LAPORAN TUGAS BESAR ALJABAR GEOMETRI

SPL, DETERMINAN, MATRIKS DAN APLIKASINYA

Disusun Oleh :
Kelompok 6

M. Hizaz Badruzaman	10222121
M Ikbal Handini	10222181
Vanny Mustaqimah	10222116
Salwa Nurazizah	10222154
Eneng Eka Nurhapipah	10222125

PROGRAM STUDI INFORMATIKA
SEKOLAH TINGGI TEKNOLOGI CIPASUNG
TASIKMALAYA

2023

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1
DESKRIPSI MASALAH

Dalam dunia matematika, pemrograman matriks memiliki peranan krusial sebagai fondasi untuk berbagai aplikasi di berbagai bidang, termasuk komputasi grafis, statistika, fisika, dan rekayasa. Keberhasilan penyelesaian tugas ini menjadi penting karena pemahaman konsep dasar aljabar linear dan penerapannya melalui operasi-operasi matriks sangat relevan.

Dalam tugas ini, fokusnya adalah mengembangkan program sederhana yang dapat menjalankan operasi-operasi matriks tersebut. Keberhasilan implementasi program ini akan membantu pemahaman konsep dasar aljabar dan penerapannya dalam pemrograman

Pemahaman konsep matriks adalah inti dari pengembangan program ini. Program yang dapat melakukan operasi matriks seperti penjumlahan, pengurangan, transpose, invers, determinan, dan penyelesaian sistem persamaan linier (SPL) menjadi alat pembelajaran yang efektif, terutama untuk mahasiswa yang sering dihadapkan pada kompleksitas aljabar geometri.

Selain itu, implementasi program ini membawa manfaat nyata dalam mendukung pemahaman konsep-konsep matriks secara praktis. Program ini dirancang untuk memberikan antarmuka yang jelas dan fungsionalitas yang handal, sehingga menjadi alat bantu yang efektif untuk mahasiswa dan profesional dalam berbagai disiplin ilmu.

Pentingnya pemahaman ini diperkuat oleh konteks pembelajaran di mana mahasiswa sering dihadapkan pada tugas dan penelitian yang melibatkan operasi-operasi matriks. Oleh karena itu, pengembangan program ini bertujuan untuk memberikan solusi interaktif bagi mereka yang ingin memahami dan mengimplementasikan operasi-operasi matriks secara praktis. Program ini tidak hanya memfasilitasi penggunaan operasi matriks melalui input keyboard, tetapi juga memberikan output yang informatif, memudahkan pengguna dalam memahami hasil operasi matriks.

Dengan demikian, program ini diharapkan dapat menjadi kontribusi positif dalam meningkatkan pemahaman dan penerapan konsep matriks, memberikan landasan kuat bagi mahasiswa dan profesional dalam menjawab tantangan matematika dan rekayasa di berbagai konteks aplikasi.

BAB II
TEORI SINGKAT

2.1 Sistem Persamaan Linier (Metode Eliminasi Gauss, Metode Eliminasi Gauss – Jordan)

Bentuk Umum SPL

- Linier: pangkat tertinggi di dalam variabelnya sama dengan 1
- Sebuah SPL dengan m buah persamaan dan n variabel x1, x2, …, xn berbentuk:

  a11x1 + a12x2 + … + a1nxn = b1

  a21x1 + a22x2 + … + a2nxn = b2

  ⋮ ⋮ ⋮ ⋮

  am1x1 + am2x2 + … + amnxn = bm

  atau dalam bentuk Ax = b

- SPL dalam bentuk matriks:

$$\begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \\ \vdots \quad\quad \vdots \quad\quad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

  atau dalam bentuk perkalian matriks: Axb

$$\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

- SPL Dalam Bentuk Kaidah Cramer

Dalam aljabar linear, **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, walau Colin Maclaurin juga menerbitkan kasus khusus dari kaidah ini pada tahun 1748 (dan mungkin ia sudah mengetahuinya sejak 1729).

Rumus aturan Cramer dituliskan sebagai berikut: xi = det Bi/det A
keterangan :
Xi = Variabel yang ingin diketahui nilainya det
B = Determinan dari matriks di mana vektor kolom ke - i diganti vektor b.
det A = Determinan matriks A

Contoh Soal Metode Cramer Tentukan sistem persamaan linear dua variabel menggunakan sistem metode cramer.
2x - 3y = -13
x + 2y = 4
Penyelesaian sistem persamaan linear dua variabel (SPDLV) dinyatakan dalam bentuk matriks.

SPLDV dalam soal di atas dapat dinyatakan dalam bentuk matriks, yakni

$$\begin{bmatrix} 2 & -3 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -13 \\ 4 \end{bmatrix}$$

Dengan demikian, kita peroleh hasil berikut ini.

$$\begin{aligned} D &= \begin{vmatrix} 2 & -3 \\ 1 & 2 \end{vmatrix} = 2(2) - (-3)(1) \\ &= 4 + 3 = 7 \\ D_x &= \begin{vmatrix} -13 & -3 \\ 4 & 2 \end{vmatrix} = -13(2) - (-3)(4) \\ &= -26 + 12 = -14 \\ D_y &= \begin{vmatrix} 2 & -13 \\ 1 & 4 \end{vmatrix} = 2(4) - (-13)(1) \\ &= 8 + 13 = 21 \end{aligned}$$

Berdasarkan Aturan Cramer, kita peroleh hasil berikut.

$$x = \frac{D_x}{D} = \frac{-14}{7} = -2$$

$$y = \frac{D_y}{D} = \frac{21}{7} = 3$$

Contoh Soal Metode Cramer Jadi nilai x dan y yang memenuhi SPLDV di atas yaitu x = -2 dan y = 3

Matriks Augmented

- SPL dapat dinyatakan secara ringkas dalam bentuk matriks augmented:

$$[A \mid \mathbf{b}] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{array}\right]$$

- Contoh:

$$\begin{aligned} x_1 + 3x_2 - 6x_3 &= 9 \\ 2x_1 - 6x_2 + 4x_3 &= 7 \\ 5x_1 + 2x_2 - 5x_3 &= -2 \end{aligned} \longrightarrow \begin{bmatrix} 1 & 3 & -6 & 9 \\ 2 & -6 & 4 & 7 \\ 5 & 2 & -5 & -2 \end{bmatrix}$$

Operasi Baris Elementer (OBE)

- Tiga operasi baris elementer terhadap matriks augmented:
  1. Kalikan sebuah baris dengan konstanta tidak nol.
  2. Pertukaran dua buah baris
  3. Tambahkan sebuah baris dengan kelipatan baris lainnya
- Solusi sebuah SPL diperoleh dengan menerapkan OBE pada matriks augmented sampai terbentuk matriks eselon baris atau matriks eselon baris tereduksi.
- Jika berakhir pada matriks eselon baris metode eliminasi Gauss
- Jika berakhir pada matriks eselon baris tereduksi metode eliminasi Gauss- Jordan

Metode Eliminasi Gauss

- Metode Eliminasi Gauss merupakan metode yang dikembangkan dari metode eliminasi, yaitu menghilangkan atau mengurangi jumlah variable sehingga dapat diperoleh nilai dari suatu variable bebas.
- matrik diubah menjadi augmented matrik :

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{array}\right]$$

- ubah matrik menjadi matrik segitiga atas atau segitiga bawah dengan menggunakan OBE (Operasi Baris Elementer).

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} & b_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} & b_n \end{bmatrix} \longrightarrow \begin{bmatrix} c_{11} & c_{12} & c_{13} & \dots & c_{1n} & d_1 \\ 0 & c_{22} & c_{23} & \dots & c_{2n} & d_2 \\ 0 & 0 & c_{33} & \dots & c_{3n} & d_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & c_{nn} & d_n \end{bmatrix}$$

- Sehingga penyelesaian dapat diperoleh dengan:

$$x_n = \frac{d_n}{c_{nn}}$$

$$x_{n-1} = \frac{1}{c_{n-1,n-1}}\left(-c_{n-1,n}x_n + d_{n-1}\right)$$

....

$$x_2 = \frac{1}{c_{22}}\left(d_2 - c_{23}x_3 - c_{24}x_4 - \dots - c_{2n}x_n\right)$$

$$x_1 = \frac{1}{c_{11}}\left(d_1 - c_{12}x_2 - c_{13}x3 - \dots - c_{1n}x_n\right)$$

Metode Eliminasi Gauss-Jordan

- Merupakan pengembangan metode eliminasi Gauss
- Operasi baris elementer (OBE) diterapkan pada matriks augmented sehingga menghasilkan matriks eselon baris tereduksi.

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

- Tidak diperlukan lagi substitusi secara mundur untuk memperoleh nilainilai variabel. Nilai variabel langsung diperoleh dari matriks augmented akhir.
- Metode eliminasi Gauss-Jordan terdiri dari dua fase:
  1. Fase maju (forward phase) atau fase eliminasi Gauss
     - ➢ Menghasilkan nilai-nilai 0 di bawah 1 utama

$$\begin{bmatrix} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{bmatrix} \overset{\text{OBE}}{\sim \cdots \sim} \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

2. Fase mundur (backward phase)

➢ Menghasilkan nilai-nilai 0 di atas satu utama.

$$\begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix} \overset{R1-(3/2)R2}{\sim} \begin{bmatrix} 1 & 0 & -5/4 & -11/4 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix} \underset{R2-(1/2)R3}{\overset{R1+(5/4)R3}{\sim}} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

Matriks eselon baris tereduksi

Dari matriks augmented terakhir, diperoleh x1 = 1, x2 = 2, x3 = 3

2.2 Determinan

Definisi

- Misalkan A adalah matriks berukuran n x n

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

- Determinan matriks A dilambangkan dengan

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

Determinan matriks 2 x 2

Untuk matriks A berukuran 2 x 2:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

maka det(A) = a11a22 – a12a21

Determinan matriks 3 x 3

Untuk matriks A berukuran 3 x3:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{matrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{matrix}$$

maka det(A) = (a11a22 a33 + a12a21a31 + a13a21a32) – (a13a22a31 + a11a23a32 + a12a21a33)

Determinan Matriks Segitiga

1. Matriks segitiga atas (upper triangular): semua elemen di bawah diagonal utama adalah nol

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \Longrightarrow \det(A) = a_{11}a_{22}\, a_{33}a_{44}$$

2. Matriks segitiga bawah (lower triangular): semua elemen di atas diagonal utama adalah nol

$$A = \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \Longrightarrow \det(A) = a_{11}a_{22}\, a_{33}a_{44}$$

- Secara umum, untuk matriks segitiga A berukuran n x n,

  det(A) = a11a22 a33… ann

Aturan Determinan

- Misalkan A adalah matriks n x n. Matriks B adalah matriks yang diperoleh dengan memanipulasi matriks A. Bagaimana determinan B?

- $A \xrightarrow{\text{Kalikan sebuah baris dengan k}} B$, maka det(B) = k det(A)
- $A \xrightarrow{\text{Pertukarkan dua baris}} B$, maka det(B) = –det(A)
- $A \xrightarrow{\text{Sebuah baris ditambahkan dengan k kali baris yang lain}} B$, maka det(B) = det(A)

2.3 Matriks Balikan

- Matriks balikan (inverse) dari sebuah matriks A adalah matriks B sedemikian sehingga AB = BA = I
- Kita katakan A dan B merupakan balikan matriks satu sama lain
- Contoh: Misalkan

$$A = \begin{bmatrix} 2 & -5 \\ -1 & 3 \end{bmatrix} \text{ and } B = \begin{bmatrix} 3 & 5 \\ 1 & 2 \end{bmatrix}$$

$$\text{maka} \quad \begin{aligned} AB &= \begin{bmatrix} 2 & -5 \\ -1 & 3 \end{bmatrix}\begin{bmatrix} 3 & 5 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I \\ BA &= \begin{bmatrix} 3 & 5 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} 2 & -5 \\ -1 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I \end{aligned}$$

- Balikan matriks A disimbolkan dengan A –1
- Sifat: AA–1 = A –1A = I
- Untuk matriks A berukuran 2 x 2, maka A –1 dihitung sebagai berikut:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \longrightarrow A^{-1} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

dengan syarat ad – bc 0

- Nilai ad – bc disebut determinan. Jika ad – bc = 0 maka matriks A tidak memiliki balikan (not invertible)

2.4 Transpose Matriks

- Transpose matriks,

B = AT

bji = aij i = 1, 2, …m; j = 1, 2, …n

- Algoritma transpose matriks:

```
for i←1 to m do
    for j←1 to n do
        bji ← aij
    end for
end for
```

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 3 \\ 1 & 4 \\ 5 & 6 \end{bmatrix}, \quad C = [1 \quad 3 \quad 5], \quad D = [4]$$

$$A^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \\ a_{14} & a_{24} & a_{34} \end{bmatrix}, \quad B^T = \begin{bmatrix} 2 & 1 & 5 \\ 3 & 4 & 6 \end{bmatrix}, \quad C^T = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}, \quad D^T = [4]$$

- Untuk matriks persegi A berukuran n x n, transpose matriks A dapat diperoleh dengan mempertukarkan elemen yang simetri dengan diagonal utama:

$$A = \begin{bmatrix} 1 & -2 & 4 \\ 3 & 7 & 0 \\ -5 & 8 & 6 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & -2 & 4 \\ 3 & 7 & 0 \\ -5 & 8 & 6 \end{bmatrix} \rightarrow A^T = \begin{bmatrix} 1 & 3 & -5 \\ -2 & 7 & 8 \\ 4 & 0 & 6 \end{bmatrix}$$

- Sifat-sifat transpose matriks

(a) $\left(A^T\right)^T = A$

(b) $(A+B)^T = A^T + B^T$

(c) $(A-B)^T = A^T - B^T$

(d) $(kA)^T = kA^T$

(e) $(AB)^T = B^T A^T$

2.5 Penjumlahan Matriks

- Penjumlahan dua buah matriks Cm x n = Am x n + Bm x n Misal A = [aij] B = [bij] maka C = A + B = [cij] , cij = aij +bij , i = 1, 2, …, m; j = 1, 2, …, n
- Pengurangan matriks: C = A – B = [cij] , cij = aij – bij , i = 1, 2, …, m; j = 1, 2, …, n
- Algoritma penjumlahan dua buah matriks:

```
for i←1 to m do
    for j←1 to n do
        c_ij ← a_ij + b_ij
    end for
end for
```

Contoh :

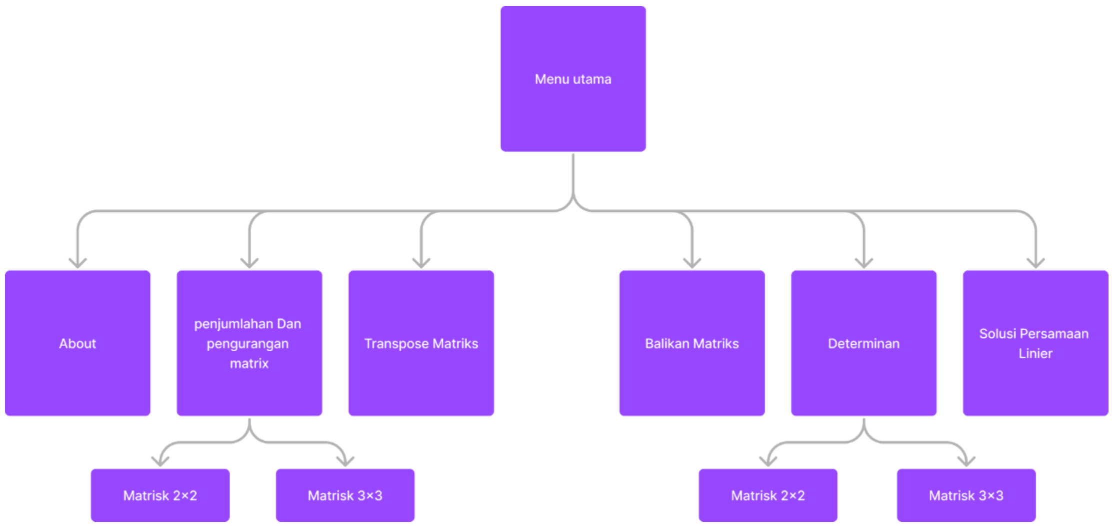
$$A = \begin{bmatrix} 2 & 1 & 0 & 3 \\ -1 & 0 & 2 & 4 \\ 4 & -2 & 7 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} -4 & 3 & 5 & 1 \\ 2 & 2 & 0 & -1 \\ 3 & 2 & -4 & 5 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}$$

- Maka,

$$A + B = \begin{bmatrix} -2 & 4 & 5 & 4 \\ 1 & 2 & 2 & 3 \\ 7 & 0 & 3 & 5 \end{bmatrix}, \quad A - B = \begin{bmatrix} 6 & -2 & -5 & 2 \\ -3 & -2 & 2 & 5 \\ 1 & -4 & 11 & -5 \end{bmatrix}$$

BAB III
PENJELASAN IMPLEMENTASI PROGRAM

Menu :



gambar 1 User flow

Panel utama akan menampilkan menu :

1. About

   Yang berisikan deskripsi Aplikasi.
2. Penjumlahan dan pengurangan Matriks

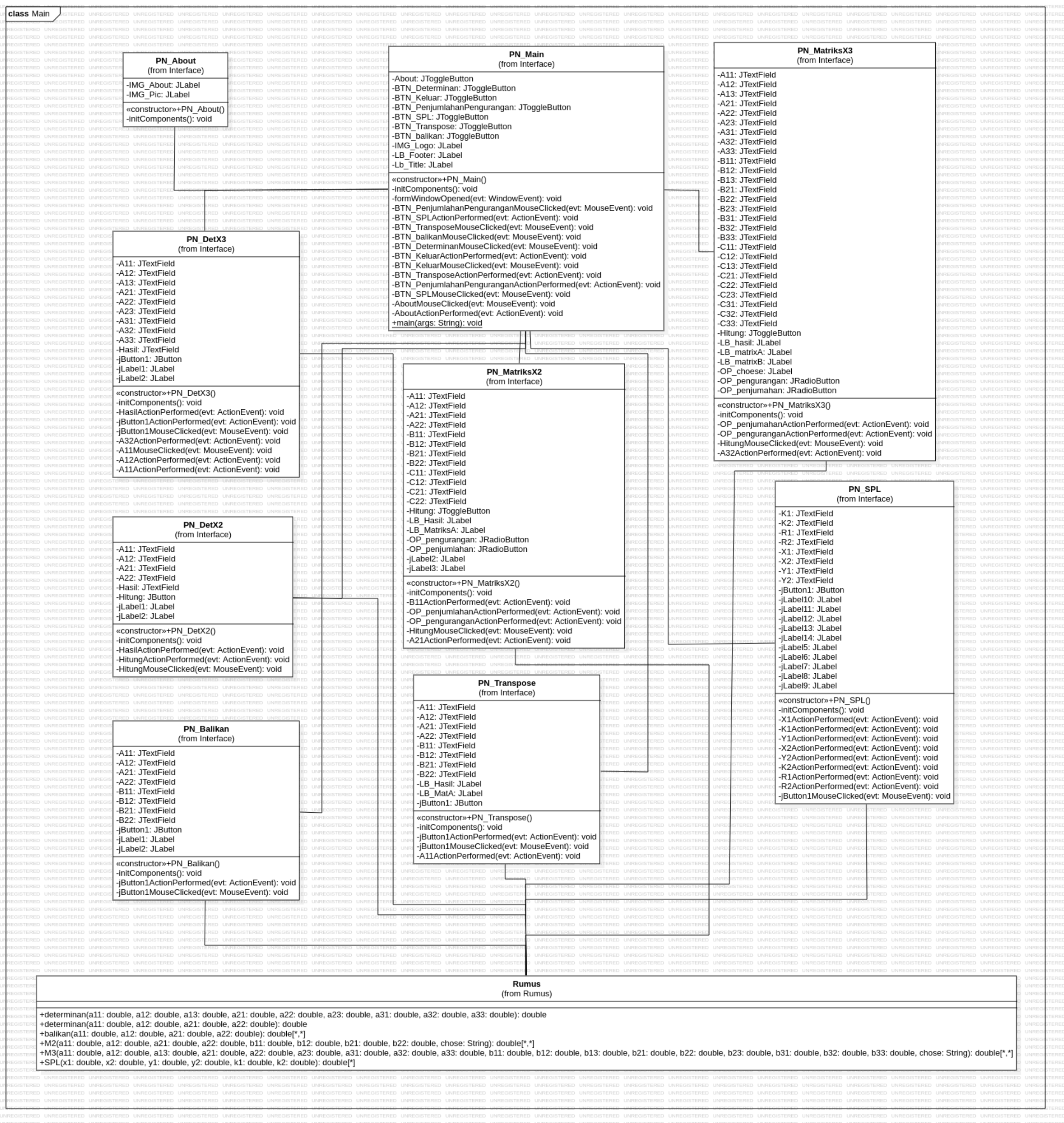
   Akan Memunculkan JOptionpane/popup untuk memilih Ukuran/Ordo matriks
   a. Matriks 2X2
   b. Matrix 3X3
3. Transpose Matriks

   Akan memunculkan Jptionpane/popup untuk memilih ukuran/ordo matriks
   a. Matriks 2x2
   b. Matriks 3x3
4. Balikan Matriks (Invers)
5. Determinan

   Akan Memunculkan JOptionpane untuk memilih Ukuran/Ordo Matriks
   a. Matriks 2X2
   b. Matrix 3X3

6. Solusi persamaan Linear
7. Keluar

Class Diagram :



gambar 2 Class Diagram

PACKAGE INTERFACE		
Class	**Komponen**	**Keterangan**
PN_Main	JToggleButton	- About: JToggleButton: Tombol toggle yang mungkin digunakan untuk menampilkan informasi "About".
	JToggleButton	- BTN_Determinan: JToggleButton: Tombol toggle untuk mengeksekusi operasi menghitung determinan.
	JToggleButton	- BTN_Keluar: JToggleButton: Tombol toggle untuk keluar dari aplikasi.
	JToggleButton	- BTN_PenjumlahanPengurangan: JToggleButton: Tombol toggle untuk mengeksekusi operasi penjumlahan/pengurangan matriks.
	JToggleButton	- BTN_SPL: JToggleButton: Tombol toggle untuk menampilkan operasi sistem persamaan linear (SPL).
	JToggleButton	- BTN_Transpose: JToggleButton: Tombol toggle untuk mengeksekusi operasi transpose matriks.
	JToggleButton	- BTN_balikan: JToggleButton: Tombol toggle untuk mengeksekusi operasi invers matriks.
	JLabel	- IMG_Logo: JLabel: Label untuk menampilkan logo.
	JLabel	- LB_Footer: JLabel: Label untuk menampilkan informasi di bagian bawah antarmuka.
	JLabel	- Lb_Title: JLabel: Label untuk menampilkan judul antarmuka.
	Constructor	+PN_Main(): Konstruktor untuk menginisialisasi objek kelas.
	initComponents()	Metode untuk menginisialisasi dan menyiapkan komponen GUI.

	formWindowOpened(evt: WindowEvent): void	Metode yang dipanggil ketika jendela antarmuka dibuka.
	BTN_PenjumlahanPenguranganMouseClicked(evt: MouseEvent): void	Metode yang dipanggil ketika tombol penjumlahan/pengurangan diklik.
	BTN_TransposeMouseClicked(evt: MouseEvent): void	Metode yang dipanggil ketika tombol transpose diklik.
	BTN_balikanMouseClicked(evt: MouseEvent): void	Metode yang dipanggil ketika tombol balikan (invers) diklik.
	BTN_DeterminanMouseClicked(evt: MouseEvent): void	Metode yang dipanggil ketika tombol determinan diklik.
	BTN_KeluarMouseClicked(evt: MouseEvent): void	Metode yang dipanggil ketika tombol keluar diklik.
	BTN_SPLMouseClicked(evt: MouseEvent): void	Metode yang dipanggil ketika tombol SPL diklik.
	AboutMouseClicked(evt: MouseEvent): void	Metode yang dipanggil ketika tombol About diklik.
	main(args: String): void	Metode utama yang memulai aplikasi.
PN_MAtrixX2	JTextFields	A11, A12, A21, ..., C22: JTextField untuk input nilai elemen matriks.
	JToggleButton	Hitung: JToggleButton, mungkin digunakan untuk memulai proses perhitungan matriks.
	JLabel	LB_Hasil, LB_MatriksA, jLabel2, jLabel3: JLabel untuk menampilkan informasi pada GUI.
	JRadioButton	OP_pengurangan, OP_penjumlahan: JRadioButton untuk memilih operasi matriks (penjumlahan atau pengurangan).
	Constructor	PN_MatriksX2(): Konstruktor publik untuk menginisialisasi objek kelas.
	initComponents()	Metode untuk menginisialisasi dan menyiapkan komponen GUI.
	B11ActionPerformed(evt:ActionEvent):void	Metode yang dipanggil ketika ada tindakan pada B11 JTextField.

	OP_penjumlahanActionPerformed(evt: ActionEvent): void	Metode yang dipanggil ketika ada tindakan pada tombol penjumlahan.
	OP_penguranganActionPerformed(evt: ActionEvent): void	Metode yang dipanggil ketika ada tindakan pada tombol pengurangan.
	HitungMouseClicked(evt: MouseEvent): void	Metode yang dipanggil ketika ada klik pada tombol "Hitung".
	A21ActionPerformed(evt: ActionEvent): void	Metode yang dipanggil ketika ada tindakan pada A21 JTextField.
PN-MatriksX3	JTextFields	- A11, A12, ..., C33: JTextField untuk input nilai elemen matriks.
	JToggleButton	- Hitung: JToggleButton, mungkin digunakan untuk memulai proses perhitungan matriks.
	JLabel	- LB_hasil: JLabel untuk menampilkan hasil perhitungan.
	JLabel	- LB_matrixA: JLabel untuk menampilkan informasi tentang matriks A.
	JLabel	- LB_matrixB: JLabel untuk menampilkan informasi tentang matriks B.
	JLabel	- OP_choese: JLabel yang mungkin digunakan untuk menunjukkan pilihan operasi (penjumlahan/pengurangan).
	JRadioButton	- OP_pengurangan: JRadioButton untuk memilih operasi pengurangan matriks.
	JRadioButton	- OP_penjumahan: JRadioButton untuk memilih operasi penjumlahan matriks.
	Constructor	>+PN_MatriksX3(): Konstruktor untuk menginisialisasi objek kelas.
	initComponents()	Metode untuk menginisialisasi dan menyiapkan komponen GUI.
	HitungMouseClicked(evt: MouseEvent): void	Metode yang dipanggil ketika tombol "Hitung" diklik.
	Komponen	**Keterangan**

PN_Transpose	JTextFields	- A11, A12, A21, A22, B11, B12, B21, B22: JTextField untuk input nilai elemen matriks.
	JLabel	- LB_Hasil: JLabel untuk menampilkan hasil operasi transpose.
	JLabel	- LB_MatA: JLabel untuk menampilkan informasi tentang matriks A.
	JButton	- jButton1: JButton, mungkin digunakan untuk memulai proses operasi transpose.
	Constructor	>+PN_Transpose(): Konstruktor untuk menginisialisasi objek kelas.
	initComponents()	Metode untuk menginisialisasi dan menyiapkan komponen GUI.
	jButton1MouseClicked(evt: MouseEvent): void	Metode yang dipanggil ketika terjadi tindakan klik pada jButton1.
PN_SPL	**Komponen**	**Keterangan**
	JTextFields	- K1, K2, R1, R2, X1, X2, Y1, Y2: JTextField untuk input nilai.
	JButton	- jButton1: JButton, mungkin untuk memulai proses SPL.
	JLabel	- jLabel5, jLabel6, ..., jLabel19: JLabel untuk tampilan informasi atau label pada GUI.
	Constructor	>+PN_SPL(): Konstruktor untuk menginisialisasi objek kelas.
	initComponents()	Metode untuk menginisialisasi dan menyiapkan komponen GUI.
	jButton1MouseClicked(evt: MouseEvent): void	Metode yang dipanggil ketika terjadi tindakan klik pada jButton1.
PN_DetX3	**Komponen**	**Keterangan**
	JTextFields	- A11, A12, A13, A21, A22, A23, A31, A32, A33: JTextField untuk input nilai.
	JTextField	- Hasil: JTextField untuk menampilkan hasil determinan.

	JButton	- jButton1: JButton, mungkin untuk memulai proses menghitung determinan.
	JLabel	- jLabel1, jLabel2: JLabel untuk tampilan informasi atau label pada GUI.
	Constructor	>+PN_DetX3(): Konstruktor untuk menginisialisasi objek kelas.
	initComponents()	Metode untuk menginisialisasi dan menyiapkan komponen GUI.
	jButton1MouseClicked(evt: MouseEvent): void	Metode yang dipanggil ketika terjadi tindakan klik pada jButton1.
PN_DetX2	**Komponen**	**Keterangan**
	JTextFields	- A11, A12, A21, A22: JTextField untuk input nilai.
	JTextField	- Hasil: JTextField untuk menampilkan hasil determinan.
	JButton	- Hitung: JButton, mungkin untuk memulai proses menghitung determinan.
	JLabel	- jLabel1, jLabel2: JLabel untuk tampilan informasi atau label pada GUI.
	Constructor	>+PN_DetX2(): Konstruktor untuk menginisialisasi objek kelas.
	initComponents()	Metode untuk menginisialisasi dan menyiapkan komponen GUI.
	HitungMouseClicked(evt: MouseEvent): void	Metode yang dipanggil ketika terjadi tindakan klik pada tombol Hitung.
PN_Balikan	**Komponen**	**Keterangan**
	JTextFields	- A11, A12, A21, A22, B11, B12, B21, B22: JTextField untuk input nilai.
	JButton	- jButton1: JButton, mungkin untuk memulai proses menghitung invers.
	JLabel	- jLabel1, jLabel2: JLabel untuk tampilan informasi atau label pada GUI.
	Constructor	>=+PN_Balikan(): Konstruktor untuk menginisialisasi objek kelas.

	initComponents()	Metode untuk menginisialisasi dan menyiapkan komponen GUI.
	jButton1MouseClicked(evt: MouseEvent): void	Metode yang dipanggil ketika terjadi tindakan klik pada tombol jButton1.

Tabel 1 Package Interface

PACKAGE RUMUS			
Class	**Method**	**Return Type**	**Deskripsi**
Rumus	determinan(a11, a12, a13, a21, a22, a23, a31, a32, a33): double	double	Menghitung determinan dari matriks 3x3 dengan elemen-elemen yang diberikan.
	determinan(a11, a12, a21, a22): double	double	Menghitung determinan dari matriks 2x2 dengan elemen-elemen yang diberikan.
	balikan(a11, a12, a21, a22): double[*,*]	double [ ][ ]	Menghitung matriks invers dari matriks 2x2 dengan elemen-elemen yang diberikan. Hasilnya berupa array dua dimensi.
	M2(a11, a12, a21, a22, b11, b12, b21, b22, chose): double[*,*]	double [ ][ ]	Melakukan operasi matriks (penjumlahan atau pengurangan) antara dua matriks 2x2 yang elemennya diberikan. Hasilnya berupa array dua dimensi.
	M3(a11, a12, a13, a21, a22, a23, a31, a32, a33, b11, b12, b13, b21, b22, b23, b31, b32, b33, chose): double[*,*]	double [ ][ ]	Melakukan operasi matriks (penjumlahan atau pengurangan) antara dua matriks 3x3 yang elemennya diberikan. Hasilnya berupa array dua dimensi.
	SPL(x1, x2, y1, y2, k1, k2): double[*]	double [ ][ ]	Menyelesaikan sistem persamaan linear (SPL) dengan dua persamaan dan dua variabel menggunakan metode substitusi atau eliminasi. Hasilnya berupa array satu dimensi solusi SPL.

Tabel 2 Rumus Package

Penjumlahan Dan pengurangan

Pada bagian ini menggunakan rumus yang ada di modul pembelajaran Aljabar geometri :

1.1. Penjumlahan dan pengurangan dua buah matriks 2X2

Cm x n = Am x n + Bm x n

Implementasi program :

```
public double[][] M2(double a11, double a12, double a21, double a22, double b11, double b12, double b21, double b22, String chose) {
    double a[][] = {{a11, a12}, {a21, a22}};
    double b[][] = {{b11, b12}, {b21, b22}};
    double c[][] = new double[2][2];

    if ("Penjumlahan".equals(anObject:chose)) {
        for (int i = 0; i < c.length; i++) {
            for (int j = 0; j < c.length; j++) {
                c[i][j] = a[i][j] + b[i][j];
            }
        }
        return c;
    } else {
        for (int i = 0; i < c.length; i++) {
            for (int j = 0; j < c.length; j++) {
                c[i][j] = a[i][j] - b[i][j];
            }
        }
        return c;
    }
}
```

gambar 3 source code rumus.M2()

1.2. Penjumlahan dan pengurangan dua buah matriks 3X3

Cm x n = Am x n - Bm x n

Implementasi program :

```
  public double[][] M3(double a11, double a12, double a13, double a21, double a22, double a23, double a31, double a32, double a33, double b11, double
b12, double b13, double b21, double b22, double b23, double b31, double b32, double b33, String chose) {
      double a[][] = {{a11, a12, a13}, {a21, a22, a23}, {a31, a32, a33}};
      double b[][] = {{b11, b12, b13}, {b21, b22, b23}, {b31, b32, b33}};

      double c[][] = new double[3][3];

      if ("Penjumlahan".equals(anObject:chose)) {
          for (int i = 0; i < c.length; i++) {
              for (int j = 0; j < c.length; j++) {
                  c[i][j] = a[i][j] + b[i][j];
              }
          }
          return c;
      } else {
          for (int i = 0; i < c.length; i++) {
              for (int j = 0; j < c.length; j++) {
                  c[i][j] = a[i][j] - b[i][j];
              }
          }
          return c;
      }
```

gambar 4 source code rumus.M3

Transpose Matriks

Pada bagian transpose rumus yang digunakan adalah :

B = AT

bji = aij i = 1, 2, …m; j = 1, 2, …n

Implementasi Program :

- Transpose Matriks 2x2

```java
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:

    B11.setText(t: A11.getText());
    B21.setText(t: A12.getText());
    B12.setText(t: A21.getText());
    B22.setText(t: A22.getText());

}
```

gambar 5 source code Interface.Transpose

- Transpose Matriks 3x3

```java
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:

    B11.setText(t: A11.getText());
    B12.setText(t: A21.getText());
    B13.setText(t: A31.getText());
    B21.setText(t: A12.getText());
    B22.setText(t: A22.getText());
    B23.setText(t: A32.getText());
    B31.setText(t: A13.getText());
    B32.setText(t: A23.getText());
    B33.setText(t: A33.getText());


}
```

gambar 6 source code interface Transpose

Balikan Matriks (Invers)

Pada bagian balikan matriks atau invers mengambil rumus dari modul aljabar geometri yang diberikan :

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \longrightarrow A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Implementasi pada program :

```java
public double[][] balikan(double a11, double a12, double a21, double a22) {
    double Adjoin[][] = {
        {a22, (a12 - 2 * a12)},
        {(a21 - 2 * a21), a11}};
    double det = determinan(a11, a12, a21, a22);
    double result[][] = new double[2][2];

    if (det == 0) {
        JOptionPane.showMessageDialog(parentComponent: null, message: "Matriks ini tidak memiliki balikan karena det(a)=0");
    } else {

        for (int i = 0; i < result.length; i++) {
            for (int j = 0; j < result.length; j++) {
                System.out.println(result[i][j] = Adjoin[i][j] / det);
            }
        }
        return result;

    }
    return null;

}
```

gambar 7 source code rumus.balikan

Determinan

Determinan matriks 2 x 2

Untuk matriks A berukuran 2 x 2:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

maka det(A) = (a11*a22) – (a12*a21)

Implementasi pada program :

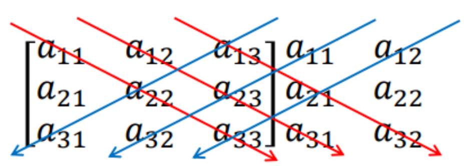
```java
public double determinan(double a11, double a12, double a21, double a22) {
    double result = (a11 * a22) - (a12 * a21);
    return result;
}
```

gambar 8 source code rumus.determinan2

Determinan matriks 3 x 3

Untuk matriks A berukuran 3 x3:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$



maka det(A) = ((a11*a22* a33) + (a12*a21*a31) + (a13*a21*a32)) – ((a13*a22*a31) + (a11*a23*a32) + (a12*a21*a33))

Implemetasi pada program

```
public double determinan(double a11, double a12, double a13, double a21, double a22, double a23, double a31, double a32, double a33) {
    double result = ((a11 * a22 * a33) + (a12 * a23 * a31) + (a13 * a21 * a32)) - ((a13 * a22 * a31) + (a11 * a23 * a32) + (a12 * a21 * a33));
    return result;
}
```

gambar 9 source code rumus.determinan3()

SPL (Sistem Persamaan Linear)

Dalam memecahkan masalah SPL kita menggunakan kaidah cramer

xi = det Bi/det A

keterangan :

Xi = Variabel yang ingin diketahui nilainya det

Bi = Determinan dari matriks di mana vektor kolom ke - i diganti vektor b.

det A = Determinan matriks A

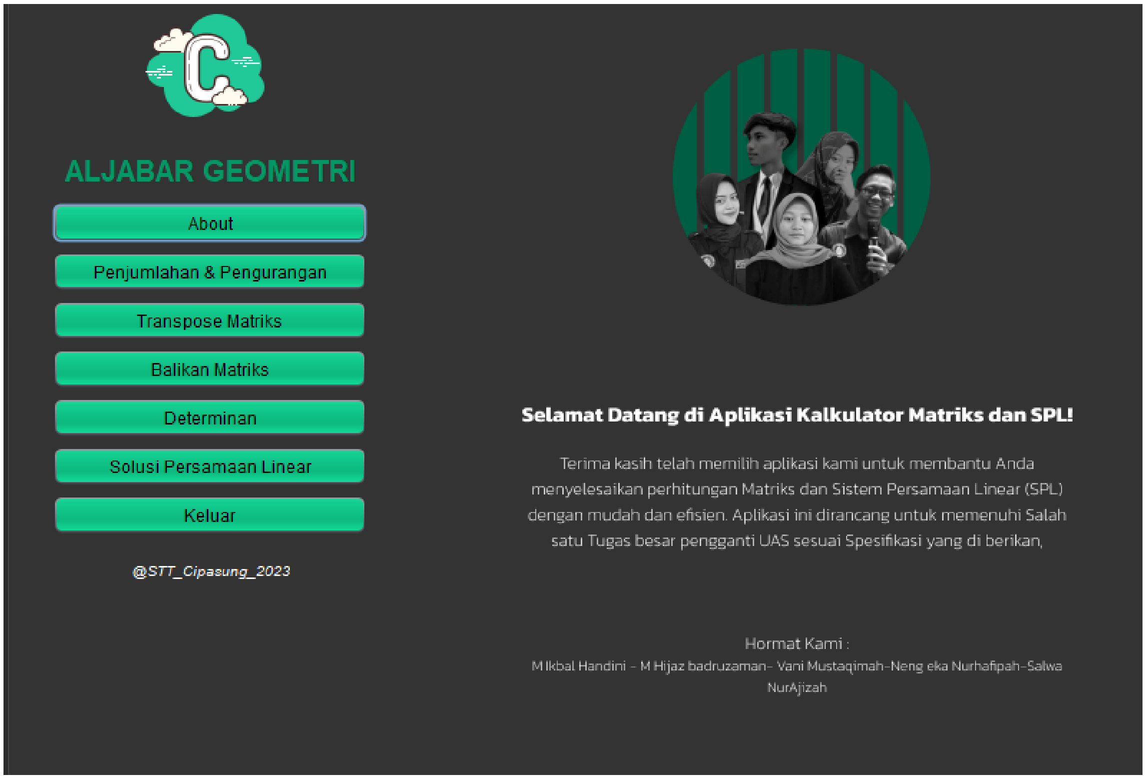
Implementasi pada program :

```
public double[] SPL(double x1, double x2, double y1, double y2, double k1, double k2) {
    double det = determinan(a11: x1, a12: y2, a21: x2, a22: y1);
    double result[] = {((k1 * y2) - (y1 * k2)) / det, ((x1 * k2) - (x2 * k1)) / det};
    return result;
}
```

gambar 10 source code rumus.SPL()

BAB IV
PENGUJIAN

4.1 Menu Program

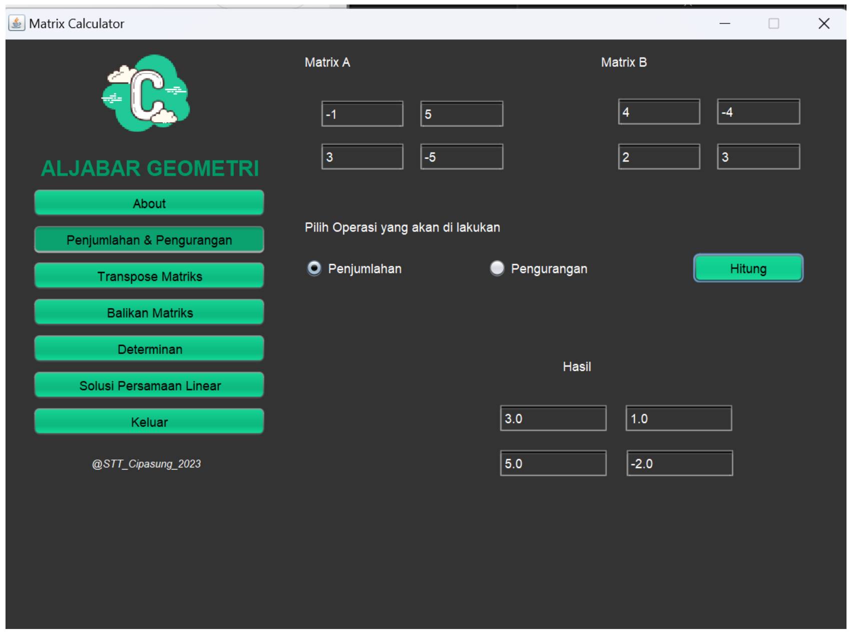


gambar 11 home

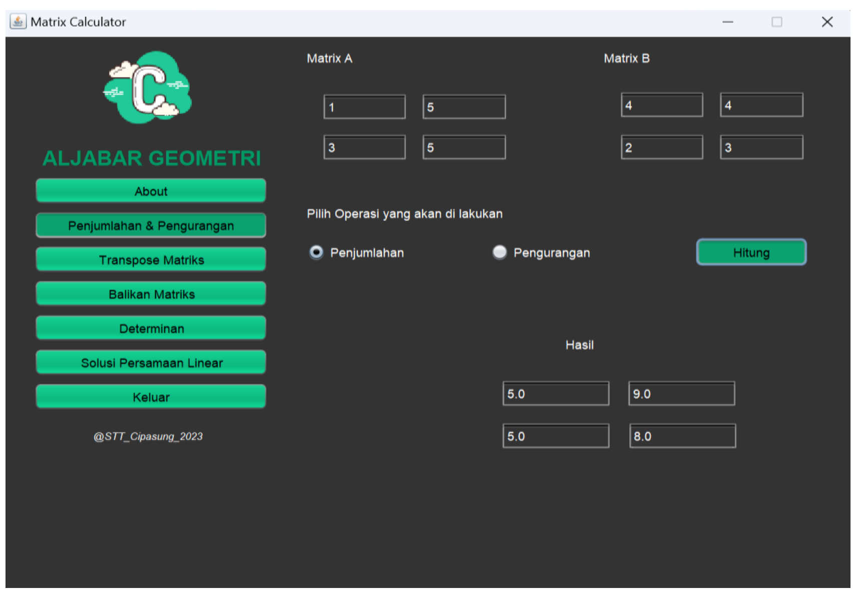
4.3 Penjumlahan dan Pengurangan Matriks

a. Penjumlahan matriks

- Penjumlahan 2x2

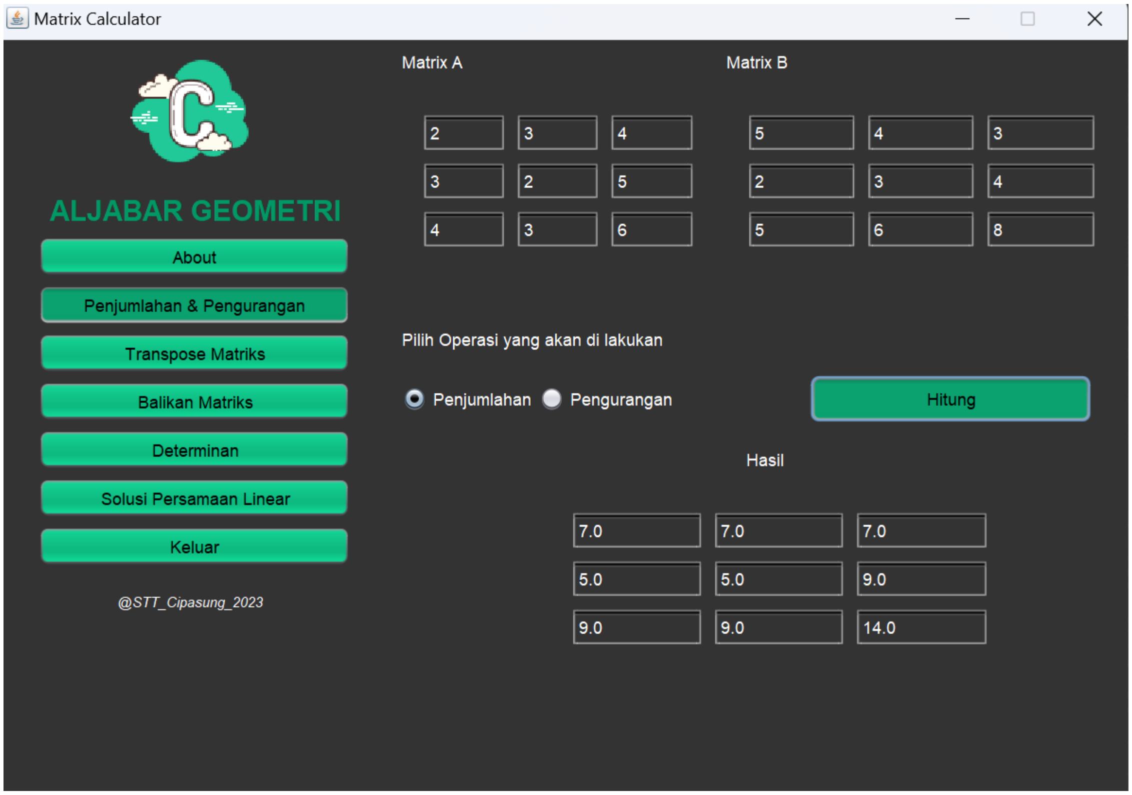


gambar 12 test penjumlahan 1

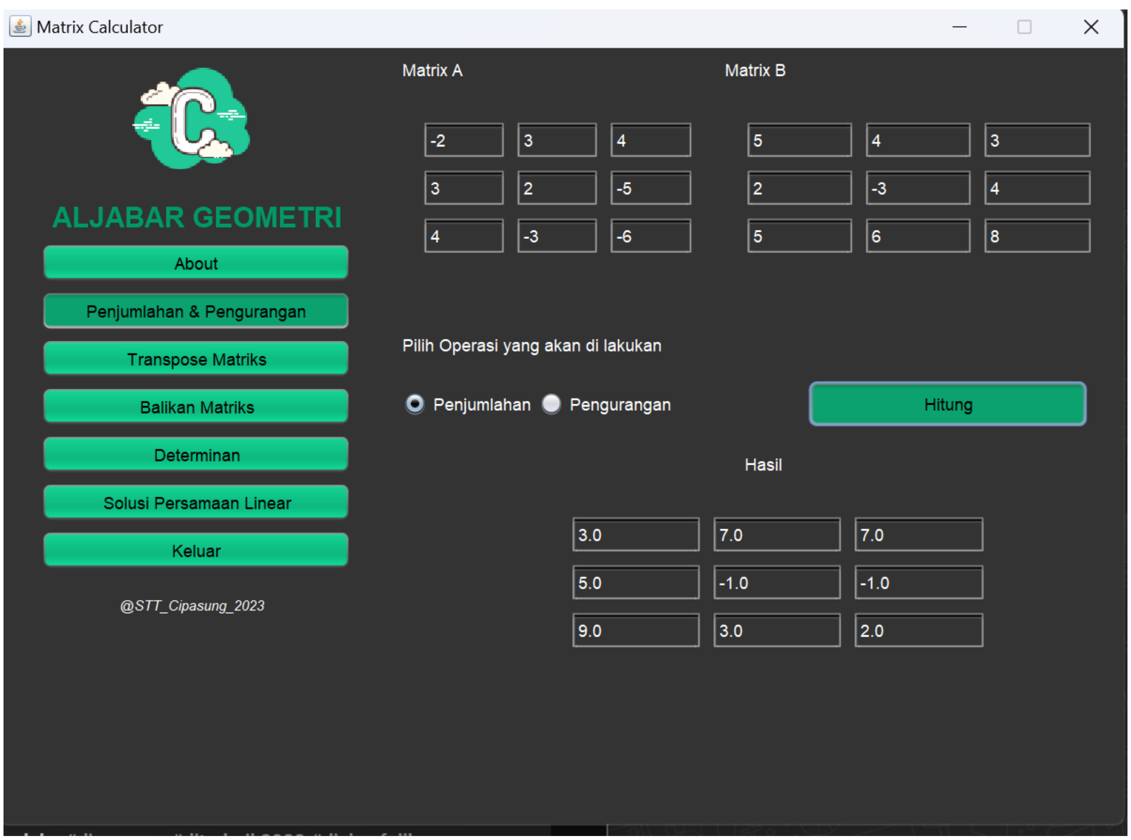


gambar 13 test penjumlahan 2

4.3 Penjumlahan 3x3



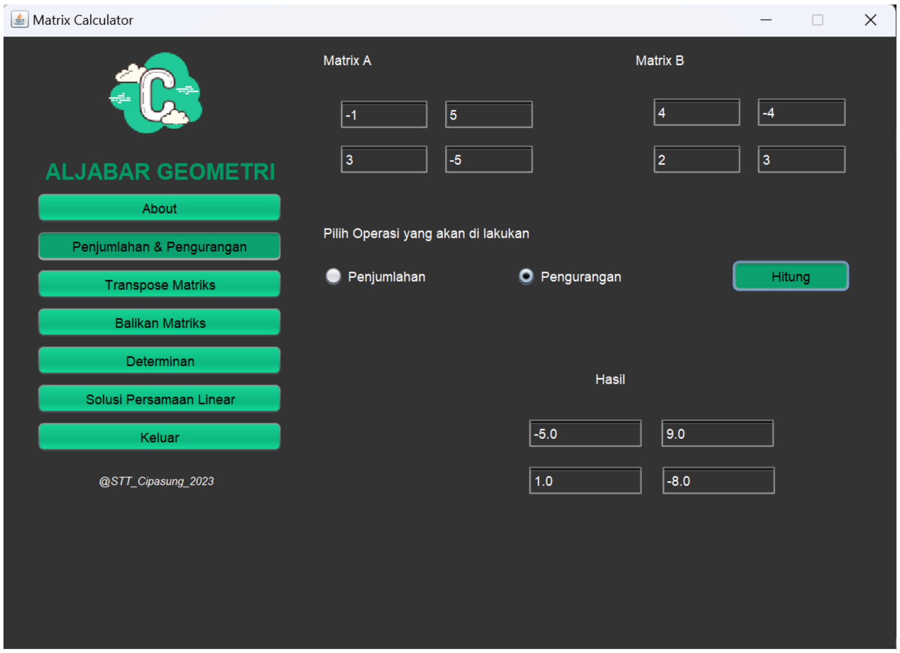
gambar 14 test penjumlahan matriks 3x3



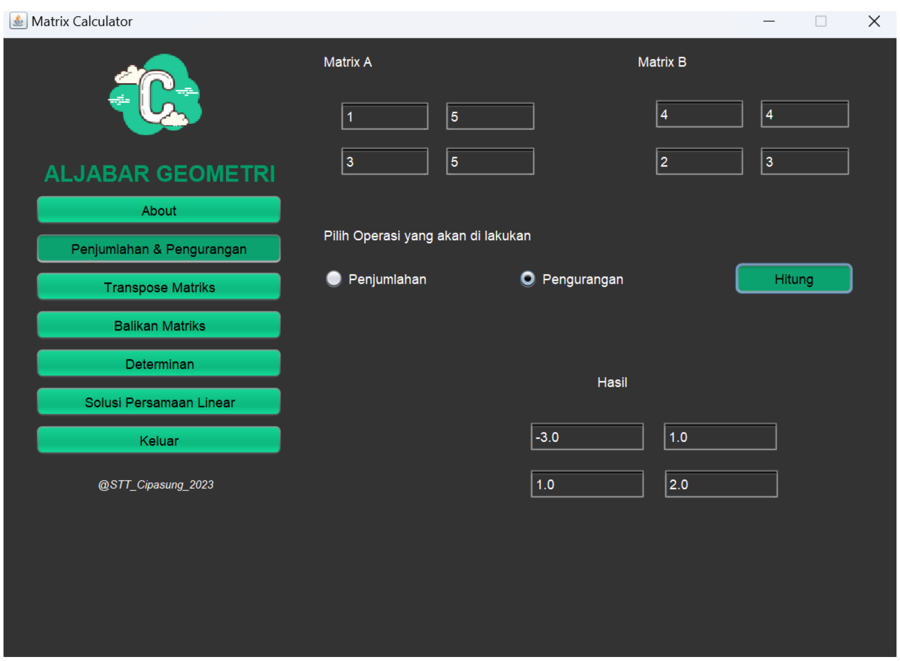
gambar 15 test penjumlahan matriks 3x3

4.4 Pengurangan matriks

- Pengurangan 2x2

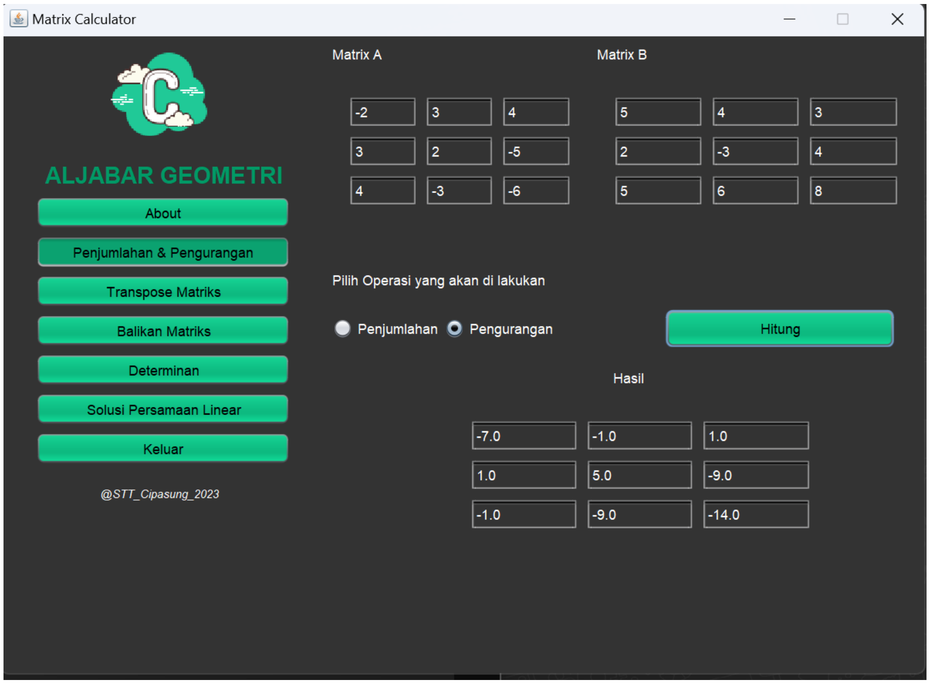


gambar 16 test pengurangan matriks 2x2

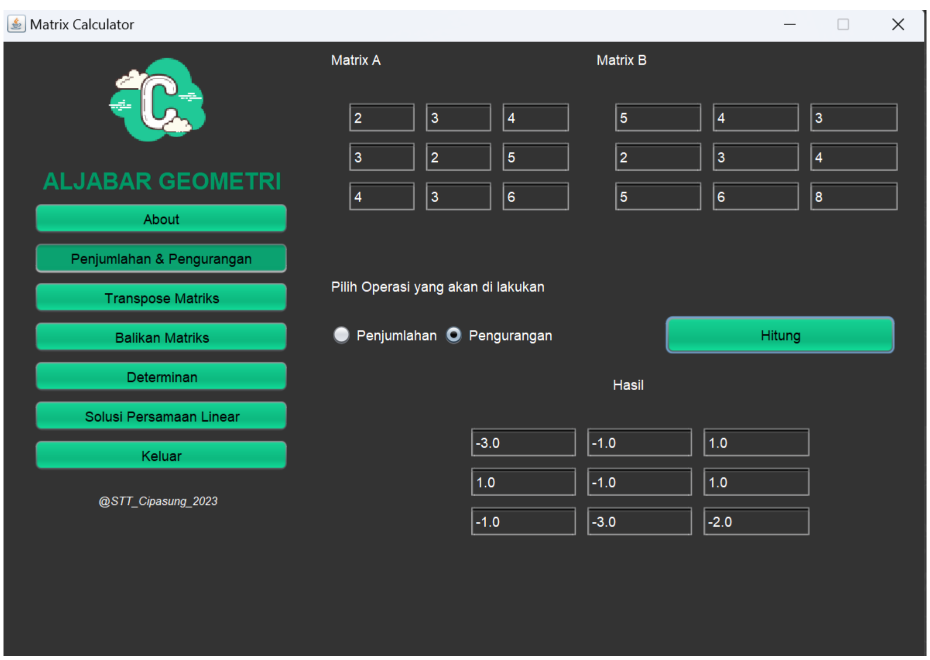


gambar 17 test pengurangan matriks 2x2

- Pengurangan 3x3



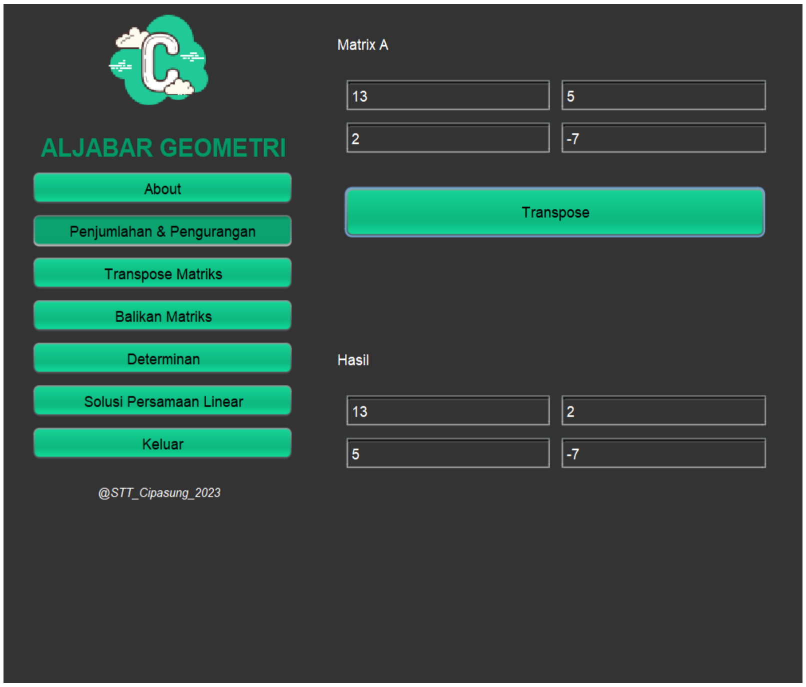
gambar 18 test pengurangan matriks 3x3



gambar 19 test pengurangan matriks 3x3

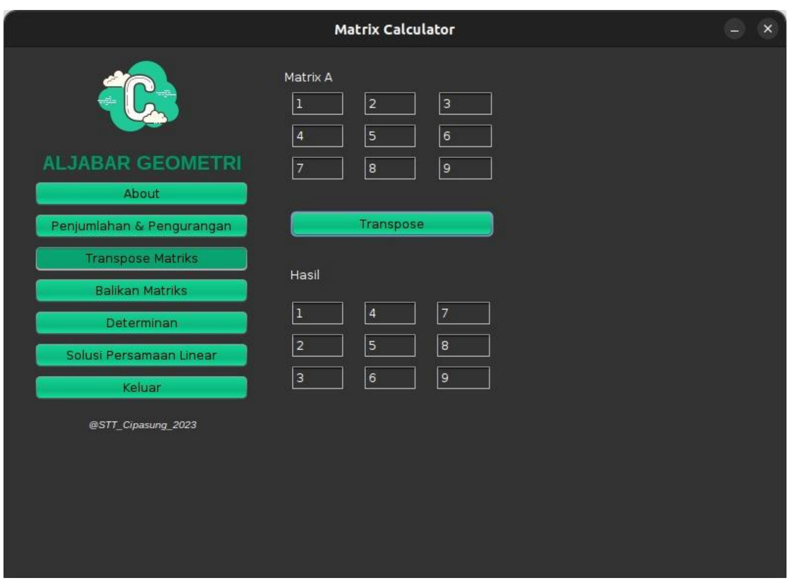
4.2 Matriks Transpose

Matriks 2x2



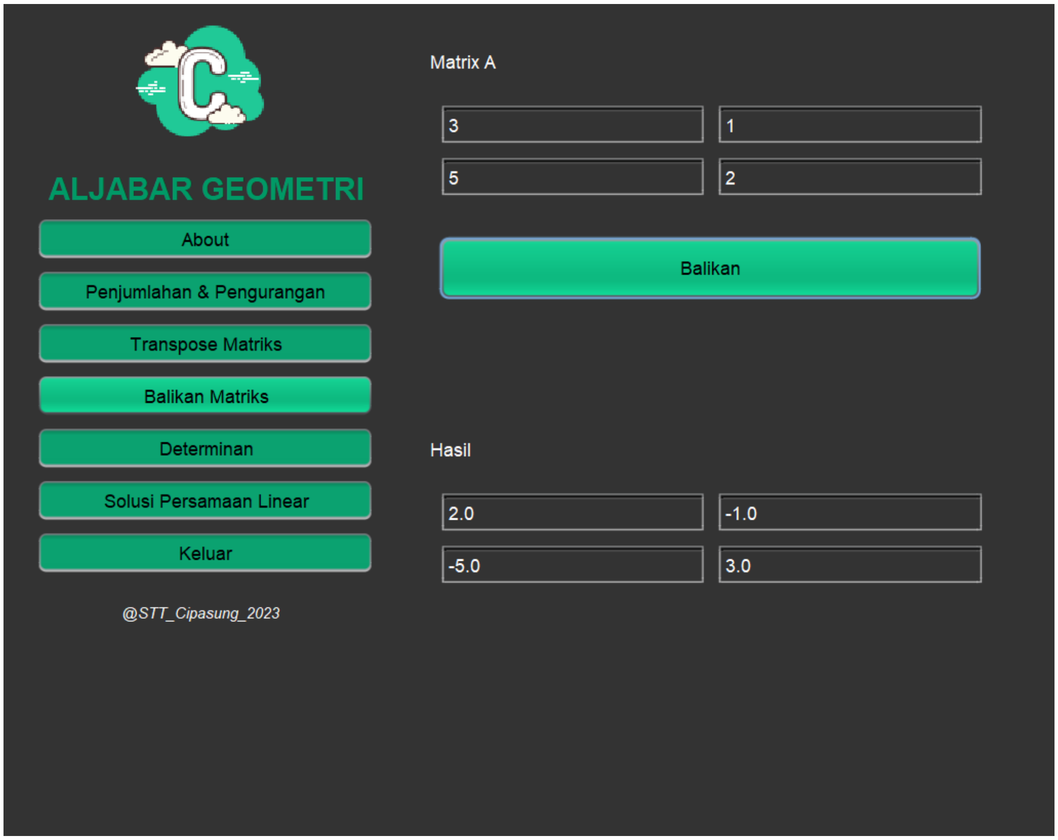
gambar 20 test Transpose matriks

Matriks 3x3



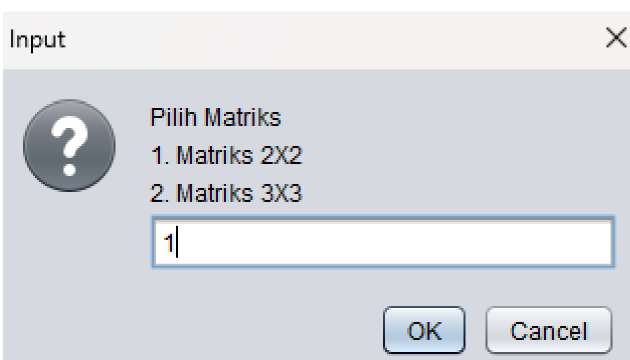
gambar 21 test Transpose matriks 3x3

4.5 Matriks Balikan (invers) (2 x 2)



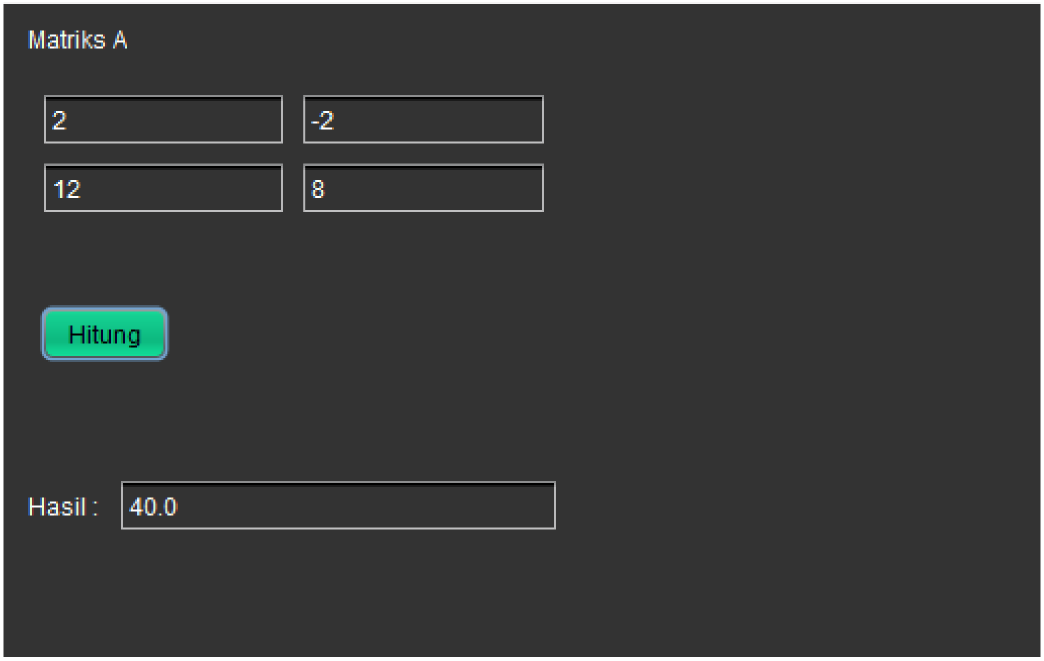
gambar 22 test Matriks balikan

4.6 Determinan Matriks (2 x 2) dan (3x3)



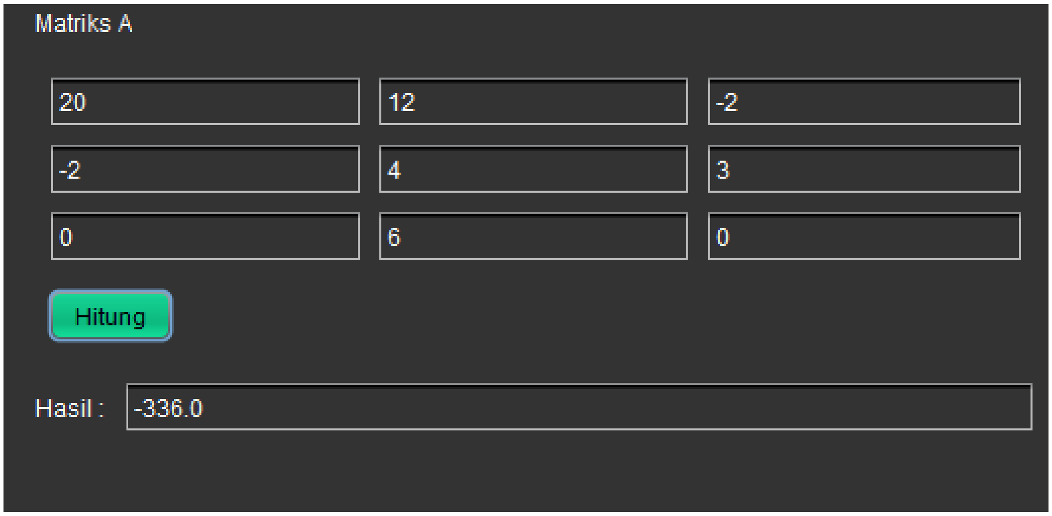
gambar 23 JOptionpane chose

Determinan Matriks 2x2 :



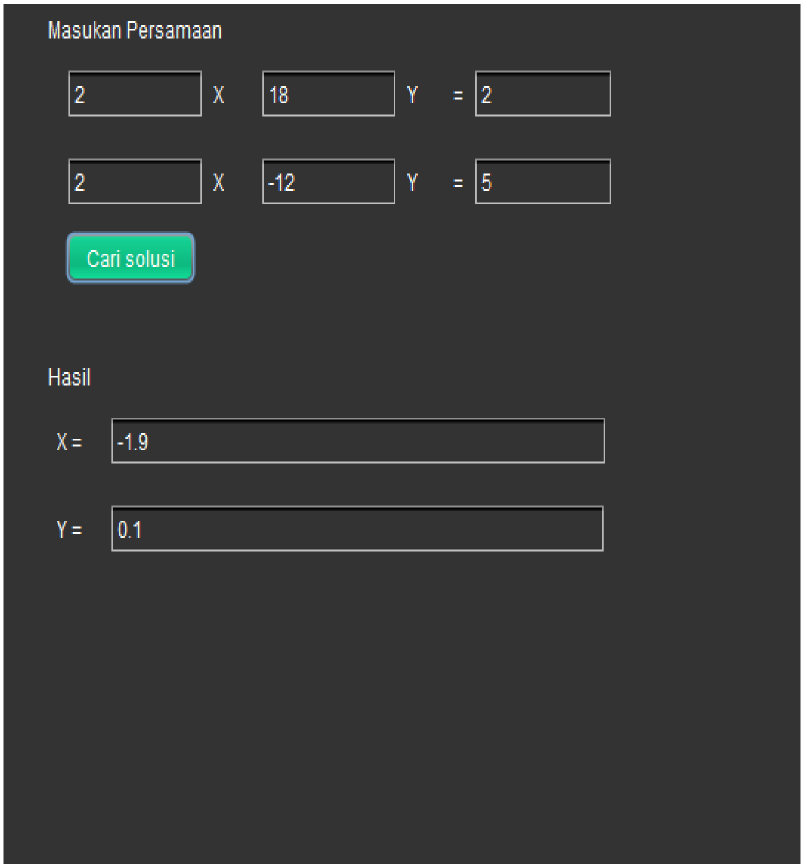
gambar 24 test determinan matriks 2x2

Determinan Matriks (3x 3) :



gambar 25 test determinan matriks 3x3

4.7 Solusi Persamaan Linear



gambar 26 test Sistem Persamaan Linier

BAB V
KESIMPULAN , SARAN DAN REFLEKSI

Kesimpulan

Program-program yang dibuat pada tugas ini:

1. Menghitung penjumlahan dan pengurangan matriks (2 x 2)
2. Menghitung matriks transpose (2 x 2)
3. Menghitung matriks balikan (invers) (2 x 2)
4. Menghitung determinan matriks (2 x 2) dan (3x3)
5. Menghitung solusi Sistem Persamaan Linier (SPL) (2x3)

Saran

Program ini memiliki beberapa kekurangan yang perlu diperhatikan. Pertama, program belum mampu menampilkan detail cara pengerjaan secara langsung setelah menyelesaikan soal. Hal ini dapat meningkatkan pemahaman pengguna terhadap langkah-langkah yang dilakukan oleh program.

Penting juga untuk menambahkan opsi metode pengerjaan, sehingga pengguna dapat memilih metode yang paling sesuai dengan kebutuhan atau preferensi mereka. Dengan memberikan pilihan ini, program dapat menjadi lebih beragam dan dapat diakses oleh berbagai tingkat pengguna.

Terakhir, untuk meningkatkan keterhubungan program dengan pengguna, sebaiknya ditambahkan kemampuan untuk menerima inputan bertipe dokumen, seperti file .txt atau Word. Dengan adanya fitur ini, pengguna dapat dengan mudah menggunakan data yang sudah ada dalam format dokumen tanpa perlu menyalin atau mengetik ulang informasi tersebut.

Dengan mengatasi kekurangan-kekurangan ini, program dapat menjadi lebih komprehensif, mudah digunakan, dan dapat memenuhi berbagai kebutuhan pengguna dengan lebih baik.

Refleksi

Tugas ini membuat kita semua belajar untuk berkolaborasi secara tim, menggunakan bahasa pemrograman Java untuk menyelesaikan permasalahan linier, dan penggunaan Github untuk berkolaborasi secara tim.

DAFTAR PUSTAKA

Intan, & Fajri, D. L. (2023, February 24). Memahami Rumus Metode Cramer, Contoh Soal, Dan Pembahasan. Retrieved from https://katadata.co.id/intan/lifestyle/63f8aad56d3a1/memahami-rumus-metode-cramer-contoh-soal-dan-pembahasan?page=2

Munir. (2022). AljabarGeometri. Retrieved from https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2022-2023/algeo22-23.htm

Nursyamsi. (2022). *MODUL PRAKTIKUM OBJECT ORIENTED PROGRAMMING JAVA PROGRAMING LANGUAGE. FAKULTAS KOMUNIKASI DAN INFORMSASI UNIVERSITAS GARUT* [PDF].