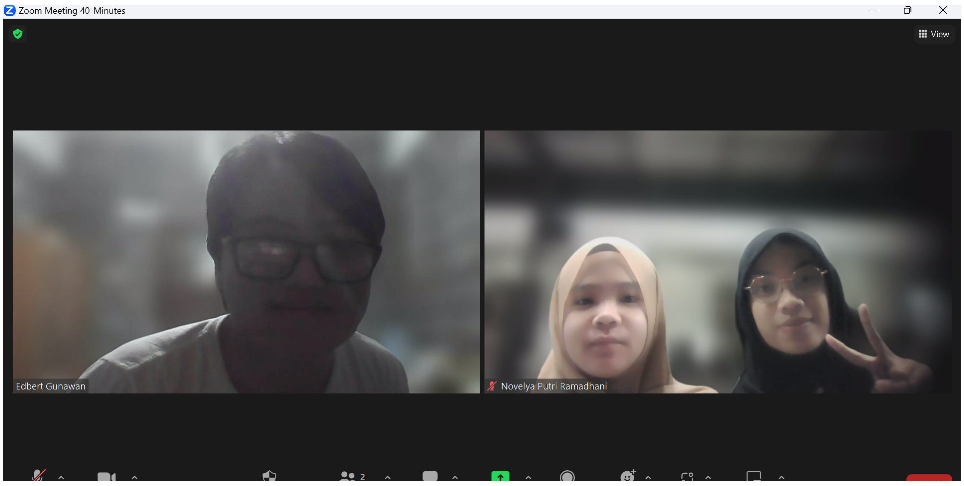
IF2123 Aljabar Linear dan Geometri

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

Laporan Tugas Besar 1

Disusun untuk memenuhi tugas mata kuliah Aljabar Linear dan Geometri
pada Semester 1 (satu) Tahun Akademik 2022/2023.



Oleh

Edbert Eddyson Gunawan (13522039)
Novelya Putri Ramadhani (13522096)
Diana Tri Handayani (13522104)

> Kelompok AsLabPro <

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

Daftar Isi

Daftar Gambar

sad

Daftar Tabel

Sad

BAB I

Deskripsi Masalah

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Anda sudah mempelajari berbagai metode untuk menyelesaikan SPL, termasuk menghitung determinan matriks. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan (x = A-1b), 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).

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

Gambar 1. Eliminasi Gauss dilakukan dengan matriks eselon baris dan eliminasi Gauss-Jordan dengan matriks eselon baris tereduksi.

Di dalam Tugas Besar 1 ini, Anda 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, gunakan library tersebut di dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi. Penjelasan tentang interpolasi dan regresi adalah seperti di bawah ini.

1. Buatlah pustaka (library atau package) 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.

2. 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:

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 aij. 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) dari determinan dan matriks balikan disesuaikan dengan persoalan masing-masing.
3. Untuk persoalan interpolasi, masukannya jika dari keyboard adalah n, (x0, y0), (x1, y1), …, (xn, yn), 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
4. Untuk persoalan regresi, masukannya jika dari keyboard adalah n (jumlah peubah x), m (jumlah sampel), semua nilai-nilai x1i, x2i, …, xni, nilai yi, dan nilai-nilai xk yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.
5. 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 x4 = -2, x3 = 2s – t, x2 = s, dan x1 = t).
6. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan. Contoh luaran untuk interpolasi adalah

   $f(x) = -0.0064x^2 + 0.2266x + 0.6762, f(5) = ...$

dan untuk regresi adalah

$f(x) = -9.5872 + 1.0732x_1, f(xk) = ...$

7. 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), fx(0, 0), fx(1, 0), fx(0, 1), fx(1, 1), fy(0, 0), fy(1, 0), fy(0, 1), fy(1, 1), fxy(0, 0), fxy(1, 0), fxy(0, 1), fxy(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 berturutturut 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).
8. Luaran program harus dapat ditampilkan pada layar komputer dan dapat disimpan ke dalam file.
9. Bahasa program yang digunakan adalah Java. Anda bebas untuk menggunakan versi java apapun dengan catatan di atas java versi 8 (8/9/11/15/17/19/20).
10. Program tidak harus berbasis GUI, cukup text-based saja, namun boleh menggunakan GUI (memakai kakas Eclipse misalnya).
11. Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan dirancang masing-masing. Misalnya, 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 II

Teori Singkat

Dalam implementasi pustaka Tugas Besar 1 ini, terdapat 10 hal yang akan dibahas yakni: metode eliminasi Gauss, metode eliminasi Gauss-Jordan, determinan, matriks balikan, matriks kofaktor, matriks adjoin, kaidah Cramer, interpolasi polinom, interpolasi bicubic spline, regresi linier berganda.

2.1 Metode Eliminasi Gauss

Eliminasi Gauss adalah sebuah metode yang digunakan dalam aljabar linear untuk menyelesaikan sistem persamaan linear. Tujuan utama dari eliminasi Gauss adalah mengubah matriks koefisien dalam sistem persamaan linear menjadi bentuk matriks segitiga atas (upper triangular matrix) atau bentuk matriks segitiga bawah (lower triangular matrix) agar mudah untuk menemukan solusi sistem persamaan linear tersebut.

2.2 Metode Eliminasi Gauss-Jordan

Eliminasi Gauss-Jordan adalah variasi dari metode eliminasi Gauss yang digunakan untuk menyelesaikan sistem persamaan linear dan menghitung invers matriks. Tujuannya adalah untuk mentransformasi matriks augmented dari sistem persamaan linear menjadi bentuk matriks identitas atau matriks balikan (inverse matrix) dari matriks koefisien. Eliminasi Gauss-Jordan sangat berguna dalam pemecahan sistem persamaan linear yang melibatkan matriks persegi.

2.3 Determinan

Determinan adalah suatu konsep dalam aljabar linear yang digunakan untuk mengukur sifat-sifat matriks, terutama dalam konteks matriks persegi (matriks dengan jumlah baris dan kolom yang sama). Determinan diberikan oleh sebuah bilangan nyata yang menggambarkan "ukuran" atau "volume" dari matriks tersebut. Untuk matriks 2x2, determinan didefinisikan sebagai perbedaan dari hasil perkalian diagonal utama dengan diagonal kedua. Sedangkan untuk ukuran yang lebih besar terdapat dua cara untuk mengetahui determinannya, yaitu dengan reduksi baris dan ekspansi kofaktor.

*Anton, H., & Rorres, C. (2010). Elementary Linear Algebra with Applications. John Wiley & Sons.

2.4 Matriks balikan

Matriks balikan dikenal juga sebagai matriks invers. Matrik balikan adalah matriks yang ketika dikalikan dengan matriks asal, menghasilkan matriks identitas. Dalam simbol matematika, jika A adalah matriks asal, maka A^(-1) adalah matriks balikan dari A. Matriks balikan hanya ada untuk matriks persegi yang non-singular, artinya determinan matriks tersebut tidak sama dengan nol. Jika sebuah matriks A tidak memiliki invers, maka matriks tersebut disebut sebagai matriks singular.

2.5 Matriks Kofaktor

Matriks kofaktor adalah suatu matriks yang dihasilkan dari matriks asal dengan menggantikan setiap elemen dengan kofaktor yang sesuai. Kofaktor suatu elemen matriks adalah produk dari minor aljabar dari elemen tersebut dengan faktor penentu (determinan) matriks yang dihasilkan dari matriks asal dengan menghapus baris dan kolom yang mengandung elemen tersebut. Matriks kofaktor sering digunakan dalam berbagai aspek aljabar linear dan kalkulus untuk melakukan berbagai operasi, seperti menghitung invers matriks, menghitung determinan, dan menentukan solusi sistem persamaan linear.

2.6 Matriks Adjoin

Matriks adjoin adalah matriks yang dihasilkan dari matriks asal dengan menukar baris dan kolom, dan kemudian menghitung determinan dari matriks tersebut. Matriks adjoin digunakan dalam berbagai aplikasi matematika, terutama dalam menghitung matriks invers dan solusi sistem persamaan linear.

2.7 Kaidah Cramer

Kaidah Cramer adalah metode untuk menemukan solusi sistem persamaan linear berdasarkan determinan. Metode ini hanya dapat diterapkan pada sistem persamaan linear yang memiliki jumlah persamaan yang sama dengan jumlah variabel, dan matriks koefisien dari sistem tersebut harus non-singular (tidak memiliki determinan nol). Dalam metode ini, solusi untuk masing-masing variabel ditemukan dengan menggunakan perbandingan dari determinan matriks yang berbeda.

2.8 Interpolasi Polinom

Interpolasi polinom adalah metode matematis yang digunakan untuk membangun suatu polinom yang melewati sejumlah titik data yang diketahui. Tujuan dari interpolasi polinom adalah untuk memperkirakan nilai-nilai di antara titik-titik data yang diberikan. Pola dasar interpolasi polinom adalah mengambil sejumlah titik data yang diketahui (x1, y1), (x2, y2), ..., (xn, yn), dan mencari polinom P(x) dengan derajat n-1.

2.9 Interpolasi Bicubic Spline

Interpolasi bicubic spline adalah salah satu metode interpolasi spline yang digunakan untuk memperkirakan data yang terletak di antara titik-titik data yang diketahui dengan menggunakan polinom kubik. Ini adalah metode yang umum digunakan dalam berbagai bidang, seperti ilmu komputer, ilmu data, pemrosesan citra, dan grafika komputer. Dalam interpolasi bicubic spline, setiap potongan antara dua titik data disusun sebagai fungsi polinom kubik, dan potongan-potongan ini disambungkan dengan baik sehingga interpolasi halus di antara titik-titik data.

2.10 Regresi Linear Berganda

Regresi linear berganda merupakan model regresi yang melibatkan lebih dari satu variabel independen. Model regresi ini digunakan untuk menganalisis respons suatu variabel dependant y yang mungkin bergantung pada sejumlah n variabel regressor. Bentuk umum dari model regresi linier berganda dengan sejumlah n variabel regressor yaitu:

$$y = b_0 + b_1x_1 + b_2x_2 + \ ... \ + b_nx_n + \varepsilon$$

Sebagai keterangan, $b_0, b_1, \dots , b_n$ merupakan koefisien regresi dan ϵ adalah nilai error.

Bab III

Implementasi Pustaka dan Program dalam Java

Dalam Tugas Besar I, terdapat 4 folder utama: src, doc, bin, dan test. Folder src memuat 2 folder lagi yakni folder Library dan Persoalan SPL. Pada Folder PersoalanSPL terdapat 3 file yakni: BicubicSplineInterpolation.java, InterpolasiPolinomial.java, RegresiLinearBerganda.java yang merupakan permasalahan utama dalam tubes ini. Pada folder Library terdapat 6 file yakni: Determinan.java, MatriksBalikan.java, MetodeEliminasi.java, OperasiDasarGambar.java, dan OperasiDasarMatrix.java yang merupakan file-file pendukung operasi matriks untuk menyelesaikan permasalahan utama dalam Tugas Besar ini. Berikut merupakan rincian serta deskripsi (atribut dan metode) dari setiap file tersebut.

3.1 Class Matrix

3.1.1 Atribut

`ROW_CAP`	Integer yang menyimpan nilai kapasitas banyaknya baris pada matriks.
`COL_CAP`	Integer yang menyimpan nilai kapasitas banyaknya kolom pada matriks.
`rowEff`	Integer yang menyimpan nilai efektif banyaknya baris pada matriks.
`colEff`	Integer yang menyimpan nilai efektif banyaknya kolom pada matriks.
`mem`	Array 2D bertipe double yang digunakan untuk menyimpan elemen matriks.

3.1.2 Metode

`int get_ROW_EFF()` `{mengembalikan nilai baris efektif}`	Selektor untuk mendapatkan nilai dari baris efektif matriks.
`int get_COL_EFF()` `{mengembalikan nilai kolom efektif}`	Selektor untuk mendapatkan nilai kolom efektif matriks.
`double get_ELMT(int i, int j)` `{mengembalikan nilai elemen dari matriks pada posisi (i,j)}`	Selektor untuk mendapatkan nilai elemen dari matriks.

`void set_ROW_EFF(int nRow)`
`{I.S. nilai rowEff sembarang`
`F.S. nilai rowEff = nRow}`
	Selektor set untuk mengubah nilai baris efektif baru.

`void set_COL_EFF(int nCol)` `{I.S. nilai colEff sembarang`	Selektor set untuk mengubah nilai kolom efektif baru.

`F.S. nilai colEff = nCol}`	

`void set_ELMT(int i, int j, int x)`
`{I.S. elemen (i,j) pada matriks sembarang`
`F.S. Elemen (i,j) pada matriks = x}`
	Selektor set untuk mengubah nilai elemen matriks pada (i, j) dengan nilai x.

3.2 Class Determinan

3.2.1 Atribut

Tidak ada atribut.

3.2.2 Metode

`public double DetReduksiBaris(Matrix m)` `{mengembalikan nilai determinan dengan metode reduksi baris}`	Fungsi untuk mencari determinan menggunakan metode reduksi baris yaitu dibuat matriks segitiga dan dikalikan setiap elemen diagonalnya.
`public double determinan2x2(Matrix m)` `{mengembalikan nilai determinan dari matriks yang berukuran 2x2}`	Fungsi untuk mencari determinan khusus matriks dengan ukuran 2x2.
`public double Kofaktor(Matrix m, int i, int j)` `{mengembalikan nilai kofaktor dari sebuah matriks m, tanpa baris i dan kolom j}`	Fungsi untuk mencari nilai kofaktor baris i dan kolom j dari matriks m dengan mencari determinan matriks m tetapi mengecualikan baris i dan kolom j.
`public double DetEkspansiKofaktor(Matrix m)` `{mengembalikan nilai determinan dengan metode ekspansi kofaktor}`	Fungsi untuk mencari determinan menggunakan metode ekspansi kofaktor yaitu menjumlahkan perkalian antara kofaktor dan elemen terkait dari suatu baris
`public Matrix adjoinMatrix(Matrix m` `{mengembalikan sebuah matriks yang merupakan adjoin dari matriks m}`	Fungsi untuk menghasilkan matriks yang merupakan adjoin dari matriks input.

`public void splitMatrixSPL(Matrix m, Matrix a, Matrix b)`
`{I.S. matriks m memiliki x kolom, matrix a dan b kosong`
`F.S. matriks a dan b potongan dari matrix m dengan matrix a memiliki x-1 kolom dan b 1 kolom}`
	Prosedur untuk memisahkan matriks m yang augmented menjadi matriks a sebagai A dan matriks b sebagai b dalam persamaan Ax = b.

`public Matrix MatrixAj(Matrix a, int j, Matrix b)` `{mengembalikan matriks adjoin a dengan kolom j digantikan matriks b}`	Fungsi untuk mendapatkan matriks yang merupakan bentuk adjoin dari matriks input dengan menggantikan setiap elemen pada kolom j menjadi terisi elemen matriks b.
`public void KaidahCramer(Matrix m)` `{mengembalikan hasil perhitungan SPL dari matriks m menggunakan Kaidah Cramer}`	Prosedur untuk mendapatkan hasil sistem persamaan linier menggunakan kaidah cramer yaitu dengan memanfaatkan fungsi MatrixAj dan determinan.

3.3 Class Matriks Balikan

3.3.1 Atribut

Tidak ada atribut.

3.3.2 Metode

`public boolean isMatrixHaveInv(Matrix m)` `{mengembalikan status sebuah matriks apakah memiliki inverse}`	Fungsi untuk menentukan apakah sebuah matriks memiliki inverse atau tidak.
`public Matrix createIdentityMatrix(int n)` `{mengembalikan sebuah matriks identitas berukuran nxn)`	Fungsi untuk mendapatkan matriks identitas dengan ukuran nxn.
`public Matrix mergeSidetoSideMatrix(Matrix m1, Matrix m2)` `{mengembalikan matriks yang merupakan gabungan matriks m1 dan m2 yang ukurannya sama}`	Fungsi untuk mendapat matriks hasil menggabungkan dua matriks.
`public Matrix cropMatrix(Matrix m)` `{mengembalikan matriks setengah kolom terakhir dari matriks m}`	Fungsi untuk mendapatkan matriks setengah kolom terakhir.
`public Matrix cropLastColOfMatrix(Matrix m)` `{mengembalikan matriks dengan menghapus kolom terakhir dari matriks m}`	Fungsi untuk mendapatkan matriks yang telah dihapus kolom terakhirnya.
`public Matrix inverse2x2(Matrix m)` `{mengembalikan inverse dari matriks m yang berukuran 2x2}`	Fungsi untuk mendapatkan inverse dari sebuah matriks 2x2.

`public Matrix inverseWithAdjoin(Matrix m)` `{mengembalikan inverse matriks m menggunakan metode adjoin}`	Fungsi untuk mendapatkan matriks inverse dengan menggunakan metode adjoin, memanfaatkan fungsi adjoin.
`public Matrix inverseWithGaussJordan(Matrix m)` `{mengembalikan inverse matriks m menggunakan metode Gauss Jordan}`	Fungsi untuk mendapatkan matriks inverse dengan metode gauss jordan yaitu memanfaatkan fungsi gauss
`public Matrix solveSPLwithInverse(Matrix m)` `{mengembalikan matriks hasil SPL menggunakan metode inverse}`	Fungsi untuk mendapatkan matriks hasil SPL sebuah matriks menggunakan metode inverse.

3.4 Class Metode Eliminasi

3.1.1 Atribut

Tidak ada atribut.

3.2.2 Metode

`public void swapRows(Matrix m, int a, int b)`
`{I.S. matriks m terisi,`
`F.S. matriks m pada baris ke-a ditukar dengan baris ke-b}`
	Prosedur untuk menukarkan antara dua baris sebuah matriks.

`public void divideRow(Matrix m, int idx, double k)`
`{I.S. matriks m terisi,`
`F.S. matriks m pada baris ke-idx dibagi dengan k}`
	Prosedur untuk membagi setiap elemen sebuah baris matriks dengan suatu double.

`public void substractRows(Matrix m, int a, int pivot, double div)`
`{I.S. matriks m terisi,`
`F.S. matriks m pada baris ke-a dikurangi dengan baris ke-pivot yang dikalikan dengan div}`
	Prosedur untuk mengurangi setiap elemen di sebuah baris matriks dengan elemen terkait pada baris lain yang sudah dikalikan dengan sebuah double.

`public void toEselon(Matrix m)` `{I.S. matriks m terisi,`	Prosedur untuk membuat matriks menjadi dalam

`F.S. matriks m menjadi dalam bentuk matriks eselon}`	bentuk eselon.

`public void toEselonRed(Matrix m)`
`{I.S. matriks m terisi,`
`F.S. matriks m menjadi dalam bentuk matriks eselon tereduksi}`
	Prosedur untuk membuat matriks menjadi dalam bentuk eselon tereduksi.

`public boolean isSegitiga(Matrix m)` `{mengembalikan boolean true yang menyatakan bahwa matriks sudah dalam bentuk matriks segitiga dan sebaliknya}`	Fungsi untuk mengecek sebuah matriks apakah berbentuk matriks segitiga.
`public boolean isSolveable(Matrix m)` `{mengembalikan boolean true yang menyatakan bahwa matriks dapat diselesaikan SPL-nya dan sebaliknya}`	Fungsi untuk mengecek sebuah matriks apakah dapat diselesaikan.
`public int Gauss(Matrix m)` `{mengembalikan integer antara 0, 1, atau 2 (tidak dapat diselesaikan, unik, atau parametrik) yang menyatakan status matriks m}`	Fungsi untuk menyatakan apakah matriks menghasilkan SPL yang unik, parametrik, atau tidak diselesaikan.

`public void SolvesSPLParametrik(Matrix m)`
`{I.S. matriks m terisi,`
`F.S. tercetak hasil SPL berupa parametrik}`
	Prosedur untuk mencetak hasil SPL sebuah matriks yang berupa parametrik.

`public Matrix SolveSPLUnik(Matrix m)` `{mengembalikan matriks yang berisi hasil SPL yang unik dari matriks m}`	Fungsi untuk mendapatkan matriks yang berisi hasil SPL yang unik dari sebuah matriks lain.
`public Matrix SolvesSPLUnikRed(Matrix m)` `{mengembalikan matriks yang berisi hasil SPL yang unik dari matriks m tereduksi}`	Fungsi untuk mendapatkan matriks yang berisi
`public double toSegitiga(Matrix m)` `{mengembalikan double yang merupakan determinan dari matriks m yang telah dibuat menjadi matriks segitiga}`	Fungsi untuk mendapatkan nilai determinan dengan membentuk matriks menjadi matriks segitiga.

3.5 Class Operasi Dasar Matriks

3.1.1 Atribut

Tidak ada atribut.

3.2.2 Metode

`void createMatrix(Matrix m, int nRows, int nCols);`
`/* Membentuk sebuah Matrix "kosong" yang siap diisi berukuran nRow x nCol di "ujung kiri" memori */`
`/* I.S. nRow dan nCol adalah valid untuk memori matriks yang dibuat */`
`/* F.S. Matriks m sesuai dengan definisi di atas terbentuk */`
	Inisialisasi elemen pada matriks dan atribut.

`void readMatrix(Matrix m, int nRow, int nCol)`
`/* I.S. isIdxValid(nRow,nCol) */`
`/* F.S. m terdefinisi nilai elemen efektifnya, berukuran nRow x nCol */`
`/* Proses: Melakukan createMatrix(m,nRow,nCol) dan mengisi nilai efektifnya */`
`/* Selanjutnya membaca nilai elemen per baris dan kolom */`
`/* Contoh: Jika nRow = 3 dan nCol = 3, maka contoh cara membaca isi matriks :`
	Menerima input dari terminal menuju dan mengubahnya menjadi tipe data matriks.

`void readMatrixFile(String filename, Matrix m)`
`{I.S. Menerima masukan berupa nama file dan matriks kosong.`
`F.S. Membaca dari filename dan matriks m terisi dari file}`
	Membaca input dari file.txt lalu mengubah menjadi matriks

`void readMatrixFileInterpolate(String filename, Matrix m, double x)`
`{I.S. Menerima masukan berupa nama file dan matriks kosong.`
`F.S. Membaca dari filename dan matriks m terisi dari file}`
	Membaca input dari file.txt lalu mengubah menjadi matriks untuk permasalahn interpolasi

`Matrix readSPL()` `{Mengeluarkan matriks augmented}`	Menerima input dari 2 matriks SPL dan membuat matriks augmented
`Matrix readSPLCramer()` `{Mengeluarkan matriks augmented untuk metode cramer}`	Menerima input dari 2 matriks dan membuat matriks untuk metode SPL Cramer.
`void displayMatrix(Matrix m)`	Menghasilkan output dari

{I.S. Matriks m terdefinisi F.S. Output dari m ke layar}	elemen Matriks m ke layar
void displayMatrixtoFile(Matrix m, String filename) {I.S. Matriks m terdefinisi dan String filename sesuai dengan nama file F.S. Menuliskan isi matriks m ke dalam file }	Menghasilkan file dengan nama filename dengan isi matriks m.
boolean isMatrixIdxValid(int i, int j) {Mengirimkan true jika i, j adalah index yang valid untuk matriks apa pun}	Cek apakah indeks dari Matriks valid
int getLastIdxRow(Matrix m) {Mengirimkan Index baris terbesar m}	Mengirimkan Index baris terbesar m
boolean isIdxEff(Matrix m, int i, int j)	Mengirimkan true jika i, j adalah Index efektif bagi m
double getElmtDiagonal(Matrix m, int i) {Mengirimkan elemen m(i,i)}	Mengirimkan elemen m(i,i)
Matrix copyMatrix(Matrix mIn) {Melakukan assignment mOut <- mIn}	Melakukan assignment mOut <- mIn
Matrix addMatrix(Matrix m1, Matrix m2) {Prekondisi : m1 berukuran sama dengan m2 }	Mengirim hasil penjumlahan matriks: m1 + m2
Matrix subtractMatrix(Matrix m1, Matrix m2) {Mengembalikan hasil dari matriks m1 - m2}	Mengirim hasil pengurangan matriks: salinan m1 – m2
Matrix multiplyMatrix(Matrix m1, Matrix m2) {Prekondisi : Ukuran kolom efektif m1 = ukuran baris efektif m2}	Mengembalikan perkalian antara matriks m1 dengan m2
Matrix multiplyMatrixWithMod(Matrix m1, Matrix m2, int mod) {Prekondisi : Ukuran kolom efektif m1 = ukuran baris efektif m2}	Mengembalikan perkalian antara matriks m1 dengan m2 dengan dimodulo mod
Matrix multiplyByConst(Matrix m, double x) {Mengirim salinan matriks hasil perkalian setiap elemen m dengan x}	Mengirim hasil perkalian setiap elemen m dengan x

`void pMultiplyByConst(Matrix m, double k)` `{Mengembalikan hasil perkalian setiap elemen m dengan x}`	Mengirim hasil perkalian setiap elemen m dengan x
`boolean isMatrixEqual(Matrix m1, Matrix m2)` `{mengirimkan true jika matriks m1 == m2}`	mengirimkan true jika matriks m1 == m2
`boolean isMatrixNotEqual(Matrix m1, Matrix m2)` `{mengirimkan true jika matriks m1 != m2}`	mengirimkan true jika matriks m1tidak sama dengan m2
`boolean isMatrixSizeEqual(Matrix m1, Matrix m2` `{Mengirimkan trus jika ukuran m1 == ukuran m2}`	Fungsi untuk cek jika ukuran matriks m1 sama atau beda dengan matriks m2
`int countElmt(Matrix m)` `{mengembalikan nilai banyaknya elemen matriks m}`	Fungsi mengembalikan banyaknya elemen apda matriks m
`boolean isSquare(Matrix m)` `{mengembalikan true jika matrik m adalah matriks persegi}`	Fungsi cek jika matriks m persegi atau bukan
`boolean isSymmetric(Matrix m)` `{mengembalikan true jika matriks m adalah matriks simetri}`	Fungsi cek jika matriks m simetri atau tidak

`boolean isIdentity(Matrix m)`
`{Mengirimkan true jika m adalah matriks satuan: isSquare(m) dan`
`   setiap elemen diagonal m bernilai 1 dan elemen yang bukan diagonal bernilai 0`
`}`
	Fungsi cek jika matriks M adalah matriks identitas

`boolean isSparse(Matrix m)`
`{Mengirimkan true jika m adalah matriks sparse: matriks “jarang” dengan definisi:`
`   hanya maksimal 5% dari memori matriks yang efektif bukan bernilai 0`
`}`
	Fungsi cek jika matriks M adalah matriks “jarang”

`Matrix negation(Matrix m)` `{Menghasilkan salinan m dengan setiap elemen dinegasikan (dikalikan -1)}`	Mengembalikan matriks dengan tiap elemen dikalikan -1

`void Negation(Matrix m)`
`{/* I.S. m terdefinisi */`
`/* F.S. m di-invers, yaitu setiap elemennya dinegasikan (dikalikan -1) */`
	Memproses matriks m menjadi matriks negasi.

`}`	

`double determinant(Matrix m)`
`/* Prekondisi: isSquare(m) */`
`/* Menghitung nilai determinan m */`
	Mengembalikan nilai determinan dari matriks m

`Matrix mergeMatrix(Matrix m1, Matrix m2)` `{mengembalikan matriks gabungan m1:m2}`	Mengembalikan matriks gabungan dari m1 dan m2.

3.6 Class Interpolasi Polinomial

3.1.1 Atribut

3.2.2 Metode

`public Matrix InputtoMatrix(int n, int input)` `{mengembalikan matriks yang telah dibaca}`	Fungsi untuk mendapatkan matriks hasil dari pembacaan input keyboard.
`public Matrix MatrixtoMatrixInt(Matrix m)` `{mengembalikan matriks dalam bentuk yang sudah siap diinterpolasikan}`	Fungsi untuk mendapatkan matriks yang telah diubah dari matriks yang berisi titik-titik menjadi matriks yang augmented untuk diinterpolasikan.
`public double Interpolasi(Matrix m, double x)` `{mengembalikan nilai interpolasi titik x terhadap matrix m}`	Fungsi untuk mendapatkan hasil interpolasi suatu titik dari sebuah matriks augmented.

3.6 Class Regresi Linier Berganda

3.1.1 Atribut

3.2.2 Metode

`public double sumOfAllElmtInCol(Matrix m, int idxCol)` `{mengembalikan double yang merupakan jumlah semua elemen dari suatu kolom}`	Fungsi untuk mendapatkan nilai jumlah semua elemen dari suatu kolom matriks.

`public void rowTimesConst(Matrix m, int idxRow, double k)`
`{I.S. matriks m terisi, k tidak nol`
`F.S. sebuah baris dikalikan dengan konstanta tidak nol}`
	Prosedur untuk mengubah suatu baris menjadi dikalikan dengan suatu konstanta tidak nol.

`public double sumOfProductTwoCols(Matrix`	Fungsi untuk

`m, int idxCol1, int idxCol2)` `{mengembalikan nilai jumlah dari hasil kali setiap 2 elemen sebaris dalam 2 kolom}`	mendapatkan suatu nilai jumlah dari hasil kali setiap 2 elemen sebaris dalam 2 kolom.
`public Matrix PersamaanHasilRegresiLinearBerganda(Matrix m)` `{mengembalikan matriks yang berisi semua konstanta dari persamaan hasil regresi linear berganda}`	Fungsi untuk mendapatkan matriks yang berisi semua konstanta dari persamaan hasil regresi linear berganda.

3.6 Class Bicubic Spline Interpolation

3.1.1 Atribut

`MBesar`	Matriks yang menyimpan matriks X pada persamaan y=Xa.
`MSoal`	Matriks yang menyimpan input dari soal

3.2.2 Metode

`public void createInitialMatrix(Matrix m)`
`{I.S. matriks sembarang,`
`F.S. matriks x pada bicubic spline}`
	Membuat matriks X pada persamaan
	y = Xa

`Public void StartBSI(int pilihan)`
`{I.S. Memulai proses Bicubic Spline Interpolation`
`F.S. Proses SBI selesai}`
	Fungsi untuk menjalankan Bicubic Spline Interpolation

BAB IV

Eksperimen

4.1 Solusi Persamaan Linear Ax = B

<table>
<tr><th colspan="2">Soal</th><th>Metode</th><th>Output</th></tr>
<tr><td rowspan="4">a)</td><td rowspan="4">$$A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$$</td><td>Gauss</td><td>Masukkan nama file:
1a.txt

1,00 1,00 -1,00 -1,00 1,00
2,00 5,00 -7,00 -5,00 -2,00
2,00 -1,00 1,00 3,00 4,00
5,00 2,00 -4,00 2,00 6,00
Tidak Memiliki Solusi
SPL tidak memiliki solusi
------------------------------</td></tr>
<tr><td>Gauss - Jordan</td><td>Masukkan nama file:
1a.txt

1,00 1,00 -1,00 -1,00 1,00
2,00 5,00 -7,00 -5,00 -2,00
2,00 -1,00 1,00 3,00 4,00
5,00 2,00 -4,00 2,00 6,00
Tidak Memiliki Solusi
SPL tidak memiliki solusi</td></tr>
<tr><td>Matriks Balikan</td><td>Matrix tersebut tidak punya inverse
Matrix tidak dapat diselesaikan menggunakan metode matriks balikan.
Gunakan metode lain
------------------------------</td></tr>
<tr><td>Kaidah Cramer</td><td>Matrix tersebut tidak dapat diselesaikan menggunakan Kaidah Cramer.
Determinan matriks A = 0.
------------------------------</td></tr>
<tr><td colspan="4">Analisis: Hasil keluaran setiap metode sama yaitu tidak memiliki solusi, karena ketika pada metode Gauss dan Gauss-Jordan terdapat baris pada matriks augmented yang berisi elemen 0 semua dan ketika pada metode matriks balikan dan kaidah cramer, matriks tidak dapat di-inverse-kan dan determinan sama dengan nol.</td></tr>
</table>

Soal	Metode	Output

<table>
<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>Solusi Parametrik
X1 = 3.0 + X5
X2 = 2.0 X5
X3 = X3
X4 = -1.0 + X5
X5 = X5
X1 = 3.0 + X5
X2 = 2.0 X5
X3 = X3
X4 = -1.0 + X5
X5 = X5
-----------------------------</td></tr>
<tr><td>Gauss -
Jordan</td><td>-----------------------------
1,00 -1,00 0,00 0,00 1,00 3,00
1,00 1,00 0,00 -3,00 0,00 6,00
2,00 -1,00 0,00 1,00 -1,00 5,00
-1,00 2,00 0,00 -2,00 -1,00 -1,00
Solusi Parametrik
X1 = 3.0 + X5
X2 = 2.0 X5
X3 = X3
X4 = -1.0 + X5
X5 = X5
X1 = 3.0 + X5
X2 = 2.0 X5
X3 = X3
X4 = -1.0 + X5
X5 = X5</td></tr>
<tr><td>Matriks
Balikan</td><td>Matrix tersebut tidak punya inverse
Matrix tidak dapat diselesaikan menggunakan metode matriks balikan.
Gunakan metode lain
-----------------------------</td></tr>
<tr><td>Kaidah
Cramer</td><td>Matrix tersebut tidak dapat diselesaikan menggunakan Kaidah Cramer.
Determinan matriks A = 0.
-----------------------------</td></tr>
<tr><td colspan="4">Analisis: Ketika menggunakan metode Gauss dan Gauss-Jordan menghasilkan output yang sama yaitu berupa parametrik. Sedangkan ketika menggunakan metode matriks balikan dan kaidah cramer tidak dapat diselesaikan karena terdapat syarat harus berupa matriks persegi.</td></tr>
</table>

Soal	**Metode**	**Output**

<table>
<tr><td rowspan="4">c)</td><td rowspan="4">$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$</td><td>Gauss</td><td>Solusi Parametrik
X1 = 0.0
X2 = 1.0 - 1.0 X6
X3 = X3
X4 = -2.0 - 1.0 X6
X5 = 1.0 + X6
X6 = X6
X1 = 0.0
X2 = 1.0 - 1.0 X6
X3 = X3
X4 = -2.0 - 1.0 X6
X5 = 1.0 + X6
X6 = X6
------------------------------</td></tr>
<tr><td>Gauss - Jordan</td><td>Solusi Parametrik
X1 = 0.0
X2 = 1.0 - 1.0 X6
X3 = X3
X4 = -2.0 - 1.0 X6
X5 = 1.0 + X6
X6 = X6
X1 = 0.0
X2 = 1.0 - 1.0 X6
X3 = X3
X4 = -2.0 - 1.0 X6
X5 = 1.0 + X6
X6 = X6
------------------------------</td></tr>
<tr><td>Matriks Balikan</td><td>Matrix tersebut tidak punya inverse
Matrix tidak dapat diselesaikan menggunakan metode matriks balikan.
Gunakan metode lain
------------------------------</td></tr>
<tr><td>Kaidah Cramer</td><td>Matrix tersebut tidak dapat diselesaikan menggunakan Kaidah Cramer.
Determinan matriks A = 0.
------------------------------</td></tr>
<tr><td colspan="4">Analisis: Ketika menggunakan metode Gauss dan Gauss-Jordan menghasilkan output yang sama yaitu berupa parametrik. Sedangkan ketika menggunakan metode matriks balikan dan kaidah cramer tidak dapat diselesaikan karena terdapat syarat harus berupa matriks persegi.</td></tr>
</table>

Soal	**Metode**	**Output**

d)	$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n+1} \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$	Gauss	n=6 n=10
		Gauss - Jordan	n=6 n=10
		Matriks Balikan	n=6 n=10
		Kaidah Cramer	n=6 n=10
Analisis:			

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	Solusi Parametrik X1 = -1.0 + X4 X2 = 2.0 X3 X3 = X3 X4 = X4 X1 = -1.0 + X4 X2 = 2.0 X3 X3 = X3 X4 = X4 ------------------------------

<table>
<tr><td rowspan="3"></td><td rowspan="3"></td><td>Gauss - Jordan</td><td><pre>Solusi Parametrik
X1 = -1.0 + X4
X2 = 2.0 X3
X3 = X3
X4 = X4
X1 = -1.0 + X4
X2 = 2.0 X3
X3 = X3
X4 = X4
-------------------------------</pre></td></tr>
<tr><td>Matriks Balikan</td><td></td></tr>
<tr><td>Kaidah Cramer</td><td></td></tr>
<tr><td colspan="4">Analisis: Ketika menggunakan metode Gauss dan Gauss-Jordan menghasilkan output yang sama yaitu berupa parametrik. Sedangkan ketika menggunakan metode matriks balikan dan kaidah cramer tidak dapat diselesaikan karena terdapat syarat harus berupa matriks persegi.</td></tr>
</table>

<table>
<tr><th colspan="2">Soal</th><th>Metode</th><th>Output</th></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></td></tr>
<tr><td>Gauss - Jordan</td><td></td></tr>
<tr><td>Matriks Balikan</td><td></td></tr>
<tr><td>Kaidah Cramer</td><td></td></tr>
<tr><td colspan="4">Analisis: Ketika menggunakan metode Gauss dan Gauss-Jordan menghasilkan output yang sama yaitu berupa parametrik. Sedangkan ketika menggunakan metode matriks balikan dan kaidah cramer tidak dapat diselesaikan karena terdapat syarat harus berupa matriks persegi.</td></tr>
</table>

4.3 SPL Berbentuk

Soal		**Metode**	**Output**
d)	$\begin{aligned} 8x_1 + x_2 + 3x_3 + 2x_4 &= 0 \\ 2x_1 + 9x_2 - x_3 - 2x_4 &= 1 \\ x_1 + 3x_2 + 2x_3 - x_4 &= 2 \\ x_1 + 6x_3 + 4x_4 &= 3 \end{aligned}$	Gauss	
		Gauss - Jordan	
		Matriks Balikan	
		Kaidah Cramer	
Analisis:			

Soal		**Metode**	**Output**
d)	$\begin{aligned} x_7 + x_8 + x_9 &= 13.00 \\ x_4 + x_5 + x_6 &= 15.00 \\ x_1 + x_2 + x_3 &= 8.00 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_6 + x_8) + 0.61396x_9 &= 14.79 \\ 0.91421(x_3 + x_5 + x_7) + 0.25(x_2 + x_4 + x_6 + x_8) &= 14.31 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_2 + x_4) + 0.61396x_1 &= 3.81 \\ x_3 + x_6 + x_9 &= 18.00 \\ x_2 + x_5 + x_8 &= 12.00 \\ x_1 + x_4 + x_7 &= 6.00 \\ 0.04289(x_1 + x_5 + x_9) + 0.75(x_2 + x_6) + 0.61396x_3 &= 10.51 \\ 0.91421(x_1 + x_5 + x_9) + 0.25(x_2 + x_4 + x_6 + x_8) &= 16.13 \\ 0.04289(x_1 + x_5 + x_9) + 0.75(x_4 + x_8) + 0.61396x_7 &= 7.04 \end{aligned}$	Gauss	3b.txt ------------------------------ 0,00 0,00 0,00 0,00 0,00 0,00 1,00 1,00 1,00 13,00 0,00 0,00 0,00 1,00 1,00 1,00 0,00 0,00 0,00 15,00 1,00 1,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 8,00 0,00 0,00 0,04 0,00 0,04 0,75 0,04 0,75 0,61 14,79 0,00 0,25 0,91 0,25 0,91 0,25 0,91 0,25 14,31 0,00 0,61 0,75 0,04 0,75 0,04 0,00 0,04 0,00 0,00 3,81 0,00 0,00 1,00 0,00 0,00 1,00 0,00 0,00 1,00 18,00 0,00 1,00 0,00 0,00 1,00 0,00 0,00 1,00 0,00 12,00 1,00 0,00 0,00 1,00 0,00 0,00 1,00 0,00 0,00 6,00 0,04 0,75 0,61 0,00 0,04 0,75 0,00 0,00 0,04 10,51 0,91 0,25 0,00 0,25 0,91 0,25 0,00 0,25 0,91 16,31 0,04 0,00 0,00 0,75 0,04 0,00 0,61 0,75 0,04 7,04 Tidak Memiliki Solusi SPL tidak memiliki solusi
		Gauss - Jordan	3b.txt ------------------------------ 0,00 0,00 0,00 0,00 0,00 0,00 1,00 1,00 1,00 13,00 0,00 0,00 0,00 1,00 1,00 1,00 0,00 0,00 0,00 15,00 1,00 1,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 8,00 0,00 0,00 0,04 0,00 0,04 0,75 0,04 0,75 0,61 14,79 0,00 0,25 0,91 0,25 0,91 0,25 0,91 0,25 14,31 0,00 0,61 0,75 0,04 0,75 0,04 0,00 0,04 0,00 0,00 3,81 0,00 0,00 1,00 0,00 0,00 1,00 0,00 0,00 1,00 18,00 0,00 1,00 0,00 0,00 1,00 0,00 0,00 1,00 0,00 12,00 1,00 0,00 0,00 1,00 0,00 0,00 1,00 0,00 0,00 6,00 0,04 0,75 0,61 0,00 0,04 0,75 0,00 0,00 0,04 10,51 0,91 0,25 0,00 0,25 0,91 0,25 0,00 0,25 0,91 16,31 0,04 0,00 0,00 0,75 0,04 0,00 0,61 0,75 0,04 7,04 Tidak Memiliki Solusi SPL tidak memiliki solusi
		Matriks Balikan	Matrix tersebut tidak punya inverse Matrix tidak dapat diselesaikan menggunakan metode matriks balikan. Gunakan metode lain
		Kaidah Cramer	Matrix tersebut tidak dapat diselesaikan menggunakan Kaidah Cramer. Banyaknya persamaan harus sama dengan banyaknya variabel.

Analisis: Hasil keluaran setiap metode sama yaitu tidak memiliki solusi, karena ketika pada metode Gauss dan Gauss-Jordan terdapat baris pada matriks augmented yang berisi elemen 0 semua dan ketika pada metode matriks balikan dan kaidah cramer, matriks tidak dapat di-inverse-kan dan determinan sama dengan nol.

4.4 Studi Kasus Interpolasi

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

x	0.1	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

Lakukan pengujian pada nilai-nilai berikut:

$x = 0.2$ $f(x) = ?$

$x = 0.55$ $f(x) = ?$

$x = 0.85$ $f(x) = ?$

$x = 1.28$ $f(x) = ?$

Pengujian nilai-nilai default

x	**f(x)**
0.2	Masukkan ukuran matriks: 7 Masukkan titik: 0.1 0.003 0.3 0.067 0.5 0.148 0.7 0.248 0.9 0.370 1.1 0.518 1.3 0.697 Masukkan nilai x yang ingin diinterpolasi: 0.2 Solusi Unik Nilai interpolasi x = 0.2 menghasilkan nilai y = 0.032960937500000016 ------------------------------

0.55	Masukkan titik: 0.1 0.003 0.3 0.067 0.5 0.148 0.7 0.248 0.9 0.370 1.1 0.518 1.3 0.697 Masukkan nilai x yang ingin diinterpolasi: 0.55 Solusi Unik Nilai interpolasi x = 0.55 menghasilkan nilai y = 0.17111865234374998
0.85	Masukkan ukuran matriks: 7 Masukkan titik: 0.1 0.003 0.3 0.067 0.5 0.148 0.7 0.248 0.9 0.370 1.1 0.518 1.3 0.697 Masukkan nilai x yang ingin diinterpolasi: 0.85 Solusi Unik Nilai interpolasi x = 0.85 menghasilkan nilai y = 0.33723583984375 ------------------------------
1.28	Masukkan ukuran matriks: 7 Masukkan titik: 0.1 0.003 0.3 0.067 0.5 0.148 0.7 0.248 0.9 0.370 1.1 0.518 1.3 0.697 Masukkan nilai x yang ingin diinterpolasi: 1.28 Solusi Unik Nilai interpolasi x = 1.28 menghasilkan nilai y = 0.6775418374999999 ------------------------------
Analisis: Didapatkan nilai interpolasi di range yang masuk akal.	

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:

Tanggal (desimal)	**f(x)**
0.2	
0.55	
0.85	
1.28	
Analisis:	

a. 16/07/2022

Insert photo here

b. 10/08/2022

Insert Photo Here

c. 05/09/2022

Insert Photo Here

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

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	Nila fungsi	Hasil Interpolasi
4		
5		
6		

4.5 Studi Kasus Regresi Linear Belanda

Diberikan sekumpulan data sesuai pada tabel berikut ini.

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.

Gunakan *Normal Estimation Equation for Multiple Linear Regression* untuk mendapatkan regresi linear berganda dari data pada tabel di atas, kemudian estimasi nilai Nitrous Oxide apabila Humidity bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30.

Dari data-data tersebut, apabila diterapkan *Normal Estimation Equation for Multiple Linear Regression*, maka diperoleh sistem persamaan linear sebagai berikut.

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

4.6 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.

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

Hasil Interpolasi.

f(x, y)	Hasil Interpolasi
f(0, 0)	Selamat Datang Di Bicubic Spline Interpolation Masukkan Nama File: 7i.txt f(0.000000, 0.000000) = 21.000000 ---
f(0.5, 0.5)	Selamat Datang Di Bicubic Spline Interpolation Masukkan Nama File: 7ii.txt f(0.500000, 0.500000) = 87.796875 ---

f(0.25, 0.75)	`Selamat Datang Di Bicubic Spline Interpolation`
	`Masukkan Nama File: 7iii.txt`
	`f(0.250000, 0.750000) = 117.732178 ---`

f(0.1, 0.9)	`Selamat Datang Di Bicubic Spline Interpolation`
	`Masukkan Nama File: 7iv.txt`
	`f(0.100000, 0.900000) = 128.575187 ---`

BAB V

Kesimpulan, Saran, dan Refleksi

5.1 Kesimpulan

Tugas Besar I IF2123 Aljabar Linier dan Geometri ini merupakan bentuk implementasi materi matriks dan operasi-operasinya dalam bentuk algoritma program dalam bahasa Java. Operasi-operasi tersebut diubah dalam bentuk-bentuk fungsi dan prosedur serta dijadikan suatu package dalam program sehingga dapat digunakan untuk mencari nilai-nilai lainnya yang menggunakan matriks. Nilai yang dimaksud adalah interpolasi polinomial, regresi linier berganda, dan bicubic spline interpolation. Sehingga program kami saat ini memiliki 6 fitur utama yaitu:

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

Untuk format input dan output program kami sudah dapat menerima dari input keyboard maupun file serta mengeluarkan melalui terminal maupun bentuk file baru. Berdasarkan beberapa test case yang telah kami lakukan didapatkan analisis berupa ketika penyelesaian SPL menggunakan metode Gauss dan Gauss-Jordan tidak ada pembatasan harus berupa matriks persegi, sehingga dapat menghasilkan solusi dalam bentuk parametrik. Sedangkan metode matriks balikan dan kaidah cramer terdapat pembatas harus merupakan matriks persegi karena hanya dapat mengeluarkan SPL dengan solusi unik.

Disimpulkan melalui Tugas Besar I IF2123 Aljabar Linier dan Geometri dapat dibuat sebuah algoritma implementasi dari operasi dan perhitungan matriks dengan berbagai metode.

5.2 Saran

Saran untuk pengembangan selanjutnya adalah sebagai berikut:

1. Membuat implementasi lebih lanjut seperti untuk image scaling maupun implementasi lainnya yang lebih fungsional.
2. Membuat format aplikasi program yang lebih interaktif dan lebih mudah digunakan bagi pengguna seperti menggunakan GUI.
3. Membuat package, fungsi, maupun prosedur yang modular dan lebih jelas agar lebih mudah ketika dilakukan pengembang lanjutan.
4. Menyediakan waktu yang lebih banyak untuk membuat format input dan output setiap kemungkinan dikarenakan spesifikasi yang cukup rinci.

5.3 Refleksi

Dalam mengerjakan tugas besar ini, kami memperoleh beragam pengalaman berharga. Tugas ini menjadi peluang untuk meningkatkan keterampilan kerja sama, komunikasi, dan kemampuan memecahkan berbagai tantangan. Selain itu, kami melatih disiplin dalam menyelesaikan tugas masing-masing. Kami merasa bahwa pembagian tugas dilakukan secara adil dan setara. Selama proses ini, kami juga mendapatkan pemahaman yang lebih mendalam tentang bahasa pemrograman Java.