix mIn)`	Metode untuk menghitung determinan dengan cara melakukan Operasi Baris Elementer dari sebuah matriks mIn dengan prekondisi matriks harus berbentuk persegi (mxm).
`public static void detFile(matrix mIn, double det)`	Metode untuk membuat file yang berisi hasil dari perhitungan determinan.
`public static double cofactor(matrix mIn, int i, int j)`	Metode yang digunakan untuk menghitung kofaktor yang akan digunakan untuk mencari determinan dengan cara ekspansi kofaktor.

3.8 Class balikan

3.8.1 Atribut

Class ini tidak memiliki atribut

3.8.2 Metode

Implementasi Kode	Garis Besar Program
`public static matrix inverseIdentitas(matrix mIn)`	Metode untuk melakukan invers dengan menggunakan implementasi dari matriks identitas.
`public static matrix inverseAdjoint`	Metode untuk melakukan invers matriks dengan menggunakan inverse Adjoint.

BAB 4 EKSPERIMEN

4.1. Temukan solusi SPL Ax = b

Soal		Metode	Output
a)	$A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$	Gauss	Mengambil file dari folder test. Nama file: 1_a 1.0 1.0 -1.0 -1.0 1.0 2.0 5.0 -7.0 -5.0 -2.0 2.0 -1.0 1.0 3.0 4.0 5.0 2.0 -4.0 2.0 6.0 Solusi tidak ada.
		Gauss-Jordan	Mengambil file dari folder test. Nama file: 1_a 1.0 1.0 -1.0 -1.0 1.0 2.0 5.0 -7.0 -5.0 -2.0 2.0 -1.0 1.0 3.0 4.0 5.0 2.0 -4.0 2.0 6.0 Solusi tidak ada.
		Matriks Balikan	Mengambil file dari folder test. Nama file: 1_a 1.0 1.0 -1.0 -1.0 1.0 2.0 5.0 -7.0 -5.0 -2.0 2.0 -1.0 1.0 3.0 4.0 5.0 2.0 -4.0 2.0 6.0 Matriks tidak memiliki inverse.

<table>
<tr><td></td><td></td><td>Kaidah Creamer</td><td>Mengambil file dari folder test.
Nama file: 1_a
1.0 1.0 -1.0 -1.0 1.0
2.0 5.0 -7.0 -5.0 -2.0
2.0 -1.0 1.0 3.0 4.0
5.0 2.0 -4.0 2.0 6.0

Tidak dapat menggunakan kaidah Cramer!</td></tr>
<tr><td colspan="4">Analisis:
Untuk SPL yang menggunakan metode Gauss dan Gauss-Jordan, memang tidak ada solusinya karena ada baris matriks augmented yang mengandung nilai 0 semua. Dengan menggunakan metode matriks balikan juga tidak bisa ditemukan solusinya karena matriks tidak memiliki balikan, begitu pula dengan metode Cramer, metode Cramer tidak dilakukan karena determinan matriks = 0.</td></tr>
<tr><td rowspan="4">b)</td><td rowspan="4">$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$</td><td>Gauss</td><td>Mengambil file dari folder test.
Nama file: 1_b
1.0 -1.0 0.0 0.0 1.0 3.0
1.0 1.0 0.0 -3.0 0.0 6.0
2.0 -1.0 0.0 1.0 -1.0 5.0
-1.0 2.0 0.0 -2.0 -1.0 -1.0

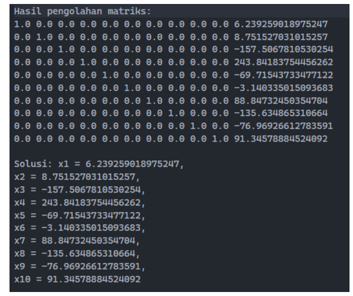
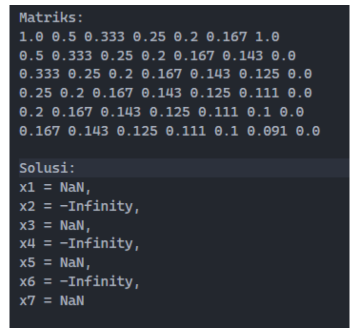
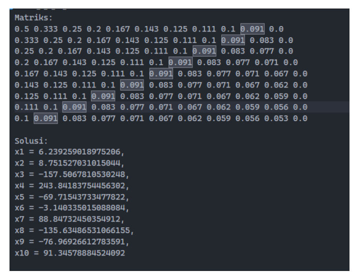
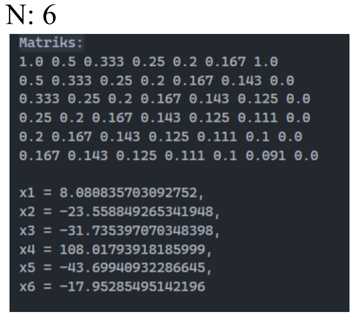
x1 = 3.0+1.00S,
x2 = +2.00S,
x3 = T,
x4 = -1.0+1.00S,
x5 = S,</td></tr>
<tr><td>Gauss-Jordan</td><td>Mengambil file dari folder test.
Nama file: 1_b
1.0 -1.0 0.0 0.0 1.0 3.0
1.0 1.0 0.0 -3.0 0.0 6.0
2.0 -1.0 0.0 1.0 -1.0 5.0
-1.0 2.0 0.0 -2.0 -1.0 -1.0

x1 = 3.0+1.00S,
x2 = +2.00S,
x3 = T,
x4 = -1.0+1.00S,
x5 = S,</td></tr>
<tr><td>Matriks Balikan</td><td>Mengambil file dari folder test.
Nama file: 1_b
1.0 -1.0 0.0 0.0 1.0 3.0
1.0 1.0 0.0 -3.0 0.0 6.0
2.0 -1.0 0.0 1.0 -1.0 5.0
-1.0 2.0 0.0 -2.0 -1.0 -1.0

Matriks memerlukan 5 persamaan untuk dapat diselesaikan</td></tr>
<tr><td>Kaidah Cramer</td><td>Mengambil file dari folder test.
Nama file: 1_b
1.0 -1.0 0.0 0.0 1.0 3.0
1.0 1.0 0.0 -3.0 0.0 6.0
2.0 -1.0 0.0 1.0 -1.0 5.0
-1.0 2.0 0.0 -2.0 -1.0 -1.0

Matriks memerlukan 5 persamaan untuk dapat diselesaikan.</td></tr>
<tr><td colspan="4">Analisis:
SPL dapat diselesaikan dengan menggunakan metode eliminasi Gauss dan Gauss-Jordan, akan tetapi berupa solusi parametrik (banyak), sedangkan untuk menggunakan matriks balikan dan kaidah Cramer, tidak dapat diselesaikan, karena di kaidah Cramer dan matriks balikan, kolom dan baris matriks harus sama, hal ini pula penyebab matriks mendapatkan solusi parametrik saat menggunakan metode Gauss dan Gauss-Jordan.</td></tr>
<tr><td>c)</td><td>$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$</td><td>Gauss</td><td>x1 = U,
x2 = 1.0-1.00S,
x3 = T,
x4 = -2.0-1.00S,
x5 = 1.0+1.00S,
x6 = S,</td></tr>
</table>

		Gauss-Jordan	x1 = U, x2 = 1.0-1.00S, x3 = T, x4 = -2.0-1.00S, x5 = 1.0+1.00S, x6 = S,
		Matriks Balikan	Mengambil file dari folder test. Nama file: 1_c 0.0 1.0 0.0 0.0 1.0 0.0 2.0 0.0 0.0 0.0 1.0 1.0 0.0 -1.0 0.0 1.0 0.0 0.0 0.0 1.0 1.0 Matriks memerlukan 6 persamaan untuk dapat diselesaikan
		Kaidah Cramer	Mengambil file dari folder test. Nama file: 1_c 0.0 1.0 0.0 0.0 1.0 0.0 2.0 0.0 0.0 0.0 1.0 1.0 0.0 -1.0 0.0 1.0 0.0 0.0 0.0 1.0 1.0 Matriks memerlukan 6 persamaan untuk dapat diselesaikan.
Analisis: Sama seperti soal sebelumnya, matriks bisa diselesaikan dengan menggunakan metode Gauss dan Gauss-Jordan dengan menghasilkan solusi parametrik. Jika menggunakan kaidah cramer dan matriks balikan, tidak bisa diselesaikan, hal ini dikarenakan jumlah baris dan kolom matriks berbeda			
d)		Gauss	N: 6 Hasil pengolahan matriks: 1.0 0.5 0.333 0.25 0.2 0.167 1.0 0.0 1.0 1.0060240963855418 0.9036144578313252 0.8072289156626504 0.7168674698795177 -6.024096385542167 0.0 0.0 1.0 1.6247659717560585 1.948455002736908 2.146502628827725 20.086291791660724 0.0 0.0 0.0 1.0 2.2698835475478267 3.0876152490148923 -46.606139793259004 0.0 0.0 0.0 0.0 1.0 0.9893726750648215 -61.46147345165293 0.0 0.0 0.0 0.0 0.0 1.0 -17.952854955151796 x1 = 8.080835703347605, x2 = -23.558849267122852, x3 = -31.735397068465872, x4 = 108.01793918179216, x5 = -43.699409319623655, x6 = -17.952854955151796 N: 10 Solusi: x1 = 6.239259018975206, x2 = 8.751527031015044, x3 = -157.5067810530248, x4 = 243.84183754456302, x5 = -69.71543733477822, x6 = -3.140335015088084, x7 = 88.84732450354912, x8 = -135.63486531066155, x9 = -76.96926612783591, x10 = 91.34578884524092
		Gauss-Jordan	N: 6 Hasil pengolahan matriks: 1.0 0.0 0.0 0.0 0.0 0.0 8.080835703347597 0.0 1.0 0.0 0.0 0.0 0.0 -23.558849267122838 0.0 0.0 1.0 0.0 0.0 0.0 -31.73539706846589 0.0 0.0 0.0 1.0 0.0 0.0 108.01793918179217 0.0 0.0 0.0 0.0 1.0 0.0 -43.699409319623655 0.0 0.0 0.0 0.0 0.0 1.0 -17.952854955151796 Solusi: x1 = 8.080835703347597, x2 = -23.558849267122838, x3 = -31.73539706846589, x4 = 108.01793918179217, x5 = -43.699409319623655, x6 = -17.952854955151796 N: 10

			 Hasil pengolahan matriks: 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 6.239259018975247 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 8.751527031015257 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -157.5067810530254 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 243.84183754456262 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 -69.71543733477122 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 -3.140335015093683 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 88.84732450354704 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 -135.634865310664 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 -76.96926612783591 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 91.34578884524092 Solusi: x1 = 6.239259018975247, x2 = 8.751527031015257, x3 = -157.5067810530254, x4 = 243.84183754456262, x5 = -69.71543733477122, x6 = -3.140335015093683, x7 = 88.84732450354704, x8 = -135.634865310664, x9 = -76.96926612783591, x10 = 91.34578884524092
		Matriks Balikan	N: 6  Matriks: 1.0 0.5 0.333 0.25 0.2 0.167 1.0 0.5 0.333 0.25 0.2 0.167 0.143 0.0 0.333 0.25 0.2 0.167 0.143 0.125 0.0 0.25 0.2 0.167 0.143 0.125 0.111 0.0 0.2 0.167 0.143 0.125 0.111 0.1 0.0 0.167 0.143 0.125 0.111 0.1 0.091 0.0 Solusi: x1 = NaN, x2 = -Infinity, x3 = NaN, x4 = -Infinity, x5 = NaN, x6 = -Infinity, x7 = NaN N: 10  Matriks: 0.5 0.333 0.25 0.2 0.167 0.143 0.125 0.111 0.1 0.091 0.0 0.333 0.25 0.2 0.167 0.143 0.125 0.111 0.1 0.091 0.083 0.0 0.25 0.2 0.167 0.143 0.125 0.111 0.1 0.091 0.083 0.077 0.0 0.2 0.167 0.143 0.125 0.111 0.1 0.091 0.083 0.077 0.071 0.0 0.167 0.143 0.125 0.111 0.1 0.091 0.083 0.077 0.071 0.067 0.0 0.143 0.125 0.111 0.1 0.091 0.083 0.077 0.071 0.067 0.062 0.0 0.125 0.111 0.1 0.091 0.083 0.077 0.071 0.067 0.062 0.059 0.0 0.111 0.1 0.091 0.083 0.077 0.071 0.067 0.062 0.059 0.056 0.0 0.1 0.091 0.083 0.077 0.071 0.067 0.062 0.059 0.056 0.053 0.0 Solusi: x1 = 6.239259018975206, x2 = 8.751527031015044, x3 = -157.5067810530248, x4 = 243.84183754456302, x5 = -69.71543733477822, x6 = -3.140335015088084, x7 = 88.84732450354912, x8 = -135.63486531066155, x9 = -76.96926612783591, x10 = 91.34578884524092
		Kaidah Cramer	N: 6  Matriks: 1.0 0.5 0.333 0.25 0.2 0.167 1.0 0.5 0.333 0.25 0.2 0.167 0.143 0.0 0.333 0.25 0.2 0.167 0.143 0.125 0.0 0.25 0.2 0.167 0.143 0.125 0.111 0.0 0.2 0.167 0.143 0.125 0.111 0.1 0.0 0.167 0.143 0.125 0.111 0.1 0.091 0.0 x1 = 8.080835703092752, x2 = -23.558849265341948, x3 = -31.735397070348398, x4 = 108.01793918185999, x5 = -43.69940932286645, x6 = -17.95285495142196 N: 10

			Matriks: 0.5 0.333 0.25 0.2 0.167 0.143 0.125 0.111 0.1 0.091 0.0 0.333 0.25 0.2 0.167 0.143 0.125 0.111 0.1 0.091 0.083 0.0 0.25 0.2 0.167 0.143 0.125 0.111 0.1 0.091 0.083 0.077 0.0 0.2 0.167 0.143 0.125 0.111 0.1 0.091 0.083 0.077 0.071 0.0 0.167 0.143 0.125 0.111 0.1 0.091 0.083 0.077 0.071 0.067 0.0 0.143 0.125 0.111 0.1 0.091 0.083 0.077 0.071 0.067 0.062 0.0 0.125 0.111 0.1 0.091 0.083 0.077 0.071 0.067 0.062 0.059 0.0 0.111 0.1 0.091 0.083 0.077 0.071 0.067 0.062 0.059 0.056 0.0 0.1 0.091 0.083 0.077 0.071 0.067 0.062 0.059 0.056 0.053 0.0 Solusi: x1 = 6.239259018975206, x2 = 8.751527031015044, x3 = -157.5067810530248, x4 = 243.84183754456302, x5 = -69.71543733477822, x6 = -3.140335015088084, x7 = 88.84732450354912, x8 = -135.63486531066155, x9 = -76.96926612783591, x10 = 91.34578884524092
Analisis: Untuk N = 6, SPL dapat diselesaikan dengan menggunakan metode Gauss, Gauss-Jordan, dan Kaidah Cramer dengan hasil yang sama berupa solusi unik, namun dengan metode matriks balikan, hasilnya bilangan NaN dan Infinity, penulis berasumsi hal ini dikarenakan adanya pembagian dengan 0 di matriks balikan. Dan untuk N = 10, semua solusi dapat ditemukan walaupun membutuhkan waktu lama, hal ini dikarenakan harus dilakukan rekursif 10.			

4.2. Sistem Persamaan Linear Berbentuk Matriks Augmented

Soal		Metode	Output
a)	$\begin{bmatrix} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{bmatrix}.$	Gauss	Mengambil file dari folder tes Nama file: 2_a 1.0 -1.0 2.0 -1.0 -1.0 2.0 1.0 -2.0 -2.0 -2.0 -1.0 2.0 -4.0 1.0 1.0 3.0 0.0 0.0 -3.0 -3.0 x1 = -1.0+1.00T, x2 = +2.00S, x3 = S, x4 = T,
		Gauss-Jordan	Nama file: 2_a 1.0 -1.0 2.0 -1.0 -1.0 2.0 1.0 -2.0 -2.0 -2.0 -1.0 2.0 -4.0 1.0 1.0 3.0 0.0 0.0 -3.0 -3.0 x1 = -1.0+1.00T, x2 = +2.00S, x3 = S, x4 = T,

<table>
<tr><td rowspan="2"></td><td rowspan="2"></td><td>Matriks Balikan</td><td>Mengambil file dari folder test.
Nama file: 2_a
1.0 -1.0 2.0 -1.0 -1.0
2.0 1.0 -2.0 -2.0 -2.0
-1.0 2.0 -4.0 1.0 1.0
3.0 0.0 0.0 -3.0 -3.0

Matriks tidak memiliki inverse.</td></tr>
<tr><td>Kaidah Creamer</td><td>Mengambil file dari folder test.
Nama file: 2_a
1.0 -1.0 2.0 -1.0 -1.0
2.0 1.0 -2.0 -2.0 -2.0
-1.0 2.0 -4.0 1.0 1.0
3.0 0.0 0.0 -3.0 -3.0

Tidak dapat menggunakan kaidah Cramer!</td></tr>
<tr><td colspan="4">Analisis:
Solusi dari matriks ini dapat didapatkan dengan metode eliminasi Gauss dan Gauss-Jordan berupa solusi parametrik, dengan menggunakan matriks balikan dan kaidah Cramer tidak dapat diselesaikan karena baris dan kolom matriks berbeda</td></tr>
<tr><td rowspan="4">b)</td><td rowspan="4">$\begin{bmatrix} 2 & 0 & 8 & 0 & 8 \\ 0 & 1 & 0 & 4 & 6 \\ -4 & 0 & 6 & 0 & 6 \\ 0 & -2 & 0 & 3 & -1 \\ 2 & 0 & -4 & 0 & -4 \\ 0 & 1 & 0 & -2 & 0 \end{bmatrix}.$</td><td>Gauss</td><td>Nama file: 2_b
2.0 0.0 8.0 0.0 8.0
0.0 1.0 0.0 4.0 6.0
-4.0 0.0 6.0 0.0 6.0
0.0 -2.0 0.0 3.0 -1.0
2.0 0.0 -4.0 0.0 -4.0
0.0 1.0 0.0 -2.0 0.0

x1 = 0.0,
x2 = 2.0,
x3 = 1.0,
x4 = 1.0,</td></tr>
<tr><td>Gauss-Jordan</td><td>Mengambil file dari folde
Nama file: 2_b
2.0 0.0 8.0 0.0 8.0
0.0 1.0 0.0 4.0 6.0
-4.0 0.0 6.0 0.0 6.0
0.0 -2.0 0.0 3.0 -1.0
2.0 0.0 -4.0 0.0 -4.0
0.0 1.0 0.0 -2.0 0.0

x1 = 0.0,
x2 = 2.0,
x3 = 1.0,
x4 = 1.0,</td></tr>
<tr><td>Matriks Balikan</td><td>Mengambil file dari folder test.
Nama file: 2_b
2.0 0.0 8.0 0.0 8.0
0.0 1.0 0.0 4.0 6.0
-4.0 0.0 6.0 0.0 6.0
0.0 -2.0 0.0 3.0 -1.0
2.0 0.0 -4.0 0.0 -4.0
0.0 1.0 0.0 -2.0 0.0

Matriks memerlukan 4 persamaan untuk dapat diselesaikan</td></tr>
<tr><td>Kaidah Cramer</td><td>Mengambil file dari folder test.
Nama file: 2_b
2.0 0.0 8.0 0.0 8.0
0.0 1.0 0.0 4.0 6.0
-4.0 0.0 6.0 0.0 6.0
0.0 -2.0 0.0 3.0 -1.0
2.0 0.0 -4.0 0.0 -4.0
0.0 1.0 0.0 -2.0 0.0

Matriks memerlukan 4 persamaan untuk dapat diselesaikan.</td></tr>
</table>

Analisis: Dengan menggunakan Gauss dan Gauss-Jordan, bisa didapatkan solusi uniknya, sedangkan dengan menggunakan Kaidah Cramer dan juga matriks balikan, tidak bisa diselesaikan, hal ini dikarenakan besar baris matriks tidak sama dengan kolom.

4.3. SPL Terbentuk

<table>
<tr><td rowspan="4">a)</td><td rowspan="4">$$\begin{aligned} 8x_1 + x_2 + 3x_3 + 2x_4 &= 0 \\ 2x_1 + 9x_2 - x_3 - 2x_4 &= 1 \\ x_1 + 3x_2 + 2x_3 - x_4 &= 2 \\ x_1 + 6x_3 + 4x_4 &= 3 \end{aligned}$$</td><td>Gauss</td><td>Mengambil file dari folder test.
Nama file: 3_a
8.0 1.0 3.0 2.0 0.0
2.0 9.0 -1.0 -2.0 1.0
1.0 3.0 2.0 -1.0 2.0
1.0 0.0 6.0 4.0 3.0

x1 = -0.2243243243243243,
x2 = 0.18243243243243246,
x3 = 0.7094594594594594,
x4 = -0.258108108108108,</td></tr>
<tr><td>Gauss-Jordan</td><td>Mengambil file dari folder test.
Nama file: 3_a
8.0 1.0 3.0 2.0 0.0
2.0 9.0 -1.0 -2.0 1.0
1.0 3.0 2.0 -1.0 2.0
1.0 0.0 6.0 4.0 3.0

x1 = -0.22432432432432436,
x2 = 0.18243243243243246,
x3 = 0.7094594594594594,
x4 = -0.258108108108108,</td></tr>
<tr><td>Matriks Balikan</td><td>Mengambil file dari folder
Nama file: 3_a
8.0 1.0 3.0 2.0 0.0
2.0 9.0 -1.0 -2.0 1.0
1.0 3.0 2.0 -1.0 2.0
1.0 0.0 6.0 4.0 3.0

Solusi:
x1 = 0.0,
x2 = 0.0,
x3 = 0.0,
x4 = 0.0,</td></tr>
<tr><td>Kaidah Cramer</td><td>Mengambil file dari folder test.
Nama file: 3_a
8.0 1.0 3.0 2.0 0.0
2.0 9.0 -1.0 -2.0 1.0
1.0 3.0 2.0 -1.0 2.0
1.0 0.0 6.0 4.0 3.0

Solusi:
x1 = -0.22432432432432434,
x2 = 0.18243243243243243,
x3 = 0.7094594594594594,
x4 = -0.2581081081081081,</td></tr>
<tr><td colspan="4">Analisis:
Solusi unik SPL dapat didapatkan dengan menggunakan metode Gauss, Gauss-Jordan, dan Kaidah Cramer. Namun dengan menggunakan metode matriks balikan, solusi yang didapatkan adalah 0, hal ini</td></tr>
</table>

<table>
<tr><td colspan="4">bisa disebabkan oleh terjadinya pembagian dengan 0 di proses matriks balikan dan juga bisa karena determinan dari matriks tersebut yang mendekati 0.</td></tr>
<tr><td rowspan="4">b)</td><td rowspan="4">$$\begin{aligned} x_7 + x_8 + x_9 &= 13.00 \\ x_4 + x_5 + x_6 &= 15.00 \\ x_1 + x_2 + x_3 &= 8.00 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_6 + x_8) + 0.61396x_9 &= 14.79 \\ 0.91421(x_3 + x_5 + x_7) + 0.25(x_2 + x_4 + x_6 + x_8) &= 14.31 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_2 + x_4) + 0.61396x_1 &= 3.81 \\ x_3 + x_6 + x_9 &= 18.00 \\ x_2 + x_5 + x_8 &= 12.00 \\ x_1 + x_4 + x_7 &= 6.00 \\ 0.04289(x_1 + x_5 + x_9) + 0.75(x_2 + x_6) + 0.61396x_3 &= 10.51 \\ 0.91421(x_1 + x_5 + x_9) + 0.25(x_2 + x_4 + x_6 + x_8) &= 16.13 \\ 0.04289(x_1 + x_5 + x_9) + 0.75(x_4 + x_8) + 0.61396x_7 &= 7.04 \end{aligned}$$</td><td>Gauss</td><td>Mengambil file dari folder test.
Nama file: 3_b
Invalid matrix format: Rows have different column counts.

x1 = 9.0-2.00S-3.00T-4.00U-5.00V-6.00W-7.00X-8.00Y,
x2 = S,
x3 = T,
x4 = U,
x5 = V,
x6 = W,
x7 = X,
x8 = Y,</td></tr>
<tr><td>Gauss-Jordan</td><td>Mengambil file dari folder test.
Nama file: 3_b
Invalid matrix format: Rows have different column counts.

x1 = 9.0-2.00S-3.00T-4.00U-5.00V-6.00W-7.00X-8.00Y,
x2 = S,
x3 = T,
x4 = U,
x5 = V,
x6 = W,
x7 = X,
x8 = Y,</td></tr>
<tr><td>Matriks Balikan</td><td>Mengambil file dari folder test.
Nama file: 3_b
Invalid matrix format: Rows have different column counts.

Matriks memerlukan 8 persamaan untuk dapat diselesaikan</td></tr>
<tr><td>Kaidah Cramer</td><td>Mengambil file dari folder test.
Nama file: 3_b
Invalid matrix format: Rows have different column counts.

Matriks memerlukan 8 persamaan untuk dapat diselesaikan.</td></tr>
<tr><td colspan="4">Analisis:
Solusi SPL berupa solusi parametrik dihasilkan dengan menggunakan metode Gauss dan Gauss-Jordan, untuk menggunakan kaidah Cramer dan matriks balikan, matriks tidak bisa diselesaikan, hal ini dikarenakan jumlah persamaan dan jumlah kolom tidak sama.</td></tr>
</table>

4.4 Sistem Reaktor

<table>
<tr><td rowspan="2"></td><td rowspan="2">$$\begin{aligned} &\text{A:} \quad m_{A_{in}} + Q_{BA}x_B - Q_{AB}x_A - Q_{AC}x_A = 0 \\ &\text{B:} \quad Q_{AB}x_A - Q_{BA}x_B - Q_{BC}x_B = 0 \\ &\text{C:} \quad m_{C_{in}} + Q_{AC}x_A + Q_{BC}x_B - Q_{C_{out}}x_C = 0 \end{aligned}$$
Tentukan solusi x_A, x_B, x_C dengan menggunakan parameter berikut : $Q_{AB} = 40$, $Q_{AC} = 80$, $Q_{BA} = 60$, $Q_{BC} = 20$ dan $Q_{Cout} = 150$ m^3/s dan $m_{Ain} = 1300$ dan $m_{Cin} = 200$ mg/s.</td><td>Gauss</td><td>Mengambil file dari folder test.
Nama file: 4
1300.0 60.0 -40.0 -80.0 0.0
0.0 40.0 -60.0 -20.0 0.0
200.0 80.0 20.0 -150.0 0.0

x1 = x2 = x3 = x4 = 0atau
x1 = +0.01S,
x2 = +1.66S,
x3 = +0.77S,
x4 = S,</td></tr>
<tr><td>Gauss-Jordan</td><td>Mengambil file dari folder test.
Nama file: 4
1300.0 60.0 -40.0 -80.0 0.0
0.0 40.0 -60.0 -20.0 0.0
200.0 80.0 20.0 -150.0 0.0

x1 = x2 = x3 = x4 = 0atau
x1 = +0.01S,
x2 = +1.66S,
x3 = +0.77S,
x4 = S,
Simpan dalam bentuk file?</td></tr>
</table>

		Matriks Balikan	Mengambil file dari folder test. Nama file: 4 1300.0 60.0 -40.0 -80.0 0.0 0.0 40.0 -60.0 -20.0 0.0 200.0 80.0 20.0 -150.0 0.0 Matriks memerlukan 4 persamaan untuk dapat diselesaikan
		Kaidah Cramer	Mengambil file dari folder test. Nama file: 4 1300.0 60.0 -40.0 -80.0 0.0 0.0 40.0 -60.0 -20.0 0.0 200.0 80.0 20.0 -150.0 0.0 Matriks memerlukan 4 persamaan untuk dapat diselesaikan.
Analisis: Solusi SPL berupa solusi parametrik dapat didapatkan dari persamaan dengan menggunakan metode eliminasi Gauss dan eliminasi Gauss-Jordan, sementara untuk matriks balikan dan kaidah cramer tidak dapat dilakukan karena kedua metode tersebut memiliki syarat khusus yaitu matriks harus persegi.			

4.5 Studi Kasus Interpolasi Polinom

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

x	0.1	0.3	0.5	0.7	0.9	1.1	1.3
f(x)	0.003	0.067	0.148	0.248	0.370	0.518	0.697

x = 0.2

```
Mengambil file dari folder test.
Nama file: 5_a_1

Hasil Perhitungan Interpolasi Polinom
Penjabaran f(x):
f(x) = - 3.360838505161542E-14x^6 + 1.395966765938392E-13x^5 + 0.026041666666442493x^4 + 1.748103953800361E-13x^3 + 0.19739583333326544x^2 + 0.24000000000001184x - 0.02297656250000066
Hasil substitusi dengan nilai x dari masukan:
f(0.2) = 0.032960937500000086
```

x = 0.55

```
Mengambil file dari folder test.
Nama file: 5_a_2

Hasil Perhitungan Interpolasi Polinom
Penjabaran f(x):
f(x) = - 3.360838505161542E-14x^6 + 1.395966765938392E-13x^5 + 0.026041666666442493x^4 + 1.748103953800361E-13x^3 + 0.19739583333326544x^2 + 0.24000000000001184x - 0.02297656250000066
Hasil substitusi dengan nilai x dari masukan:
f(0.55) = 0.17111865234374998
```

x = 0.85

```
Mengambil file dari folder test.
Nama file: 5_a_3

Hasil Perhitungan Interpolasi Polinom
Penjabaran f(x):
f(x) = - 3.360838505161542E-14x^6 + 1.395966765938392E-13x^5 + 0.026041666666442493x^4 + 1.748103953800361E-13x^3 + 0.19739583333326544x^2 + 0.24000000000001184x - 0.02297656250000066
Hasil substitusi dengan nilai x dari masukan:
f(0.85) = 0.33723583984374994
```

x = 1.28

```
Mengambil file dari folder test.
Nama file: 5_a_4

Hasil Perhitungan Interpolasi Polinom
Penjabaran f(x):
f(x) = - 3.360838505161542E-14x^6 + 1.395966765938392E-13x^5 + 0.026041666666442493x^4 + 1.748103953800361E-13x^3 + 0.19739583333326544x^2 + 0.24000000000001184x - 0.02297656250000066
Hasil substitusi dengan nilai x dari masukan:
f(1.28) = 0.6775418375
```

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

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

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

a. 16/07/2022

```
Hasil Perhitungan Interpolasi Polinom
Penjabaran f(x):
f(x) = - 148.08779815815305x^9 + 1084189.3635916898x^8 - 6.825273380019188E7x^7 + 1.883074614890625E9x^6 - 2.9666427456E10x^5 + 2.91748970496E11x^4 - 1.833661079145362E12x^3 + 7.192077367914573E12x^2 - 1.609430786595506E13x + 1.5731538
028763451E13
Hasil substitusi dengan nilai x dari masukan:
f(7.516) = 47220.419090270996
```

b. 10/08/2022

```
Mengambil file dari folder test.
Nama file: 5_b_b

Hasil Perhitungan Interpolasi Polinom
Penjabaran f(x):
f(x) = - 148.08779815815305x^9 + 1084189.3635916898x^8 - 6.825273380019188E7x^7 + 1.883074614890625E9x^6 - 2.9666427456E10x^5 + 2.91748970496E11x^4 - 1.833661079145362E12x^3 + 7.192077367914573E12x^2 - 1.609430786595506E13x + 1.5731538
028763451E13
Hasil substitusi dengan nilai x dari masukan:
f(8.323) = 112356.52448654175
```

c. 05/09/2022

```
Mengambil file dari folder test.
Nama file: 5_b_c

Hasil Perhitungan Interpolasi Polinom
Penjabaran f(x):
f(x) = - 148.08779815815305x^9 + 1084189.3635916898x^8 - 6.825273380019188E7x^7 + 1.883074614890625E9x^6 - 2.9666427456E10x^5 + 2.91748970496E11x^4 - 1.833661079145362E12x^3 + 7.192077367914573E12x^2 - 1.609430786595506E13x + 1.5731538
028763451E13
Hasil substitusi dengan nilai x dari masukan:
f(5.3) = 1.452827583537745E9
```

d. Masukan user lainnya berupa **tanggal (desimal) yang sudah diolah** dengan asumsi prediksi selalu dilakukan untuk tahun 2022.

Data pengguna yang dipilih adalah 10 Oktober 2022 dengan angka desimal 10.3:

```
Hasil Perhitungan Interpolasi Polinom
Penjabaran f(x):
f(x) = - 148.08779815815305x^9 + 1084189.3635916898x^8 - 6.825273380019188E7x^7 + 1.883074614890625E9x^6 - 2.9666427456E10x^5 + 2.91748970496E11x^4 - 1.833661079145362E12x^3 + 7.19207736
7914573E12x^2 - 1.609430786595506E13x + 1.5731538028763451E13
Hasil substitusi dengan nilai x dari masukan:
f(10.3) = 7.051195597677307E8
```

c. Sederhanakan fungsi $f(x)$ yang memenuhi kondisi

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

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

n = 4

```
Hasil Perhitungan Interpolasi Polinom
Penjabaran f(x):
f(x) = - 0.19933333333333342x^4 + 1.000666666666667x^3 - 1.856166666666667x^2 + 1.5928333333333333x
Hasil substitusi dengan nilai x dari masukan:
f(1.0) = 0.538
```

n = 5

```
Nama file: 5_c_5

Hasil Perhitungan Interpolasi Polinom
Penjabaran f(x):
f(x) = 0.23356119791666677x^5 - 1.4062500000000013x^4 + 3.2063802083333375x^3 - 3.5250000000000026x^2 + 2.0260000000000002x
Hasil substitusi dengan nilai x dari masukan:
f(5.0) = 173.77501953125
```

n = 6

```
Mengambil file dari folder test.
Nama file: 5_c_6

Hasil Perhitungan Interpolasi Polinom
Penjabaran f(x):
f(x) = 0
Hasil substitusi dengan nilai x dari masukan:
f(0.0) = 0.0
```

4.6 Studi Kasus Regresi Linear Berganda

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.

Penyelesaian Regresi Linear Berganda untuk di atas menggunakan *Normal Estimation Equation for Multiple Linear Regression* untuk mendapatkan regresi linear berganda dari data pada tabel di atas, diperoleh sistem persamaan linear sebagai berikut.

$$
\begin{array}{lllll}
20b_0 & + 863.1b_1 & + 1530.4b_2 & + 587.84b_3 & = 19.42 \\
863.1b_0 & + 54876.89b_1 & + 67000.09b_2 & + 25283.395b_3 & = 779.477 \\
1530.4b_0 & + 67000.09b_1 & + 117912.32b_2 & + 44976.867b_3 & = 1483.437 \\
587.84b_0 & + 25283.395b_1 & + 44976.867b_2 & + 17278.5086b_3 & = 571.1219
\end{array}
$$

(jika ingin melihat representasi persamaan di atas dapat menambahkan writeMatrix mfuc)

Kemudian diestimasi nilai Nitrous Oxide(f(x)) dengan Humidity bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30 didapatkan hasil sebagai berikut :

```
f(x) = -3.5077781409 - 0.0026249907x1 + 0.0007989410x2 + 0.1541550302x3
Hampiran (taksiran) nilai f(x): 0.9384342262216672
```

4.7. Studi Kasus Interpolasi Bicubic Spline

Diberikan matriks input dengan bentuk sebagai berikut. Format matriks masukan bukan mewakili nilai matriks, tetapi mengikuti format masukan pada bagian "Spesifikasi Tugas" nomor 7.

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

f(0, 0) :

```
Mengambil file dari folder test.
Nama file: 7_a

Hasil Bicubic Interpolation
Nilai x dan y dari masukan disubstitusikan, dan hasilnya adalah
f(0.0,0.0) = 21.0
```

f(0.5, 0.5) :

```
Mengambil file dari folder test.
Nama file: 7_b

Hasil Bicubic Interpolation
Nilai x dan y dari masukan disubstitusikan, dan hasilnya adalah
f(0.5,0.5) = 87.796875
```

f(0.25, 0.75) :

```
Mengambil file dari folder test.
Nama file: 7_c

Hasil Bicubic Interpolation
Nilai x dan y dari masukan disubstitusikan, dan hasilnya adalah
f(0.25,0.75) = 117.732177734375
```

f(0.1, 0.9) :

```
Mengambil file dari folder test.
Nama file: 7_d

Hasil Bicubic Interpolation
Nilai x dan y dari masukan disubstitusikan, dan hasilnya adalah
f(0.1,0.9) = 128.57518700000003
```

BAB 5 KESIMPULAN

5. 1. Kesimpulan

Dalam proyek Tugas Besar 1 IF2123 Aljabar Linier dan Geometri ini, kami telah mengimplementasikan objek matriks beserta sejumlah fungsi pengolahan matriks yang meliputi metode eliminasi Gauss, eliminasi Gauss-Jordan, penentuan matriks balikan, perhitungan determinan, kaidah Cramer, dan berbagai fungsi pendukung lainnya. Implementasi ini berhasil diintegrasikan ke dalam sebuah aplikasi berbasis Java yang mampu menyelesaikan berbagai permasalahan yang direpresentasikan dalam bentuk Sistem Persamaan Linier (SPL). Hasil penyelesaian SPL dalam proyek ini dapat berupa tiga jenis solusi: solusi unik, solusi banyak, atau bahkan ketiadaan solusi. Tingkat efisiensi dalam penyelesaian SPL sangat tergantung pada jenis matriks yang diinputkan. Misalnya, metode kaidah Cramer dan metode matriks balikan hanya dapat diterapkan pada matriks persegi, sementara eliminasi Gauss dan eliminasi Gauss-Jordan dapat digunakan tanpa pembatasan jenis matriks dan mungkin menghasilkan solusi banyak. Pengujian dengan matriks Hilbert mengungkapkan bahwa metode eliminasi Gauss dan eliminasi Gauss-Jordan jauh lebih cepat dan efisien dibandingkan dengan kaidah Cramer. Namun, metode Cramer juga memiliki keterbatasan saat determinan mendekati nol. Selanjutnya, metode penyelesaian SPL ini digunakan sebagai dasar untuk menangani berbagai permasalahan seperti interpolasi polinom, interpolasi bicubic, dan regresi linier berganda.

Semua solusi ini disatukan dalam program utama berbasis Java yang mencakup beragam fitur, antara lain:

1. Penyelesaian Sistem Persamaan Linier (SPL) menggunakan metode eliminasi Gauss, eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer.
2. Penyelesaian determinan suatu matriks dengan metode OBE (Operasi Baris Elementer), metode kofaktor baris, dan metode kofaktor kolom.
3. Penyelesaian matriks balikan suatu matriks dengan metode identitas dan metode matriks adjoin.
4. Penyelesaian interpolasi polinom.
5. Penyelesaian interpolasi bicubic.
6. Penyelesaian regresi linier berganda

5.2 Saran

Pelaksanaan Tugas Besar 1 IF2123 Aljabar Linier dan Geometri di Semester I Tahun 2022/2023 merupakan pengalaman yang sangat berharga bagi saya. Dari pengalaman ini, saya ingin berbagi beberapa saran kepada pembaca yang mungkin akan menghadapi tugas serupa di masa depan:

1. Pemahaman Bahasa Pemrograman: Tugas ini mengharuskan penulisan program dalam bahasa Java, yang mungkin belum familiar bagi sebagian mahasiswa. Saya sangat menyarankan untuk meluangkan waktu yang cukup guna mempelajari bahasa ini dengan baik, terutama jika Anda belum memiliki pengalaman sebelumnya. Karena tugas ini melibatkan banyak fitur yang perlu dikuasai dengan rinci.

2. Kerja Sama Tim: Efektivitas dalam kerja sama tim memiliki peran penting dalam menyelesaikan tugas ini. Kolaborasi secara real-time melalui alat seperti VSCode LiveShare untuk pembuatan program dan kolaborasi langsung pada MS Word untuk pembuatan laporan bisa sangat membantu. Selain itu, dalam pengembangan source code, konflik antara anggota tim dapat terjadi. Oleh karena itu, menggunakan alat pengelola versi seperti Github sangat direkomendasikan agar mempermudah manajemen proyek secara asinkron.

3. Pemahaman Struktur Data: Sebagai tugas berbasis pemrograman berorientasi objek, penggunaan Abstract Data Type (ADT) dan struktur data yang tepat sangat penting. Di dalam tugas ini, penggunaan struktur data Matriks sangat dianjurkan karena dapat mempermudah dan merapikan kode program serta memungkinkan penggunaan fungsi secara berulang.

Semoga saran-saran ini membantu pembaca dalam menyiapkan diri untuk menangani tugas serupa di masa depan.

5.2 Refleksi

Tugas Besar 1 IF2123 Aljabar Linier dan Geometri di Semester I Tahun 2023/2024 adalah salah satu tantangan awal yang kami hadapi pada semester ini. Proses penyelesaian tugas ini tidak datang tanpa hambatan. Melalui tugas ini, kami semakin menghargai penggunaan matriks dan penerapannya dalam berbagai aspek kehidupan, terutama dalam konteks matematika dan teknologi. Sifat matriks dan atribut-atributnya memiliki peran penting dalam menyelesaikan masalah, khususnya dalam konteks sistem persamaan linier.

Keberagaman aplikasi matriks juga dapat ditemukan dalam berbagai bidang ilmu, termasuk dalam Informatika. Sebagai contoh, matriks sering digunakan dalam pengolahan citra atau dalam bidang kriptografi. Pada tugas ini, kami diberikan kesempatan untuk mengimplementasikan pengolahan citra menggunakan matriks, dan ini memberikan kami apresiasi yang mendalam terhadap betapa kuatnya matriks dalam menciptakan citra baru dari data numerik. Dalam perjalanan kami mengerjakan tugas ini, kami juga merasakan kegembiraan dalam mempelajari bahasa pemrograman baru, yaitu Java, dan menggunakannya dengan efektif dalam waktu singkat, yaitu kurang dari tiga minggu.

Kami menghadapi tantangan dalam memahami dan menguasai konsep matriks selama beberapa minggu ini, tetapi berkat kerja sama tim yang erat dan kontribusi aktif setiap anggota kelompok, kami berhasil mengatasi rintangan tersebut. Setiap anggota kelompok saling mendukung dan menyelesaikan tugas mereka dengan baik, menciptakan sinergi yang efektif dalam menyelesaikan Tugas Besar ini.

DAFTAR PUSTAKA

Rinaldi Munir. "Aljabar dan Geometri untuk Informatika 2023/2024."
informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/algeo23-24.html
Rowe, John B. "Bicubic Interpolation of Digital Images."
https://www.mssc.mu.edu/~daniel/pubs/RoweTalkMSCS_BiCubic.pdf