

**LAPORAN TUGAS BESAR**  
**IF2123 - ALJABAR LINEAR DAN GEOMETRI**  
**SEMESTER I 2023/2024**

Kelompok 25 / BiggerWeapons 🇲🇵

Ibrahim Ihsan Rasyid	13522018
Zaki Yudhistira Candra	13522031
Muhammad Syaraf Akmal	13522076



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

## DAFTAR ISI

<b>BAB 1.....</b>	<b>4</b>
<b>BAB 2.....</b>	<b>5</b>
A. Matriks, determinan matriks, dan matriks balikan.....	5
B. Metode-metode dalam menyelesaikan sistem persamaan linier.....	6
C. Interpolasi Polinomial.....	6
D. Regresi Linear Berganda.....	7
E. Bicubic Spline Interpolation.....	7
<b>BAB 3.....</b>	<b>9</b>
A. Folder Matrix.....	9
1. Matrix.java.....	9
2. Matrix_scan.java.....	12
3. Matrix_save.java.....	13
B. Folder Functions.....	14
1. BcbSpline.java.....	14
2. Interpolasi.java.....	15
3. Inverse.java.....	15
4. Regresi.java.....	16
5. SPL.java.....	17
C. Folder Main.....	18
<b>BAB 4.....</b>	<b>20</b>
A. Inisiasi Program.....	20
B. Sistem Persamaan Linear.....	21
a. Metode Gauss.....	21
b. Metode Gauss Jordan.....	25
c. Aturan Crammer.....	29
d. Metode Inverse.....	29
C. Inverse.....	29
a. Metode Operasi Baris Elementer.....	29
b. Metode Minor Kofaktor.....	29
D. Determinan.....	30
a. Metode Reduksi Eselon.....	30
b. Metode Minor Kofaktor.....	30
E. Interpolasi Polinomial.....	31
F. Interpolasi Bicubic Spline.....	36
G. Regresi Linier Berganda.....	39
<b>BAB 5.....</b>	<b>42</b>

A. Hasil yang Dicapai.....	42
B. Saran Pengembangan.....	42
C. Refleksi.....	42
<b>Daftar Referensi.....</b>	<b>43</b>

## BAB 1

### Deskripsi Masalah

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Telah dipelajari 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, dan kaidah Cramer (khusus untuk SPL dengan  $n$  peubah dan  $n$  persamaan). Solusi sebuah SPL mungkin tidak ada, banyak (tidak berhingga), atau hanya satu (unik/tunggal).

Di dalam Tugas Besar 1 ini, kami membuat beberapa library aljabar linier dalam Bahasa Java. Library tersebut berisi fungsi-fungsi seperti eliminasi Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung determinan, kaidah Cramer (kaidah Cramer khusus untuk SPL dengan  $n$  peubah dan  $n$  persamaan). Selanjutnya, library tersebut digunakan di dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi.

## BAB 2

### Teori Singkat

#### A. Matriks, determinan matriks, dan matriks balikan

Matriks adalah *array* atau tabel berisikan nilai-nilai berupa angka, symbol, dan ekspresi lainnya. Matriks tersusun atas baris dan kolom yang biasanya digunakan untuk merepresentasikan suatu objek matematika.

Suatu matriks yang memiliki baris dan kolom yang sama, biasa disebut matriks persegi, memiliki suatu nilai yang merepresentasikan matriks tersebut yaitu determinan. Pencarian matriks dapat dicari dengan 2 metode yaitu kofaktor dan reduksi barisan. Pada matriks  $2 \times 2$ ,

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

**Gambar 2.1** Matriks  $2 \times 2$

Determinan dapat dicari dengan  $\det A = (ad - bc)$ , melalui basis ini dapat dicari matriks dengan baris dan kolom  $> 2$  dengan minor-kofaktor. Proses minor-kofaktor akan menghasilkan penjumlahan matriks baru sebanyak jumlah kolom dengan baris dan kolom yang berkurang 1 pada matriks barunya. Melalui proses ini dapat dicari determinan secara rekursif dengan matriks  $2 \times 2$  sebagai basis.

Reduksi barisan merupakan metode untuk mencari determinan dengan mengalikan elemen-elemen diagonal dari matriks (matriks dengan baris dan kolom  $> 2$ ). Bentuk reduksi matriks adalah sebagai berikut,

$$\begin{bmatrix} y_{11} & y_{12} & y_{13} \\ 0 & y_{22} & y_{23} \\ 0 & 0 & y_{33} \end{bmatrix}$$

**Gambar 2.2** Matriks  $3 \times 3$  yang telah direduksi

Dengan mengalikan  $y_{11}$ ,  $y_{22}$ , dan  $y_{33}$  kita bisa mendapatkan nilai determinan dari sebuah matriks. Proses mencari bentuk reduksi barisan matriks dilakukan dengan OBE pada matriks.

Matriks memiliki bentuk balikan/invers yang memiliki sifat secara operasi matriks sebagai kebalikan bagi matriks awal. Matriks invers dapat dicari melalui 2 metode yaitu kofaktor dan reduksi barisan. Secara definisi umum matriks invers adalah sebagai berikut,

$$A^{-1} = \frac{1}{\det A} \times \text{Adj}(A)$$

Metode kofaktor digunakan untuk mencari adjoin matriks A dan juga determinan. Pada adjoin, hasil determinan dari kofaktor matriks akan menjadi elemen sesuai dengan lokasi singgungan kofaktor. Saat sudah ditemukan nilai determinan untuk jadi tiap elemen, matriks akan ditranspose dan hasil akhir akan menjadi matriks adjoin. Matriks adjoin akan dikalikan dengan 1/det matriks.

## B. Metode-metode dalam menyelesaikan sistem persamaan linier

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

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

**Gambar 2.3** Matriks perhitungan dengan operasi barisan elementer

## C. Interpolasi Polinomial

Interpolasi polinom adalah penggabungan sejumlah  $n$  titik menjadi persamaan polinom yang memiliki lintasan tertentu. Pola pada persamaan polinomial berupa,

$$P_n(x) = a_0 + a_1x + a_1x^2 + \dots + a_nx^n$$

Persamaan di atas dapat digunakan untuk mengaproksimasi sebuah nilai  $y$  berdasarkan nilai  $x$  pada range  $[x_0, x_n]$ . Untuk menghasilkan persamaan dengan pangkat tertinggi  $n$ , diperlukan titik sebanyak  $n+1$ . Untuk mendapatkan nilai  $a_0, a_1, a_2, \dots$ , kita perlu penyelesaian dengan SPL. Pada tubes ini kita mengaplikasikan konsep penyelesaian SPL dengan matriks yang ter-augmentasi dan menggunakan berbagai metode penyelesaian SPL (Gauss, Gauss-Jordan, Cramer) sehingga didapatkan dengan nilai-nilai  $a$  fungsi  $P_n(x) = a_0 + a_1x + a_1x^2 + \dots + a_nx^n$  untuk mengaproksimasi nilai  $y$  dari sebuah  $x$ .

#### D. Regresi Linear Berganda

Regresi Linear Berganda merupakan salah satu metode untuk memprediksi nilai. Meskipun sudah ada persamaan jadi untuk menghitung regresi linear sederhana, terdapat persamaan umum dari regresi linear yang bisa digunakan untuk regresi linear berganda, yaitu

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

**Gambar 2.4** Rumus umum regresi linier

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

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

**Gambar 2.5** Rumus perhitungan *Normal Estimation Equation*

Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode penyelesaian SPL dengan matriks, seperti eliminasi Gauss, eliminasi Gauss-Jordan, dan lainnya.

#### E. Bicubic Spline Interpolation

*Bicubic Spline Interpolation* adalah metode untuk mengaproksimasi nilai fungsi sekitarnya melalui turunan berarah. Metode ini menggunakan 4 titik sebagai referensi yaitu  $f(0,0)$ ,  $f(1,0)$ ,  $f(0,1)$ , dan  $f(1,1)$  dengan pemodelan  $f(x,y)$  sebagai berikut

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

Nilai-nilai  $a$  dapat kita cari melalui pemodelan matriks  $y = Xa$  dan menggunakan metode SPL matriks, dimana  $y$  adalah matriks  $16 \times 1$  berisi nilai-nilai  $f(0,0)$ ,  $f(1,0)$ ,

$f(0,1)$ ,  $f(1,1)$ ,  $f(0,0)_x$ ,  $f(1,0)_x$ ,  $f(0,1)_x$ ,  $f(1,1)_x$ ,  $f(0,0)_y$ ,  $f(1,0)_y$ ,  $f(0,1)_y$ ,  $f(1,1)_y$ ,  $X$  berisi nilai-nilai dari pemodelan diatas dari nilai-nilai di  $Y$  dan  $a$  berisi solusi-solusi.



## BAB 3

### Implementasi Pustaka dan Program dalam Java

Pada program ini, terdapat 9 *class* yang terbagi ke dalam 3 folder, yaitu Matrix, Functions, dan Main.

#### A. Folder Matrix

Folder Matrix berisi 3 class yaitu Matrix, Matrix\_scan, dan Matrix\_save.

##### 1. Matrix.java

Matrix.java berisi tipe data bentukan Matrix beserta fungsi-fungsi operasi ataupun manipulasi matriks. Secara umum, class ini terdiri dari :

- Atribut

Atribut	Penjelasan
public double[][] data	Atribut yang menyimpan elemen-elemen Matrix yang bertipe double
public int row	Atribut berupa banyaknya baris yang bertipe integer
public int collumns	Atribut berupa banyaknya kolom yang bertipe integer

- Konstruktor

Konstruktor	Penjelasan
public Matrix(double[][] matriks, int row_input, int collumn_input)	Mendefinisikan sebuah Matrix dengan atribut-atribut tersebut

- Metode

Metode	Penjelasan
public static void set_row(Matrix m, int row)	Menyimpan banyaknya baris dari Matrix dengan row
public static void	Menyimpan banyaknya kolom dari

set_columns(Matrix m, int columns)	Matriks dengan columns
public double getELMT(int i, int j)	Mengambil elemen dengan indeks (i, j)
public void setELMT(int i, int j, double val)	Menyimpan elemen pada indeks (i, j) dengan val
public void display()	Mencetak tiap elemen Matrix sesuai dengan indeksnya
public boolean isPersegi()	Mengembalikan true jika Matrix adalah matriks persegi, dan sebaliknya
public static boolean isRowIndexValid(Matrix m, int row)	Mengembalikan true jika indeks baris row valid untuk Matrix m, dan sebaliknya
public static boolean isCollumnIndexValid(Matrix m, int collumn)	Mengembalikan true jika indeks kolom collumn valid untuk Matrix m, dan sebaliknya
public static boolean isValid(Matrix m, int row, int collumn)	Mengembalikan true jika indeks tidak valid untuk Matrix m, dan sebaliknya
public static boolean isIdentity(Matrix m)	Mengembalikan true jika Matrix m merupakan matriks identitas, dan sebaliknya
public boolean isRowZero(int row, int batas_collumn)	Mengembalikan true jika hingga kolom tertentu, elemen Matrix pada baris tersebut adalah 0, dan sebaliknya
public boolean isCollumnZero(int collumn)	Mengembalikan true jika setiap elemen Matrix pada kolom tertentu adalah 0, dan sebaliknya
public static boolean isMatrixZero(Matrix m)	Mengembalikan true jika setiap elemen Matrix adalah 0, dan sebaliknya
public boolean isAllRowNotZero()	Mengembalikan true jika semua baris Matriks tidak ada yang 0

public static Matrix transpose(Matrix m)	Mengembalikan sebuah Matrix yang merupakan hasil transpos dari Matrix m
public void self_transpose()	Melakukan transpose untuk Matrix m dan di simpan di dalamnya
public void multiplyByConst(double val)	Mengalikan suatu Matrix dengan suatu konstanta
public void subtract_baris (int row1, int row2, double factor)	Mengurangi suatu row baris dengan row baris lainnya dengan faktor pengali
public void divideBaris(int row, double pembagi)	Membagi baris Matrix dengan suatu pembagi
public void tukar_baris(int row1, int row2)	Menukar suatu baris pada Matrix dengan baris lainnya
public void removeCollumn(int collumn)	Menghilangkan kolom dari suatu Matrix
public void removeZeroCollumn()	Menghilangkan kolom yang bernilai 0 pada suatu Matrix
public void normalize()	Menjadikan Matrix agar berbentuk (m-1)x(m) dengan menambahkan baris kosong
public static Matrix addRow(Matrix m, boolean under)	Menambahkan row pada sebuah Matrix
public static Matrix addCol(Matrix m, boolean right)	Menambahkan kolom pada suatu Matrix
public void divideByPivot()	Membagi per baris Matrix berdasarkan elemen pemimpin setelah reduksi
public void p_reduksi_eselon(boolean btm)	Mereduksi eselon suatu Matrix
public static double getDeterminanReduksi(Matrix m)	Mencari nilai determinan Matrix dengan metode reduksi eselon
public static double getDeterminanKofaktor(Matrix m)	Mencari nilai determinan Matrix dengan metode kofaktor

public static Matrix createIdentity(int n)	Membentuk Matrix identitas
public static int countRowNotZero(Matrix m, int row)	Menghitung jumlah baris yang tidak 0 pada suatu Matrix
public Matrix copyMatrix()	Menduplikat suatu Matrix
public static String MatrixtoString(Matrix m)	Mengubah Matrix menjadi suatu String
public static void splitAugmentMtx(Matrix mAug, Matrix mSPL, Matrix mJwb)	Membagi augmented Matrix menjadi Matrix Aij dan B
public static void splitDataAndApprox(Matrix mAwal, Matrix mData, Matrix mApprox)	Membagi Matrix mAwal (mxn) menjadi Matrix mData (m-1xn) dan mApprox (n-1x1)
public static Matrix multiplyMatrix(Matrix m1, Matrix m2)	Melakukan perkalian Matrix
public static Matrix gantiKolom(Matrix m1, Matrix m2, int col)	Mensubsitusi kolom pada Matrix dengan suatu kolom
public static double sumColumnWithOp(Matrix m, int col1, int col2)	Mendapatkan jumlah kolom m1*m2
public static double sumColumnWithoutOP(Matrix m, int col)	Mendapatkan jumlah dari kolom tanpa dikalikan faktor
public static double[] kali_array(double[] array, double faktor)	Mengalikan Array dengan suatu faktor
public static void kurang_array(double[] array1, double[] array2)	Mengurangi array1 dengan array2

## 2. Matrix\_scan.java

Matrix\_scan.java berisi fungsi-fungsi yang menyediakan metode untuk memasukkan input dari keyboard (scan\_matriks\_keyboard) atau dari sebuah file text (scan\_file).

Input dari keyboard secara garis besar meminta scan dari keyboard dan memasukkan inputan pada sebuah matriks dengan isi tipe elemen double.

Input dari file text secara garis besar meminta nama file lalu fungsi akan mencari nama file tersebut pada directory yang sudah ditentukan, lalu file akan dibaca dengan membaca per *line* dan memasukkan semua string, kecuali spasi, ke array string dengan metode split(). Elemen per line akan dibaca selama object scanner memiliki nextline dengan while looping, array string didefinisikan per looping sehingga akan diperbarui sesuai dengan line, lalu setiap elemen string pada array string tadi akan diparse menjadi tipe elemen double dan akan dimasukkan ke sebuah matriks dengan tipe elemen sama. Komponen jumlah baris dapat dicari dengan iterasi sebuah variabel setiap looping dan komponen kolom didapatkan melalui komparasi *length* maksimum tiap array string.

Secara umum, class terdiri atas :

- Atribut  
Tidak ada atribut pada class ini
- Konstruktor  
Tidak ada konstruktor pada class ini
- Metode

Atribut	Penjelasan
public static void scan_matriks_keyboard(Matrix m, String jenis)	Menerima masukan ukuran serta masing-masing elemen matriks dengan metode input bergantung jenis, lalu menyimpannya ke Matrix m
public static Matrix scan_file(String file_name)	Mengembalikan sebuah Matrix dengan membaca file file_name

## 3. Matrix\_save.java

Matrix\_save.java berisi prosedur untuk menerima suatu array String dan menuliskannya pada suatu file. File dituliskan dalam format .txt. Dimanfaatkan library file pada java. File disimpan per sesi pemakaian alat, pengguna akan diminta untuk memasukkan nama sebagai identitas sesi, dan memiliki opsi untuk

menyimpan proses pada file apabila diperlukan. Maksimal terdapat 100 kali penggunaan fungsi dalam satu kali pemakaian program.

Secara umum, class ini terdiri atas:

- Atribut  
Tidak ada atribut pada class ini
- Konstruktor  
Tidak ada konstruktor pada class ini
- Metode

Atribut	Penjelasan
public static void saveFile(String[] array_solusi, String directory, String Nama_file, int count)	Melakukan penyimpanan dari keluaran-keluaran pada sesi penjalanan program

## B. Folder Functions

Folder Functions berisi 5 class yaitu BcbSpline, Interpolasi, Inverse, Regresi, dan SPL.

### 1. BcbSpline.java

BcbSpline.java berisi fungsi-fungsi untuk menyelesaikan persoalan bicubic spline interpolation dengan menerima input berupa matriks 4x4 dan nilai (x, y) yang ingin ditaksir. Secara umum, class ini terdiri atas :

- Atribut  
Tidak ada atribut pada class ini.
- Konstruktor  
Tidak ada konstruktor pada class ini.
- Metode

Metode	Penjelasan
public static void Interpolasi_bcb(Matrix fxy, Matrix xy, String [] array_solusi, int idx_array)	Melakukan pencarian solusi (variabel $a$ ) pada pemodelan aproksimasi nilai $f(x,y)$ untuk mencari nilai aproksimasi terhadap $x$ dan $y$ yang diberikan.
public static double PangkatN(double a, int b)	Melakukan perpangkatan terhadap elemen double $a$ sebanyak $b$ kali

## 2. Interpolasi.java

Interpolasi.java berisi fungsi-fungsi untuk menyelesaikan persoalan interpolasi polinomial dengan menerima input tiap pasang (xi, yi) dan nilai x yang ingin ditaksir hasilnya. Secara umum, class ini terdiri atas :

- Atribut  
Tidak ada atribut pada class ini.
- Konstruktor  
Tidak ada konstruktor pada class ini.
- Metode

Metode	Penjelasan
public static void Interpolasi_Polinom(Matrix m, double val, String [] array_solusi, int idx_array)	Melakukan perancangan rumus interpolasi polinom melalui titik-titik yang diberikan dan juga memberikan hasil nilai aproksimasi y dari x yang diberikan
public static double PangkatN(double a, int b)	Melakukan perpangkatan terhadap elemen double a sebanyak b kali

## 3. Inverse.java

Inverse.java berisi fungsi-fungsi untuk mencari balikan dari suatu matriks, baik dengan metode reduksi baris maupun dengan metode kofaktor matriks. Secara umum, class ini terdiri atas :

- Atribut  
Tidak ada atribut pada class ini.
- Konstruktor  
Tidak ada konstruktor pada class ini.
- Metode

Metode	Penjelasan
public static void Inverse_matrix_reduksi(Matrix m, boolean utama, String[]	Melakukan penghitungan balikan matriks m dengan metode reduksi baris, dan apabila utama maka

array_solusi, int idx_array)	hasil dan penghitungan ditampilkan
public static Matrix adjoint(Matrix m)	Menentukan dan mengembalikan matriks adjoin dari matriks m
public static void Inverse_Kofaktor(Matrix m, String [] array_solusi, int idx_array)	Melakukan penghitungan balikan matriks m dengan metode kofaktor

#### 4. Regresi.java

Regresi.java berisi fungsi-fungsi untuk memecahkan persoalan regresi linier berganda dengan menerima input sebanyak sebarang data dan nilai yang ingin ditaksir, lalu mengeluarkan hasil regresi beserta nilai taksiran yang ingin dicari. Secara umum, class terdiri atas :

- Atribut

Atribut	Penjelasan
public static Matrix solusi_regresi	Atribut berupa matriks nx1 yang berisi koefisien-koefisien hasil regresi
public static double hasil_taksir	Atribut berupa hasil perhitungan nilai taksiran yang ingin dicari bertipe double

- Konstruktor  
Tidak ada konstruktor pada class ini



- Metode

Metode	Penjelasan
public static Matrix normalEstEq(Matrix m)	Melakukan penghitungan Normal Estimation Equation dari matriks data m dan mengembalikan matriks hasil penghitungan
public static void solusiRegresi(Matrix mUtama, Matrix mTaksir, String [] array_solusi, int idx_array)	Melakukan kalkulasi regresi terhadap mUtama dan menyimpannya ke solusi_regresi, lalu menghitung nilai taksiran dari mTaksir dan menyimpannya ke hasil_taksir
public static void printHasil(Matrix m)	Mencetak hasil regresi dan penaksiran yang tersimpan dalam Matrix m dengan format

## 5. SPL.java

Library SPL berisi prosedur untuk menyelesaikan suatu sistem persamaan linear, memanfaatkan beberapa fungsi-fungsi dari library Matrix.java dan beberapa library lainnya. Library SPL ini berisi beberapa metode penyelesaian SPL yaitu, penyelesaian SPL dengan metode Gauss, Gauss Jordan, Aturan Crammer, dan Inverse Matrix. Class SPL terdiri atas :

- Atribut  
Tidak ada atribut pada class ini
- Konstruktor  
Tidak ada konstruktor pada class ini
- Metode

Atribut	Penjelasan
public static void Gauss(Matrix m, String[] array_solusi, int idx_array)	Perhitungan SPL dengan metode eliminasi Gauss
public static void Gauss_Jordan(Matrix m, String[] array_solusi, int idx_array)	Perhitungan SPL dengan metode eliminasi Gauss Jordan
public static double[][]	Memperoleh Matriks jawaban

getJawabanSPL2(Matrix m){	yang nanti akan diproses apabila SPL tidak memiliki jawaban tunggal (parametrik).
public static void Inverse(Matrix m, boolean utama, String[] array_solusi, int idx_array)	Perhitungan SPL dengan metode matriks balikan dan menampilkan hasilnya apabila utama
public static void Cramer(Matrix m, String[] array_solusi, int idx_array)	Perhitungan SPL dengan kaidah Cramer dan menampilkan hasilnya

### C. Folder Main

Folder Main berisi hanya 1 class, yaitu Main. Main.java berisi fungsi-fungsi untuk melakukan eksekusi utama program ini. Ketika dijalankan, program akan meminta input dari pengguna sesuai dengan menu-menu yang tersedia yang akan ditampilkan di layar seperti menghitung solusi SPL, menghitung determinan matriks, melakukan regresi, dan sebagainya, dan akan berhenti ketika pengguna memilih menu keluar. Secara umum, class terdiri atas :

- Atribut  
Tidak ada atribut pada class ini.
- Konstruktor  
Tidak ada konstruktor pada class ini.
- Metode

Metode	Penjelasan
public static void delay(long milisecond)	Memberikan efek jeda saat inisialisasi program
public static void pressAnyKeytoContinue()	Meminta pengguna untuk menekan tombol enter untuk melanjutkan program setelah mengeksekusi suatu fungsi
public static void printMainMenu(long delay_length, String sesi, String date, int usage)	Mencetak main menu untuk ditampilkan pada layar
public static void	Mencetak menu masukan untuk ditampilkan

<code>printJenisMasukan()</code>	pada layar
<code>public static void printBiggerWeapon(String file_name)</code>	Mencetak “BiggerWeapons” (tulisan beserta gambar senjata) untuk tampilan awal program

## Eksperimen

### A. Inisiasi Program

Dilakukan inisiasi program terlebih dahulu.



### Gambar 4.1 Inisiasi Program

Program akan meminta untuk memasukkan nama sesi yang nantinya akan digunakan sebagai penamaan file yang akan disimpan. Program akan menampilkan menu dan pengguna dapat memasukkan sebuah integer yang akan menjadi acuan pilihan pengguna.

## B. Sistem Persamaan Linear

### a. Metode Gauss

Berikut adalah eksperimen program menggunakan metode eliminasi Gauss untuk persamaan - persamaan berikut.

#### 1. Persamaan SPL 1

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

**Gambar 4.3.a** Persamaan SPL 1

```
Masukkan tujuan Anda : 1
==Pilih metode perhitungan SPL==
--> 1. Eliminasi Gauss
--> 2. Eliminasi Gauss Jordan
--> 3. Metode matriks balikan
--> 4. Aturan Crammer
Masukkan tujuan Anda : 1
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 2
Masukkan nama file : spl1.txt
Menghitung SPL dengan metode GAUSS.
1.0000 1.0000 -1.0000 -1.0000 1.0000
2.0000 5.0000 -7.0000 -5.0000 -2.0000
2.0000 -1.0000 1.0000 3.0000 4.0000
5.0000 2.0000 -4.0000 2.0000 6.0000
SPL tidak memiliki solusi.
<Tekan ENTER untuk melanjutkan>
```

**Gambar 4.3.b** Solusi persamaan SPL 1

#### 2. Persamaan SPL 2

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

**Gambar 4.4.a** Persamaan SPL 2

```

Masukkan tujuan Anda : 1
==Pilih metode perhitungan SPL==
--> 1. Eliminasi Gauss
--> 2. Eliminasi Gauss Jordan
--> 3. Metode matriks balikan
--> 4. Aturan Cramer
Masukkan tujuan Anda : 1
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 2
Masukkan nama file : spl2.txt
Menghitung SPL dengan metode GAUSS.
1.0000 -1.0000 0.0000 0.0000 1.0000 3.0000
1.0000 1.0000 0.0000 -3.0000 0.0000 6.0000
2.0000 -1.0000 0.0000 1.0000 -1.0000 5.0000
-1.0000 2.0000 0.0000 -2.0000 -1.0000 -1.0000
SPL tidak memiliki solusi tunggal.
x0 = 3.0000+1.0000a
x1 = 2.0000a
x2 = -1.0000+1.0000a
x3 = 1.0000a
<Tekan ENTER untuk melanjutkan>

```

**Gambar 4.4.b** Solusi persamaan SPL 2

### 3. Persamaan SPL 3

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

**Gambar 4.5.a** Persamaan SPL 3

```

Masukkan tujuan Anda : 1
==Pilih metode perhitungan SPL==
--> 1. Eliminasi Gauss
--> 2. Eliminasi Gauss Jordan
--> 3. Metode matriks balikan
--> 4. Aturan Cramer
Masukkan tujuan Anda : 1
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 2
Masukkan nama file : spl3.txt
Menghitung SPL dengan metode GAUSS.
0.0000 1.0000 0.0000 0.0000 1.0000 0.0000 2.0000
0.0000 0.0000 0.0000 1.0000 1.0000 0.0000 -1.0000
0.0000 1.0000 0.0000 0.0000 0.0000 1.0000 1.0000
SPL tidak memiliki solusi tunggal.
x0 = 1.0000-1.0000a
x1 = -2.0000-1.0000a
x2 = 1.0000+1.0000a
x3 = 1.0000a
<Tekan ENTER untuk melanjutkan>

```

**Gambar 4.5.b** Solusi persamaan SPL 3

#### 4. Persamaan SPL 4

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

**Gambar 4.6.a** Persamaan SPL 4

```
Masukkan tujuan Anda : 1
==Pilih metode perhitungan SPL==
--> 1. Eliminasi Gauss
--> 2. Eliminasi Gauss Jordan
--> 3. Metode matriks balikan
--> 4. Aturan Crammer
Masukkan tujuan Anda : 1
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 2
Masukkan nama file : spl4.txt
Menghitung SPL dengan metode GAUSS.
1.0000 -1.0000 2.0000 -1.0000 -1.0000
2.0000 1.0000 -2.0000 -2.0000 -2.0000
-1.0000 2.0000 -4.0000 1.0000 1.0000
3.0000 0.0000 0.0000 -3.0000 -3.0000
SPL tidak memiliki solusi tunggal.
x0 = -1.0000+1.0000b
x1 = 2.0000a
x2 = 1.0000a
x3 = 1.0000b
<Tekan ENTER untuk melanjutkan>
```

**Gambar 4.6.b** Solusi persamaan SPL 4

## 5. Matrix Hilbert 1

```
1.0 0.5 0.3333333333333333 0.25 0.2 0.1666666666666666 0
0.5 0.3333333333333333 0.25 0.2 0.1666666666666666 0.14285714285714285 0
0.3333333333333333 0.25 0.2 0.1666666666666666 0.14285714285714285 0.125 0
0.25 0.2 0.1666666666666666 0.14285714285714285 0.125 0.1111111111111111 0
0.2 0.1666666666666666 0.14285714285714285 0.125 0.1111111111111111 0.1 0
0.1666666666666666 0.14285714285714285 0.125 0.1111111111111111 0.1 0.090909090909091 0
```

**Gambar 4.7.a** Matrix Hilbert 1 (n = 6)

```
Masukkan tujuan Anda : 1
==Pilih metode perhitungan SPL==
--> 1. Eliminasi Gauss
--> 2. Eliminasi Gauss Jordan
--> 3. Metode matriks balikan
--> 4. Aturan Crammer
Masukkan tujuan Anda : 1
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 2
Masukkan nama file : hilbert1.txt
Menghitung SPL dengan metode GAUSS.
1.0000 0.5000 0.3333 0.2500 0.2000 0.1667 0.0000
0.5000 0.3333 0.2500 0.2000 0.1667 0.1429 0.0000
0.3333 0.2500 0.2000 0.1667 0.1429 0.1250 0.0000
0.2500 0.2000 0.1667 0.1429 0.1250 0.1111 0.0000
0.2000 0.1667 0.1429 0.1250 0.1111 0.1000 0.0000
0.1667 0.1429 0.1250 0.1111 0.1000 0.0909 0.0000
Berikut solusi dari SPL :
x0 = 0.0040
x1 = -0.1190
x2 = 0.8333
x3 = -2.2222
x4 = 2.5000
```

**Gambar 4.7.b** Solusi Matrix Hilbert 1 (n = 6)



b. Metode Gauss Jordan

Berikut adalah eksperimen program menggunakan metode eliminasi Gauss untuk persamaan - persamaan berikut.

1. Persamaan SPL 5

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

**Gambar 4.8.a** Persamaan SPL 5

```
Masukkan tujuan Anda : 1
==Pilih metode perhitungan SPL==
--> 1. Eliminasi Gauss
--> 2. Eliminasi Gauss Jordan
--> 3. Metode matriks balikan
--> 4. Aturan Cramer
Masukkan tujuan Anda : 2
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 2
Masukkan nama file : spl5.txt
Kalkulasi matrix gagal, progress tidak disimpan.
<Tekan ENTER untuk melanjutkan>
```

**Gambar 4.8.b** Hasil perhitungan persamaan SPL 5

Program tidak dapat melakukan perhitungan pada SPL 5 karena program hanya dapat melakukan perhitungan SPL dengan Matriks ukuran  $(n \leq m-1) \times m$ , hal ini merupakan salah satu keterbatasan program SPL yang kami buat.

2. Persamaan SPL 6

$$\begin{aligned} 8x_1 + x_2 + 3x_3 + 2x_4 &= 0 \\ 2x_1 + 9x_2 - x_3 - 2x_4 &= 1 \\ x_1 + 3x_2 + 2x_3 - x_4 &= 2 \\ x_1 + 6x_3 + 4x_4 &= 3 \end{aligned}$$

**Gambar 4.9.a** Persamaan SPL 6

```

Masukkan tujuan Anda : 1
==Pilih metode perhitungan SPL==
--> 1. Eliminasi Gauss
--> 2. Eliminasi Gauss Jordan
--> 3. Metode matriks balikan
--> 4. Aturan Crammer
Masukkan tujuan Anda : 2
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 2
Masukkan nama file : spl6.txt
Menghitung SPL dengan metode GAUSS JORDAN.
8.0000 0.0000 3.0000 2.0000 0.0000
2.0000 9.0000 -1.0000 -2.0000 1.0000
1.0000 3.0000 2.0000 -1.0000 2.0000
1.0000 0.0000 6.0000 4.0000 3.0000
Berikut solusi dari SPL :
x0 = -0.2000
x1 = 0.1765
x2 = 0.7059
x3 = -0.2588
<Tekan ENTER untuk melanjutkan>

```

**Gambar 4.9.b** Solusi persamaan SPL 6

### 3. Persamaan SPL 7

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

**Gambar 4.10.a** Persamaan SPL 7

```

Masukkan tujuan Anda : 1
==Pilih metode perhitungan SPL==
--> 1. Eliminasi Gauss
--> 2. Eliminasi Gauss Jordan
--> 3. Metode matriks balikan
--> 4. Aturan Crammer
Masukkan tujuan Anda : 2
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 2
Masukkan nama file : spl7.txt
Kalkulasi matrix gagal, progress tidak disimpan.
<Tekan ENTER untuk melanjutkan>

```

**Gambar 4.10.b** Hasil perhitungan persamaan SPL 7

#### 4. Persamaan SPL 8

$$\begin{aligned}
 \text{A: } m_{A_{in}} + Q_{BA}x_B - Q_{AB}x_A - Q_{AC}x_A &= 0 \\
 \text{B: } Q_{AB}x_A - Q_{BA}x_B - Q_{BC}x_B &= 0 \\
 \text{C: } m_{C_{in}} + Q_{AC}x_A + Q_{BC}x_B - Q_{C_{out}}x_C &= 0
 \end{aligned}$$

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  $Q_{C_{out}} = 150 \text{ m}^3/\text{s}$  dan  $m_{A_{in}} = 1300$  dan  $m_{C_{in}} = 200 \text{ mg/s}$ .

**Gambar 4.11.a** Persamaan SPL 8

```

Masukkan tujuan Anda : 1
==Pilih metode perhitungan SPL==
--> 1. Eliminasi Gauss
--> 2. Eliminasi Gauss Jordan
--> 3. Metode matriks balikan
--> 4. Aturan Crammer
Masukkan tujuan Anda : 2
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 2
Masukkan nama file : spl8.txt
Menghitung SPL dengan metode GAUSS JORDAN.
60.0000 -40.0000 -80.0000 -1300.0000
40.0000 -60.0000 20.0000 0.0000
80.0000 20.0000 -150.0000 -200.0000
Berikut solusi dari SPL :
x0 = 42.6271
x1 = 38.1356
x2 = 29.1525
<Tekan ENTER untuk melanjutkan>

```

**Gambar 4.11.b** Solusi persamaan SPL 8

## 5. Matrix Hilbert 2

```
1.0 0.5 0.333333333333333 0.25 0.2 0.166666666666666 0.14285714285714285 0.125 0.111111111111111 0.1 1
0.5 0.333333333333333 0.25 0.2 0.166666666666666 0.14285714285714285 0.125 0.111111111111111 0.1 0.090909090909091 0
0.333333333333333 0.25 0.2 0.166666666666666 0.14285714285714285 0.125 0.111111111111111 0.1 0.090909090909091 0.083333333333333 0
0.25 0.2 0.166666666666666 0.14285714285714285 0.125 0.111111111111111 0.1 0.090909090909091 0.083333333333333 0.07692307692307693 0
0.2 0.166666666666666 0.14285714285714285 0.125 0.111111111111111 0.1 0.090909090909091 0.083333333333333 0.07692307692307693 0.07142857142857142 0
0.166666666666666 0.14285714285714285 0.125 0.111111111111111 0.1 0.090909090909091 0.083333333333333 0.07692307692307693 0.07142857142857142 0.0666666666666667 0
0.14285714285714285 0.125 0.111111111111111 0.1 0.090909090909091 0.083333333333333 0.07692307692307693 0.07142857142857142 0.0666666666666667 0.0625 0
0.125 0.111111111111111 0.1 0.090909090909091 0.083333333333333 0.07692307692307693 0.07142857142857142 0.0666666666666667 0.0625 0.058823529411764705 0
0.111111111111111 0.1 0.090909090909091 0.083333333333333 0.07692307692307693 0.07142857142857142 0.0666666666666667 0.0625 0.058823529411764705 0.0555555555555555 0
0.1 0.090909090909091 0.083333333333333 0.07692307692307693 0.07142857142857142 0.0666666666666667 0.0625 0.058823529411764705 0.0555555555555555 0.05263157894736842
```

### Gambar 4.12.a Persamaan Hilbert 2

```
Masukkan tujuan Anda : 1
==Pilih metode perhitungan SPL==
--> 1. Eliminasi Gauss
--> 2. Eliminasi Gauss Jordan
--> 3. Metode matriks balikan
--> 4. Aturan Crammer
Masukkan tujuan Anda : 2
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 2
Masukkan nama file : hilbert2.txt
Menghitung SPL dengan metode GAUSS JORDAN.
1.0000 0.5000 0.3333 0.2500 0.2000 0.1667 0.1429 0.1250 0.1111 0.1000 1.0000
0.5000 0.3333 0.2500 0.2000 0.1667 0.1429 0.1250 0.1111 0.1000 0.0909 0.0000
0.3333 0.2500 0.2000 0.1667 0.1429 0.1250 0.1111 0.1000 0.0909 0.0833 0.0000
0.2500 0.2000 0.1667 0.1429 0.1250 0.1111 0.1000 0.0909 0.0833 0.0769 0.0000
0.2000 0.1667 0.1429 0.1250 0.1111 0.1000 0.0909 0.0833 0.0769 0.0714 0.0000
0.1667 0.1429 0.1250 0.1111 0.1000 0.0909 0.0833 0.0769 0.0714 0.0667 0.0000
0.1429 0.1250 0.1111 0.1000 0.0909 0.0833 0.0769 0.0714 0.0667 0.0625 0.0000
0.1250 0.1111 0.1000 0.0909 0.0833 0.0769 0.0714 0.0667 0.0625 0.0588 0.0000
0.1111 0.1000 0.0909 0.0833 0.0769 0.0714 0.0667 0.0625 0.0588 0.0556 0.0000
0.1000 0.0909 0.0833 0.0769 0.0714 0.0667 0.0625 0.0588 0.0556 0.0526 0.0000
Berikut solusi dari SPL :
x0 = 99.9963
x1 = -4949.6828
x2 = 79193.2195
x3 = -600538.1930
x4 = 2522224.5682
x5 = -6305486.4995
x6 = 762560.5909
x7 = 169.9749
x8 = 19.9970
x9 = 1.0000
<Tekan ENTER untuk melanjutkan>
```

### Gambar 4.12.b Solusi persamaan Hilbert 2

c. Aturan Crammer

```
Masukkan tujuan Anda : 1
==Pilih metode perhitungan SPL==
--> 1. Eliminasi Gauss
--> 2. Eliminasi Gauss Jordan
--> 3. Metode matriks balikan
--> 4. Aturan Crammer
Masukkan tujuan Anda : 4
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan jumlah baris : 4
Masukkan jumlah kolom : 5
Masukkan matrix :
2 4 3 2 2
1 1 1 1 1
10 12 14 5 2
2 3 9 1 8
Berikut Solusi dari SPL menggunakan kaidah Cramer :
x0 = -2.0169
x1 = 0.6441
x2 = 1.2881
x3 = 2.3729
<Tekan ENTER untuk melanjutkan>
```

**Gambar 4.12.1** Perhitungan SPL dengan aturan cramer

d. Metode Inverse

```
Masukkan tujuan Anda : 1
==Pilih metode perhitungan SPL==
--> 1. Eliminasi Gauss
--> 2. Eliminasi Gauss Jordan
--> 3. Metode matriks balikan
--> 4. Aturan Crammer
Masukkan tujuan Anda : 3
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan jumlah baris : 3
Masukkan jumlah kolom : 4
Masukkan matrix :
5 4 3 9
5 8 3 3
1 1 1 1
Berikut Solusi dari SPL menggunakan metode inverse :
x0 = 3.7500
x1 = -1.5000
x2 = -1.2500
<Tekan ENTER untuk melanjutkan>
```

**Gambar 4.12.2** Perhitungan SPL dengan metode inverse

### C. Inverse

#### a. Metode Operasi Baris Elementer

```
Masukkan tujuan Anda : 3
==Pilih metode balikan matriks==
--> 1. Invers reduksi
--> 2. Invers determinan kofaktor
Masukkan tujuan Anda : 1
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan jumlah kolom : 5
Masukkan matrix :
1 8 9 2 10
5 2 3 9 78
9 7 6 5 4
4 3 6 8 1
4 3 9 8 1
Menghitung inverse matriks dengan reduksi baris.
Berikut hasil matriks balikan :
-0.1071 0.0069 0.1533 -0.2380 0.1613
0.1338 -0.0170 0.0013 0.3416 -0.3567
0.0000 0.0000 -0.0000 -0.3333 0.3333
0.0030 0.0014 -0.0770 0.3688 -0.1980
0.0031 0.0127 -0.0010 -0.0232 0.0088
<Tekan ENTER untuk melanjutkan>
```

**Gambar 4.13** Perhitungan matriks balikan dengan reduksi eselon

#### b. Metode Minor Kofaktor

```
Masukkan tujuan Anda : 3
==Pilih metode balikan matriks==
--> 1. Invers reduksi
--> 2. Invers determinan kofaktor
Masukkan tujuan Anda : 2
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan jumlah kolom : 3
Masukkan matrix :
2 5 4
9 9 2
2 4 1
Menghitung inverse matriks dengan kofaktor.
Berikut hasil matriks inverse balikan :
0.0204 0.2245 -0.5306
-0.1020 -0.1224 0.6531
0.3673 0.0408 -0.5510
<Tekan ENTER untuk melanjutkan>
```

**Gambar 4.14** Perhitungan matriks balikan dengan metode minor kofaktor

#### D. Determinan

##### a. Metode Reduksi Eselon

```
==Pilih metode perhitungan Determinan==
--> 1. Ekspansi Kofaktor
--> 2. Reduksi baris Matriks
Masukkan tujuan Anda : 1
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan jumlah kolom : 4
Masukkan matrix :
2 3 4 2
100 9 8 3
4 3 2 7
2 3 5 19
Mencari determinan matriks dengan kofaktor.
2.0000 3.0000 4.0000 2.0000
100.0000 9.0000 8.0000 3.0000
4.0000 3.0000 2.0000 7.0000
2.0000 3.0000 5.0000 19.0000
Determinan : 10608.0000
<Tekan ENTER untuk melanjutkan>
```

**Gambar 4.15** Perhitungan determinan Matriks dengan metode reduksi eselon

##### b. Metode Minor Kofaktor

```
Masukkan tujuan Anda : 2
==Pilih metode perhitungan Determinan==
--> 1. Ekspansi Kofaktor
--> 2. Reduksi baris Matriks
Masukkan tujuan Anda : 1
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan jumlah kolom : 3
Masukkan matrix :
1 8 2
4 6 9
9 7 5
Mencari determinan matriks dengan kofaktor.
1.0000 8.0000 2.0000
4.0000 6.0000 9.0000
9.0000 7.0000 5.0000
Determinan : 403.0000
<Tekan ENTER untuk melanjutkan>
```

**Gambar 4.16** Perhitungan determinan Matriks dengan metode minor kofaktor

## E. Interpolasi Polinomial

- 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:

$$\begin{array}{ll} x = 0.2 & f(x) = ? \\ x = 0.55 & f(x) = ? \\ x = 0.85 & f(x) = ? \\ x = 1.28 & f(x) = ? \end{array}$$

**Gambar 4.17** Tabel test case interpolasi polinomial

```
| Selamat datang di program BiggerWeapons !
| Silahkan masukan nama sesi : tes
| Initializing - - -
| Sesi : tes 10-05-23 [0]
| =====MENU UTAMA=====
| 1. Sistem Persamaan Linier
| 2. Determinan
| 3. Matriks balikan
| 4. Interpolasi Polinom
| 5. Interpolasi Bicubic Spline
| 6. Regresi Linier Berganda
| 7. Keluar
| Masukkan tujuan Anda : 4
| Pilih jenis masukan
| ---> 1. Masukan dari keyboard
| ---> 2. Masukan dari file
| Masukkan tujuan Anda : 1
| Masukkan jumlah n titik : 7
| Masukkan titik sebanyak 7 :
| x0 : 0.1
| y0 : 0.003
| x1 : 0.3
| y1 : 0.067
| x2 : 0.5
| y2 : 0.148
| x3 : 0.7
| y3 : 0.248
| x4 : 0.9
| y4 : 0.370
| x5 : 1.1
| y5 : 0.518
| x6 : 1.3
| y6 : 0.697
| Masukkan nilai x yang ingin ditaksir :
| 0.2
| f(x) = -1.4380360689856027E-14x^6+6.174727131032124E-14x^5+0.026041666666564076x^4+8.257232547210652E-14x^3+0.19739583333330044x^2+0.2400000000000005
| 9x+-0.02297656250000034,f(0.2) = 0.032960937500000044
```

**Gambar 4.18** Hasil untuk  $x = 0,2$



```

Sesi : tes 10-05-23 [1]
=====MENU UTAMA=====
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic Spline
6. Regresi Linier Berganda
7. Keluar
Masukkan tujuan Anda : 4
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan jumlah n titik : 7
Masukkan titik sebanyak 7 :
x0 : 0.1
y0 : 0.003
x1 : 0.3
y1 : 0.067
x2 : 0.5
y2 : 0.148
x3 : 0.7
y3 : 0.248
x4 : 0.9
y4 : 0.370
x5 : 1.1
y5 : 0.518
x6 : 1.3
y6 : 0.697
Masukkan nilai x yang ingin ditaksir :
0.55
f(x) = -1.4380360689856027E-14x^6+6.174727131032124E-14x^5+0.026041666666564076x^4+8.257232547210652E-14x^3+0.19739583333330044x^2+0.2400000000000059x+0.02297656250
000034,f(0.55) = 0.17111865234375007

```

**Gambar 4.19** Hasil untuk  $x = 0,55$

```

=====MENU UTAMA=====
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic Spline
6. Regresi Linier Berganda
7. Keluar
Masukkan tujuan Anda : 4
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan jumlah n titik : 7
Masukkan titik sebanyak 7 :
x0 : 0.1
y0 : 0.003
x1 : 0.3
y1 : 0.067
x2 : 0.5
y2 : 0.148
x3 : 0.7
y3 : 0.248
x4 : 0.9
y4 : 0.370
x5 : 1.1
y5 : 0.518
x6 : 1.3
y6 : 0.697
Masukkan nilai x yang ingin ditaksir :
0.85
f(x) = -1.4380360689856027E-14x^6+6.174727131032124E-14x^5+0.026041666666564076x^4+8.257232547210652E-14x^3+0.19739583333330044x^2+0.2400000000000059x+0.02297656250
000034,f(0.85) = 0.33723583984375
<tekan ENTER untuk melanjutkan>

```

**Gambar 4.20** Hasil untuk  $x = 0,85$

```

=====MENU UTAMA=====
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic Spline
6. Regresi Linier Berganda
7. Keluar
Masukkan tujuan Anda : 4
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan jumlah n titik : 7
Masukkan titik sebanyak 7 :
x0 : 0.1
y0 : 0.003
x1 : 0.3
y1 : 0.067
x2 : 0.5
y2 : 0.148
x3 : 0.7
y3 : 0.248
x4 : 0.9
y4 : 0.370
x5 : 1.1
y5 : 0.518
x6 : 1.3
y6 : 0.697
Masukan nilai x yang ingin ditaksir :
1.28
f(x) = -1.4380360689856027E-14x^6+6.174727131032124E-14x^5+0.0260416666666564076x^4+8.257232547210652E-14x^3+0.19739583333330044x^2+0.2400000000000059x+-0.02297656250
000034,f(1.28) = 0.6775418375
<Tekan ENTER untuk melanjutkan>

```

**Gambar 4.21** Hasil untuk  $x = 1,28$

- 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

Tanggal (desimal) adalah tanggal yang sudah diolah ke dalam bentuk desimal 3 angka di belakang koma dengan memanfaatkan perhitungan sebagai berikut:

**Gambar 4.22** Tabel test case kasus covid

tanggal(desimal) sebagai berikut:

$$\text{Tanggal (desimal)} = 6 + (17/30) = 6,567$$

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

- 16/07/2022
- 10/08/2022
- 05/09/2022
- Masukan user lainnya berupa **tanggal (desimal)** yang sudah diolah dengan asumsi prediksi selalu dilakukan untuk tahun 2022.

a.  $16/07/2022 = 7,516$

```
Pilih jenis masukan
--> 1. Masukan dari keyboard
--> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan jumlah n titik : 10
Masukkan titik sebanyak 10 :
x0 : 6.567
y0 : 12624
x1 : 7
y1 : 21807
x2 : 7.258
y2 : 38391
x3 : 7.451
y3 : 54517
x4 : 7.548
y4 : 51952
x5 : 7.839
y5 : 28228
x6 : 8.161
y6 : 35764
x7 : 8.484
y7 : 20813
x8 : 8.709
y8 : 12408
x9 : 9
y9 : 10534
Masukkan nilai x yang ingin ditaksir :
7.516
f(x) = -141121.13609816754x^9+9381819.863749687x^8+2.7575487443719405E8x^7+4.700910325686923E9x^6+5.119154187746152E10x^5+3.690152113407484E11x^4+1.7592169642251704E12x^3+5.342211415807217E12x^2+-9.362517627045893E12x+7.200424656191251E12,f(7.516) = 53537.853515625
<Tekan ENTER untuk melanjutkan>
```

**Gambar 4.23** Hasil untuk poin a

b.  $10/08/2022 = 8,322$

```
Masukkan tujuan Anda : 4
Pilih jenis masukan
--> 1. Masukan dari keyboard
--> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan jumlah n titik : 10
Masukkan titik sebanyak 10 :
x0 : 6.567
y0 : 12624
x1 : 7
y1 : 21807
x2 : 7.258
y2 : 38391
x3 : 7.451
y3 : 54517
x4 : 7.548
y4 : 51952
x5 : 7.839
y5 : 28228
x6 : 8.161
y6 : 35764
x7 : 8.484
y7 : 20813
x8 : 8.709
y8 : 12408
x9 : 9
y9 : 10534
Masukkan nilai x yang ingin ditaksir :
8.322
f(x) = -141121.13609816754x^9+9381819.863749687x^8+2.7575487443719405E8x^7+4.700910325686923E9x^6+5.119154187746152E10x^5+3.690152113407484E11x^4+1.7592169642251704E12x^3+5.342211415807217E12x^2+-9.362517627045893E12x+7.200424656191251E12,f(8.322) = 36344.8046875
<Tekan ENTER untuk melanjutkan>
```

**Gambar 4.24** Hasil untuk poin b

c.  $05/09/2022 = 9.166$

```

Masukkan tujuan Anda : 4
Pilih jenis masukan
--> 1. Masukan dari keyboard
--> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan jumlah n titik : 10
Masukkan titik sebanyak 10 :
x0 : 6.567
y0 : 12624
x1 : 7
y1 : 21807
x2 : 7.258
y2 : 38391
x3 : 7.451
y3 : 54517
x4 : 7.548
y4 : 51952
x5 : 7.839
y5 : 28228
x6 : 8.161
y6 : 35764
x7 : 8.484
y7 : 20813
x8 : 8.709
y8 : 12408
x9 : 9
y9 : 10534
Masukan nilai x yang ingin ditaksir :
9.166
f(x) = -141121.13609816754x^9+9381819.863749687x^8+2.7575487443719405E8x^7+4.700910325686923E9x^6+5.119154187746152E10x^5+3.690152113407484E11x^4+1.75921696422517
04E12x^3+5.342211415807217E12x^2+9.362517627045893E12x+7.200424656191251E12,f(9.166) = -659187.90625
<tekan ENTER untuk melanjutkan>

```

**Gambar 4.25** Hasil untuk poin c

d.  $10/10/2022 = 10.322$

```

Masukkan tujuan Anda : 4
Pilih jenis masukan
--> 1. Masukan dari keyboard
--> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan jumlah n titik : 10
Masukkan titik sebanyak 10 :
x0 : 6.567
y0 : 12624
x1 : 7
y1 : 21807
x2 : 7.258
y2 : 38391
x3 : 7.451
y3 : 54517
x4 : 7.548
y4 : 51952
x5 : 7.839
y5 : 28228
x6 : 8.161
y6 : 35764
x7 : 8.484
y7 : 20813
x8 : 8.709
y8 : 12408
x9 : 9
y9 : 10534
Masukan nilai x yang ingin ditaksir :
10.322
f(x) = -141121.13609816754x^9+9381819.863749687x^8+2.7575487443719405E8x^7+4.700910325686923E9x^6+5.119154187746152E10x^5+3.690152113407484E11x^4+1.75921696422517
04E12x^3+5.342211415807217E12x^2+9.362517627045893E12x+7.200424656191251E12,f(10.322) = -8.2415059409375E8
<tekan ENTER untuk melanjutkan>

```

**Gambar 4.26** Hasil untuk poin d

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  $h = (2 - 0)/5 = 0.4$ .

**Gambar 4.27** Test case interpolasi bagian c

```

=====MENU UTAMA=====
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic Spline
6. Regresi Linier Berganda
7. Keluar
Masukkan tujuan Anda : 4
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan jumlah n titik : 6
Masukkan titik sebanyak 6 :
x0 : 0
y0 : 0
x1 : 0.4
y1 : 0.418
x2 : 0.8
y2 : 0.507
x3 : 1.2
y3 : 0.560
x4 : 1.6
y4 : 0.583
x5 : 2.0
y5 : 0.576
Masukan nilai x yang ingin ditaksir :
0.5
f(x) = 0.22867838541666619x^5+1.3818359374999967x^4+3.1647135416666607x^3+3.4976562499999955x^2+2.0202916666666657x+0.0,f(0.5) = 0.4521024169921876
<Tekan ENTER untuk melanjutkan>

```

**Gambar 4.28** Hasil untuk test case interpolinom bagian c

#### F. Interpolasi Bicubic Spline

```

=====MENU UTAMA=====
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic Spline
6. Regresi Linier Berganda
7. Keluar
Masukkan tujuan Anda : 5
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan matriks 4x4 :
21 98 125 153
51 101 161 59
0 42 72 210
16 12 81 96
Masukan nilai x,y yang ingin ditaksir
x : 0
y : 0
f(0.0,0.0) = 21.0
<Tekan ENTER untuk melanjutkan>

```

**Gambar 4.29** Hasil untuk test case  $x = 0, y = 0$

```
Sesi : tes 10-05-23 [0]
=====MENU UTAMA=====
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic Spline
6. Regresi Linier Berganda
7. Keluar
Masukkan tujuan Anda : 5
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan matriks 4x4 :
21 98 125 153
51 101 161 59
0 42 72 210
16 12 81 96
Masukan nilai x,y yang ingin ditaksir
x : 0.5
y : 0.5
f(0.5,0.5) = 87.796875
<Tekan ENTER untuk melaniutkan>
```

**Gambar 4.30** Hasil untuk test case  $x = 0.5$ ,  $y = 0.5$

```
Sesi : tes 10-05-23 [1]
=====MENU UTAMA=====
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic Spline
6. Regresi Linier Berganda
7. Keluar
Masukkan tujuan Anda : 5
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan matriks 4x4 :
21 98 125 153
51 101 161 59
0 42 72 210
16 12 81 96
Masukan nilai x,y yang ingin ditaksir
x : 0.25
y : 0.75
f(0.25,0.75) = 117.732177734375
<Tekan ENTER untuk melanjutkan>
```

**Gambar 4.31** Hasil untuk test case  $x = 0.25$ ,  $y = 0.75$

```

=====MENU UTAMA=====
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic Spline
6. Regresi Linier Berganda
7. Keluar
Masukkan tujuan Anda : 5
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan matriks 4x4 :
21 98 125 153
51 101 161 59
0 42 72 210
15 12 81 96
Masukan nilai x,y yang ingin ditaksir
x : 0.1
y : 0.9
f(0.1,0.9) = 128.574458
<Tekan ENTER untuk melanjutkan>

```

**Gambar 4.32** Hasil untuk test case  $x = 0.1$ ,  $y = 0.9$

#### G. Regresi Linier Berganda

Masukan dari keyboard :



```

=====MENU UTAMA=====
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic Spline
6. Regresi Linier Berganda
7. Keluar
Masukkan tujuan Anda : 6
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 1
Masukkan jumlah baris : 5
Masukkan jumlah kolom : 4
Masukkan matrix :
8 7 5 9
6 4 8 6
7 6 3 7
8 7 5 3
2 4 7 5
Masukkan nilai yang ingin ditaksir :
3
6
7
Berikut adalah hasil kalkulasi regresi linier berganda dari data yang diberikan:
 $f(x) = 9.3143 + 0.3429b_0 + -0.6000b_1 + -0.3714b_2$ 
Sedangkan hasil taksiran nilai yang dicari adalah :
 $f(3.0, 6.0, 7.0) = 4.1428571428570855$ 

```

**Gambar 4.33** Hasil percobaan dengan masukan keyboard (hanya digunakan untuk mengecek apakah program dapat dijalankan sesuai dengan yang seharusnya)

Studi kasus :

Table 12.1: Data for Example 12.1

Nitrous Oxide, $y$	Humidity, $x_1$	Temp., $x_2$	Pressure, $x_3$	Nitrous Oxide, $y$	Humidity, $x_1$	Temp., $x_2$	Pressure, $x_3$
0.90	72.4	76.3	29.18	1.07	23.2	76.8	29.38
0.91	41.6	70.3	29.35	0.94	47.4	86.6	29.35
0.96	34.3	77.1	29.24	1.10	31.5	76.9	29.63
0.89	35.1	68.0	29.27	1.10	10.6	86.3	29.56
1.00	10.7	79.0	29.78	1.10	11.2	86.0	29.48
1.10	12.9	67.4	29.39	0.91	73.3	76.3	29.40
1.15	8.3	66.8	29.69	0.87	75.4	77.9	29.28
1.03	20.1	76.9	29.48	0.78	96.6	78.7	29.29
0.77	72.2	77.7	29.09	0.82	107.4	86.8	29.03
1.07	24.0	67.7	29.60	0.95	54.9	70.9	29.37

Source: Charles T. Hare, "Light-Duty Diesel Emission Correction Factors for Ambient Conditions," EPA-600/2-77-116. U.S. Environmental Protection Agency.

**Gambar 4.34** Data studi kasus yang ingin diregresi

Masukan dari file :

```
=====MENU UTAMA=====
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic Spline
6. Regresi Linier Berganda
7. Keluar
Masukkan tujuan Anda : 6
Pilih jenis masukan
---> 1. Masukan dari keyboard
---> 2. Masukan dari file
Masukkan tujuan Anda : 2
Masukkan nama file : reg.txt
Berikut adalah hasil kalkulasi regresi linier berganda dari data yang diberikan:
 $f(x) = -3.5078 + -0.0026b0 + 0.0008b1 + 0.1542b2$ 
Sedangkan hasil taksiran nilai yang dicari adalah :
 $f(0.5, 76.0, 29.3) = 1.06837126815791$ 
```

**Gambar 4.35** Hasil perhitungan regresi data studi kasus dengan masukan file

## BAB 5

### Kesimpulan

#### A. Hasil yang Dicapai

Sistem Persamaan Linier banyak dijumpai di persoalan sehari-hari, dan terdapat beberapa metode yang bisa digunakan untuk menyelesaikannya, yaitu eliminasi Gauss, eliminasi Gauss-Jordan, matriks balikan, dan kaidah Cramer. Lebih lanjut, penyelesaian SPL juga dapat digunakan untuk melakukan interpolasi polinomial, regresi linier berganda, dan interpolasi bicubic spline.

Pada tugas besar ini, kami telah membuat implementasi sistem persamaan linier dengan membuat library perhitungan matriks untuk menyelesaikan persoalan-persoalan tersebut.

#### B. Saran Pengembangan

Terdapat beberapa saran yang dapat disampaikan terkait pengembangan program, yaitu :

- Melakukan diskusi terkait struktur dari program, seperti bagaimana dekomposisi persoalan atau spesifikasi pada program yang akan dibuat.
- Membuat program SPL yang lebih baik dalam menangani kasus-kasus error dalam perhitungan SPL
- Lebih baik lagi dalam melakukan perencanaan pengerjaan tugas

#### C. Refleksi

Ada banyak hal yang bisa kami dapatkan ketika merefleksikan diri setelah pengerjaan tugas besar ini. Inisiasi dari tiap anggota terbilang cukup baik, karena masing-masing dari kami sadar bahwa kami masih butuh banyak belajar, sehingga untuk meminimalisasi ke-chaos-an di dekat deadline, kami semua segera mengerjakan tugas besar ini, setidaknya berprogres setiap hari.

Link Repository : <https://github.com/ZakiYudhistira/Algeo01-22018>

## Daftar Referensi

Anton, Howard, and Chris Rorres. *Elementary Linear Algebra: Applications Version*. Wiley, 2013.

“Bicubic Spline -- from Wolfram MathWorld.” *Wolfram MathWorld*,  
<https://mathworld.wolfram.com/BicubicSpline.html>.

Hobart and William Smith Colleges. *Introduction to Programming Using Java*. 5.02 ed., New York, Hobart and William Smith Colleges, 1996.

W3School. “Java Tutorial.” *W3Schools*, <https://www.w3schools.com/java/>.