

**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 ISI.....	i
DAFTAR GAMBAR.....	ii
DAFTAR TABEL.....	iii
BAB 1 DESKRIPSI MASALAH.....	1
BAB II TEORI SINGKAT	3
2.1 Sistem Persamaan Linier (Metode Eliminasi Gauss, Metode Eliminasi Gauss – Jordan).....	3
2.2 Determinan.....	8
2.3 Matriks Balikan	10
2.4 Transpose Matriks	10
2.5 Penjumlahan Matriks	12
BAB III PENJELASAN IMPLEMENTASI PROGRAM.....	13
BAB IV PENGUJIAN.....	28
4.1 Menu Program.....	28
4.3 Penjumlahan dan Pengurangan Matriks	29
4.3 Penjumlahan 3x3.....	30
4.4 Pengurangan matriks.....	31
4.5 Matriks Balikan (invers) (2 x 2)	35
4.6 Determinan Matriks (2 x 2) dan (3x3).....	35
4.7 Solusi Persamaan Linear.....	38
BAB V KESIMPULAN , SARAN DAN REFLEKSI.....	39
DAFTAR PUSTAKA.....	41

DAFTAR GAMBAR

gambar 1 User flow.....	13
gambar 2 Class Diagram.....	16
gambar 3 source code rumus.M2()	24
gambar 4 source code rumus.M3	24
gambar 5 source code Interface.Transpose	25
gambar 6 source code interface Transpose	25
gambar 7 source code rumus.balikan	26
gambar 8 source code rumus.determinan2.....	26
gambar 9 source code rumus.determinan3()	27
gambar 10 source code rumus.SPL()	27
gambar 11 home.....	28
gambar 12 test penjumlahan 1	29
gambar 13 test penjumlahan 2	29
gambar 14 test penjumlahan matriks 3x3	30
gambar 15 test penjumlahan matriks 3x3	30
gambar 16 test pengurangan matriks 2x2	31
gambar 17 test pengurangan matriks 2x2	31
gambar 18 test pengurangan matriks 3x3	32
gambar 19 test pengurangan matriks 3x3	32
gambar 20 test Transpose matriks.....	33
gambar 21 test Transpose matriks 3x3.....	33
gambar 22 test Matriks balikan	35
gambar 23 JOptionpane chose	35
gambar 24 test determinan matriks 2x2	36
gambar 25 test determinan matriks 3x3	37
gambar 26 test Sistem Persamaan Linier	38

DAFTAR TABEL

Tabel 1 Package Interface	22
Tabel 2 Rumus Package.....	23

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 x_1, x_2, \dots, x_n berbentuk:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m$$

atau dalam bentuk $Ax = b$

- SPL dalam bentuk matriks:

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

atau dalam bentuk perkalian matriks: Axb

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

- 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: $x_i = \det B_i / \det A$

keterangan :

x_i = Variabel yang ingin diketahui nilainya

B_i = 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.

$$\begin{aligned} D &= \begin{vmatrix} 2 & -3 \\ 1 & 2 \end{vmatrix} = 2(2) - (-3)(1) \\ &= 4 + 3 = 7 \end{aligned}$$

$$\begin{aligned} D_x &= \begin{vmatrix} -13 & -3 \\ 4 & 2 \end{vmatrix} = -13(2) - (-3)(4) \\ &= -26 + 12 = -14 \end{aligned}$$

$$\begin{aligned} D_y &= \begin{vmatrix} 2 & -13 \\ 1 & 4 \end{vmatrix} = 2(4) - (-13)(1) \\ &= 8 + 13 = 21 \end{aligned}$$

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:

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

• Contoh:

$$\begin{array}{rcl} x_1 + 3x_2 - 6x_3 & = & 9 \\ 2x_1 - 6x_2 + 4x_3 & = & 7 \\ 5x_1 + 2x_2 - 5x_3 & = & -2 \end{array} \quad \longrightarrow \quad \left[\begin{array}{ccc|c} 1 & 3 & -6 & 9 \\ 2 & -6 & 4 & 7 \\ 5 & 2 & -5 & -2 \end{array} \right]$$

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 :

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

- 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:

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

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

.....

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

$$x_1 = \frac{1}{c_{11}} (d_1 - c_{12}x_2 - c_{13}x_3 - \dots - c_{1n}x_n)$$

Metode Eliminasi Gauss-Jordan

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

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

- Tidak diperlukan lagi substitusi secara mundur untuk memperoleh nilai-nilai 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

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

2. Fase mundur (backward phase)

➤ Menghasilkan nilai-nilai 0 di atas satu utama.

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

Matriks eselon baris tereduksi

Dari matriks augmented terakhir, diperoleh $x_1 = 1$, $x_2 = 2$, $x_3 = 3$

2.2 Determinan

Definisi

- Misalkan A adalah matriks berukuran $n \times n$

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

- Determinan matriks A dilambangkan dengan

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

Determinan matriks 2 x 2

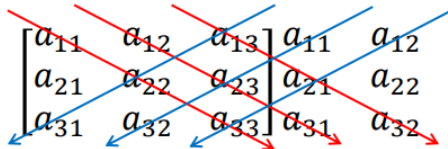
Untuk matriks A berukuran 2×2 :

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

maka $\det(A) = a_{11}a_{22} - a_{12}a_{21}$

Determinan matriks 3 x 3

Untuk matriks A berukuran 3 x 3:

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


maka $\det(A) = (a_{11}a_{22}a_{33} + a_{12}a_{21}a_{31} + a_{13}a_{21}a_{32}) - (a_{13}a_{22}a_{31} + a_{11}a_{23}a_{32} + a_{12}a_{21}a_{33})$

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 $n \times n$,

$$\det(A) = a_{11}a_{22}a_{33}\dots a_{nn}$$

Aturan Determinan

- Misalkan A adalah matriks $n \times n$. 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 $\xrightarrow{\text{Sebuah baris ditambahkan dengan } k \text{ kali baris yang lain}}$ 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

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

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

- Balikan matriks A disimbolkan dengan A^{-1}
- Sifat: $AA^{-1} = A^{-1}A = I$
- Untuk matriks A berukuran 2×2 , 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 $ad - bc \neq 0$

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

2.4 Transpose Matriks

- Transpose matriks,

$$B = A^T$$

$$b_{ji} = a_{ij} \quad i = 1, 2, \dots, m; j = 1, 2, \dots, n$$

- Algoritma transpose matriks:

```

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

```

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

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

- Untuk matriks persegi A berukuran $n \times n$, 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

$$(a) \quad (A^T)^T = A$$

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

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

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

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

2.5 Penjumlahan Matriks

- Penjumlahan dua buah matriks $C_{m \times n} = A_{m \times n} + B_{m \times n}$ Misal $A = [a_{ij}]$ $B = [b_{ij}]$ maka $C = A + B = [c_{ij}]$, $c_{ij} = a_{ij} + b_{ij}$, $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$
- Pengurangan matriks: $C = A - B = [c_{ij}]$, $c_{ij} = a_{ij} - b_{ij}$, $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$
- Algoritma penjumlahan dua buah matriks:

```
for i ← 1 to m do
  for j ← 1 to n do
     $c_{ij} \leftarrow a_{ij} + b_{ij}$ 
  end for
end for
```

Contoh :

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

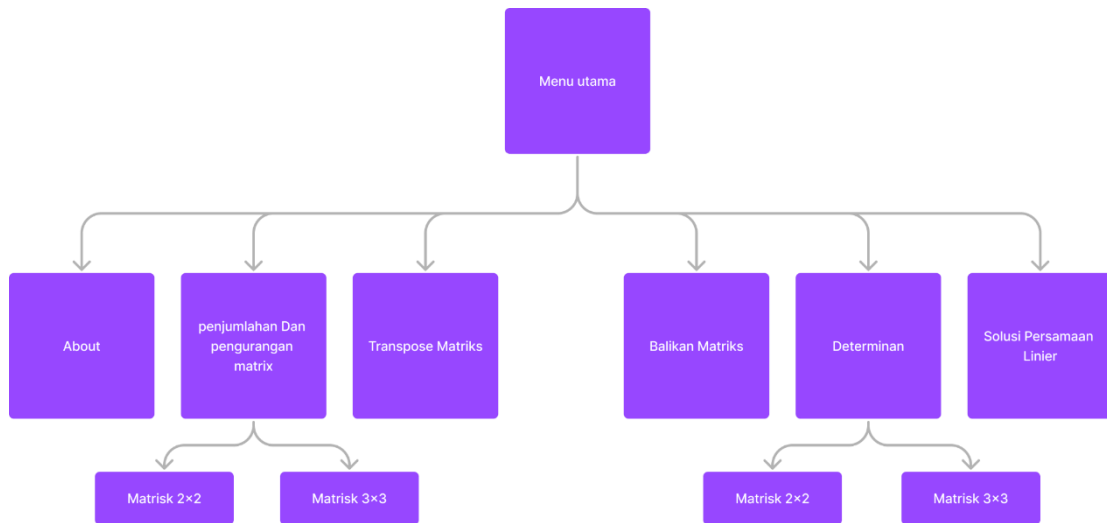
- Maka,

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

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

Berikut adalah garis besar user flow atau alur pengguna untuk aplikasi kalkulator matriks dan SPL:

- a) Pengguna membuka aplikasi:
 - Aplikasi membuka antarmuka pengguna yang menampilkan berbagai panel untuk operasi matriks dan SPL.
- b) Pengguna memilih operasi atau panel yang diinginkan:
 - Pengguna memilih salah satu panel (misalnya, penjumlahan matriks, SPL, determinan, dll.) atau memilih jenis operasi yang ingin dilakukan.
- c) Pengguna memasukkan nilai:
 - Jika memilih panel matriks, pengguna memasukkan nilai elemen matriks sesuai dengan panel yang dipilih.
 - Jika memilih SPL, pengguna memasukkan nilai variabel dan konstanta pada masing-masing persamaan.
- d) Pengguna memilih tindakan:
 - Pengguna memilih tindakan yang ingin dilakukan, seperti menghitung determinan, menyelesaikan SPL, atau melakukan operasi lainnya.
- e) Pengguna mengeksekusi perhitungan:
 - Pengguna menekan tombol "Hitung" atau tindakan serupa untuk memulai perhitungan.
- f) Aplikasi melakukan perhitungan:
 - Aplikasi menggunakan rumus dan logika yang sesuai untuk melakukan perhitungan berdasarkan nilai yang dimasukkan.
- g) Hasil ditampilkan:
 - Hasil perhitungan ditampilkan pada panel atau area yang sesuai dalam antarmuka pengguna.
 - Hasil dapat berupa nilai determinan, solusi SPL, hasil operasi matriks, atau informasi lainnya.
- h) Pengguna dapat melakukan operasi berikutnya atau keluar:
 - Pengguna dapat memilih untuk melakukan operasi berikutnya dengan memilih panel atau jenis operasi lainnya.
 - Pengguna dapat keluar dari aplikasi jika sudah selesai.
- i) Aplikasi memberikan feedback:

- Aplikasi memberikan feedback kepada pengguna, seperti pesan kesalahan jika ada, atau konfirmasi berhasil jika perhitungan selesai dengan sukses.
- j) Aplikasi tetap responsif dan interaktif:
 - Aplikasi tetap responsif terhadap input pengguna dan memberikan respons visual yang jelas.

Garis besar ini memberikan gambaran umum tentang cara pengguna dapat berinteraksi dengan aplikasi dari awal hingga akhir proses perhitungan. Setiap langkah dapat diuraikan lebih lanjut berdasarkan kompleksitas dan detail implementasi aplikasi yang diinginkan.

class Main

```

    -About: JToggleButton
    -BTN_Determinan: JToggleButton
    -BTN_Keluar: JToggleButton
    -BTN_PenjumlahanPengurangan: JToggleButton
    -BTN_SPL: JToggleButton
    -BTN_Transpose: JToggleButton
    -BTN_Balkan: JToggleButton
    -IMG_Logo: JLabel
    -LB_Footer: JLabel
    -Lb_Title: JLabel

    +constructor=PN_About()
    +initComponents(); void

    +PN_About
        (from Interface)

    +PN_Main
        (from Interface)

    +PN_MatriksX3
        (from Interface)

    +PN_DetX3
        (from Interface)

    +PN_DetX2
        (from Interface)

    +PN_MatriksX2
        (from Interface)

    +PN_SPL
        (from Interface)

    +PN_Balkan
        (from Interface)

    +PN_Transpose
        (from Interface)

    +Rumus
        (from Rumus)
  
```

PN_About
(from Interface)

```

    -IMG_About: JLabel
    -IMG_Pic: JLabel

    +constructor=PN_About()
    +initComponents(); void
  
```

PN_Main
(from Interface)

```

    -About: JToggleButton
    -BTN_Determinan: JToggleButton
    -BTN_Keluar: JToggleButton
    -BTN_PenjumlahanPengurangan: JToggleButton
    -BTN_SPL: JToggleButton
    -BTN_Transpose: JToggleButton
    -BTN_Balkan: JToggleButton
    -IMG_Logo: JLabel
    -LB_Footer: JLabel
    -Lb_Title: JLabel

    +constructor=PN_Main()
    +initComponents(); void
    +formWindowOpened(evt: WindowEvent): void
    +BTN_PenjumlahanPenguranganMouseClicked(evt: MouseEvent): void
    +BTN_SPLActionPerformed(evt: ActionEvent): void
    +BTN_TransposeMouseClicked(evt: MouseEvent): void
    +BTN_BalkanMouseClicked(evt: MouseEvent): void
    +BTN_DeterminanMouseClicked(evt: MouseEvent): void
    +BTN_KeluarActionPerformed(evt: ActionEvent): void
    +BTN_KeluarMouseClicked(evt: MouseEvent): void
    +BTN_PenjumlahanPenguranganActionPerformed(evt: ActionEvent): void
    +BTN_SPLMouseClicked(evt: MouseEvent): void
    +AboutHouseClicked(evt: MouseEvent): void
    +AboutActionPerformed(evt: ActionEvent): void
    +main(args: String[]): void
  
```

PN_MatriksX3
(from Interface)

```

    -A11: JTextField
    -A12: JTextField
    -A13: JTextField
    -A21: JTextField
    -A22: JTextField
    -A23: JTextField
    -A31: JTextField
    -A32: JTextField
    -A33: JTextField
    -Hasil: JTextField
    -Button1: JButton
    -Label1: JLabel
    -Label2: JLabel

    +constructor=PN_MatriksX3()
    +initComponents(); void
    +OP_penjumlahanActionPerformed(evt: ActionEvent): void
    +OP_penguranganActionPerformed(evt: ActionEvent): void
    +HitungMouseClicked(evt: MouseEvent): void
    +A32ActionPerformed(evt: ActionEvent): void
  
```

PN_DetX3
(from Interface)

```

    -A11: JTextField
    -A12: JTextField
    -A13: JTextField
    -A21: JTextField
    -A22: JTextField
    -A23: JTextField
    -A31: JTextField
    -A32: JTextField
    -A33: JTextField
    -Hasil: JTextField
    -Button1: JButton
    -Label1: JLabel
    -Label2: JLabel

    +constructor=PN_DetX3()
    +initComponents(); void
    +HasilActionPerformed(evt: ActionEvent): void
    +Button1ActionPerformed(evt: ActionEvent): void
    +A32ActionPerformed(evt: ActionEvent): void
    +A33ActionPerformed(evt: ActionEvent): void
    +A11MouseClicked(evt: MouseEvent): void
    +A12ActionPerformed(evt: ActionEvent): void
    +A11ActionPerformed(evt: ActionEvent): void
  
```

PN_DetX2
(from Interface)

```

    -A11: JTextField
    -A12: JTextField
    -A21: JTextField
    -A22: JTextField
    -Hasil: JTextField
    -Hitung: JButton
    -Label1: JLabel
    -Label2: JLabel

    +constructor=PN_DetX2()
    +initComponents(); void
    +HasilActionPerformed(evt: ActionEvent): void
    +HitungActionPerformed(evt: ActionEvent): void
    +HitungMouseClicked(evt: MouseEvent): void
  
```

PN_MatriksX2
(from Interface)

```

    -A11: JTextField
    -A12: JTextField
    -A21: JTextField
    -B11: JTextField
    -B12: JTextField
    -B21: JTextField
    -B22: JTextField
    -C11: JTextField
    -C12: JTextField
    -C21: JTextField
    -C22: JTextField
    -Hitung: JToggleButton
    -LB_Hasil: JLabel
    -LB_MatriksA: JLabel
    -OP_pengurangan: JRadioButton
    -OP_penjumlahan: JRadioButton
    -Label1: JLabel
    -Label2: JLabel

    +constructor=PN_MatriksX2()
    +initComponents(); void
    +B11ActionPerformed(evt: ActionEvent): void
    +OP_penjumlahanActionPerformed(evt: ActionEvent): void
    +OP_penguranganActionPerformed(evt: ActionEvent): void
    +HitungMouseClicked(evt: MouseEvent): void
    +A21ActionPerformed(evt: ActionEvent): void
  
```

PN_SPL
(from Interface)

```

    -K1: JTextField
    -K2: JTextField
    -R1: JTextField
    -R2: JTextField
    -X1: JTextField
    -X2: JTextField
    -Y1: JTextField
    -Y2: JTextField
    -Button1: JButton
    -Label10: JLabel
    -Label11: JLabel
    -Label12: JLabel
    -Label13: JLabel
    -Label14: JLabel
    -Label6: JLabel
    -Label7: JLabel
    -Label8: JLabel
    -Label9: JLabel

    +constructor=PN_SPL()
    +initComponents(); void
    +X1ActionPerformed(evt: ActionEvent): void
    +K1ActionPerformed(evt: ActionEvent): void
    +Y1ActionPerformed(evt: ActionEvent): void
    +X2ActionPerformed(evt: ActionEvent): void
    +Y2ActionPerformed(evt: ActionEvent): void
    +K2ActionPerformed(evt: ActionEvent): void
    +R1ActionPerformed(evt: ActionEvent): void
    +R2ActionPerformed(evt: ActionEvent): void
    +Button1MouseClicked(evt: MouseEvent): void
  
```

PN_Balkan
(from Interface)

```

    -A11: JTextField
    -A12: JTextField
    -A21: JTextField
    -A22: JTextField
    -B11: JTextField
    -B12: JTextField
    -B21: JTextField
    -B22: JTextField
    -Button1: JButton
    -Label1: JLabel
    -Label2: JLabel

    +constructor=PN_Balkan()
    +initComponents(); void
    +Button1ActionPerformed(evt: ActionEvent): void
    +Button1MouseClicked(evt: MouseEvent): void
  
```

PN_Transpose
(from Interface)

```

    -A11: JTextField
    -A12: JTextField
    -A21: JTextField
    -A22: JTextField
    -B11: JTextField
    -B12: JTextField
    -B21: JTextField
    -B22: JTextField
    -LB_Hasil: JLabel
    -LB_MatA: JLabel
    -Button1: JButton

    +constructor=PN_Transpose()
    +initComponents(); void
    +Button1ActionPerformed(evt: ActionEvent): void
    +Button1MouseClicked(evt: MouseEvent): void
    +A11ActionPerformed(evt: ActionEvent): void
  
```

Rumus
(from Rumus)

```

    +determinan(a11: double, a12: double, a13: double, a21: double, a22: double, a23: double, a31: double, a32: double, a33: double): double
    +determinan(a11: double, a12: double, a21: double, a22: double): double
    +balkan(a11: double, a12: double, a21: double, a22: double, b11: double, b12: double, b21: double, b22: double, chose: String): double[""]
    +M2(a11: double, a12: double, a21: double, a22: double, b11: double, b12: double, b21: double, b22: double, chose: String): double[""]
    +M3(a11: double, a12: double, a31: 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: double, chose: String): double[""]
    +SPL(x1: double, x2: double, y1: double, y2: double, k1: double, k2: double): double[""]
  
```

16

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_penjumlahan: 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

$$C_{m \times n} = A_{m \times n} + B_{m \times n}$$

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(chose)) {
        for (int i = 0; i < c.length; i++) {
            for (int j = 0; j < c[0].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[0].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

$$C_{m \times n} = A_{m \times n} - B_{m \times n}$$

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(chose)) {
        for (int i = 0; i < c.length; i++) {
            for (int j = 0; j < c[0].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[0].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 :

$$B = A^T$$

$$b_{ji} = a_{ij} \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n$$

Implementasi Program :

- Transpose Matriks 2x2

```
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

```
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 :

```

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 $\det(A) = (a_{11} * a_{22}) - (a_{12} * a_{21})$

Implementasi pada program :

```

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

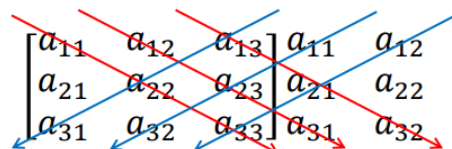
```

gambar 8 source code rumus.determinan2

Determinan matriks 3 x 3

Untuk matriks A berukuran 3 x 3:

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



maka $\det(A) = ((a_{11} \cdot a_{22} \cdot a_{33}) + (a_{12} \cdot a_{21} \cdot a_{31}) + (a_{13} \cdot a_{21} \cdot a_{32})) - ((a_{13} \cdot a_{22} \cdot a_{31}) + (a_{11} \cdot a_{23} \cdot a_{32}) + (a_{12} \cdot a_{21} \cdot a_{33}))$

Implementasi 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

$x_i = \det B_i / \det A$

keterangan :

X_i = Variabel yang ingin diketahui nilainya \det

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

$\det A$ = Determinan matriks A

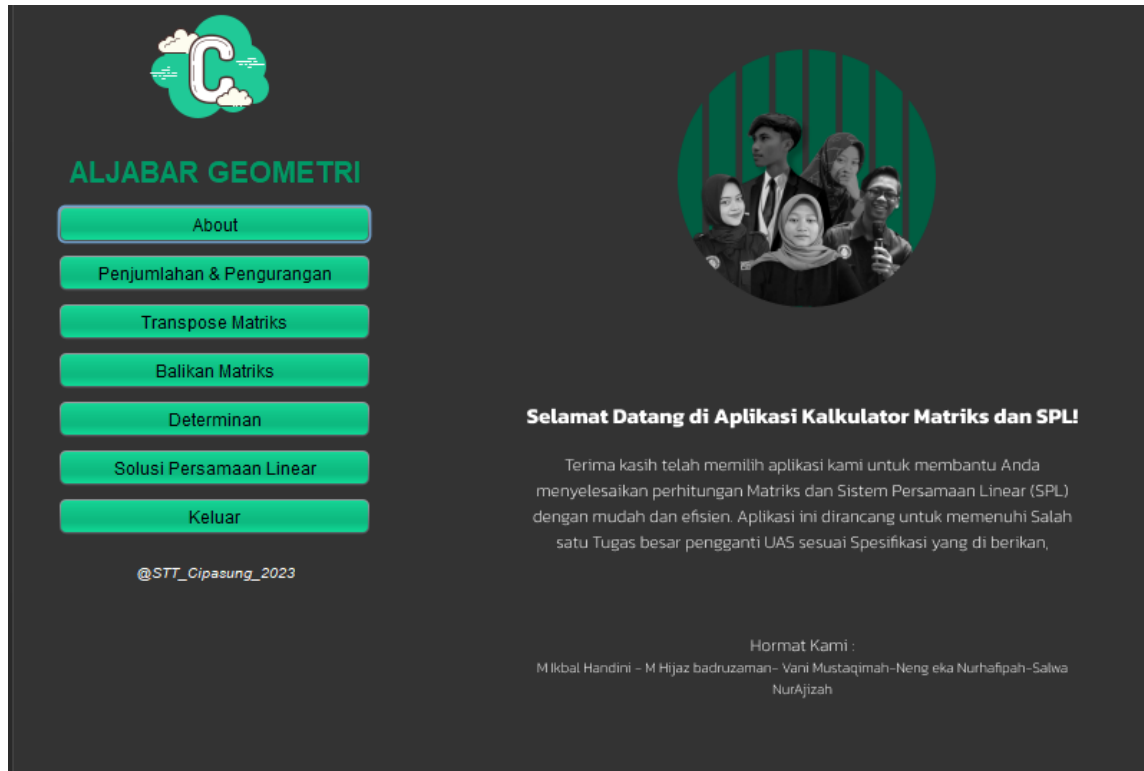
Implementasi pada program :

```
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

The screenshot shows the 'Matrix Calculator' application window. On the left is a sidebar with a logo and a menu of operations: About, Penjumlahan & Pengurangan (selected), Transpose Matriks, Balikan Matriks, Determinan, Solusi Persamaan Linear, and Keluar. The main area is titled 'Matrix A' and 'Matrix B'. Matrix A contains the values [-1, 5; 3, -5]. Matrix B contains the values [4, -4; 2, 3]. Below the matrices, there are radio buttons for 'Penjumlahan' (selected) and 'Pengurangan'. A 'Hitung' button is to the right. The 'Hasil' section shows the result of the addition: [3.0, 1.0; 5.0, -2.0]. The footer of the sidebar includes the text '@STT_Cipasung_2023'.


gambar 12 test penjumlahan 1

This screenshot shows the same 'Matrix Calculator' application. In this instance, Matrix A contains the values [1, 5; 3, 5] and Matrix B contains the values [4, 4; 2, 3]. The 'Penjumlahan' radio button is still selected. The 'Hitung' button is visible. The 'Hasil' section displays the result of the addition: [5.0, 9.0; 5.0, 8.0]. The sidebar menu and footer remain the same as in the previous screenshot.

gambar 13 test penjumlahan 2

4.3 Penjumlahan 3x3

Matrix Calculator



ALJABAR GEOMETRI

- About
- Penjumlahan & Pengurangan
- Transpose Matriks
- Balikan Matriks
- Determinan
- Solusi Persamaan Linear
- Keluar

@STT_Cipasung_2023

Matrix A

2	3	4
3	2	5
4	3	6

Matrix B

5	4	3
2	3	4
5	6	8

Pilih Operasi yang akan dilakukan

☒ Penjumlahan ☐ Pengurangan


Hitung

Hasil

7.0	7.0	7.0
5.0	5.0	9.0
9.0	9.0	14.0

gambar 14 test penjumlahan matriks 3x3

Matrix Calculator



ALJABAR GEOMETRI

- About
- Penjumlahan & Pengurangan
- Transpose Matriks
- Balikan Matriks
- Determinan
- Solusi Persamaan Linear
- Keluar

@STT_Cipasung_2023

Matrix A

-2	3	4
3	2	-5
4	-3	-6

Matrix B

5	4	3
2	-3	4
5	6	8

Pilih Operasi yang akan dilakukan

☒ Penjumlahan ☐ Pengurangan

Hitung

Hasil

3.0	7.0	7.0
5.0	-1.0	-1.0
9.0	3.0	2.0

gambar 15 test penjumlahan matriks 3x3

4.4 Pengurangan matriks

- Pengurangan 2x2

The screenshot shows the 'Matrix Calculator' application window. On the left is a sidebar with a logo and a menu titled 'ALJABAR GEOMETRI' containing buttons for 'About', 'Penjumlahan & Pengurangan', 'Transpose Matriks', 'Balikan Matriks', 'Determinan', 'Solusi Persamaan Linear', and 'Keluar'. The main area is divided into sections for 'Matrix A', 'Matrix B', operation selection, and the result. Matrix A is $\begin{bmatrix} -1 & 5 \\ 3 & -5 \end{bmatrix}$ and Matrix B is $\begin{bmatrix} 4 & -4 \\ 2 & 3 \end{bmatrix}$. The 'Pengurangan' (Subtraction) operation is selected. The result, labeled 'Hasil', is $\begin{bmatrix} -5.0 & 9.0 \\ 1.0 & -8.0 \end{bmatrix}$.

Matrix A	Matrix B
-1	4
5	-4
3	2
-5	3

Pilih Operasi yang akan di lakukan

☐ Penjumlahan ☒ Pengurangan

Hasil

-5.0	9.0
1.0	-8.0

gambar 16 test pengurangan matriks 2x2

This screenshot shows the same 'Matrix Calculator' application with different input values. Matrix A is $\begin{bmatrix} 1 & 5 \\ 3 & 5 \end{bmatrix}$ and Matrix B is $\begin{bmatrix} 4 & 4 \\ 2 & 3 \end{bmatrix}$. The 'Pengurangan' operation is still selected. The result, labeled 'Hasil', is $\begin{bmatrix} -3.0 & 1.0 \\ 1.0 & 2.0 \end{bmatrix}$.

Matrix A	Matrix B
1	4
5	4
3	2
5	3

Pilih Operasi yang akan di lakukan

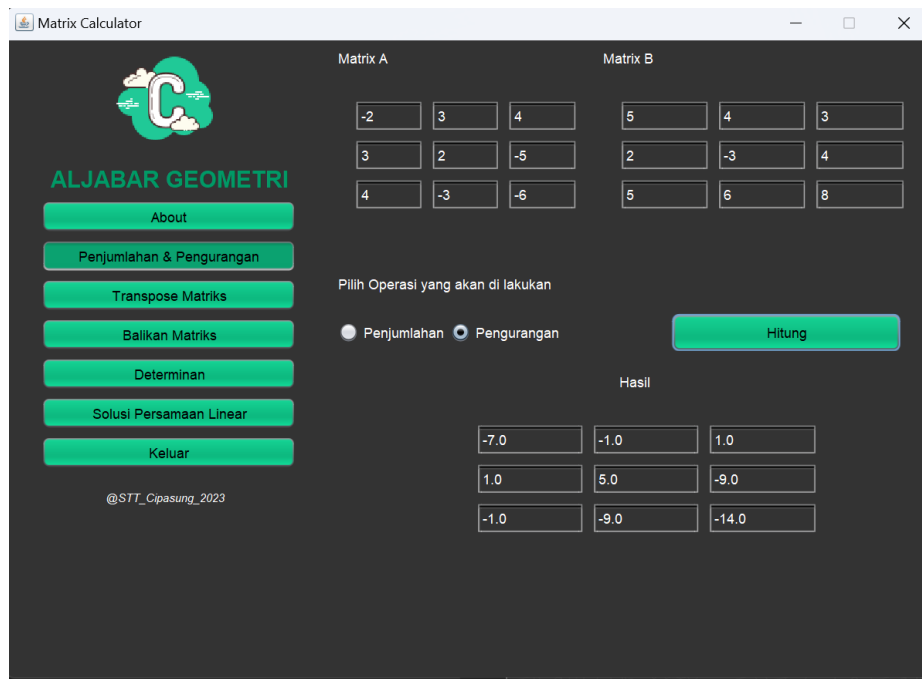
☐ Penjumlahan ☒ Pengurangan

Hasil

-3.0	1.0
1.0	2.0

gambar 17 test pengurangan matriks 2x2

- Pengurangan 3x3



Matrix Calculator

ALJABAR GEOMETRI

- About
- Penjumlahan & Pengurangan
- Transpose Matriks
- Balikan Matriks
- Determinan
- Solusi Persamaan Linear
- Keluar

@STT_Cipasing_2023

Matrix A

-2	3	4
3	2	-5
4	-3	-6

Matrix B

5	4	3
2	-3	4
5	6	8

Pilih Operasi yang akan di lakukan

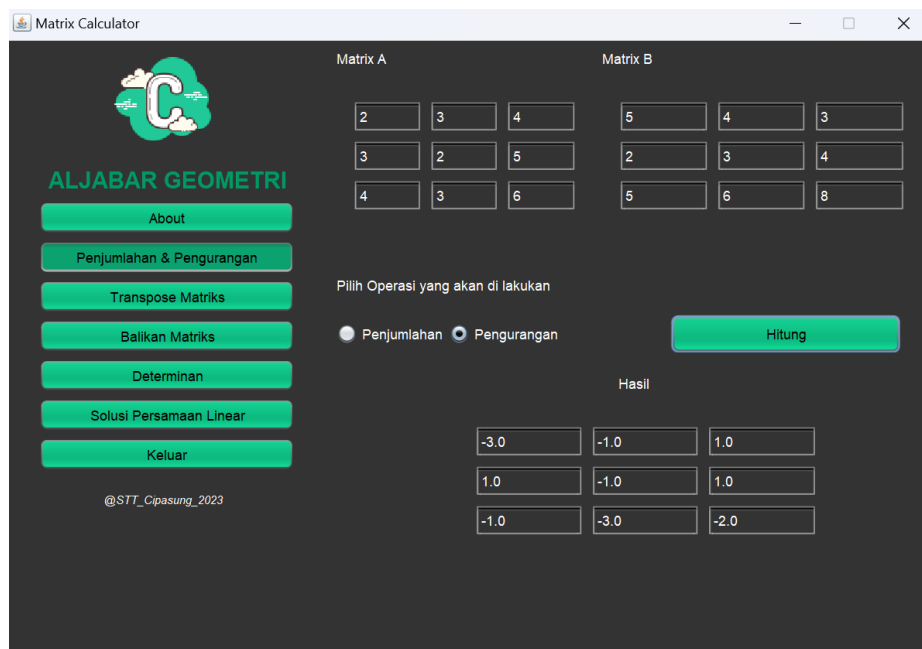
☐ Penjumlahan ☒ Pengurangan

Hitung

Hasil

-7.0	-1.0	1.0
1.0	5.0	-9.0
-1.0	-9.0	-14.0

gambar 18 test pengurangan matriks 3x3



Matrix Calculator

ALJABAR GEOMETRI

- About
- Penjumlahan & Pengurangan
- Transpose Matriks
- Balikan Matriks
- Determinan
- Solusi Persamaan Linear
- Keluar

@STT_Cipasing_2023

Matrix A

2	3	4
3	2	5
4	3	6

Matrix B

5	4	3
2	3	4
5	6	8

Pilih Operasi yang akan di lakukan

☐ Penjumlahan ☒ Pengurangan

Hitung

Hasil

-3.0	-1.0	1.0
1.0	-1.0	1.0
-1.0	-3.0	-2.0

gambar 19 test pengurangan matriks 3x3

4.2 Matriks Transpose

Matriks 2x2



The screenshot shows the ALJABAR GEOMETRI web application interface. On the left is a sidebar with a logo and a menu of mathematical functions. The main area is titled 'Matrix A' and contains a 2x2 grid of input fields with the values 13, 5, 2, and -7. A large green 'Transpose' button is positioned below the input fields. Below the button, the 'Hasil' (Result) section displays the transposed matrix in a 2x2 grid with values 2, -7, 13, and 5. The footer includes the text '@STT_Cipasing_2023'.

ALJABAR GEOMETRI

- About
- Penjumlahan & Pengurangan
- Transpose Matriks
- Balikan Matriks
- Determinan
- Solusi Persamaan Linear
- Keluar

Matrix A

13	5
2	-7

Transpose

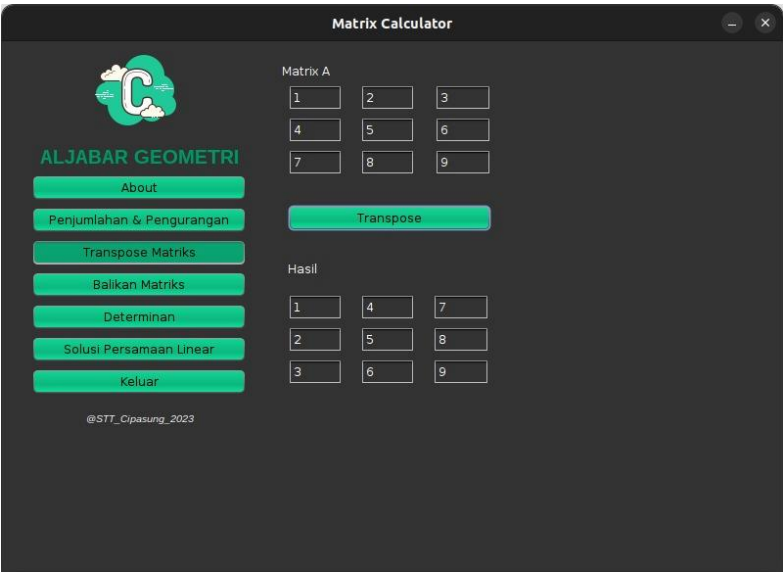
Hasil

2	-7
13	5

@STT_Cipasing_2023

gambar 20 test Transpose matriks

Matriks 3x3



The screenshot shows the Matrix Calculator application window. The sidebar on the left contains the same menu as the previous image. The main area is titled 'Matrix A' and contains a 3x3 grid of input fields with the values 1, 2, 3, 4, 5, 6, 7, 8, and 9. A green 'Transpose' button is located below the input fields. Below the button, the 'Hasil' (Result) section displays the transposed matrix in a 3x3 grid with values 4, 7, 9, 1, 5, 6, 2, 3, and 8. The footer includes the text '@STT_Cipasing_2023'.

Matrix Calculator

ALJABAR GEOMETRI

- About
- Penjumlahan & Pengurangan
- Transpose Matriks
- Balikan Matriks
- Determinan
- Solusi Persamaan Linear
- Keluar

Matrix A

1	2	3
4	5	6
7	8	9

Transpose

Hasil

4	7	9
1	5	6
2	3	8

@STT_Cipasing_2023

gambar 21 test Transpose matriks 3x3

4.5 Matriks Balikan (invers) (2 x 2)

ALJABAR GEOMETRI

- About
- Penjumlahan & Pengurangan
- Transpose Matriks
- Balikan Matriks
- Determinan
- Solusi Persamaan Linear
- Keluar

Matrix A

3	1
5	2

Balikan

Hasil

2.0	-1.0
-5.0	3.0

@STT_Cipasung_2023

gambar 22 test Matriks balikan

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

Input

Pilih Matriks

- 1. Matriks 2X2
- 2. Matriks 3X3

1

OK Cancel

gambar 23 JOptionpane chose

Determinan Matriks 2x2 :

Matriks A

<input type="text" value="2"/>	<input type="text" value="-2"/>
<input type="text" value="12"/>	<input type="text" value="8"/>

Hasil :

gambar 24 test determinan matriks 2x2

Determinan Matriks (3x 3) :

Matriks A

20	12	-2
-2	4	3
0	6	0

Hitung

Hasil : -336.0

gambar 25 test determinan matriks 3x3

4.7 Solusi Persamaan Linear

Masukan Persamaan

$$\begin{array}{rclcl} \boxed{2} & \times & \boxed{18} & Y & = \boxed{2} \\ \boxed{2} & \times & \boxed{-12} & Y & = \boxed{5} \end{array}$$

Hasil

X =

Y =

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 (2×2)
2. Menghitung matriks transpose (2×2)
3. Menghitung matriks balikan (invers) (2×2)
4. Menghitung determinan matriks (2×2) dan (3×3)
5. Menghitung solusi Sistem Persamaan Linier (SPL) (2×3)

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].