LAPORAN TUGAS BESAR ALJABAR GEOMETRI

Disusun Oleh:

Adzril Fauzi	10222093
Anisa	10222134
Eko Nugraha S	10222094
Handayani	10222153
Silva Putri Fadila N	10222117

PROGRAM STUDI INFORMATIKA
SEKOLAH TINGGI TEKNOLOGI CIPASUNG
KABUPATEN TASIKMALAYA

2023

BAB I DESKRIPSI MASALAH

Berikut adalah spesifikasi tugas yang ada di dalam laporan ini.

A. Buatlah pustaka dalam Bahasa Java untuk menemukan solusi SPL dengan metode eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n peubah dan n persamaan), menghitung determinan matriks dengan reduksi baris dan dengan ekspansi kofaktor, dan menghitung balikan matriks.

B. Gunakan pustaka di atas untuk membuat program penyelesaian berbagai persoalan dalam bentuk SPL, menyelesaikan persoalan interpolasi dan regresi linier, menghitung matriks balikan, menghitung determinan matriks dengan berbagai metode (reduksi baris dan ekspansi kofaktor).

Spesifikasi program adalah sebagai berikut:

1. Program dapat menerima masukan (input) baik dari keyboard maupun membaca masukan dari file text. Untuk SPL, masukan dari keyboard adalah m, n, koefisien aij , dan bi. Masukan dari file berbentuk matriks augmented tanpa tanda kurung,setiap elemen matriks dipisah oleh spasi. Misalnya,

   3 4.5 2.8 10 12
   -3 7 8.3 11 -4
   0.5 -10 -9 12 0

2. Untuk persoalan menghitung determinan dan matriks balikan, masukan dari keyboard adalah n dan koefisien aij. Masukan dari file berbentuk matriks, setiap elemen matriks dipisah oleh spasi. Misalnya,

   3 4.5 2.8 10
   -3 7 8.3 11
   0.5 -10 -9 12

3. Untuk persoalan interpolasi, masukannya jika dari keyboard adalah n, (x0, y0), (x1, y1), ..., (xn, yn), dan nilai x yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung. Misalnya jika titik-titik datanya adalah (8.0, 2.0794), (9.0,2.1972), dan (9.5, 2.2513), maka di dalam file text ditulis sebagai berikut:

   8.0 2.0794
   9.0 2.1972
   9.5 2.2513

4. Untuk persoalan regresi, masukannya jika dari keyboard adalah n (jumlah peubah x), semua nilai-nilai x1i, x2i, ..., xni, nilai yi, dan nilai-nilai xi yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.
5. Untuk persoalan SPL, luaran (output) program adalah solusi SPL. Jika solusinya tunggal, tuliskan nilainya. Jika solusinya tidak ada, tuliskan solusi tidak ada, jika solusinya banyak, maka tuliskan solusinya dalam bentuk parametrik (misalnya x4 = -2, x3 = 2s – t, x2 = s, dan x1 = t.)
6. Untuk persoalan determinan dan matriks balikan, maka luarannya sesuai dengan persoalan masing-masing.
7. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan.
8. Luaran program harus dapat ditampilkan pada layar komputer dan dapat disimpan ke dalam file.
9. Bahasa program yang digunakan adalah Java.
10. Program tidak harus berbasis GUI, cukup text-based saja, namun boleh menggunakan GUI (memakai kakas Eclipse misalnya).
11. Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan dirancang masing-masing. Misalnya, menu:

    MENU

    1. Penjumlahan dan Pengurangan Matriks
    2. Matriks Transpose
    3. Matriks Balikan
    4. Determinan
    5. Sistem Persamaan Linear
    0. Keluar

    Pilihan menu nomor 1 ada sub-menu lagi yaitu pilihan metode:

    1. Penjumlahan
    2. Pengurangan

    Pilihan menu nomer 2 dan 4 ada sub-menu juga yaitu:

    1. 2x2
    2. 3x3

BAB II LANDASAN TEORI

1. SISTEM PERSAMAAN LINEAR

Sistem persamaan linear merupakan gabungan dari beberapa persamaan linear yang saling berkorelasi beberapa metode digunakan seperti eliminasi, substitusi, eliminasi Gauss Jordan, Matriks dan lain sebagainya.Metode-metode tersebut jika dikerjakan secara manual seringkali membutuhkan waktu yang lama untuk menyelesaikan sistem persamaan linear sehingga untuk memudahkan pengerjaan maka digunakan software MATLAB.

Jika terdapat sejumlah persamaan linear dalam variabel $x_1, x_2, \ldots, x_n$ maka dinamakan sistem persamaan linear *(system of linear equations).* Seperti yang telah dijelaskan sebelumnya, bahwa Sistem Persamaan Linear (SPL) adalah suatu sistem yang didalamnya terdiri dari dua atau lebih persamaan linear.[19] Bentuk umumSistem Persamaan Linear (SPL) yang terdiri dari i buah persamaan linier dan j buah peubah dapat dituliskan sebagai berikut:

$$\begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots\ldots + a_{1j}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots\ldots + a_{2j}x_n = b_2 \\ \vdots \qquad \vdots \qquad \vdots \qquad\qquad \vdots \qquad \vdots \\ a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 + \ldots\ldots + a_{ij}x_n = b_i \end{array}$$

Dalam bentuk Matriks **AX** = **B** dapat dituliskan :

$$\boldsymbol{A}\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1j} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & a_{i3} & \cdots & a_{ij} \end{bmatrix}, \boldsymbol{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \end{bmatrix}$$

Dimana $a_1, a_2, \ldots a_n$,dan b merupakan bilangan-bilangan real, dan $x_1, x_2, \ldots,$ x_n adalah peubah. a dan b dengan subskrip merupakan konstanta. Jika mempunyai beberapa persamaan linear maka sekumpulan persamaan linear itu disebut Sistem Persamaan Linear. Suatu pasangan beberapa bilangan disebut solusi dari suatu Sistem Persamaan Linear (SPL) jika pasangan tersebut memenuhi kebenaran masing-masing persamaan dari Sistem Persamaan Linear (SPL) tersebut.

Solusi dari persamaan linier $a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_jx_n = b$ adalah suatu urutan dari n bilangan $s_1, s_2, \ldots, s_n$, sedemikian sehingga persamaan tersebut akan terpenuhi jika menggaitkan $x_1 = s_1$, $x_2 = s_2, \ldots, x_n = s_n$. Kumpulan semua solusi dari persamaan itu disebut himpunan solusi atau kadang- kadang disebut sebagai solusi umum dari persamaan tersebut.

2. ELIMINASI GAUSS

Eliminasi Gauss mengubah persamaan linear menjadi bentuk matriks, kemudian diubah ke bentuk Eselon Baris melalui Operasi Baris Elementer. Setelah itu bentuk matriks diselesaikan dengan substitusi balik. Eliminasi Gauss ini berasal dari operasi matematika pada baris matriks yang dilanjutkan sampai tersisa satu variabel. Metode eliminasi Gauss dipakai untuk menyelesaikan persamaan dalam bidang astronomi.

Contoh Soal Eliminasi Gauss Selesaikan sistem persamaan berikut:

x1 + 2x2 + 3x3 = 6

2x1 + 5 x2 + 10 x 3 = 17

x1 + 3x2 + 10 x3 = 18

Augmented matrik pada persamaan lanjar diatas adalah:

$$\begin{bmatrix} 1 & 2 & 3 & 6 \\ 2 & 5 & 10 & 17 \\ 1 & 3 & 10 & 17 \end{bmatrix}$$

Buat operasi baris elementer berikut ini :

$$\begin{matrix} B_2 = B_2 - 2B_1 \\ B_3 = B_3 - B_1 \end{matrix} \begin{bmatrix} 1 & 2 & 3 & 6 \\ 0 & 1 & 4 & 5 \\ 0 & 1 & 7 & 11 \end{bmatrix}$$

$$B_3 = B_3 - B_2 \begin{bmatrix} 1 & 2 & 3 & 6 \\ 0 & 1 & 4 & 5 \\ 0 & 0 & 3 & 6 \end{bmatrix}$$

Oleh sebab itu akan diperoleh penyelesaian:

$$x_3 = \frac{6}{3} = 2$$

$$x_2 = \frac{1}{1}(5 - 4 * 2) = -3$$

$$x_1 = \frac{1}{1}(6 - 2 * (-3) - 3 * (2)) = 6$$

3. ELIMINASI GAUSS-JORDAN

Eliminasi gauss-jordan ini adalah pengembangan dari eliminasi gauss yang hasilnya lebih disederhanakan lagi. Metode ini dimodifikasi oleh Wilhelm Jordan seorang insinyur Jerman pada tahun 1887. Dengan metode ini selain dapat digunakan untuk menyelesaikan sistem persamaan linear juga dapat digunakan untuk mencari invers dari sebuah matriks.

Sehingga untuk mengoperasikan persamaan linear cara penyelesaiannya pun hampir sama dengan metode gauss, namun pada metode gauss kita hanya menghasilkan matriks yang eselon baris sedangkan metode eliminasi gauss-jordan ini perbedaanya hanya kita harus membuat elemen elemen diatas maupun dibawah diagonal utama menjadi bernilai nol.

Sehingga hasilnya menjadi matriks eselon yang tereduksi yaitu menjadi sebuah matriks dengan diagonal satuan atau matriks identitas (semua elemen pada diagonal utama bernilai 1, sedangkan elemen lainnya bernilai nol). Tahap pengerjaanya sama dengan metode sebelumnya yaitu eliminasi gauss menggunakan cara elementer.

Diketahui SPL 3 variabel

x + 3y + 2z = 4
2x + 7y + 4z = 6
2x + 9y + 7z = 4

Tentukan nilai dari variabel - variabel persamaan linear diatas !

Penyelesaian :

Sama seperti metode gauss, pertama kita harus mengubah persamaan linear 3 variabel diatas menjadi sebuah matriks yang teraugmentasi.

Tahap1

$$\begin{array}{c} b1 \\ b2 \\ b3 \end{array} \left[\begin{array}{ccc|c} 1 & 3 & 2 & 4 \\ 2 & 7 & 4 & 6 \\ 3 & 9 & 7 & 4 \end{array}\right]$$

Baris ke-1 didapat dari x + 3y + 2z = 4

Baris ke-2 didapat dari 2x + 7y + 4z = 6

Baris ke-3 didapat dari 3x + 9y + 7z = 4

> Notes : untuk variabel x, y dan z jika didepan variabel tidak ada angkanya maka dilambangkan dengan angka 1dan ikut disertakan kedalam matriks. Misalnya x menjadi 1 .

Karena baris pertama pada kolom pertama (a11) sudah bernilai 1. Maka, kita akan mengubah baris ke-2 (b2) terlebih dahulu.

Tahap 2

$$\begin{array}{c} b1 \\ b2 \\ b3 \end{array} \left[\begin{array}{ccc|c} 1 & 3 & 2 & 4 \\ 2 & 7 & 4 & 6 \\ 3 & 9 & 7 & 4 \end{array}\right] \xrightarrow{} \text{b2 − 2(b1)} \left[\begin{array}{ccc|c} 1 & 3 & 2 & 4 \\ 0 & 1 & 0 & -2 \\ 3 & 9 & 7 & 4 \end{array}\right]$$

Pembahasan :

$b2 - 2(b) = 2 - 2(1) = 2 - 2 = 0 \rightarrow a21$

$b2 - 2(b1) = 7 - 2(3) = 7 - 6 = 1 \rightarrow a22$

$b2 - 2(b1) = 4 - 2(2) = 4 - 4 = 0 \rightarrow a23$

$b2 - 2(b1) = 6 - 2(4) = 6 - 8 = \text{-}2 \rightarrow a24$

Hasil dari a21, a22, a23 dan a24 akan menjadi baris ke-2 (b2) dan untuk elemen lainnya tetap sama.

Tahap Ketiga

Selanjutnya kita akan mengubah nilai pada baris ke-3 (b3).

$$\begin{array}{c} b1 \\ b2 \\ b3 \end{array}\left[\begin{array}{ccc|c} 1 & 3 & 2 & 4 \\ 0 & 1 & 0 & -2 \\ 3 & 9 & 7 & 4 \end{array}\right] \Rightarrow \mathrm{b3} - 3(\mathrm{b1}) \left[\begin{array}{ccc|c} 1 & 3 & 2 & 4 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & -8 \end{array}\right]$$

Pembahasan :

$b3 - 3(b1) = 3 - 3(1) = 3 - 3 = 0 \rightarrow a31$

$b3 - 3(b1) = 9 - 3(3) = 9 - 9 = 0 \rightarrow a32$

$b3 - 3(b1) = 7 - 3(2) = 7 - 6 = 1 \rightarrow a33$

$b3 - 3(b1) = 4 - 3(4) = 4 - 12 = -8 \rightarrow a34$

Hasil dari a31, a32, a33 dan a34 akan menjadi baris ke-3 (b3).

Tahap Keempat

Karena baris pertama pada kolom ke-2 (a12) dan baris pertama pada kolom ke-3 (a13) belum bernilai nol, maka kita masih harus mengoperasikannya agar bernilai nol sehingga menjadi matriks yang tereduksi.

$$\begin{array}{c} b1 \\ b2 \\ b3 \end{array}\left[\begin{array}{ccc|c} 1 & 3 & 2 & 4 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & -8 \end{array}\right] \Rightarrow \mathrm{b1} - 3(\mathrm{b2}) \left[\begin{array}{ccc|c} 1 & 0 & 2 & 10 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & -8 \end{array}\right] \Rightarrow \mathrm{b1} - 2(\mathrm{b3}) \left[\begin{array}{ccc|c} 1 & 0 & 0 & 26 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & -8 \end{array}\right]$$

Pembahasan untuk mengubah $a12$:

$b1 - 3(b2) = 1 - 3(0) = 1 - 0 = 1 \rightarrow a11$

$b1 - 3(b2) = 3 - 3(1) = 3 - 3 = 0 \rightarrow a12$

$b1 - 3(b2) = 2 - 3(0) = 2 - 0 = 2 \rightarrow a13$

$b1 - 3(b2) = 4 - 3(-2) = 4 - (-6) = 10 \rightarrow a14$

Hasil dari a11,a12,a13 dan a14 ini akan menjadi baris ke-1 (b1).

Pembahasan untuk mengubah $a13$:

$b1 - 2(b3) = 1 - 2(0) = 1 - 0 = 1 \rightarrow a11$

$b1 - 2(b3) = 0 - 2(0) = 0 - 0 = 0 \rightarrow a12$

$b1 - 2(b3) = 2 - 2(1) = 2 - 2 = 0 \rightarrow a13$

$b1 - 2(b3) = 10 - 2(-8) = 10 - (-16) = 26 \rightarrow a14$

Hasil dari a11, a12, a13 dan a14 ini akan menjadi tahap terakhir untuk mengoperasikan baris pertama (b1), sehingga kita sudah mendapatkan matriks yang tereduksi.

Tahap Terakhir

$$\begin{array}{c} b1 \\ b2 \\ b3 \end{array} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 26 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & -8 \end{array}\right]$$

Setelah kita menghasilkan matriks eselon tereduksi yang membentuk sebuah matriks identitas seperti diatas, maka kita tidak perlu mensubstitusikannya seperi pada eliminasi gauss karena, sudah dapat diketahui nilai variabelnya yaitu : x = 26, y = -2 dan z = -8

4. DETERMINAN

merupakan selisih antara perkalian elemen-elemen pada diagonal utama dengan perkalian elemen-elemen pada diagonal sekunder. Determinan matriks hanya dapat dicari dengan matriks persegi. Determinan dari matriks A dapat ditulis det(A) atau |A|. Determinan matriks dapat ditemukan dalam matriks persegi ordo 2x2 dan 3x3.

a. Determinan Matriks Ordo 2x2, Determinan matriks persegi dengan ordo 2x2 dapat dihitung dengan cara berikut:

$$\det(A) = |A| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = a.d - b.c$$

b. Determinan Matriks Ordo 3x3

Determinan matriks persegi dengan ordo 3x3 dapat dihitung dengan menggunakan dua cara, yaitu kaidah sarrus dan ekspansi kofaktor, Namun, cara yang paling sering digunakan dalam menentukan determinan matriks ordo 3x3 adalah dengan kaidah sarrus.

Langkah-langkah mencari determinan matriks ordo 3x3 dengan kaidah sarrus

1. Meletakan kolom pertama dan kolom kedua di sebelah kanan garis vertikal determinan.
2. Jumlahkan hasil kali elemen-elemen yang terletak pada diagonal utama dengan hasil kali elemen-elemen yang sejajar diagonal utama pada arah kanan kemudian kurangi dengan jumlah hasil kali elemen-elemen yang terletak pada diagonal samping dengan elemen-elemen yang sejajar dengan diagonal samping.

$$|A| = \left|\begin{array}{ccc|cc} a & b & c & a & b \\ d & e & f & d & e \\ g & h & i & g & h \end{array}\right.$$

|A| = (a.e.i) + (b.f.g) + (c.d.h) - (c.e.g) - (a.f.h) - (b.d.i)

|A| = (a.e.i + b.f.g + c.d.h) - (c.e.g + a.f.h + b.f.i)

Invers Matriks

Invers matriks adalah kebalikan (invers) dari sebuah matriks yang apabila matriks tersebut dikalikan dengan inversnya, akan menjadi matriks identitas. Invers matriks

dilambangkan dengan A-1. Suatu matriks dikatakan memiliki invers jika determinan dari matriks tersebut tidak sama dengan nol.
Untuk menentukan invers dari sebuah matriks, terdapat dua aturan berdasarkan ordonya, yaitu ordo 2x2 dan ordo 3x3.

Invers Matriks Ordo 2x2

Invers matriks persegi dengan ordo 2x2 dapat dicari dengan cara sebagai berikut:

$$A^{-1} = \frac{1}{|A|} \times Adj\ A, \text{ dengan syarat } |A| \neq 0$$

$$\text{Jika } A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \text{ maka } A^{-1} = \frac{1}{ad-bc} \times \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}, \text{ dengan } |A| \neq 0$$

Invers Matriks Ordo 3x3

Untuk mencari invers matriks pada ordo 3x3, dapat digunakan metode eliminasi Gauss Jordan.
Secara sistematis, eliminasi Gauss Jordan dapat dinyatakan sebagai berikut:

$$[A|I] \rightarrow [I|A^{-1}]$$

5. MATRIKS BALIKAN

JIka A dan B matriks bujur sangkar sedemikian rupa sehingga A B = B A = I , maka B disebut balikan atau *invers* dari A dan dapat dituliskan $B = A^{-1}$ (B sama dengan *invers* A). Matriks B juga mempunyai *invers* yaitu A maka dapat dituliskan $A = B^{-1}$. Jika tidak ditemukan matriks B, maka A dikatakan **matriks tunggal** (singular). Jika matriks B dan C adalah *invers* dari A maka B = C.

Matriks A = a b $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ dapat di-*invers* apabila ad - b
c ≠ 0 c

6. MATRIKS TRANSPOSE

Matriks transpose adalah operasi yang menghasilkan matriks baru dengan menukar baris dan kolom dari matriks asal. Dengan kata lain, elemen-elemen yang berada pada baris i, kolom j dari matriks awal akan menjadi elemen pada baris j, kolom i dalam matriks transpose. Operasi ini umumnya dilambangkan dengan simbol \(A^T\), di mana \(A\) adalah matriks asal.
Contoh, jika matriks asal adalah:

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

Maka matriks transposenya adalah:

$$A^T = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

Transpose sering digunakan dalam berbagai bidang matematika dan ilmu komputer.

Contoh untuk matriks 2x2:

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

Contoh untuk matriks 3x3:

$$\begin{bmatrix} p & q & r \\ s & t & u \\ v & w & x \end{bmatrix}^T = \begin{bmatrix} p & s & v \\ q & t & w \\ r & u & x \end{bmatrix}$$

7. PENJUMLAHAN MATRIKS

Penjumlahan matriks adalah operasi matematika yang dilakukan dengan menggabungkan dua matriks menjadi satu matriks baru. Untuk menjumlahkan dua matriks, elemen-elemen yang berada pada posisi yang sama di matriks yang akan dijumlahkan dijumlahkan satu sama lain.

Contoh:

Misalkan kita memiliki dua matriks:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

dan

$$B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

Maka penjumlahan matriks (A + B) akan menghasilkan matriks baru:

$$A + B = \begin{bmatrix} 1+5 & 2+6 \\ 3+7 & 4+8 \end{bmatrix} = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}$$

Jadi, hasil penjumlahan matriks A dan B adalah matriks baru

$$\begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}.$$

BAB III PENJELASAN IMPLEMENTASI PROGRAM

Disini kami telah melakukan eksekusi pemograman menggunakan bahasa Java, yang dimulai dengan tampilan menu.

1. Syntax Tampilan Menu

```
public class Menu{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int choice;

        System.out.println("Kelompok 3");
        System.out.println("Adzril Fauzi");
        System.out.println("Anisa");
        System.out.println("Eko Nugraha S");
        System.out.println("Handayani");
        System.out.println("Silva Putri P N");
        System.out.println("");

        do {
            System.out.println("=== Menu ===");
            System.out.println("1. Matriks Penjumlahan dan Pengurangan "
                    + "\n2. Matriks Transpose"
                    + "\n3. Matriks Balikan"
                    + "\n4. Determinan"
                    + "\n5. Sistem Persamaan Linear\n"
                    + "0. Keluar ");
            System.out.print("Apa yang ingin anda hitung : ");
            choice = scanner.nextInt();


            switch (choice) {
                case 1:
```

Pada kondisi *switch case* dilanjut dengan memasukan pemograman perhitungan matriks yang lain.

Output - **Matriks (run)**

```
run:
Kelompok 3
Adzril Fauzi
Anisa
Eko Nugraha S
Handayani
Silva Putri P N

=== Menu ===
1. Matriks Penjumlahan dan Pengurangan
2. Matriks Transpose
3. Matriks Balikan
4. Determinan
5. Sistem Persamaan Linear
0. Keluar
Apa yang ingin anda hitung :
```

2. Syntax Penjumlahan dan Pengurangan Matriks

```
case 1:// Penjumlahan + Pengurangan
          System.out.println("1. Penjumlahan\n2. Pengurangan");
          System.out.print("Masukan Pilihan Anda : ");
          int choice2 = scanner.nextInt();
          int [][]a = new int [2][2];
          int [][]b = new int [2][2];
          int [][]c = new int [2][2];
          switch (choice2) {
            case 1:// sub switch 1
              for (int i = 0; i < a.length; i++) {
                for (int j = 0; j <a.length; j++) {
                  System.out.print("Masukan Nilai A["+i+"]["+j+"]:");
                  a[i][j] = scanner.nextInt();
                }
              }

              for (int i = 0; i < b.length; i++) {
                for (int j = 0; j <b.length; j++) {
                  System.out.print("Masukan Nilai B["+i+"]["+j+"]:");
                  b[i][j] = scanner.nextInt();
                }
              }

              //operasi
              c = plus.Penjumlahan(a, b);

              for(int i=0; i<b.length; i++){
                for(int j=0; j<b.length; j++){
                  System.out.print(+(c[i][j])+" ");
                }
```

```
        System.out.println(" ");
      }
      System.out.println(" ");

      break;//end Of sub switch1

    case 2:// sub switch 2
      for (int i = 0; i < a.length; i++) {
        for (int j = 0; j <a.length; j++) {
          System.out.print("Masukan Nilai A["+i+"]["+j+"]:");
          a[i][j] = scanner.nextInt();
        }
      }

      for (int i = 0; i < b.length; i++) {
        for (int j = 0; j <b.length; j++) {
          System.out.print("Masukan Nilai B["+i+"]["+j+"]:");
          b[i][j] = scanner.nextInt();
        }
      }

      c = plus.Pengurangan(a, b);

      for (int i = 0; i < b.length; i++) {
        for (int j = 0; j <b.length; j++) {
          System.out.print(c[i][j] + " ");
        }
        System.out.println(" ");
      }
      System.out.println(" ");

      break;// end of sub switch 2

    default:
      throw new AssertionError();
  }//end of sub-switch No.1
  break;//end of plus and min
```

3. Syntax Matriks Transpose 2x2 dan 3x3

```
case 1://2x2
      int [][]A= new int[2][2];
      int [][]B= new int[2][2];

      for (int i = 0; i < A.length; i++) {
        for (int j = 0; j < A.length; j++) {
          System.out.print("Masukan Nilai A["+i+"]["+j+"] :");
          A[i][j] = scanner.nextInt();
          }
        }
```

```
                    B[0][0]=A[0][0];
                      B[0][1]=A[1][0];
                      B[1][0]=A[0][1];
                      B[1][1]=A[1][1];

                      for (int i = 0; i < A.length; i++) {
                        for (int j = 0; j < A.length; j++) {
                          System.out.println("Hasil Tranpose AT ["+i+"]["+j+"] : "
+B[i][j]);
                          }
                      }
                  break;//end of transpose 2x2

                    case 2://3x3
                      int [][]A1= new int[3][3];
                      int [][]B1= new int[3][3];

                      for (int i = 0; i < A1.length; i++) {
                        for (int j = 0; j < A1.length; j++) {
                          System.out.print("Masukan Nilai A["+i+"]["+j+"] :");
                          A1[i][j] = scanner.nextInt();
                          }
                        }

                      B1[0][0]=A1[0][0];
                      B1[0][1]=A1[1][0];
                      B1[1][0]=A1[0][1];
                      B1[1][1]=A1[1][1];
//                      B1[2][0]=A1[][];

                      for (int i = 0; i < A1.length; i++) {
                        for (int j = 0; j < A1.length; j++) {
                          System.out.println("Hasil Tranpose AT ["+i+"]["+j+"] : "
+B1[i][j]);
                          }
                        }
                    break;
                    default:
                      throw new AssertionError();
                    }//end of sub-switch No.1
                  break;//end of plus and min
```

4. Syntax Matriks Balikan

```
import java.util.Scanner;

public class Invers {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
```

```
        System.out.print("Masukkan ukuran matriks (n): ");
        int n = input.nextInt();

        System.out.println("Masukkan nilai matriks");
        double[][] matrix = new double[n][n];

        // Membaca elemen matriks dari pengguna
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                System.out.print("Masukkan elemen matriks[" + (i + 1) + "][" + (j + 1)
+ "]: ");
                matrix[i][j] = input.nextDouble();
            }
        }

        // Menghitung matriks balikan
        double[][] inverseMatrix = calculateInverse(matrix);

        // Menampilkan matriks balikan
        System.out.println("Matriks Balikan:");
        displayMatrix(inverseMatrix);

        input.close();
    }

    // Metode untuk menghitung matriks balikan menggunakan eliminasi Gauss-
Jordan
    private static double[][] calculateInverse(double[][] matrix) {
        int n = matrix.length;
        double[][] augmentedMatrix = new double[n][2 * n];

        // Membuat matriks augmented [matrix | I]
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                augmentedMatrix[i][j] = matrix[i][j];
                augmentedMatrix[i][j + n] = (i == j) ? 1 : 0;
            }
        }

        // Melakukan eliminasi Gauss-Jordan
        for (int i = 0; i < n; i++) {
            // Membuat leading entry menjadi 1
            double pivot = augmentedMatrix[i][i];
            for (int j = 0; j < 2 * n; j++) {
                augmentedMatrix[i][j] /= pivot;
            }

            // Menghilangkan elemen di bawah leading entry
            for (int k = 0; k < n; k++) {
```

```
                if (k != i) {
                    double factor = augmentedMatrix[k][i];
                    for (int j = 0; j < 2 * n; j++) {
                        augmentedMatrix[k][j] -= factor * augmentedMatrix[i][j];
                    }
                }
            }
        }

        // Mengambil matriks balikan dari bagian kanan matriks augmented
        double[][] inverseMatrix = new double[n][n];
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                inverseMatrix[i][j] = augmentedMatrix[i][j + n];
            }
        }

        return inverseMatrix;
    }

    // Metode untuk menampilkan matriks
    private static void displayMatrix(double[][] matrix) {
        int rows = matrix.length;
        int cols = matrix[0].length;

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

5. Syntax Determinan 2x2 dan 3x3

```
System.out.println("1. Determinan 2x2 \n2. Determinan 3x3");
                System.out.print("Masukan Pilihan Anda : ");
                int choice4 = scanner.nextInt();

                switch (choice4){
                    case 1:
                        int [][]A = new int [2][2];

                        for(int i=0;i<2;i++){
                            for(int j=0;j<2;j++){
                                System.out.print("Masukan Nilai A["+i+"]["+j+"] : ");
                            A[i][j]= scanner.nextInt();
                            }
                        }
```

```
                int hasil =(A[0][0]*A[1][1])-(A[0][1]*A[1][0]);
                System.out.println ("det(a) = "+hasil);

                break;

                case 2:
                   int [][]Aa = new int [3][3];

                   for(int i=0;i<3;i++){
                      for(int j=0;j<3;j++){
                         System.out.print("Masukan Nilai A["+i+"]["+j+"] : ");
                         Aa[i][j]= scanner.nextInt();
                         }
                      }

                   int hasil1 =(Aa[0][0]*Aa[1][1])-(Aa[0][1]*Aa[1][0]);
                   System.out.println ("det(a) = "+hasil1);

                break;
             }
```

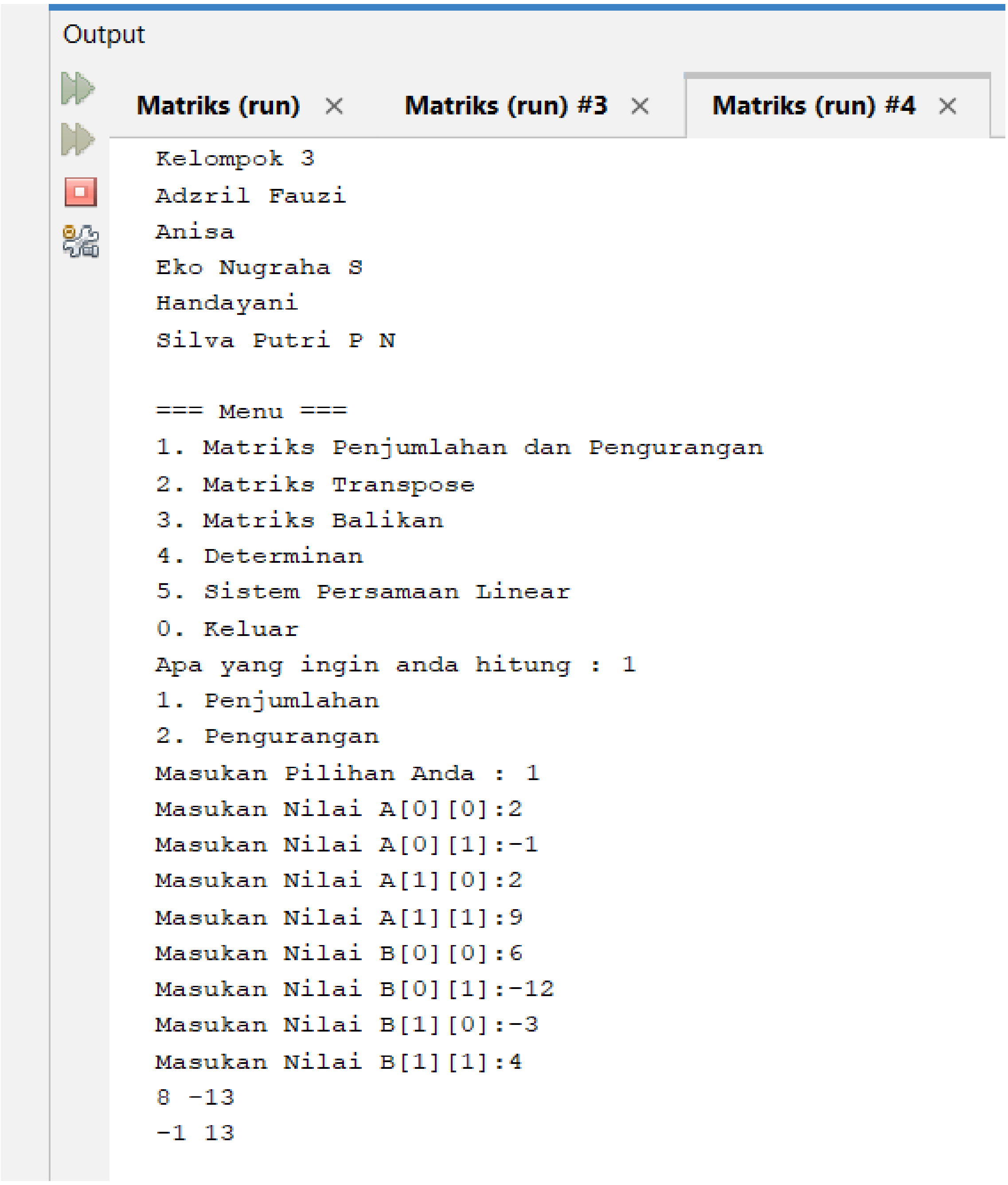
6. Syntax Sistem Persamaan Linear

```
import java.util.Scanner;

public class SPL {
   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);

      System.out.print("Masukkan jumlah variabel (n): ");
      int n = input.nextInt();

      double[][] augmentedMatrix = new double[n][n + 1];

      System.out.println("Masukkan koefisien matriks augmented:");
      for (int i = 0; i < n; i++) {
         for (int j = 0; j <= n; j++) {
            System.out.print("Masukkan elemen matriks[" + (i + 1) + "][" + (j + 1)
+ "]: ");
            augmentedMatrix[i][j] = input.nextDouble();
         }
      }

      // Menampilkan matriks augmented sebelum eliminasi
      System.out.println("Matriks Augmented sebelum eliminasi:");
      displayMatrix(augmentedMatrix);

      // Melakukan eliminasi Gauss
      for (int i = 0; i < n; i++) {
         // Membuat leading entry menjadi 1
         double pivot = augmentedMatrix[i][i];
```

```java
            for (int j = i; j <= n; j++) {
                augmentedMatrix[i][j] /= pivot;
            }

            // Menghilangkan elemen di bawah leading entry
            for (int k = i + 1; k < n; k++) {
                double factor = augmentedMatrix[k][i];
                for (int j = i; j <= n; j++) {
                    augmentedMatrix[k][j] -= factor * augmentedMatrix[i][j];
                }
            }
        }

        // Menampilkan matriks augmented setelah eliminasi
        System.out.println("\nMatriks Augmented setelah eliminasi:");
        displayMatrix(augmentedMatrix);

        // Melakukan substitusi mundur
        double[] solution = new double[n];
        for (int i = n - 1; i >= 0; i--) {
            solution[i] = augmentedMatrix[i][n];
            for (int j = i + 1; j < n; j++) {
                solution[i] -= augmentedMatrix[i][j] * solution[j];
            }
        }

        // Menampilkan solusi
        System.out.println("\nSolusi SPL:");
        for (int i = 0; i < n; i++) {
            System.out.println("x" + (i + 1) + " = " + solution[i]);
        }

        input.close();
    }

    // Metode untuk menampilkan matriks
    private static void displayMatrix(double[][] matrix) {
        int rows = matrix.length;
        int cols = matrix[0].length;

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println();
        }
        System.out.println();
    }
}
```

BAB IV PENGUMPULAN DAN PENGUJIAN DATA

1. Output RPlus



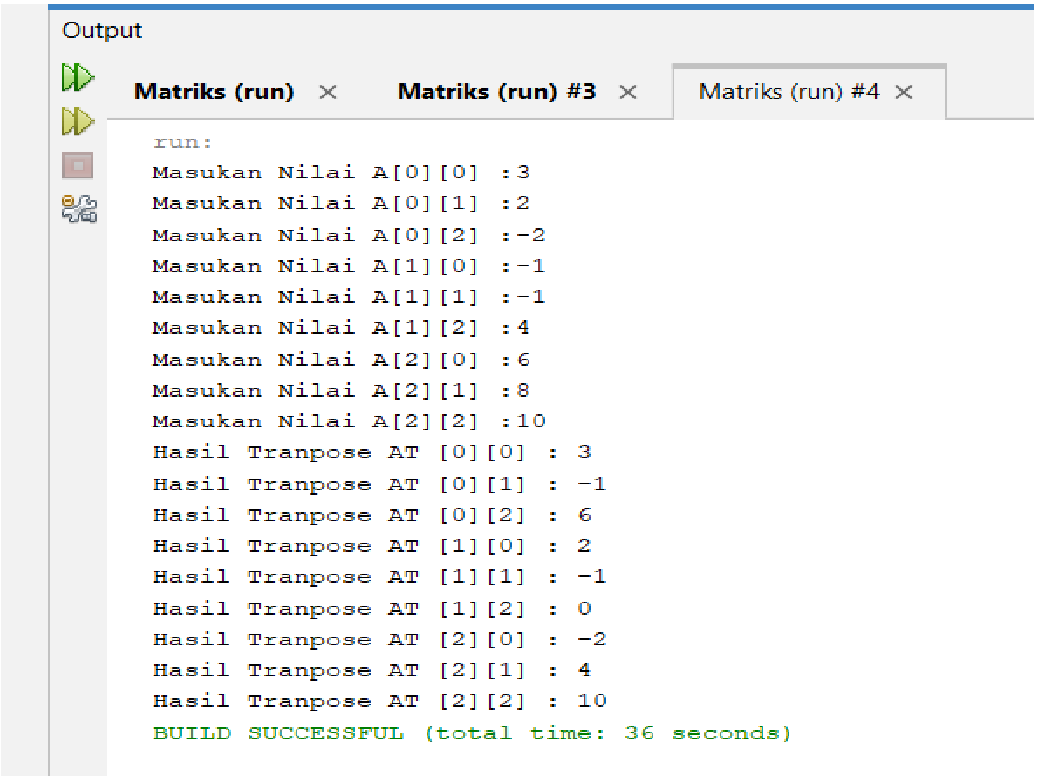
2. Output RMin

```
Output
Matriks (run) ×   Matriks (run) #3 ×   Matriks (run) #4 ×   Matriks (run) #5 ×
Kelompok 3
Adzril Fauzi
Anisa
Eko Nugraha S
Handayani
Silva Putri P N

=== Menu ===
1. Matriks Penjumlahan dan Pengurangan
2. Matriks Transpose
3. Matriks Balikan
4. Determinan
5. Sistem Persamaan Linear
0. Keluar
Apa yang ingin anda hitung : 1
1. Penjumlahan
2. Pengurangan
Masukan Pilihan Anda : 2
Masukan Nilai A[0][0]:7
Masukan Nilai A[0][1]:0
Masukan Nilai A[1][0]:-9
Masukan Nilai A[1][1]:8
Masukan Nilai B[0][0]:-4
Masukan Nilai B[0][1]:5
Masukan Nilai B[1][0]:2
Masukan Nilai B[1][1]:-6
11 -5
-11 14
```

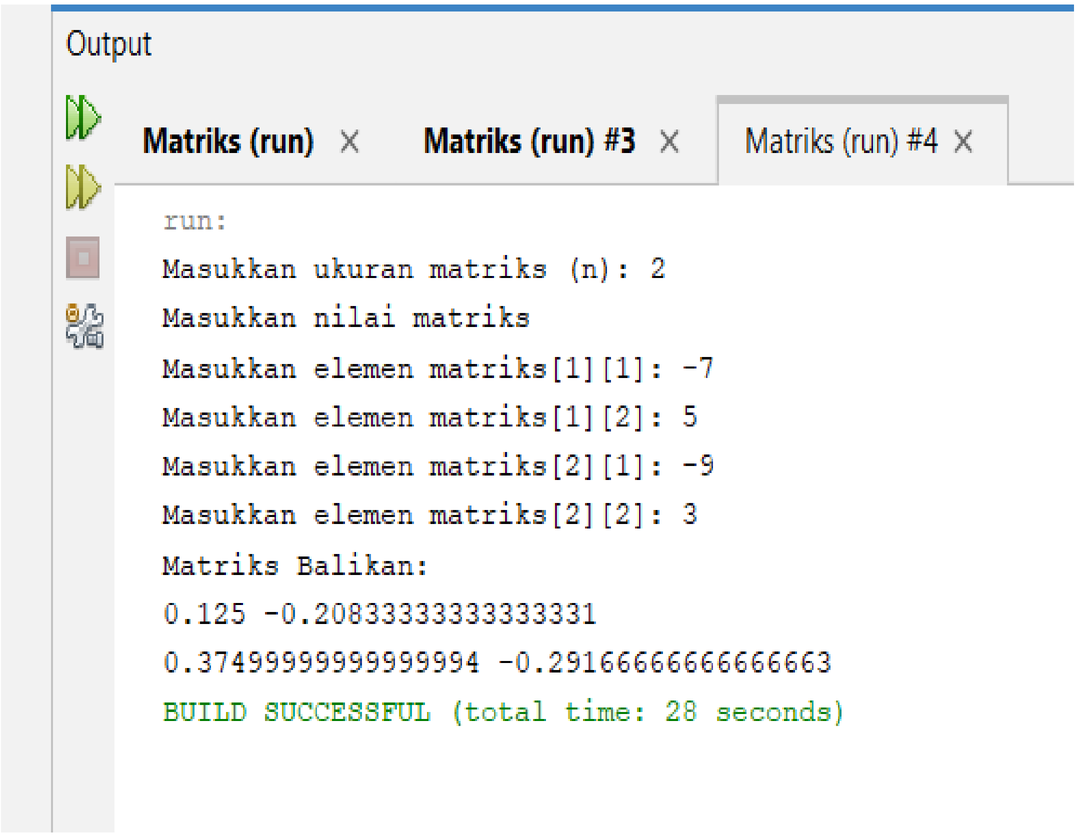
3. Output Transpose Matriks 2x2

```
Output
Matriks (run) ×   Matriks (run) #3 ×   Matriks (run) #4 ×
run:
Masukan Nilai A[0][0] :5
Masukan Nilai A[0][1] :6
Masukan Nilai A[1][0] :-1
Masukan Nilai A[1][1] :6
Hasil Tranpose AT [0][0] : 5
Hasil Tranpose AT [0][1] : -1
Hasil Tranpose AT [1][0] : 6
Hasil Tranpose AT [1][1] : 6
BUILD SUCCESSFUL (total time: 18 seconds)
```

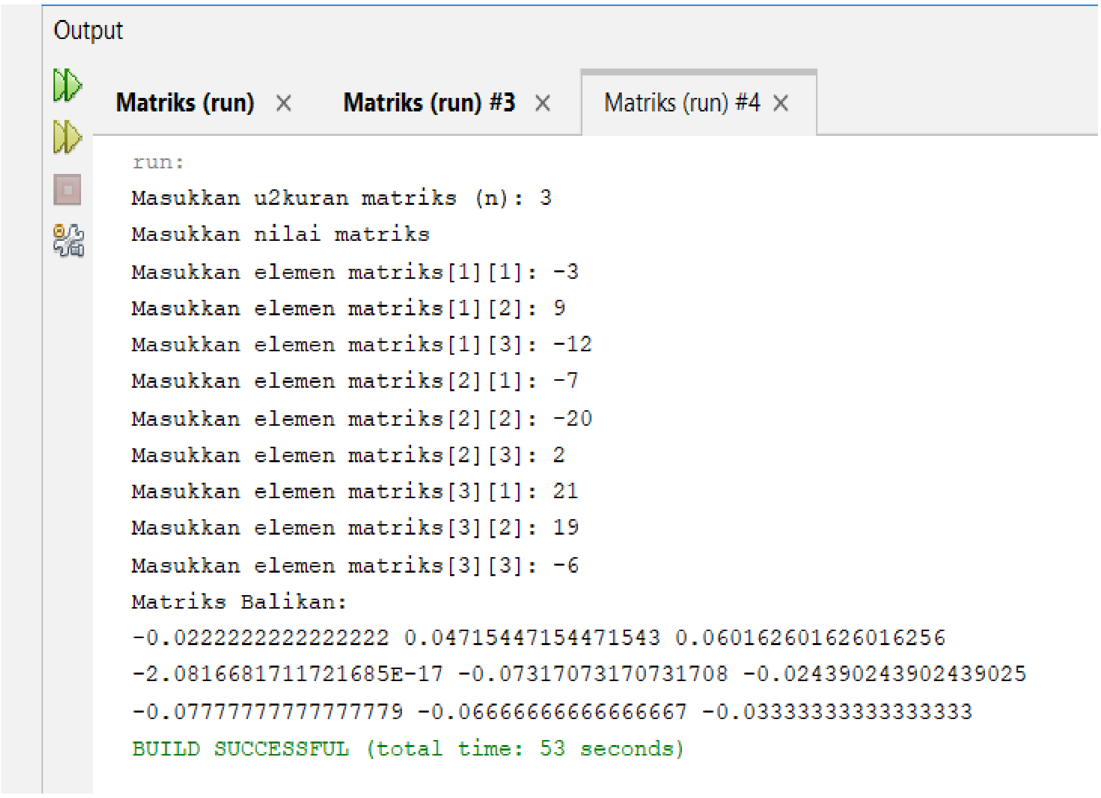
4. **Output Transpose Matriks 3x3**



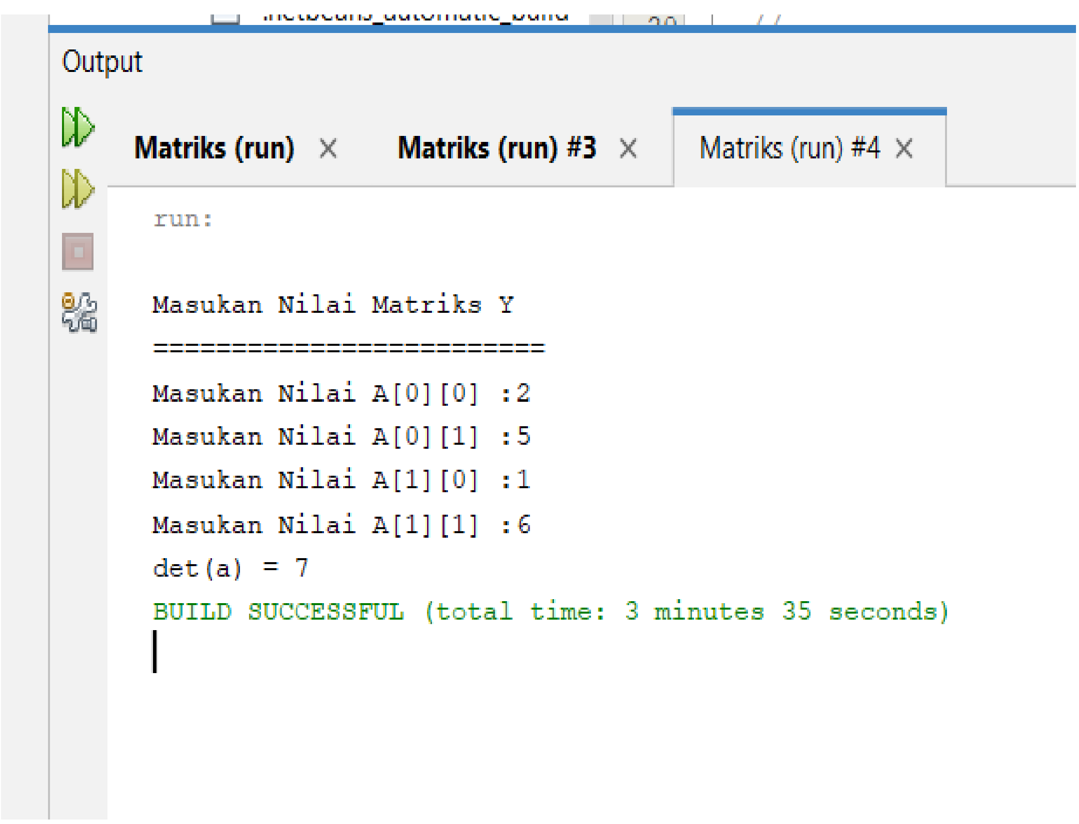
5. **Output Invers 2x2**



6. Output Invers 3x3



7. Output Determinan Matriks 2x2



8. Output Determinan 3x3

Output

Matriks (run) × Matriks (run) #3 × Matriks (run) #4 ×

```
run:

Masukan Nilai Matriks Y
=========================
Masukan Nilai A[0][0] :2
Masukan Nilai A[0][1] :3
Masukan Nilai A[0][2] :6
Masukan Nilai A[1][0] :12
Masukan Nilai A[1][1] :4
Masukan Nilai A[1][2] :6
Masukan Nilai A[2][0] :7
Masukan Nilai A[2][1] :15
Masukan Nilai A[2][2] :13
det(a) = -28
BUILD SUCCESSFUL (total time: 57 seconds)
```

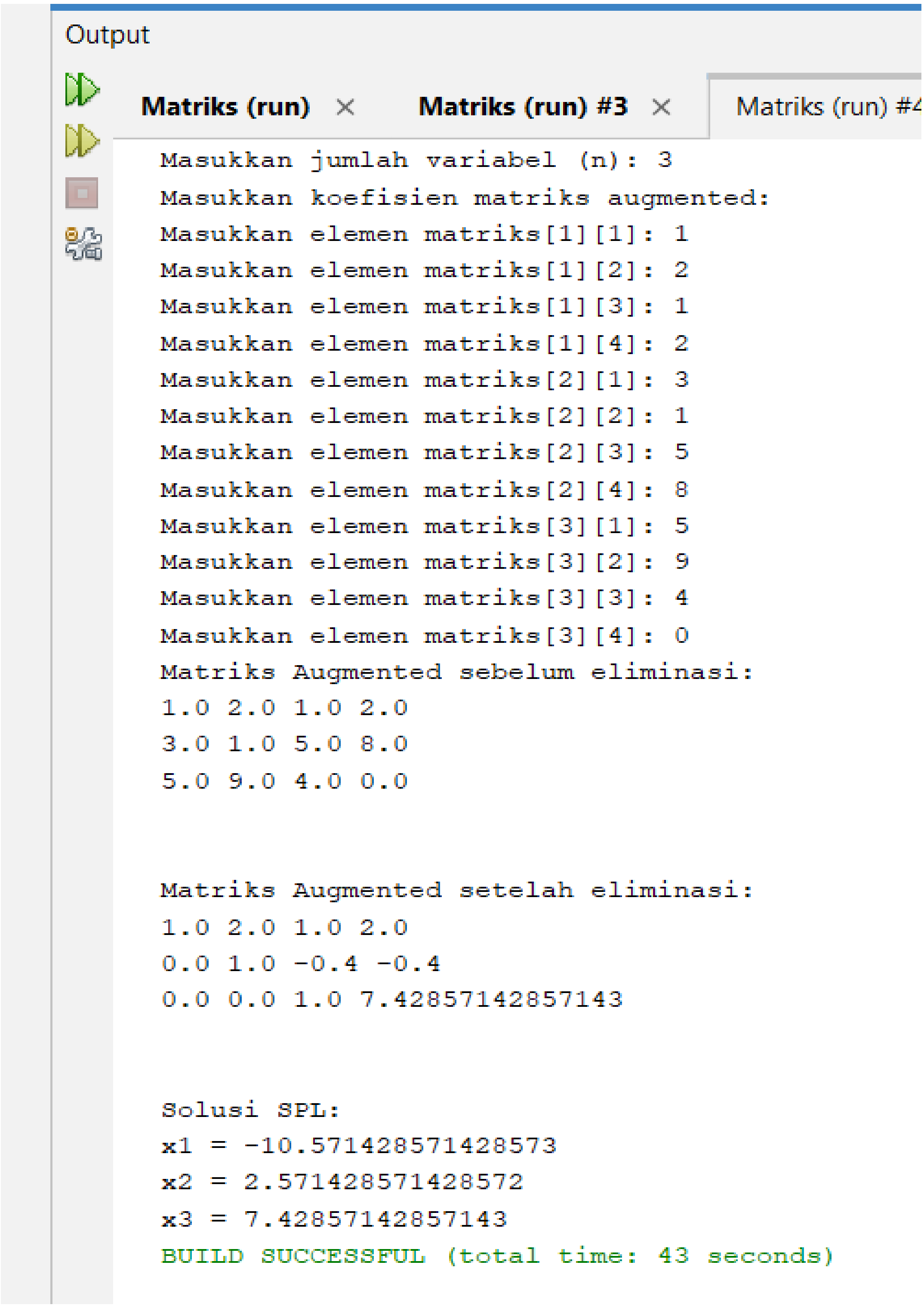
9. Output Sistem Persamaan Linier 2x2

Output

Matriks (run) × Matriks (run) #3 × Matriks (run) #4 ×

```
run:
Masukkan jumlah variabel (n): 2
Masukkan koefisien matriks augmented:
Masukkan elemen matriks[1][1]: 1
Masukkan elemen matriks[1][2]: 2
Masukkan elemen matriks[1][3]: 1
Masukkan elemen matriks[2][1]: 2
Masukkan elemen matriks[2][2]: 3
Masukkan elemen matriks[2][3]: 1
Matriks Augmented sebelum eliminasi:
1.0 2.0 1.0
2.0 3.0 1.0


Matriks Augmented setelah eliminasi:
1.0 2.0 1.0
0.0 1.0 1.0


Solusi SPL:
x1 = -1.0
x2 = 1.0
BUILD SUCCESSFUL (total time: 23 seconds)
```

10. Output Sistem Persamaan Linier 3x3



BAB V KESIMPULAN

Kesimpulan

Program ini dibuat menggunakan pemograman Java dengan file projek Matriks yang di dalam nya ada package Main dan Rumus. Di dalam package Main dibuat untuk menyatukan program pada package Rumus. Kemudian di dalam package Rumus terdapat beberapa class untuk membuat rumus-rumus matriks dengan pemograman.

Program-program yang dibuat pada tugas ini:

1. Menghitung Penjumlahan dan Pengurangan Matriks.
2. Menghitung Matriks Transpose dan didalam nya ada sub-menu 2x2 dan 3x3.
3. Menghitung Matriks Balikan.
4. Menghitung Determinan dan ada sub-menu seperti Matriks Transpose.
5. Menghitung solusi SPL dengan berbagai metode, yaitu eliminasi Gauss dan eliminasi Gauss-Jordan

DAFTAR PUSTAKA

https://akupintar.id/info-pintar/-/blogs/matriks-pengertian-operasi-determinan-invers-dan-contoh-soal

https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-02-Matriks-Eselon.pdf?authuser=0

https://fti.ars.ac.id/blog/content/matriks--jenis-jenis-matriks

https://www.detik.com/edu/detikpedia/d-5563519/pengertian-invers-matriks-dan-istilah-istilahnya

https://www.mademattka.net/2017/08/pengertian-minor-kofaktor-matriks.html

https://www.mademattka.net/2017/08/pengertian-minor-kofaktor-matriks.html

https://id.wikipedia.org/wiki/Kaidah_Cramer

https://informatika.stei.itb.ac.id/~rinaldi.munir/MetNum/2010-2011/Interpolasi%20Polinom.pdf