

# **LIBRARY ALJABAR LINIER DALAM BAHASA JAVA DENGAN FUNGSI SISTEM PERSAMAAN VARIABEL, DETERMINAN, DAN APLIKASINYA**

## **LAPORAN TUGAS BESAR**

**Disusun untuk memenuhi salah satu tugas besar  
mata kuliah Aljabar Linier dan Geometri  
IF2123-03**



**Oleh**

<b>Kelvin Rayhan Alkarim</b>	<b>13521005</b>
<b>Bernardus Willson</b>	<b>13521021</b>
<b>Kenny Benaya Nathan</b>	<b>13521023</b>

**Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
INSTITUT TEKNOLOGI BANDUNG  
2022**

## **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 ( $x = A^{-1}b$ ), dan kaidah Cramer (khusus untuk SPL dengan n peubah dan n persamaan). Solusi sebuah SPL mungkin tidak ada, banyak (tidak berhingga), atau hanya satu (unik/tunggal).

Di dalam Tugas Besar 1 ini, 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.

## **BAB II**

### **TEORI SINGKAT**

#### **1. Metode Eleminasi Gauss**

Eleminasi Gauss adalah Algoritma yang digunakan untuk menyelesaikan sistem persamaan linear. Algoritma ini terdiri dari serangkaian operasi yang dilakukan pada matriks koefisien dari sistem persamaan tersebut. Walau akan mengubah bentuk matriks, operasi-operasi tersebut tidak akan mengubah solusi dari sistem persamaan. Hal ini memungkinkan matriks koefisien dibentuk menjadi sebuah matriks segitiga atas, sehingga solusi sistem persamaan dapat ditentukan dengan cukup melakukan eliminasi variabel secara berulang. Eliminasi Gauss juga dapat digunakan untuk menghitung rank dari matriks, determinan dari matriks persegi, dan invers dari matriks nonsingular. Metode ini dinamai dari matematikawan Carl Friedrich Gauss (1777–1855), walaupun beberapa kasus khusus dari metode ini — tapi tanpa dilengkapi bukti — sudah dikenal oleh matematikawan Tionghoa semenjak tahun 179 M.

#### **2. Metode Eleminasi Gauss-Jordan**

Eliminasi Gauss-Jordan adalah pengembangan dari eliminasi Gauss yang hasilnya lebih sederhanalagi. Caranya adalah dengan meneruskan operasi baris dari eliminasi Gauss sehingga menghasilkan matriks yang Eselon-baris. Ini juga dapat digunakan sebagai salah satu metode penyelesaian persamaan linear dengan menggunakan matriks. Metode ini digunakan untuk mencari invers dari sebuah matriks.

#### **3. Determinan**

Determinan Matriks adalah nilai yang bisa dihitung dari unsur-unsur matriks. Ada beberapa cara dalam mendapat determinan matriks ini. Salah Satunya dengan menggunakan kofaktor.

#### **4. Matriks balikan**

Matriks balikan banyak dipakai dalam pengolahan matriks. Misalnya dalam pengukuran statistik, pencocokan fungsi pada data hasil pengamatan menggunakan metode kuadrat terkecil (least square). Di sini, nilai matriks memberikan informasi tentang galat mutlak yang dikandung data. Selain itu, matriks balikan juga dapat dipakai untuk menghitung solusi sistem persamaan lanjar (akan dibahas pada metode matriks balikan). Akan ditunjukkan juga bahwa matriks balikan dapat diperoleh dengan metode eliminasi Gauss-Jordan.

#### **5. Matriks kofaktor**

Sesuai nama metodenya, kofaktor, berarti ada sebuah faktor, dalam hal ini adalah faktor pengali yang sama. Jika ditelaah kembali cara ataupun rumus sebelumnya, terlihat bahwa suku-suku determinan tersebut mempunyai kesamaan beberapa faktornya. Pada ukuran  $2 \times 2$ , sudah tidak bisa difaktorkan kembali, tetapi pada ordo  $3 \times 3$  faktor-faktor yang sama bisa "dikeluarkan".

## **6. Matriks adjoint**

Pengertian dari matriks itu sendiri adalah sekumpulan bilangan yang akan disusun secara baris atau kolom di dalam suatu tanda kurung. Bilangan yang membentuk suatu matriks, akan disebut sebagai elemen-elemen matriks. Adjoin dari matriks persegi  $A = [a_{ij}]_{n \times n}$  didefinisikan sebagai transpos dari matriks  $[A_{ij}]_{n \times n}$  di mana  $A_{ij}$  adalah kofaktor dari elemen  $a_{ij}$ . Adjoin dari matriks  $A$  dilambangkan dengan  $\text{adj } A$ . Untuk mencari adjoint dari sebuah matriks, pertama-tama cari kofaktor dari matriks yang diberikan. Kemudian temukan transpos dari matriks kofaktor tersebut.

## **7. Kaidah cramer**

Kaidah Cramer adalah rumus yang dapat digunakan untuk menyelesaikan sistem persamaan linear dengan banyak persamaan sama dengan banyak variabel, dan berlaku ketika sistem tersebut memiliki solusi yang tunggal. Rumus ini menyatakan solusi dengan menggunakan determinan matriks koefisien (dari sistem persamaan) dan determinan matriks lain yang diperoleh dengan mengganti salah satu kolom matriks koefisien dengan vektor yang berada sebelah kanan persamaan. Metode ini dinamai dari matematikawan Swiss Gabriel Cramer (1704–1752), yang pada tahun 1750 menerbitkan kaidah ini untuk sebarang banyaknya variabel,[1][2] walau Colin Maclaurin juga menerbitkan kasus khusus dari kaidah ini pada tahun 1748[3] (dan mungkin ia sudah mengetahuinya sejak 1729)

## **8. Interpolasi Polinomial**

Interpolasi polinomial merupakan teknik interpolasi dengan mengasumsikan pola data yang kita miliki mengikuti pola polinomial baik berderajat satu (linier) maupun berderajat tinggi. Interpolasi dengan metode ini dilakukan dengan terlebih dahulu membentuk persamaan polinomial. Persamaan polinomial yang terbentuk selanjutnya digunakan untuk melakukan interpolasi dari nilai yang diketahui atau ekstrapolasi (prediksi) dari nilai diluar rentang data yang diketahui.

## **9. Interpolasi Bicubic**

Interpolasi bikubik adalah perpanjangan dari interpolasi kubik (jangan dikelirukan dengan interpolasi spline kubik, metode penerapan interpolasi kubik ke kumpulan data) untuk menginterpolasi titik data pada kisi reguler dua dimensi. Permukaan interpolasi (artinya bentuk kernel, bukan gambar) lebih halus daripada permukaan yang sesuai yang diperoleh dengan interpolasi bilinear atau interpolasi tetangga terdekat. Interpolasi bikubik dapat dilakukan dengan menggunakan polinomial Lagrange, spline kubik, atau algoritma konvolusi kubik.

## **10. Regresi Linear berganda**

Regresi linier (Linier Regression) adalah salah satu teknik analisis statistik yang digunakan untuk mengestimasi hubungan fungsional di antara dua atau lebih variabel, yaitu antara variabel terikat (dependen; respon; Y) dengan satu atau lebih variabel bebas (independen variable, predictor, X). Apabila kita hanya menggunakan satu variabel bebas untuk memprediksi respons, disebut

sebagai regresi linier sederhana (simple linear regression), sedangkan apabila terdapat lebih dari 1 variabel bebas, disebut sebagai regresi linier berganda (multiple linear regression).

### **BAB III**

### **IMPLEMENTASI PUSTAKA**

Pada program Java kami, terdapat 13 class yang dibagi ke 4 folder utama source code, yaitu:

#### 1. Folder Apps

Folder Apps berisi implementasi dari algoritma - algoritma yang akan dilakukan pada matriks. Folder ini berisi 11 class: DeterminantCofact.java, DeterminantReduct.java, InverseAdj.java, InverseReduct.java, SPLGauss.java, SPLGaussJordan.java, SPLInverse.java, SPLCrammer.java, PolinomInterpolate.java, LinearRegression.java, dan BicubicInterpolate.java.

##### 1.1 DeterminantCofact.java

Class ini berisi tentang algoritma perhitungan dan penyelesaian Determinan menggunakan metode ekspansi kofaktor.

- Attributes

Attribute	Description
Private int ROW	Berisi data integer yang menampung jumlah baris dari matriks
Private int COL	Berisi data integer yang menampung jumlah kolom dari matriks
Private double Determinant	Berisi nilai determinan dari suatu matriks
Private double[][] Matrix	Menampung data angka double yang merepresentasikan isi dari matriks
Private double[][] subMatrix	Menampung data angka double yang merepresentasikan isi dari submatriks

- Constructor

Constructor(Parameter)	Description
Public Matrix(int ROW, int COL)	Konstruktor akan membuat suatu matriks dengan baris sebanyak ROW dan kolom sebanyak COL
Public subMatrix(int ROW-1, int	Konstruktor akan membuat suatu

COL-1)	matriks dengan baris sebanyak ROW-1 dan kolom sebanyak COL-1
--------	--

- Methods

Method(Parameter)	Description
Static double DeterminantC(double[][] Mat)	Mengembalikan nilai determinan dengan metode ekspansi kofaktor
Public static void functionDetCofact()	Menerima masukkan ukuran matriks, elemen matriks, dan mengeluarkan hasil fungsi DeterminantC

## 1.2 DeterminantReduct.java

Class ini berisi tentang algoritma perhitungan dan penyelesaian Determinan menggunakan metode reduksi.

- Attributes

Attribute	Description
Private int ROW	Berisi data integer yang menampung jumlah baris dari matriks
Private int COL	Berisi data integer yang menampung jumlah kolom dari matriks
Private int P	Berisi data integer pangkat yang bertambah saat ada pertukaran baris
Private double Determinant	Berisi nilai determinan dari suatu matriks
Private double[][] Matrix	Menampung data angka double yang merepresentasikan isi dari matriks

- Constructor

Constructor(Parameter)	Description
Public Matrix(int ROW, int COL)	Konstruktor akan membuat suatu matriks dengan baris sebanyak ROW dan kolom sebanyak COL

- Methods

<b>Method(Parameter)</b>	<b>Description</b>
Static void DeterminantR(double[][] Mat)	Mengeluarkan hasil fungsi DeterminanR
Static void PrintEselon(double[][] Mat)	Mengeluarkan print-printan matriks
Public static void functionDetReduct()	Menerima masukkan ukuran matriks, elemen matriks, dan menjalankan fungsi DeterminanR

### 1.3 InverseAdj.java

Class ini berisi tentang algoritma perhitungan dan penyelesaian Matriks Invers metode Adjoin.

- Attributes

<b>Attribute</b>	<b>Description</b>
Private int ROW	Berisi data integer yang menampung jumlah baris dari matriks
Private int COL	Berisi data integer yang menampung jumlah kolom dari matriks
Private double Determinant	Berisi nilai determinan dari suatu matriks
Private double[][] Matrix	Menampung data angka double yang merepresentasikan isi dari matriks

- Constructor

<b>Constructor(Parameter)</b>	<b>Description</b>
Public Cofactor(int ROW, int COL)	Konstruktor akan membuat suatu matriks dengan baris sebanyak ROW dan kolom sebanyak COL
Public Inverse(int ROW, int COL)	Konstruktor akan membuat suatu matriks dengan baris sebanyak ROW dan kolom sebanyak COL

Public mOut(int ROW, int COL)	Konstruktor akan membuat suatu matriks dengan baris sebanyak ROW dan kolom sebanyak COL
Public Matrix(int ROW, int COL)	Konstruktor akan membuat suatu matriks dengan baris sebanyak ROW dan kolom sebanyak COL

- Methods

Method(Parameter)	Description
Static void Cofactor(double[][] Mat)	Mengeluarkan hasil fungsi Cofactor
Static double[][] Transpose(double[][] Mat)	Mengembalikan nilai transpose matriks
Static double[][] Inverse(double[][] Mat)	Mengembalikan nilai invers matriks
Static double[][] MultiplyByConst(double[][] Mat, double k)	Mengembalikan nilai perkalian matriks dengan konstanta
Static void PrintInverse(double[][] Mat)	Mengeluarkan hasil matriks inverse
Public static void functionInverseAdj()	Menerima masukkan ukuran matriks, elemen matriks, dan menjalankan fungsi Inverse

#### 1.4 InverseReduct.java

Class ini berisi tentang algoritma perhitungan dan penyelesaian Matriks Invers metode Reduksi.

- Attributes

Attribute	Description
Private int ROW	Berisi data integer yang menampung jumlah baris dari matriks
Private int COL	Berisi data integer yang menampung jumlah kolom dari matriks

Private double[][] Matrix	Menampung data angka double yang merepresentasikan isi dari matriks
Private double[][] IdentityMat	Menampung data angka double yang merepresentasikan matriks identitas

- Constructor

Constructor(Parameter)	Description
Public Matrix(int ROW, int COL)	Konstruktor akan membuat suatu matriks dengan baris sebanyak ROW dan kolom sebanyak COL
Public IdentityMat(int ROW, int COL)	Konstruktor akan membuat suatu matriks dengan baris sebanyak ROW dan kolom sebanyak COL

- Methods

Method(Parameter)	Description
Static void ForwardOBE(double[][] Mat, double[][] MatAns)	Menjalankan OBE matriks secara maju
Static void BackwardOBE(double[][] Mat, double[][] MatAns)	Menjalankan OBE matriks secara mundur
Static void PrintEselon(double[][] Mat)	Mengeluarkan hasil matriks yang sudah direduksi
Public static void functionInverseReduct()	Menerima masukkan ukuran matriks, elemen matriks, dan menjalankan fungsi Forward dan Backward OBE

## 1.5 SPLGauss.java

Class ini berisi tentang algoritma perhitungan dan penyelesaian SPL dengan menggunakan metode Gauss.

- Attributes

Attribute	Description
Private int ROW	Berisi data integer yang menampung jumlah baris dari matriks

Private int COL	Berisi data integer yang menampung jumlah kolom dari matriks
Private double[][] Matrix	Menampung data angka double yang merepresentasikan isi dari matriks
Private double[][] MatrixAns	Menampung data angka double yang merepresentasikan matriks jawaban

- Constructor

Constructor(Parameter)	Description
Public Matrix(int ROW, int COL)	Konstruktor akan membuat suatu matriks dengan baris sebanyak ROW dan kolom sebanyak COL
Public MatrixAns(int ROW)	Konstruktor akan membuat suatu matriks dengan baris sebanyak ROW

- Methods

Method(Parameter)	Description
Static void ForwardOBE(double[][] Mat, double[][] MatAns)	Menjalankan OBE matriks secara maju
Static void PrintEselon(double[][] Mat)	Mengeluarkan hasil matriks yang sudah direduksi
Static void PrintSolution(double[][] Mat)	Mengeluarkan hasil SPL
Public static void functionSPLGauss()	Menerima masukkan ukuran matriks, elemen matriks, dan menjalankan fungsi Forward OBE

## 1.6 SPLGaussJordan.java

Class ini berisi tentang algoritma perhitungan dan penyelesaian SPL dengan menggunakan metode Gauss-Jordan.

- Attributes

Attribute	Description

Private int ROW	Berisi data integer yang menampung jumlah baris dari matriks
Private int COL	Berisi data integer yang menampung jumlah kolom dari matriks
Private double[][] Matrix	Menampung data angka double yang merepresentasikan isi dari matriks
Private double[][] MatrixAns	Menampung data angka double yang merepresentasikan matriks jawaban

- Constructor

Constructor(Parameter)	Description
Public Matrix(int ROW, int COL)	Konstruktor akan membuat suatu matriks dengan baris sebanyak ROW dan kolom sebanyak COL
Public MatrixAns(int ROW)	Konstruktor akan membuat suatu matriks dengan baris sebanyak ROW

- Methods

Method(Parameter)	Description
Static void ForwardOBE(double[][] Mat, double[][] MatAns)	Menjalankan OBE matriks secara maju
Static void BackwardOBE(double[][] Mat, double[][] MatAns)	Menjalankan OBE matriks secara mundur
Static void PrintEselon(double[][] Mat)	Mengeluarkan hasil matriks yang sudah direduksi
Public static void functionSPLGaussJordan()	Menerima masukkan ukuran matriks, elemen matriks, dan menjalankan fungsi Forward dan Backward OBE

## 1.7 SPLInverse.java

Class ini berisi tentang algoritma perhitungan dan penyelesaian SPL dengan menggunakan metode Inverse.

- Attributes

<b>Attribute</b>	<b>Description</b>
Private int ROW	Berisi data integer yang menampung jumlah baris dari matriks
Private int COL	Berisi data integer yang menampung jumlah kolom dari matriks
Private double[][] Matrix	Menampung data angka double yang merepresentasikan isi dari matriks
Private double[][] MatrixAns	Menampung data angka double yang merepresentasikan matriks jawaban

- Constructor

<b>Constructor(Parameter)</b>	<b>Description</b>
Public Matrix(int ROW, int COL)	Konstruktor akan membuat suatu matriks dengan baris sebanyak ROW dan kolom sebanyak COL
Public MatrixAns(int ROW)	Konstruktor akan membuat suatu matriks dengan baris sebanyak ROW

- Methods

<b>Method(Parameter)</b>	<b>Description</b>
Static double[] multiplyMatrix(double[][] Mat, double[] MatAns)	Mengembalikan hasil perkalian 2 buah matriks
Static double[][] Inverse(double[][] Mat)	Mengembalikan nilai invers matriks
Public static void functionSPLInverse()	Menerima masukkan ukuran matriks, elemen matriks, dan menjalankan fungsi perkalian dari invers matriks dan matriks jawaban

## 1.8 SPLCrammer.java

Class ini berisi tentang algoritma perhitungan dan penyelesaian SPL dengan menggunakan metode Crammer.

- Attributes

Attribute	Description
Private int ROW	Berisi data integer yang menampung jumlah baris dari matriks
Private int COL	Berisi data integer yang menampung jumlah kolom dari matriks
Private double[][] Matrix	Menampung data angka double yang merepresentasikan isi dari matriks
Private double[][] MatrixAns	Menampung data angka double yang merepresentasikan matriks jawaban

- Constructor

Constructor(Parameter)	Description
Public Matrix(int ROW, int COL)	Konstruktor akan membuat suatu matriks dengan baris sebanyak ROW dan kolom sebanyak COL
Public MatrixAns(int ROW)	Konstruktor akan membuat suatu matriks dengan baris sebanyak ROW

- Methods

Method(Parameter)	Description
Static void Crammer(double[] Mat, double[] MatAns)	Mengeluarkan hasil SPL metode Crammer
Static double DeterminantC(double[][] Mat)	Mengembalikan nilai determinan dengan metode ekspansi kofaktor
Public static void functionSPLCrammer()	Menerima masukkan ukuran matriks, elemen matriks, dan menjalankan fungsi crammer matriks

## 1.9 PolinomInterpolate.java

Class ini berisi tentang algoritma perhitungan dan penyelesaian Interpolasi Polinom.

- Attributes

<b>Attribute</b>	<b>Description</b>
Private int N	Berisi data integer yang merepresentasikan jumlah titik
Private int X	Berisi data integer yang merepresentasikan titik X
Private int Y	Berisi data integer yang merepresentasikan titik Y
Private int x	Berisi data integer yang merepresentasikan nilai x yang akan ditaksir
Private double[][] Matrix	Menampung data angka double yang merepresentasikan isi dari matriks
Private double[][] MatrixAns	Menampung data angka double yang merepresentasikan matriks jawaban

- Constructor

<b>Constructor(Parameter)</b>	<b>Description</b>
Public X(int N)	Konstruktor akan membuat suatu matriks dengan baris sebanyak N
Public Y(int N)	Konstruktor akan membuat suatu matriks dengan baris sebanyak N
Public Matrix(int N, int N)	Konstruktor akan membuat suatu matriks dengan baris sebanyak N dan kolom sebanyak N
Public MatrixAns(int N)	Konstruktor akan membuat suatu matriks dengan baris sebanyak N

- Methods

<b>Method(Parameter)</b>	<b>Description</b>
Static void ForwardOBE(double[][] Mat, double[][] MatAns)	Menjalankan OBE matriks secara maju
Static void BackwardOBE(double[][] Mat, double[][] MatAns)	Menjalankan OBE matriks secara mundur

Public static void functionPolinomInterpolate()	Menerima masukkan ukuran matriks, elemen matriks, dan menjalankan fungsi interpolasi polinom
--	--

### 1.10 LinearRegression.java

Class ini berisi tentang algoritma perhitungan dan penyelesaian Interpolasi Polinom.

- Attributes

Attribute	Description
Private int row	Berisi data integer yang menampung jumlah baris dari matriks
Private int col	Berisi data integer yang menampung jumlah kolom dari matriks
Private double[] x	Menampung data angka double yang merepresentasikan nilai x
Private double[][] matriks	Menampung data angka double yang merepresentasikan isi dari matriks
Private double[][] matrikz	Menampung data angka double yang merepresentasikan hasil

- Constructor

Constructor(Parameter)	Description
Public x(int col)	Konstruktor akan membuat suatu matriks dengan baris sebanyak col
Public matriks(int row, int col+1)	Konstruktor akan membuat suatu matriks dengan baris sebanyak row dan kolom sebanyak col+1
Public matrikz(int col+1, int col+2)	Konstruktor akan membuat suatu matriks dengan baris sebanyak col+1 dan kolom sebanyak col+2

- Methods

Method(Parameter)	Description
-------------------	-------------

Static void ForwardOBE(double[][] Mat, double[][] MatAns)	Menjalankan OBE matriks secara maju
Static void BackwardOBE(double[][] Mat, double[][] MatAns)	Menjalankan OBE matriks secara mundur
Static void seperateMatrix(double[][] matriks, int row, int col)	Melakukan pemisahan matriks augmented
Public static void functionRegression()	Menerima masukkan ukuran matriks, elemen matriks, dan menjalankan fungsi regresi

### 1.11 BicubicInterpolate.java

Class ini berisi tentang algoritma perhitungan dan penyelesaian Interpolasi Bikubik.

- Attributes

Attribute	Description
Private double[][] matrix	Menampung data angka double yang merepresentasikan isi dari matriks
Private double[] matrixs	Menampung data angka double
Private double[] matriksa	Menampung data angka double

- Constructor

Constructor(Parameter)	Description
Public matrix(int 4, int 4)	Konstruktor akan membuat suatu matriks dengan baris sebanyak 4 dan kolom sebanyak 4
Public matrixs(int 16)	Konstruktor akan membuat suatu matriks dengan baris sebanyak 16
Public matriksa(int 16)	Konstruktor akan membuat suatu matriks dengan baris sebanyak 16

- Methods

Method(Parameter)	Description

Static void ForwardOBE(double[][] Mat, double[][] MatAns)	Menjalankan OBE matriks secara maju
Static void BackwardOBE(double[][] Mat, double[][] MatAns)	Menjalankan OBE matriks secara mundur

## 2. Folder Function

Folder Function berisi fungsi-fungsi yang akan dipanggil pada main. Folder ini berisi 1 class saja yaitu inputMatrix.java.

### 2.1 inputMatrix.java

Class ini berisi tentang algoritma untuk membaca file txt.

- Attributes

Attribute	Description
Private int choice	Berisi data integer yang menampung pilihan input dari user
Private boolean status	Berisi data boolean
Private int col	Berisi data integer yang menampung jumlah baris dari matriks
Private double[][] Matrix	Menampung data angka double yang merepresentasikan isi dari matriks
Private int row	Berisi data integer yang menampung jumlah kolom dari matriks

- Constructor

Constructor(Parameter)	Description
public static double[][] readfileBicubic() throws Exception	Program untuk mengubah isi file kedalam matriks

- Methods

Method(Parameter)	Description

public static int inputMenu()	Memilih menu mana yang mau dijalankan

### 3. Folder utility

Folder ini berisi fungsi untuk menjalankan program utama. Folder ini berisi hanya 1 class saja.

#### 2.1 menu.java

- Attributes

Attribute	Description
Private int choice	Berisi data integer yang menampung input pilihan user
Private boolean status	Berisi data boolean
Private string choice	Berisi data string yang menampung input pilihan dari user

- Methods

Method(Parameter)	Description
public static void mainMenu () throws Exception	Menjalankan pemilihan menu
public static void splMenu () throws Exception	Menjalankan menu lanjutan dari operasi SPL

## BAB IV

### EKSPERIMENT

1. Temukan solusi SPL  $Ax = b$ , berikut:

a.

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

Dengan Metode gauss:

```
PROGRAM ELIMINASI GAUSS
Masukkan ukuran matriks baris x kolom (gunakan spasi sebagai pemisah): 4
4
Masukkan elemen matriks (gunakan spasi sebagai pemisah dan enter untuk pindah baris):
1 1 -1 -1
2 5 -7 -5
2 -1 1 3
5 2 -4 2
Masukkan elemen matriks jawaban (gunakan spasi sebagai pemisah dan enter untuk pindah baris):
1
-2
4
6
Hasil matriks setelah menggunakan GAUSS:
1.000 2.500 -3.500 -2.500 | -1.000
-0.000 1.000 -1.333 -1.333 | -1.000
0.000 0.000 1.000 -1.000 | 1.000
NaN NaN NaN NaN | Infinity
Solusi SPL dengan metode GAUSS:
X1 = NaN
X2 = NaN
X3 = NaN
X4 = NaN
```

Dengan Metode Gauss-Jordan:

```

PROGRAM ELIMINASI GAUSS

Masukkan ukuran matriks baris X kolom (gunakan spasi sebagai pemisah): 4 4

Masukkan elemen matriks (gunakan spasi sebagai pemisah dan enter untuk pindah baris):
1 1 -1 -1
2 5 -7 -5
2 -1 1 3
5 2 -4 2

Masukkan elemen matriks jawaban (gunakan spasi sebagai pemisah dan enter untuk pindah baris):
1
-2
4
6

Hasil matriks setelah menggunakan GAUSS:
1.000 2.500 -3.500 -2.500 | -1.000
-0.000 1.000 -1.333 -1.333 | -1.000
0.000 0.000 1.000 -1.000 | 1.000
NaN NaN NaN NaN | Infinity

Solusi SPL dengan metode GAUSS:
X1 = NaN
X2 = NaN
X3 = NaN
X4 = NaN

Hasil matriks setelah menggunakan GAUSS-JORDAN:
NaN NaN NaN NaN | NaN
NaN NaN NaN NaN | NaN
NaN NaN NaN NaN | NaN
NaN NaN NaN NaN | Infinity

```

Dengan metode matriks balikan :

```

PROGRAM SPL INVERS

Masukkan ukuran matriks baris X kolom (gunakan spasi sebagai pemisah):
4 4

Masukkan elemen matriks (gunakan spasi sebagai pemisah dan enter untuk pindah baris):
1 1 -1 -1
2 5 -7 -5
2 -1 1 3
5 2 -4 2

Masukkan matriks hasil (gunakan enter untuk pindah baris):
1
-2
4
6

Solusi SPL dengan metode INVERS:
X1 = NaN
X2 = NaN
X3 = NaN
X4 = NaN

```

Dengan kaidah cramer :

```
PROGRAM SPL CRAMMER
Masukkan ukuran matriks baris X kolom (gunakan spasi sebagai pemisah): 4 4
Masukkan elemen matriks (gunakan spasi sebagai pemisah dan enter untuk pindah baris):
1 1 -1 -1
2 5 -7 -5
2 -1 1 3
5 2 -4 2
Masukkan matriks hasil (gunakan enter untuk pindah baris):
1
-2
4
6
Tidak dapat diperoleh penyelesaian karena determinannya 0!
```

b.

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

Dengan metode Gauss :

```
PROGRAM ELIMINASI GAUSS
Masukkan ukuran matriks baris X kolom (gunakan spasi sebagai pemisah): 4 5
Masukkan elemen matriks (gunakan spasi sebagai pemisah dan enter untuk pindah baris):
1 -1 0 0 1
1 1 0 -3 0
2 -1 0 1 -1
-1 2 0 -2 -1
Masukkan elemen matriks jawaban (gunakan spasi sebagai pemisah dan enter untuk pindah baris):
3
6
5
-1
Hasil matriks setelah menggunakan GAUSS:
1.000 -1.000 0.000 0.000 1.000 | 3.000
0.000 1.000 0.000 -1.500 -0.500 | 1.500
NaN NaN NaN Infinity -Infinity |
| -Infinity
NaN NaN NaN NaN NaN | NaN
Solusi SPL dengan metode GAUSS:
X1 = NaN
X2 = NaN
X3 = NaN
X4 = NaN
```

Dengan metode Gauss-Jordan :

```
PROGRAM ELIMINASI GAUSS
Masukkan ukuran matriks baris X kolom (gunakan spasi sebagai pemisah): 4 5
Masukkan elemen matriks (gunakan spasi sebagai pemisah dan enter untuk pindah baris):
1 -1 0 0 1
1 1 0 -3 0
2 -1 0 1 -1
-1 2 0 -2 -1
Masukkan elemen matriks jawaban (gunakan spasi sebagai pemisah dan enter untuk pindah baris):
3
6
5
-1
Hasil matriks setelah menggunakan GAUSS:
1.000 -1.000 0.000 0.000 1.000 | 3.000
0.000 1.000 0.000 -1.500 -0.500 | 1.500
NaN NaN NaN Infinity -Infinity | -Infinity
NaN NaN NaN NaN NaN | NaN

Solusi SPL dengan metode GAUSS:
X1 = NaN
X2 = NaN
X3 = NaN
X4 = NaN

Hasil matriks setelah menggunakan GAUSS-JORDAN:
NaN NaN NaN 1.000 | NaN
NaN NaN NaN -0.500 | NaN
NaN NaN NaN -Infinity | NaN
NaN NaN NaN NaN NaN | NaN

Solusi SPL dengan metode GAUSS-JORDAN:
X1 = NaN
X2 = NaN
X3 = NaN
X4 = NaN
```

Dengan metode Matriks balikan :

```
PROGRAM SPL INVERS
Masukkan ukuran matriks baris X kolom (gunakan spasi sebagai pemisah):
4 5
Tidak dapat diperoleh penyelesaian karena bukan matriks persegi!
```

Dengan kaidah cramer :

```
PROGRAM SPL CRAMMER
Masukkan ukuran matriks baris X kolom (gunakan spasi sebagai pemisah): 4 5
Tidak dapat diperoleh penyelesaian karena bukan matriks persegi!
```

c.

$$\left[ \begin{array}{ccc|c} 1 & 1 & 1 & 0 \\ 2 & 3 & 1 & 1 \\ 3 & 1 & 2 & 1 \end{array} \right]$$

Dengan metode Gauss :

```
PROGRAM ELIMINASI GAUSS
=====
1. Input dari console
2. Input dari file
=====
Pilihan (1-2):
1
Masukkan ukuran matriks baris X kolom (gunakan spasi sebagai pemisah):
3 3

Masukkan elemen matriks (gunakan spasi sebagai pemisah dan enter untuk pindah baris):
1 1 1
2 3 1
3 1 2

Masukkan elemen matriks jawaban (gunakan spasi sebagai pemisah dan ent er untuk pindah baris):
0 1 1

Hasil matriks setelah menggunakan GAUSS:
1.000  1.500  0.500 |   0.500
-0.000  1.000  -0.143 |   0.143
0.000  0.000  1.000 |  -1.000

Solusi SPL dengan metode GAUSS:
X1 = 1.0000
X2 = 0.0000
X3 = -1.0000
```

Dengan metode Gauss-Jordan :

```
PROGRAM ELIMINASI GAUSS-JORDAN
=====
1. Input dari console
2. Input dari file
=====
Pilihan (1-2):
1
Masukkan ukuran matriks baris X kolom (gunakan spasi sebagai pemisah):
3 3

Masukkan elemen matriks (gunakan spasi sebagai pemisah dan enter untuk pindah baris):
1 1 1
2 3 1
3 1 2

Masukkan elemen matriks jawaban (gunakan spasi sebagai pemisah dan ent er untuk pindah baris):
0 1 1

Hasil matriks setelah menggunakan GAUSS:
1.000  1.500  0.500 |   0.500
-0.000  1.000  -0.143 |   0.143
0.000  0.000  1.000 |  -1.000

Solusi SPL dengan metode GAUSS:
X1 = 1.0000
X2 = 0.0000
X3 = -1.0000

Hasil matriks setelah menggunakan GAUSS-JORDAN:
1.000  0.000  0.000 |   1.000
0.000  1.000  0.000 |   0.000
0.000  0.000  1.000 |  -1.000

Solusi SPL dengan metode GAUSS-JORDAN:
X1 = 1.0000
X2 = 0.0000
X3 = -1.0000
```

Dengan metode matriks balikan :

```
PROGRAM SPL INVERS
=====
1. Input dari console
2. Input dari file
=====
Pilihan (1-2):
1
Masukkan ukuran matriks baris x kolom (gunakan spasi sebagai pemisah):
3 3

Masukkan elemen matriks (gunakan spasi sebagai pemisah dan enter untuk
pindah baris):
1 1 1
2 3 1
3 1 2

Masukkan matriks hasil (gunakan enter untuk pindah baris):
0 1 1

Solusi SPL dengan metode INVERS:
X1 = 1.0000
X2 = 0.0000
X3 = -1.0000
```

Dengan kaidah cramer

```
PROGRAM SPL CRAMMER
=====
1. Input dari console
2. Input dari file
=====
Pilihan (1-2):
1
Masukkan ukuran matriks baris x kolom (gunakan spasi sebagai pemisah):
3 3

Masukkan elemen matriks (gunakan spasi sebagai pemisah dan enter untuk
pindah baris):
1 1 1
2 3 1
3 1 2

Masukkan matriks hasil (gunakan enter untuk pindah baris):
0 1 1

Solusi SPL dengan metode CRAMMER:
X1 = 1.0000
X2 = -0.0000
X3 = -1.0000
```

### 3. SPL berbentuk

a.

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

Dengan metode Gauss :

```
PROGRAM ELIMINASI GAUSS

Masukkan ukuran matriks baris X kolom (gunakan spasi sebagai pemisah): 4 4

Masukkan elemen matriks (gunakan spasi sebagai pemisah dan enter untuk pindah baris):
8 1 3 2
2 9 -1 -2
1 3 2 -1
1 0 6 4

Masukkan elemen matriks jawaban (gunakan spasi sebagai pemisah dan enter untuk pindah baris):
0 1 2 3

Hasil matriks setelah menggunakan GAUSS:
1.000  0.125  0.375  0.250 |  0.000
0.000  1.000  -0.200  -0.286 |  0.114
0.000  0.000  1.000  0.663 |  0.538
-0.000 -0.000  -0.000  1.000 |  -0.258

Solusi SPL dengan metode GAUSS:
X1 = -0.2243
X2 = 0.1824
X3 = 0.7095
X4 = -0.2581
```

Dengan metode Gauss-Jordan :

```
PROGRAM ELIMINASI GAUSS

Masukkan ukuran matriks baris X kolom (gunakan spasi sebagai pemisah): 4 4

Masukkan elemen matriks (gunakan spasi sebagai pemisah dan enter untuk pindah baris):
8 1 3 2
2 9 -1 -2
1 3 2 -1
1 0 6 4

Masukkan elemen matriks jawaban (gunakan spasi sebagai pemisah dan enter untuk pindah baris):
0 1 2 3

Hasil matriks setelah menggunakan GAUSS:
1.000  0.125  0.375  0.250   |   0.000
0.000  1.000  -0.200  -0.286  |   0.114
0.000  0.000  1.000  0.663   |   0.538
-0.000 -0.000 -0.000  1.000   |   -0.258

Solusi SPL dengan metode GAUSS:
X1 = -0.2243
X2 = 0.1824
X3 = 0.7095
X4 = -0.2581

Hasil matriks setelah menggunakan GAUSS-JORDAN:
1.000  0.000  0.000  0.000   |   -0.224
0.000  1.000  0.000  0.000   |   0.182
0.000  0.000  1.000  0.000   |   0.709
-0.000 -0.000 -0.000  1.000   |   -0.258

Solusi SPL dengan metode GAUSS-JORDAN:
X1 = -0.2243
X2 = 0.1824
X3 = 0.7095
X4 = -0.2581
```

Dengan metode matriks balikan :

```
PROGRAM SPL INVERS

Masukkan ukuran matriks baris X kolom (gunakan spasi sebagai pemisah):
4 4

Masukkan elemen matriks (gunakan spasi sebagai pemisah dan enter untuk pindah baris):
8 1 3 2
2 9 -1 -2
1 3 2 -1
1 0 6 4

Masukkan matriks hasil (gunakan enter untuk pindah baris):
0 1 2 3

Solusi SPL dengan metode INVERS:
X1 = -0.2243
X2 = 0.1824
X3 = 0.7095
X4 = -0.2581
```

Dengan kaidah cramer :

```
PROGRAM SPL CRAMMER
```

```
Masukkan ukuran matriks baris X kolom (gunakan spasi sebagai pemisah): 4 4
```

```
Masukkan elemen matriks (gunakan spasi sebagai pemisah dan enter untuk pindah baris):
```

```
8 1 3 2  
2 9 -1 -2  
1 3 2 -1  
1 0 6 4
```

```
Masukkan matriks hasil (gunakan enter untuk pindah baris):
```

```
0 1 2 3
```

```
Solusi SPL dengan metode CRAMMER:
```

```
X1 = -0.2243  
X2 = 0.1824  
X3 = 0.7095  
X4 = -0.2581
```

#### 4. Studi kasus Interpolasi

- 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  $f(x)$ .

$x$	0.4	0.7	0.11	0.14	0.17	0.2	0.23
$f(x)$	0.043	0.005	0.058	0.072	0.1	0.13	0.147

Lakukan pengujian pada nilai-nilai default berikut:

$$x = 0.2 \quad f(x) =$$

Masukkan jumlah titik: 7

Masukkan nilai X1: 0.4  
Masukkan nilai Y1: 0.043  
Masukkan nilai X2: 0.7  
Masukkan nilai Y2: 0.005  
Masukkan nilai X3: 0.11  
Masukkan nilai Y3: 0.058  
Masukkan nilai X4: 0.14  
Masukkan nilai Y4: 0.072  
Masukkan nilai X5: 0.17  
Masukkan nilai Y5: 0.1  
Masukkan nilai X6: 0.2  
Masukkan nilai Y6: 0.13  
Masukkan nilai X7: 0.23  
Masukkan nilai Y7: 0.147

Masukkan nilai X yang akan ditaksir: 0.2

Hasil matriks setelah menggunakan GAUSS:

1.000	0.400	0.160	0.064	0.026	0.010	0.004		0.043
0.000	1.000	1.100	0.930	0.715	0.526	0.379		-0.127
0.000	0.000	1.000	1.210	1.063	0.832	0.618		-0.127
0.000	0.000	0.000	1.000	1.380	1.298	1.053		5.927
0.000	0.000	0.000	0.000	1.000	1.580	1.614		77.871
0.000	0.000	0.000	0.000	0.000	1.000	1.810		-522.1
07								
-0.000	-0.000	-0.000	-0.000	-0.000	-0.000	1.000		-4212.
435								

Solusi SPL dengan metode GAUSS:

X1 = -0.1846  
X2 = 10.2764  
X3 = -163.9157  
X4 = 1220.8549  
X5 = -4346.3140  
X6 = 7102.3992  
X7 = -4212.4345

P(0.2) = 0.1300 Apakah anda ingin menggunakan kalkulator lagi? (y/n)

[ ]

$$x = 0.55 \quad f(x) =$$

```
Masukkan jumlah titik: 7
```

```
Masukkan nilai X1: 0.4
Masukkan nilai Y1: 0.043
Masukkan nilai X2: 0.7
Masukkan nilai Y2: 0.005
Masukkan nilai X3: 0.11
Masukkan nilai Y3: 0.058
Masukkan nilai X4: 0.14
Masukkan nilai Y4: 0.072
Masukkan nilai X5: 0.17
Masukkan nilai Y5: 0.1
Masukkan nilai X6: 0.2
Masukkan nilai Y6: 0.13
Masukkan nilai X7: 0.23
Masukkan nilai Y7: 0.147
```

```
Masukkan nilai X yang akan ditaksir: 0.55
```

```
Hasil matriks setelah menggunakan GAUSS:
```

1.000	0.400	0.160	0.064	0.026	0.010	0.004		0.043
0.000	1.000	1.100	0.930	0.715	0.526	0.379		-0.127
0.000	0.000	1.000	1.210	1.063	0.832	0.618		-0.127
0.000	0.000	0.000	1.000	1.380	1.298	1.053		5.927
0.000	0.000	0.000	0.000	1.000	1.580	1.614		77.871
0.000	0.000	0.000	0.000	0.000	1.000	1.810		-522.1
07								
-0.000	-0.000	-0.000	-0.000	-0.000	-0.000	1.000		-4212.
435								

```
Solusi SPL dengan metode GAUSS:
```

```
X1 = -0.1846
X2 = 10.2764
X3 = -163.9157
X4 = 1220.8549
X5 = -4346.3140
X6 = 7102.3992
X7 = -4212.4345
```

```
P(0.55) = 2.1376Apakah anda ingin menggunakan kalkulator lagi? (y/n)
```

$$x = 0.85 \quad f(x) =$$

Masukkan jumlah titik: 7

Masukkan nilai X1: 0.4  
Masukkan nilai Y1: 0.043  
Masukkan nilai X2: 0.7  
Masukkan nilai Y2: 0.005  
Masukkan nilai X3: 0.11  
Masukkan nilai Y3: 0.058  
Masukkan nilai X4: 0.14  
Masukkan nilai Y4: 0.072  
Masukkan nilai X5: 0.17  
Masukkan nilai Y5: 0.1  
Masukkan nilai X6: 0.2  
Masukkan nilai Y6: 0.13  
Masukkan nilai X7: 0.23  
Masukkan nilai Y7: 0.147

Masukkan nilai X yang akan ditaksir: 0.85

Hasil matriks setelah menggunakan GAUSS:

1.000	0.400	0.160	0.064	0.026	0.010	0.004		0.043
0.000	1.000	1.100	0.930	0.715	0.526	0.379		-0.127
0.000	0.000	1.000	1.210	1.063	0.832	0.618		-0.127
0.000	0.000	0.000	1.000	1.380	1.298	1.053		5.927
0.000	0.000	0.000	0.000	1.000	1.580	1.614		77.871
0.000	0.000	0.000	0.000	0.000	1.000	1.810		-522.1
07								
-0.000	-0.000	-0.000	-0.000	-0.000	-0.000	1.000		-4212.
435								

Solusi SPL dengan metode GAUSS:

X1 = -0.1846  
X2 = 10.2764  
X3 = -163.9157  
X4 = 1220.8549  
X5 = -4346.3140  
X6 = 7102.3992  
X7 = -4212.4345

P(0.85) = -66.2696Apakah anda ingin menggunakan kalkulator lagi? (y/n)

■

$$x = 1.28 \quad f(x) =$$

Masukkan jumlah titik: 7

Masukkan nilai X1: 0.4  
Masukkan nilai Y1: 0.043  
Masukkan nilai X2: 0.7  
Masukkan nilai Y2: 0.005  
Masukkan nilai X3: 0.11  
Masukkan nilai Y3: 0.058  
Masukkan nilai X4: 0.14  
Masukkan nilai Y4: 0.072  
Masukkan nilai X5: 0.17  
Masukkan nilai Y5: 0.1  
Masukkan nilai X6: 0.2  
Masukkan nilai Y6: 0.13  
Masukkan nilai X7: 0.23  
Masukkan nilai Y7: 0.147

Masukkan nilai X yang akan ditaksir: 1.28

Hasil matriks setelah menggunakan GAUSS:

1.000	0.400	0.160	0.064	0.026	0.010	0.004		0.043
0.000	1.000	1.100	0.930	0.715	0.526	0.379		-0.127
0.000	0.000	1.000	1.210	1.063	0.832	0.618		-0.127
0.000	0.000	0.000	1.000	1.380	1.298	1.053		5.927
0.000	0.000	0.000	0.000	1.000	1.580	1.614		77.871
0.000	0.000	0.000	0.000	0.000	1.000	1.810		-522.1
07								
-0.000	-0.000	-0.000	-0.000	-0.000	-0.000	1.000		-4212.
435								

Solusi SPL dengan metode GAUSS:

X1 = -0.1846  
X2 = 10.2764  
X3 = -163.9157  
X4 = 1220.8549  
X5 = -4346.3140  
X6 = 7102.3992  
X7 = -4212.4345

P(1.28) = -3485.1449Apakah anda ingin menggunakan kalkulator lagi? (y/n)

#### 4. Studi Kasus Interpolasi Bicubic

Diberikan Matriks input:

153	59	210	96
125	161	72	81
98	101	42	12
21	51	0	16

Tentukan nilai:

$$f(0,0) =$$

PROGRAM INTERPOLASI BIKUBIK

Masukkan elemen matriks (gunakan spasi sebagai pemisah dan enter untuk pindah baris):

153 59 210 96  
125 161 72 81  
98 101 42 12  
21 51 0 16

Masukan Nilai X: 0

Masukan Nilai Y: 0

$$f(0.0,0.0) = 161.000$$

$$f(0.5,0.5) =$$

PROGRAM INTERPOLASI BIKUBIK

Masukkan elemen matriks (gunakan spasi sebagai pemisah dan enter untuk pindah baris):

153 59 210 96  
125 161 72 81  
98 101 42 12  
21 51 0 16

Masukan Nilai X: 0.5

Masukan Nilai Y: 0.5

$$f(0.5,0.5) = 97.727$$

$f(0.25,0.25) =$

PROGRAM INTERPOLASI BIKUBIK

Masukkan elemen matriks (gunakan spasi sebagai pemisah dan enter untuk pindah baris):

153 59 210 96  
125 161 72 81  
98 101 42 12  
21 51 0 16

Masukan Nilai X: 0.25

Masukan Nilai Y: 0.25

$f(0.25,0.25) = 133.798$  Masukkan pilihan anda : 1

$f(0.1,0.9) =$

PROGRAM INTERPOLASI BIKUBIK

Masukkan elemen matriks (gunakan spasi sebagai pemisah dan enter untuk pindah baris):

153 59 210 96  
125 161 72 81  
98 101 42 12  
21 51 0 16

Masukan Nilai X: 0.1

Masukan Nilai Y: 0.9

$f(0.1,0.9) = 74.696$  Masukkan pilihan anda : □

## **BAB V**

### **KESIMPULAN**

#### **5.1 Kesimpulan**

SPL dapat diselesaikan menggunakan berbagai macam metode, diantaranya metode eliminasi Gauss, Gauss - Jordan, matriks balikan, dan kaidah cramer. SPL ini juga ternyata dapat digunakan untuk menaksir suatu nilai menggunakan interpolasi polinom dan regresi linear berganda. Untuk tugas besar ini, kami berhasil membuat kalkulator matriks dengan implementasi algoritma yang sudah ada ke dalam bahasa Java dan dapat menyelesaikan berbagai permasalahan menggunakan matriks, contohnya adalah menggunakan interpolasi polinom dan regresi linear berganda. Namun sayangnya, masih ada beberapa fungsi yang kurang sempurna dalam implementasinya dikarenakan durasi dan kurangnya wawasan kami terhadap bahasa java.

#### **5.2 Saran**

- Seharusnya pengerajan tidak dilakukan mepet deadline sehingga masih banyak waktu untuk memperbaiki bug dan menyelesaikan setiap fungsi yang diminta.
- Seharusnya pembagian tugas dibagi sama rata sehingga pengerajan bisa lebih efektif dan lebih cepat

#### **5.3 Refleksi**

Refleksi yang kami dapatkan dari tugas ini adalah kami bisa memperbaiki lagi kinerja kami dalam berbagai hal, contohnya pembuatan timeline kerja agar selalu ada progress dalam tugas tiap harinya. Kami juga merasa masih harus memperbaiki pembagian waktu karena pengerajan tugas masih terlalu berdekatan dengan deadline. Hal lain yang kami dapatkan dari tugas ini adalah pengalaman dan pengetahuan membuat program dalam bahasa Java.

## **REFERENSI**

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2021-2022/Pengantar-Pemrograman-dengan-Bahasa-Java-2021.pdf>