# Laporan Tugas Besar Algeo 1


Anggota:

William Andrian Dharma T. - 13523006

Nathan Jovial Hartono - 13523032

Abdullah Farhan - 13523042

# BAB I : Deskripsi Masalah

## I.I. Sistem Persamaan Linier (SPL)

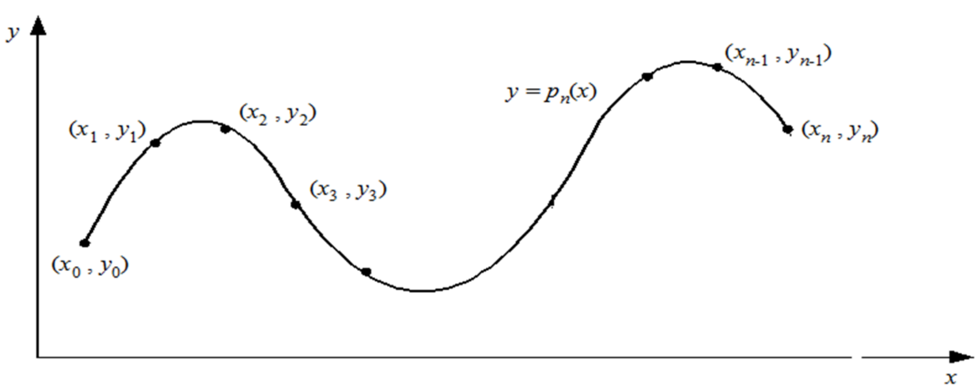
Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan ($x = A^{-1}b$), dan kaidah *Cramer* (khusus untuk SPL dengan *n* peubah dan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

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

**Gambar 1.1 Eliminasi Gauss dilakukan dengan matriks eselon baris dan eliminasi Gauss-Jordan dengan matriks eselon baris tereduksi.**

## I.II. Interpolasi Polinomial

Persoalan interpolasi polinom adalah sebagai berikut: Diberikan $n$+1 buah titik berbeda, $(x_0, y_0)$, $(x_1, y_1)$, ..., $(x_n, y_n)$. Tentukan polinom $p_n(x)$ yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga $y_i = p_n(x_i)$ untuk $i$ = 0, 1, 2, ..., $n$.




**Gambar 1.2 Ilustrasi beberapa titik yang diinterpolasi secara polinomial.**

Setelah polinom interpolasi $p_n(x)$ ditemukan, $p_n(x)$ dapat digunakan untuk menghitung perkiraan nilai $y$ di sembarang titik di dalam selang $[x_0, x_n]$.

Untuk persoalan interpolasi, masukannya jika dari *keyboard* adalah $n$, $(x_0, y_0)$, $(x_1, y_1)$, ..., $(x_n, y_n)$, dan nilai $x$ yang akan ditaksir nilai fungsinya. Jika masukannya dari *file*, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung. Masukan kemudian dilanjutkan dengan satu buah baris berisi satu buah nilai $x$ yang akan ditaksir menggunakan fungsi interpolasi yang telah didefinisikan. Misalnya jika titik-titik datanya adalah (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513) dan akan mencari nilai $y$ saat $x$ = 8.3, maka di dalam *file text* ditulis sebagai berikut:

```
8.0 2.0794
9.0 2.1972
9.5 2.2513
8.3
```

Untuk persoalan polinom interpolasi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada $x$ yang diberikan. Contoh luaran untuk interpolasi adalah

$$f(x) = -0.0064x^2 + 0.2266x + 0.6762, \quad f(5) = \ldots$$

## I.III. Regresi Berganda

Regresi (akan dipelajari lebih lanjut di Probabilitas dan Statistika) merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Pada tugas besar ini, anda diminta untuk membuat 2 jenis regresi yaitu Regresi Linier Berganda dan Regresi Kuadratik Berganda.

### 1. Regresi Linier Berganda

Meskipun sudah ada persamaan jadi untuk menghitung regresi linear sederhana, terdapat persamaan umum dari regresi linear yang bisa digunakan untuk regresi linear berganda, yaitu.

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

**Gambar 1.3 Ilustrasi persamaan umum dari regresi linear untuk regresi linear berganda.**

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

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

## 2. Regresi Kuadratik Berganda

Bentuk persamaan dari regresi kuadratik ada 3, yaitu:

a. Variabel Linier: Variabel dengan derajat satu seperti X, Y, dan Z
b. Variabel Kuadrat: Variabel dengan derajat dua seperti $X^2$
c. Variabel Interaksi: 2 Variabel dengan derajat satu yang dikalikan dengan satu sama lain seperti XY, YZ, dan XZ

## 3. Persoalan Umum

Untuk persoalan regresi, masukannya jika dari *keyboard* adalah *n* (jumlah peubah *x*), *m* (jumlah sampel), semua nilai-nilai $x_{1i}$, $x_{2i}$, ..., $x_{ni}$, nilai $y_i$, dan nilai-nilai $x_k$ yang akan ditaksir nilai fungsinya. Jika masukannya dari *file*, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.

Untuk persoalan polinom regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada *x* yang diberikan. Luaran untuk regresi adalah sebagai berikut:

$$f(x) = -9.5872 + 1.0732x_1, \quad f(x_k) = \ldots$$

untuk kasus regresi kuadratik, variabel boleh menggunakan $x_1$, $x_2$, dan lain-lain tetapi perlu dijelaskan variabel tersebut merepresentasikan apa. Berikut contoh luaran untuk kuadratik.

$x_1$ = X

.

$x_3$ = $X^2$

.

$x_5$ = XY

[Persamaan dan Solusi]

## I.VI. Bicubic Spline Interpolation

*Bicubic spline interpolation* adalah metode interpolasi yang digunakan untuk mengaproksimasi fungsi di antara titik-titik data yang diketahui. Pendekatan ini menciptakan permukaan yang halus dan kontinu, memungkinkan untuk perluasan data secara visual yang lebih akurat daripada metode interpolasi linear.

Untuk studi kasus ini, buatlah matriks $X$ menggunakan persamaan yang ada (tidak *hardcode*) serta carilah invers matriks $X$ dengan *library* yang telah kalian buat dalam penyelesaian masalah. Berikut adalah sebuah tautan yang dapat dijadikan referensi.Tugas Anda pada studi kasus ini adalah membangun persamaan $f(x, y)$ yang akan digunakan untuk melakukan interpolasi berdasarkan nilai $f(a, b)$ dari masukan matriks 4 x 4. Nilai masukan $a$ dan $b$ berada dalam rentang [0, 1]. Nilai yang akan diinterpolasi dan turunan berarah disekitarnya dapat diilustrasikan pada titik berwarna merah pada gambar di bawah.

Untuk persoalan *bicubic spline interpolation*, masukan dari *file text* (.txt) yang berisi matriks berukuran 4 x 4 yang berisi konfigurasi nilai fungsi dan turunan berarah disekitarnya, diikuti dengan nilai $a$ dan $b$ untuk mencari nilai $f(a, b)$. Misalnya jika nilai dari $f(0, 0)$, $f(1, 0)$, $f(0, 1)$, $f(1, 1)$, $f_x(0, 0)$, $f_x(1, 0)$, $f_x(0, 1)$, $f_x(1, 1)$, $f_y(0, 0)$, $f_y(1, 0)$, $f_y(0, 1)$, $f_y(1, 1)$, $f_{xy}(0, 0)$, $f_{xy}(1, 0)$, $f_{xy}(0, 1)$, $f_{xy}(1, 1)$ berturut-turut adalah 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 serta nilai $a$ dan $b$ yang dicari berturut-turut adalah 0.5 dan 0.5 maka isi *file text* ditulis sebagai berikut:

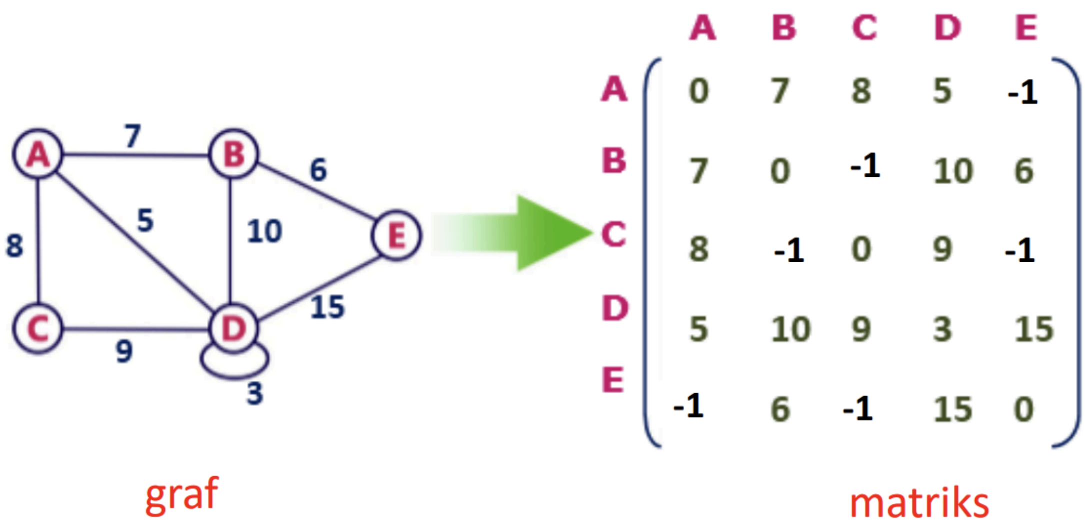
```
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
0.5 0.5
```

Luaran yang dihasilkan adalah nilai dari $f(0.5, 0.5)$.

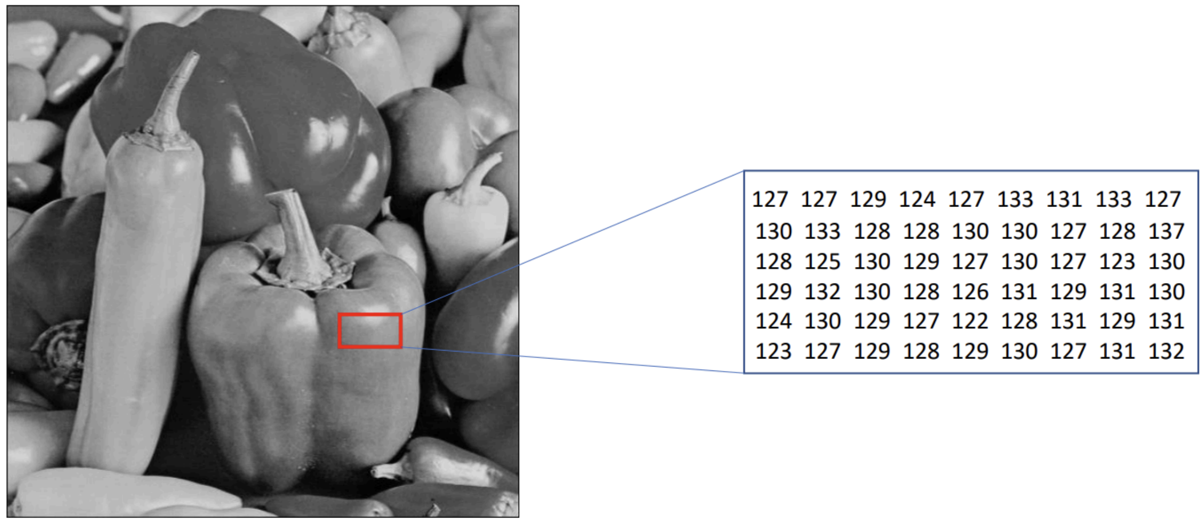
# BAB II : Teori Singkat

## II.I. Definisi Matriks

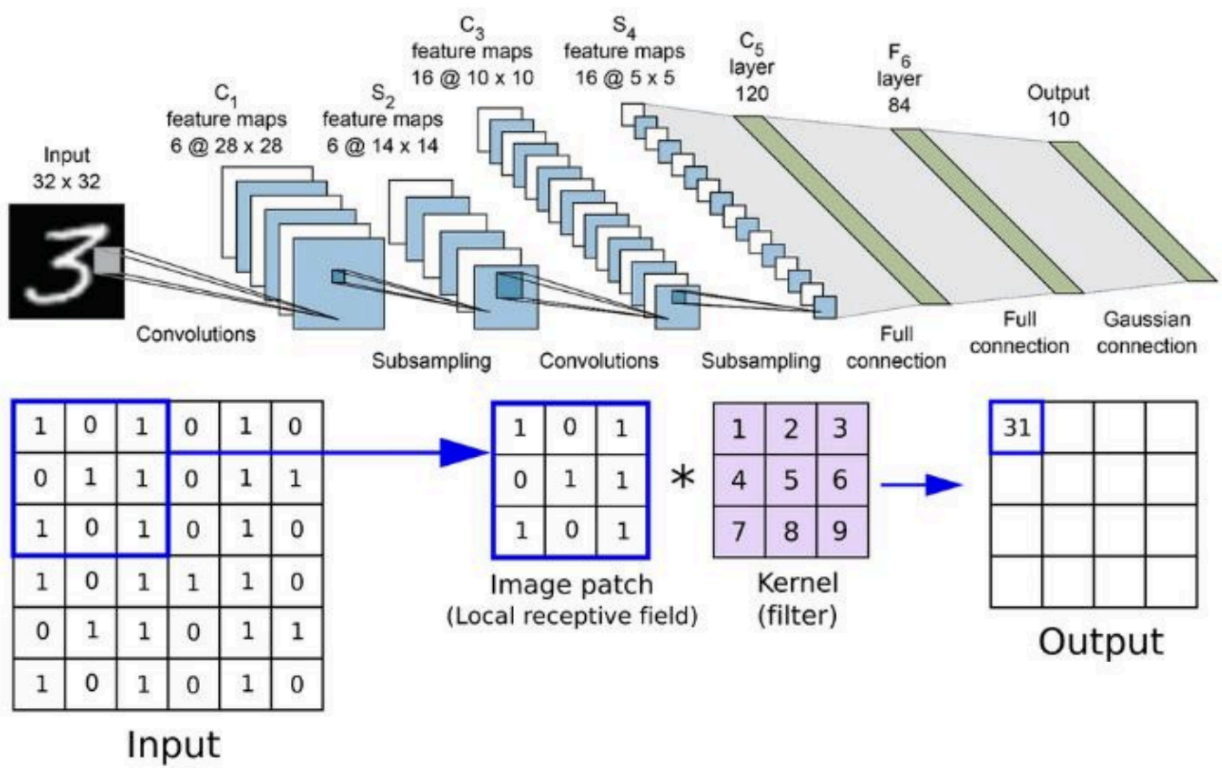
Matriks merupakan sebuah set “2 dimensional” dengan elemen, berupa bilangan dalam konteks aljabar linear, tersusun pada setiap baris dan kolom set, merepresentasikan sebuah “mathematical object”. Matriks banyak digunakan untuk merepresentasikan jenis data dalam bidang keinformatikaan. Contohnya representasi graf dengan nilai edge antar node, representasi data gambar dengan menyimpan nilai rgb-a tiap pixel, matriks kernel pada deep learning, dan banyak aplikasi lainnya. Salah satu aplikasi matriks yang menjadi dasar dalam teori aljabar linear yaitu penyelesaian sistem persamaan linear dengan matriks.




**Gambar 2.1 Representasi graf dalam bentuk matriks**




**Gambar 2.2 Representasi gambar format PNG dalam bentuk Matriks**




**Gambar 2.3 Matriks kernel dalam metode *deep learning***

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + b_1 = 0 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + b_2 = 0 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + b_m = 0, \end{cases}$$

$$\boldsymbol{A}\mathbf{x} = \mathbf{b}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

**Gambar 2.4 Representasi matriks untuk sistem persamaan linear**

## II.II. Operasi Baris Elementer

Metode paling mendasar untuk menyelesaikan Sistem Persamaan Linear (SPL) menggunakan metode eliminasi Gauss dengan Operasi Baris Elementer (OBE) pada matriks augmented seperti Gambar 2.4. Operasi baris elementer melibatkan tiga operasi untuk mengubah matriks ke bentuk yang diinginkan. Ketiga operasi tersebut yaitu, Sebuah baris dapat dikalikan dengan konstanta tidak nol, sebuah baris dapat dipertukarkan dengan baris lain, dan sebuah baris dapat ditambahkan dengan kelipatan baris lain.

## II.III. Metode Gauss

Metode eliminasi Gauss merupakan salah satu metode penyelesaian SPL menggunakan matriks. Terdapat 3 langkah untuk memperoleh solusi SPL. Pertama, SPL dinyatakan dalam bentuk matriks augmented sebagai berikut. Lalu, lakukan proses OBE pada matriks augmented sehingga terbentuk matriks eselon baris atau matriks dengan setiap baris memiliki satu utama. Terakhir, SPL yang berkorespondensi pada matriks eselon baris diselesaikan dengan teknik penyulihan mundur (*Backward Substitution*). Perlu diketahui terdapat tiga jenis solusi dalam proses eliminasi ini, yaitu solusi unik, solusi parametrik, dan tidak ada solusi. Pada SPL Homogen (*Ax = O*) hanya memiliki solusi trivial dan solusi non-trivial.

$$\begin{aligned} 2x_1 + 3x_2 - x_3 &= 5 \\ 4x_1 + 4x_2 - 3x_3 &= 3 \\ -2x_1 + 3x_2 - x_3 &= 1 \end{aligned} \quad \Rightarrow \quad \begin{bmatrix} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{bmatrix}$$

**Gambar 2.5 Transformasi SPL menjadi matriks augmented**

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

**Gambar 2.6 Transformasi matriks augmented menjadi matriks eselon baris**

$$\begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix} \quad \Rightarrow \quad \begin{aligned} x_1 + 3/2x_2 - 1/2x_3 &= 5/2 \\ x_2 + 1/2x_3 &= 7/2 \\ x_3 &= 3 \end{aligned}$$

(iii) $x_3 = 3$

(ii) $x_2 + 1/2x_3 = 7/2 \rightarrow x_2 = 7/2 - 1/2(3) = 2$

(i) $x_1 + 3/2x_2 - 1/2x_3 = 5/2 \rightarrow x_1 = 5/2 - 3/2\,(2) - 1/2\,(3) = 1$

$$x_1 = 1,\ x_2 = 2,\ x_3 = 3$$

**Gambar 2.7 Teknik Penyulihan Mundur (*Backward Substitution*)**

## II.IV. Metode Gauss-Jordan

Metode eliminasi Gauss-Jordan merupakan metode eliminasi merupakan pengembangan dari metode gauss, dimana matriks augmented menerapkan OBE menghasilkan matriks eselon baris tereduksi. Metode ini tidak memerlukan *backward substitution* untuk memperoleh nilai dari setiap variabel SPL, nilai variabel langsung diperoleh dari matriks augmented akhir. Metode eliminasi Gauss-Jordan terdiri dari dua fase, fase maju atau fase eliminasi Gauss dan fase mundur atau fase menghasilkan nol diatas satu utama.

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

**Gambar 2.8 Transformasi Matriks Augmented menjadi Matriks Eselon Baris Tereduksi**

## II.V. Matrix Inverse

Matriks balikan dapat dihitung menggunakan metode eliminasi Gauss–Jordan. Untuk matriks A berukuran n x n dengan determinan tidak sama dengan 0 memiliki balikan yang dicari dengan cara berikut:

$$[A|I] \overset{\text{G-J}}{\sim} [I|A^{-1}]$$

**Gambar 2.9 Rumus Menghitung Matriks Balikan dengan Gauss-Jordan**

dimana matriks I merupakan matriks identitas berukuran n x n. Metode eliminasi Gauss-Jordan diterapkan secara simultan untuk A maupun I dengan contoh sebagai berikut. Jika A tidak memiliki balikan, A dinamakan matriks singular. Untuk SPL *Ax = b*, apabila A memiliki balikan, maka *Ax = b* tidak memiliki solusi yang tunggal. Jika A memiliki balikkan, maka SPL memiliki solusi unik. Untuk SPL homogen *Ax = 0*, SPL memiliki solusi trivial jika A memiliki balikan dan solusi non trivial jika tidak memiliki balikan.

## II.VI. Determinan Matriks

Determinan matriks dapat ditemukan dengan bantuan proses eliminasi Gauss/Gauss-Jordan. Perlu diketahui terdapat dua jenis matriks segitiga, pertama matriks segitiga atas (*upper triangular*), dimana semua elemen di bawah diagonal utama adalah nol, dan matriks segitiga bawah (lower triangular), dimana semua elemen diatas diagonal utama adalah nol. Kedua matriks segitiga ini memiliki nilai determinan yaitu hasil kali semua elemen pada diagonal utama.

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

**Gambar 2.10 Matriks Segitiga Atas**

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

**Gambar 2.11 Matriks Segitiga Bawah**

Terdapat beberapa aturan determinan untuk matriks hasil manipulasi matriks lain. Misalkan matriks A adalah matriks n x n dan matriks B adalah matriks yang diperoleh dari manipulasi matriks A, dengan OBE. Jika sebuah baris pada A dikalikan dengan konstanta k, maka det(B) = k . det(A). Jika terdapat pertukaran baris diantara matriks A, maka det(B) = -det(A). Jika sebuah baris A ditambahkan dengan k kali baris lain, maka det(B) = det(A).

| Relationship | Operation |
| --- | --- |
| $\begin{vmatrix} ka_{11} & ka_{12} & ka_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = k \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$ <br> $\det(B) = k\det(A)$ | The first row of $A$ is multiplied by $k$. |
| $\begin{vmatrix} a_{21} & a_{22} & a_{23} \\ a_{11} & a_{12} & a_{13} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = - \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$ <br> $\det(B) = -\det(A)$ | The first and second rows of $A$ are interchanged. |
| $\begin{vmatrix} a_{11}+ka_{21} & a_{12}+ka_{22} & a_{13}+ka_{23} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$ <br> $\det(B) = \det(A)$ | A multiple of the second row of $A$ is added to the first row. |

**Gambar 2.12 Relasi OBE dengan Nilai Matriks**

$$[A] \overset{\text{OBE}}{\sim} [\text{matriks segitiga bawah}]$$

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \overset{\text{OBE}}{\sim} \begin{bmatrix} a'_{11} & a'_{12} & \dots & a'_{1n} \\ 0 & a'_{22} & \dots & a'_{2n} \\ \vdots & \vdots & \dots & a'_{3n} \\ 0 & 0 & 0 & a'_{nn} \end{bmatrix}$$

$$\text{maka } \det(A) = (-1)^p\, a'_{11}a'_{22} \dots a'_{nn} \qquad *)$$

*p* menyatakan banyaknya operasi pertukaran baris di dalam OBE

**Gambar 2.13 Determinan Matriks Setelah OBE**

## II.VII. Matriks Kofaktor

Matriks Kofaktor merupakan matriks yang terdiri atas kofaktor entri tiap elemen dari matriks A. Transpose dari matriks kofaktor menghasilkan matriks adjoin. Kofaktor matriks melibatkan minor entri dari sebuah elemen di matriks. Minor entri dari elemen merupakan determinan upa-matriks (submatrix) yang elemennya tidak berada pada baris i dan kolom j. Berikut merupakan rumus kofaktor pada elemen (i, j):

**Contoh 1**: Tinjau matriks A berikut $A = \begin{bmatrix} 6 & -3 & 1 \\ 2 & 2 & -4 \\ 1 & 5 & 3 \end{bmatrix}$

Minor entri dan kofaktor dari matriks A adalah sebagai berikut:

$M_{11}=\begin{vmatrix} 2 & -4 \\ 5 & 3 \end{vmatrix}=(2)(3)-(-4)(5) = 26$

$C_{11} = (-1)^{1+1}M_{11}=M_{11}=26$

$M_{12}=\begin{vmatrix} 2 & -4 \\ 1 & 3 \end{vmatrix}=(2)(3)-(-4)(1) = 10$

$C_{12} = (-1)^{1+2}M_{12}=-M_{12}=-10$

$M_{13}=\begin{vmatrix} 2 & 2 \\ 1 & 5 \end{vmatrix}=(2)(5)-(2)(1) = 8$

$C_{13} = (-1)^{1+3}M_{13}= M_{13}= 8$

$M_{22}=\begin{vmatrix} 6 & 1 \\ 1 & 3 \end{vmatrix}=(6)(3)-(1)(1) = 17$

$C_{22} = (-1)^{2+2}M_{22}= M_{22}= 17$

dan seterusnya untuk $M_{21}$, $M_{23}$, $M_{31}$, $M_{32}$, $M_{33}$ dihitung dengan cara yang sama

dan seterusnya untuk $C_{21}$, $C_{23}$, $C_{31}$, $C_{32}$, $C_{33}$ dihitung dengan cara yang sama

**Gambar 2.14 Rumus Kofaktor pada Entri Elemen**

**Contoh 6**: Tentukan matriks kofaktor dan adjoin dari matriks A berikut

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

Penyelesaian:

Maktriks kofaktor dari A adalah

$$\begin{bmatrix} +\begin{vmatrix} 6 & 3 \\ -4 & 0 \end{vmatrix} & -\begin{vmatrix} 1 & 3 \\ 2 & 0 \end{vmatrix} & +\begin{vmatrix} 1 & 6 \\ 2 & -4 \end{vmatrix} \\ -\begin{vmatrix} 2 & -1 \\ -4 & 0 \end{vmatrix} & +\begin{vmatrix} 3 & -1 \\ 2 & 0 \end{vmatrix} & -\begin{vmatrix} 3 & 2 \\ 2 & -4 \end{vmatrix} \\ +\begin{vmatrix} 2 & -1 \\ 6 & 3 \end{vmatrix} & -\begin{vmatrix} 3 & -1 \\ 1 & 3 \end{vmatrix} & +\begin{vmatrix} 3 & 2 \\ 1 & 6 \end{vmatrix} \end{bmatrix} = \begin{bmatrix} 12 & 6 & -16 \\ 4 & 2 & 16 \\ 12 & -10 & 16 \end{bmatrix}$$

**Gambar 2.15 Contoh Matriks Kofaktor dari A**

## II.VIII. Kaidah Cramer

Kaidah Cramer merupakan metode untuk mendapatkan solusi unik dari SPL Ax=b. Solusi unik setiap peubah melibatkan pembagian determinan matriks Aj dengan determinan matriks A, matriks Aj diperoleh dengan mengganti entri pada kolom ke-j dari A dengan entri dari matriks b, sebagai berikut:

$$x_1 = \frac{\det(A_1)}{\det(A)} \quad , \quad x_2 = \frac{\det(A_2)}{\det(A)} \quad , \ \dots \ , \quad x_n = \frac{\det(A_n)}{\det(A)}$$

**Gambar 2.16 Solusi Unik Kaidah Cramer**

$$A = \begin{bmatrix} -1 & 2 & -3 \\ 2 & 0 & 1 \\ 3 & -4 & 4 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \qquad \det(A) = \begin{vmatrix} 3 & 2 & -1 \\ 1 & 6 & 3 \\ 2 & -4 & 0 \end{vmatrix} = 10$$

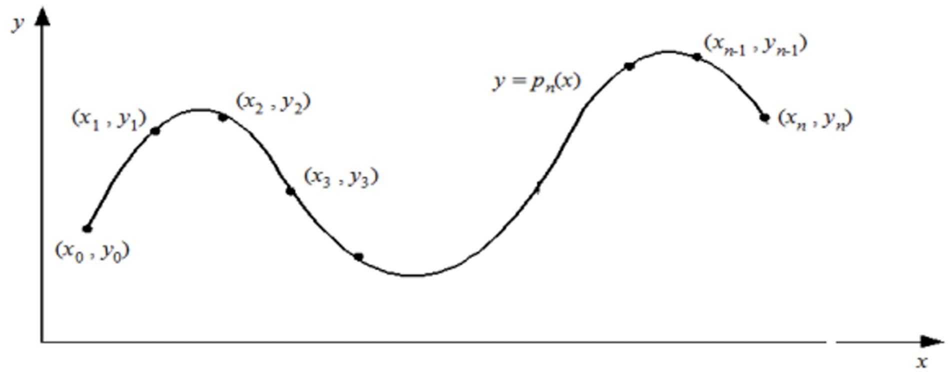
$$A_1 = \begin{bmatrix} 1 & 2 & -3 \\ 0 & 0 & 1 \\ 2 & -4 & 4 \end{bmatrix} \qquad A_2 = \begin{bmatrix} -1 & 1 & -3 \\ 2 & 0 & 1 \\ 3 & 2 & 4 \end{bmatrix} \qquad A_3 = \begin{bmatrix} -1 & 2 & 1 \\ 2 & 0 & 0 \\ 3 & -4 & 2 \end{bmatrix}$$

**Gambar 2.17 Entri Matriks Aj**

## II.IX. Interpolasi Polinomial

Interpolasi polinomial merupakan metode matematika yang digunakan untuk memperkirakan nilai antara titik-titik data yang diketahui, dengan mengasumsikan bahwa data mengikuti pola fungsi polinomial. Fungsi polinomial dapat berupa derajat satu (linear) dengan bentuk $f(x) = ax + b$, derajat dua (kuadrat) dengan bentuk $f(x) = ax^2 + bx + c$, atau derajat tinggi lainnya. Dalam proses interpolasi, langkah pertama adalah membentuk persamaan polinomial dari data yang ada, kemudian persamaan tersebut digunakan untuk interpolasi (mencari nilai di antara titik-titik data yang diketahui) dan ekstrapolasi (memprediksi nilai di luar rentang data yang ada). Bentuk matematis interpolasi polinomial untuk n+1 titik data $(x_0, y_0)$, $(x_1, y_1)$, ..., $(x_n, y_n)$ adalah persamaan polinomial derajat n: $p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$. Dalam persamaan tersebut, x adalah variabel bebas, $a_0$, $a_1$, $a_2$, ..., $a_n$ adalah koefisien yang dicari, dan n adalah derajat polinomial. Untuk mencari koefisien $a_0$, $a_1$, $a_2$, ..., $a_n$ dilakukan dengan membentuk sistem persamaan linear dan menggunakan metode eliminasi Gauss, dimana setiap titik data menghasilkan satu persamaan.

Setelah koefisien ditemukan, persamaan dapat digunakan untuk menghitung nilai y untuk sebarang nilai x dalam rentang data, memperkirakan tren atau pola data, dan membuat prediksi untuk nilai-nilai baru. Interpolasi polinomial memiliki keunggulan yaitu dapat menghasilkan kurva yang mulus dan cocok untuk data yang mengikuti pola polinomial. Namun, metode ini juga memiliki keterbatasan dimana semakin tinggi derajat polinomial maka semakin kompleks perhitungannya dan dapat menghasilkan osilasi yang tidak diinginkan untuk derajat tinggi. Interpolasi polinomial banyak digunakan dalam berbagai bidang seperti analisis data ilmiah, pemodelan matematika, prediksi nilai dalam rekayasa, dan analisis statistik. Secara keseluruhan, interpolasi polinomial adalah metode yang powerful untuk menganalisis dan memprediksi data. Namun demikian, perlu kehati-hatian dalam memilih derajat polinomial yang sesuai dengan karakteristik data yang dianalisis.




**Gambar 2.18 Visualisasi Interpolasi Polinom**

## II.X. Regresi Linier Berganda

Regresi Linier Berganda adalah teknik untuk mendapatkan model dari banyak variabel fitur terhadap suatu variabel target yang dirumuskan dengan rumus dibawah ini.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon$$

Dimana untuk sejumlah i observasi:

$y_i$ = Variabel target / dependen

$x_i$ = Variabel fitur / penjelas

$\beta_0$ = Konstanta / y - intercept / *bias*

$\beta_p$ = Koefisien variable / *weight*

$\epsilon$ = Error / residual

Cara untuk mendapatkan nilai $\beta$ adalah sebagai berikut:

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

Kemudian solusinya dapat didapat dengan menggunakan penyelesaian SPL.

## II.XI. Regresi Kuadratik Berganda

Regresi Kuadratik Berganda adalah teknik untuk mendapatkan sebuah model yang menjelaskan hubungan variabel fitur dengan variabel target selain dengan linear, juga dengan ekspresi kuadratik dan interaksi.

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_1^{\ 2} + \beta_3 x_2 + \beta_4 x_1 x_2 + \beta_5 x_2^{\ 2}$$

Penyelesaian dari regresi kuadratik berganda dapat direpresentasikan dengan

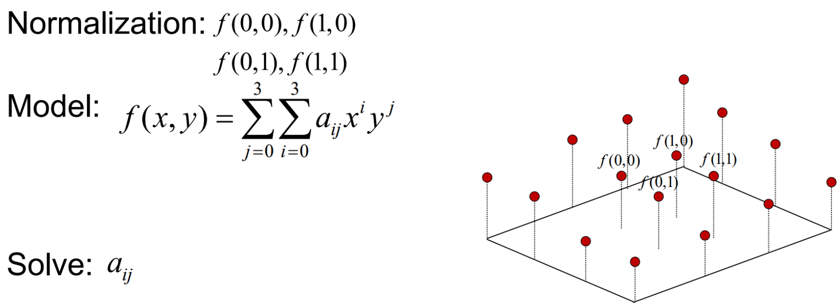
$$
\underbrace{\begin{pmatrix}
N & \sum u_i & \sum v_i & \sum u_i^2 & \sum u_i v_i & \sum v_i^2 \\
\sum v_i & \sum u_i v_i & \sum v_i^2 & \sum u_i^2 v_i & \sum u_i v_i^2 & \sum v_i^3 \\
\sum u_i^2 & \sum u_i^3 & \sum u_i^2 v_i & \sum u_i^4 & \sum u_i^3 v_i & \sum u_i^2 v_i^2 \\
\sum u_i v_i & \sum u_i^2 v_i & \sum u_i v_i^2 & \sum u_i^3 v_i & \sum u_i^2 v_i^2 & \sum u_i v_i^3 \\
\sum v_i^2 & \sum u_i v_i^2 & \sum v_i^3 & \sum u_i^2 v_i^2 & \sum u_i v_i^3 & \sum v_i^4
\end{pmatrix}}_{\mathbf{A}}
\begin{pmatrix} a^* \\ b^* \\ c^* \\ d^* \\ e^* \\ f^* \end{pmatrix}
=
\underbrace{\begin{pmatrix} \sum y_i \\ \sum y_i u_i \\ \sum y_i v_i \\ \sum y_i u_i^2 \\ \sum y_i u_i v_i \\ \sum y_i v_i^2 \end{pmatrix}}_{\mathbf{b}}
$$

Kemudian akan digunakan metode penyelesaian SPL untuk menemukan setiap nilai koefisien dari ekspresi yang ada.

## II.XII. Bicubic Spline Interpolation

*Bicubic spline interpolation* melibatkan konsep *spline* dan konstruksi serangkaian polinomial kubik di dalam setiap sel segi empat dari data yang diberikan. Pendekatan ini menciptakan permukaan yang halus dan kontinu, memungkinkan untuk perluasan data secara visual yang lebih akurat daripada metode interpolasi linear.

Dalam pemrosesan menggunakan interpolasi *bicubic spline* digunakan 16 buah titik, 4 titik referensi utama di bagian pusat, dan 12 titik di sekitarnya sebagai aproksimasi turunan dari keempat titik referensi untuk membagun permukaan bikubik. Bentuk pemodelannya adalah sebagai berikut.




**Gambar 2.x Pemodelan Interpolasi Bicubic Spline**

Selain melibatkan model dasar, juga digunakan model turunan berarah dari kedua sumbu, baik terhadap sumbu *x*, sumbu *y*, mapun keduanya. Persamaan polinomial yang digunakan adalah sebagai berikut.

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

$$f_x(x,y) = \sum_{j=0}^{3}\sum_{i=1}^{3} a_{ij} i x^{i-1} y^j$$

$$f_y(x,y) = \sum_{j=1}^{3}\sum_{i=0}^{3} a_{ij} j x^i y^{j-1}$$

$$f_{xy}(x,y) = \sum_{j=0}^{3}\sum_{i=0}^{3} a_{ij} ij x^{i-1} y^{j-1}$$

**Gambar 2.x persamaan fungsi penurunan dua variabel**

Dengan menggunakan nilai fungsi dan turunan berarah tersebut, dapat terbentuk sebuah matriks solusi *X* yang membentuk persamaan penyelesaian sebagai berikut.

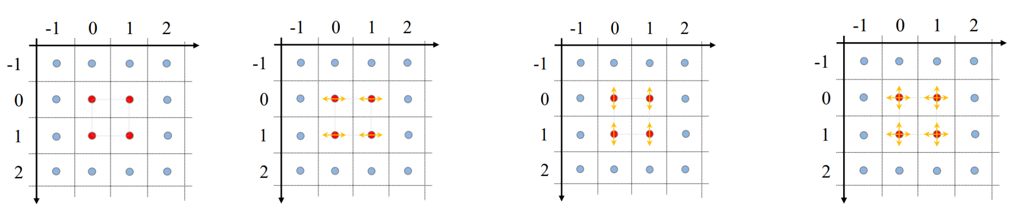
$$y = Xa$$

$$\begin{bmatrix} f(0,0) \\ f(1,0) \\ f(0,1) \\ f(1,1) \\ f_x(0,0) \\ f_x(1,0) \\ f_x(0,1) \\ f_x(1,1) \\ f_y(0,0) \\ f_y(1,0) \\ f_y(0,1) \\ f_y(1,1) \\ f_{xy}(0,0) \\ f_{xy}(1,0) \\ f_{xy}(0,1) \\ f_{xy}(1,1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 2 & 4 & 6 & 0 & 3 & 6 & 9 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30} \\ a_{01} \\ a_{11} \\ a_{21} \\ a_{31} \\ a_{02} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{03} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}$$

**Gambar 2.x Pemodelan Interpolasi Bicubic Spline**

Perlu diketahui bahwa elemen pada matriks $X$ adalah nilai dari setiap komponen koefisien $a_{ij}$ yang diperoleh dari persamaan fungsi maupun persamaan turunan yang telah dijelaskan sebelumnya. Sebagai contoh, elemen matriks $X$ pada baris 8 kolom ke 2 adalah koefisien dari $a_{10}$ pada ekspansi sigma untuk $f_x(1, 1)$ sehingga diperoleh nilai konstanta $1 \times 1^{1-1} \times 1^0 = 1$, sesuai dengan isi matriks $X$.

Nilai dari vektor $a$ dapat dicari dari persamaan $y = Xa$, lalu vektor $a$ tersebut digunakan sebagai nilai variabel dalam $f(x, y)$, sehingga terbentuk fungsi interpolasi bicubic sesuai model.




**Gambar 2.x Nilai fungsi yang akan di interpolasi pada titik merah, turunan berarah terhadap sumbu x, terhadap sumbu y, dan keduanya (kiri ke kanan)**

# BAB III : Implementasi

## 1. Class Matrix

### 1.1. Tabel 3.1.1 Attribute

| Nama dan Tipe | Deskripsi |
|---|---|
| public double[][] data | Tempat penyimpanan elemen matriks |
| public int rows | Jumlah baris pada matriks |
| public int cols | Jumlah kolom pada matrik |

### 1.2. Tabel 3.1.2 Method

| Fungsi/Prosedur | Deskripsi |
|---|---|
| public Matrix(int rows, int cols) | Konstruktor dari matriks. Membuat matriks dengan rows baris dan cols kolom. |
| public void readMatrix(Scanner scanner) | Membaca matrix dari input user dengan scanner |
| public void readMatrix(Scanner scanner) | Jumlah kolom pada matrik |
| public Matrix createIdentityMatrix() | Membuat matriks identitas |
| public Matrix copyMatrix() | Menyalin matriks ke matriks yang lain |
| public void setElmt(int row, int col, double val) | Set elemen matriks pada (row, col) public |
| public double getElmt(int row, int col) | Mendapatkan elemen matriks pada (row, col) |
| public int getRows() | Mendapatkan jumlah row matrix |

| | |
|---|---|
| public double[] getRow(int row) | Mendapatkan semua elemen matriks baris ke row |
| public void setRow(int row, double[] values) | Set semua elemen matriks baris ke row dari input sebuah array |
| public int getCols() | Mendapatkan jumlah kolom matriks |
| public double[] getCol(int col) | Mendapatkan matrix kolom ke-col |
| public void setCol(int col, double[] values) | Set matrix kolom ke-col, input array of double |
| public double[][] getData() { | Mendapatkan matrix data dalam 2d list |
| public void fillMatrix(double value) | Mengisi matriks dengan sebuah konstanta |
| public void swapRows(int row1, int row2) | Swap baris row1 dan row2 |
| public void displayMatrix() | Menampilkan matriks ke layar, dengan format 2 angka di belakang koma<br><br>Setiap elemen pada sebuah baris dipisah dengan spasi<br>public |
| **ERROR HANDLING (BOOL)** | |
| public boolean isSquare() | Cek apakah matriks bentuk persegi |
| public boolean isRowIdxValid(int row) | Cek apakah index baris valid |
| public boolean isColIdxValid(int col) | Cek apakah index kolom valid |
| public boolean isIdxValid(int row, int col) | Cek apakah index baris dan kolom valid |

| | |
|---|---|
| public boolean isMatrixSizeEqual(Matrix m2) | Cek  apakah size matriks sama |
| public boolean isMatrixEqual(Matrix m2) | Cek apakah matriks sama dengan matriks m2 |
| **ARITHMETIC APPLICATION** | |
| public void multiplyMatrixConst(double constant) | Mengalikan matriks dengan konstanta |
| public Matrix multiplyMatrix(Matrix m2) | Mengalikan matriks dengan matriks m2 |
| **MATRIX MODIFICATION** | |
| public Matrix transpose() | Mendapatkan transpose matriks ukuran rows x cols |
| public Matrix adjoint() | Mendapatkan matriks adjoint |
| public Matrix inverse() | Mendapatkan matriks inverse |
| public Matrix inverseRedRow() | Mendapatkan matriks inverse dengan matriks eselon tereduksi |
| public static Matrix augmentedMatrix(Matrix m1, Matrix m2) | Split  augmented matriks menjadi matriks m1 dan m2<br><br>Tidak return matriks, langsung set m1 dan m2 |
| public static void splitAugmentedMatrix(Matrix m, Matrix m1, Matrix m2) | Split augmented matriks menjadi matriks m1 dan m2<br><br>Tidak return matriks, langsung set m1 dan m2 |

| DETERMINAN | |
|---|---|
| public double determinant() | Mendapatkan determinant matriks, (hanya matriks square) menggunakan metode kofaktor |
| public double determinantRedRow() | Mencari determinan dengan menggunakan metode reduction row |
| public Matrix minor(int i,int j) | Mendapatkan minor matriks pada baris i dan kolom |
| **ROW ELEMENTARY OPERATIONS** | |
| public Matrix cofactor() | Mendapatkan matriks kofaktor |
| public boolean isRowAllZero(int row) | Cek apakah baris row nol semua |
| public void toRowEchelonForm() | Fungsi utama untuk mengubah matriks menjadi bentuk eselon baris |
| public int findLeftmostNonZeroColumn(int startRow) | Fungsi baru untuk mencari kolom non-zero paling kiri |
| public void ensureNonZeroPivot(int pivotRow, int pivotCol) | Fungsi baru untuk menukar baris jika elemen pivot adalah nol |
| public void dividePivotRow(int pivotRow, int pivotCol) | Fungsi untuk membagi baris dengan pivotnya |
| public void subtractPivotRow(int pivotRow, int pivotCol) | Fungsi untuk mengurangkan baris pivot dari baris-baris di bawahnya |
| public void toReducedRowEchelonForm() | Fungsi untuk mengubah matriks eselon baris menjadi bentuk eselon baris tereduksi |
| **COLUMN ELEMENTARY OPERATIONS** | |
| public int findPivotColumn(int row) | Fungsi baru untuk mencari kolom pivot dalam suatu baris |

| | |
|---|---|
| private void eliminateAbovePivot(int pivotRow, int pivotCol) | Fungsi untuk mengurangkan baris pivot dari baris-baris di bawahnya |
| private void roundToZero() | Fungsi untuk membulatkan elemen bernilai sangat kecil ke nol |
| **HELPER FUNCTION FOR SPL** | |
| public double[] backwardSubstitution(double[][] U, double[] Y, boolean isParametric) | Mendapatkan hasil jawaban SPL Gauss [x1 .. xn] |
| public int computeRank() | Hitung rank setelah matrix dibuat jadi matrix eselon baris |

## 2. Class LinearSystem

### 2.1. Tabel 3.2.1 Attribute

| Nama dan tipe | Deskripsi |
|---|---|
| private Matrix features | Matriks koefisien (A in Ax=b) |
| private Matrix target | Target value (b in Ax=b) |

### 2.2. Tabel 3.2.2 Method

| Fungsi/Prosedur | Deskripsi |
|---|---|
| public LinearSystem(Matrix matrix) | Konstruktor |
| public Matrix getFeatures() | Mendapatkan elemen matriks feature |
| public Matrix getTarget() | Mendapatkan semua elemen matriks target |

| | |
|---|---|
| public double[] gauss() | menyelesaikan SPL menggunakan metode Gauss |
| public double[] gaussJordan() | gunakan metode Gauss-Jordan |
| public Matrix CramerRule() | menyelesaikan SPL dengan metode Cramer |
| public double[] inverseMethodSPL(Matrix feature, Matrix target) | Mendapatkan jawaban SPL dengan matriks invers |
| public String checkSolutionType() | mengecek tipe solusi SPL |

## 3. Class Interpolation Polynomial

### 3.1. Tabel 3.3.1 Attribute

| Nama dan tipe | Deskripsi |
|---|---|
| private double[] coefficients | Nilai koefisien-koefisien pada polinom |
| private double[] x; | Nilai domain dari polinom |
| private double[] y; | Nilai kodomain dari polinom |
| private int n; | Dimensi dari polinom (berapa banyak variabel xn yang ada di polinom) |

### 3.2. Tabel 3.3.1 Method

| Fungsi/Prosedur | Deskripsi |
|---|---|
| public Interpolation(double[] x, double[] y) | Konstruktor |
| private void calculateCoefficients() | Mendapatkan nilai koefisien dari setiap variabel xn dengan bantuan metode gauss |

| | |
|---|---|
| public double[] getPolynomial() | Mendapatkan semua nilai koefisien yang ada |
| public double interpolate(double x) | Memprediksi nilai x pada polinom |

4. **Class LinearRegressor**

4.1. **Tabel 3.4.1 Attribute**

| Nama dan tipe | Deskripsi |
|---|---|
| private Matrix model | Mode yang disimpan sebagai c + x1 +..+xn |

4.2. **Tabel 3.4.1 Method**

| Fungsi/Prosedur | Deskripsi |
|---|---|
| public LinearRegressor() | Konstruktor |
| public Matrix biasedMatrix(Matrix matrix) | Menaikkan dimensi matriks dengan menambahkan 1 di setiap baris |
| public void fit(Matrix features, Matrix target) | Fit matriks dengan model linear (X^T * X) θ = X^T * y |
| public Matrix getModel() | Mendapatkan model |
| public void printModel() | Menampilkan model |
| public String toStringModel() | Mendapatkan model dalam bentuk string |
| public Matrix predict(Matrix features) | Prediksi model terhadap matrix fitur |

5. **Class BicubicSpline**

5.1. **Tabel 3.5.1 Attribute**

| Nama dan tipe | Deskripsi |
|---|---|

| | |
|---|---|
| private Matrix model | Hasil dari fitting data |

## 5.2. Tabel 3.5.2 Method

| Fungsi/Prosedur | Deskripsi |
|---|---|
| public BicubicalSpline() | Konstruktor |
| public Matrix setX() | Mengembalikan matriks X dengan ukuran 16x16 sesuai dengan formula bicubic spline |
| public Matrix resizeY | Mengubah matriks input menjadi matriks 16 x 1 sesuai dengan formula bicubic spline |
| public void fit(Matrix y) | Fit data dengan bicubic spline |
| public Matrix getResult() | Mengembalikan hasil fitting |
| public Matrix predict(Matrix xy) | Prediksi data dengan bicubical spline yang sudah di fit |

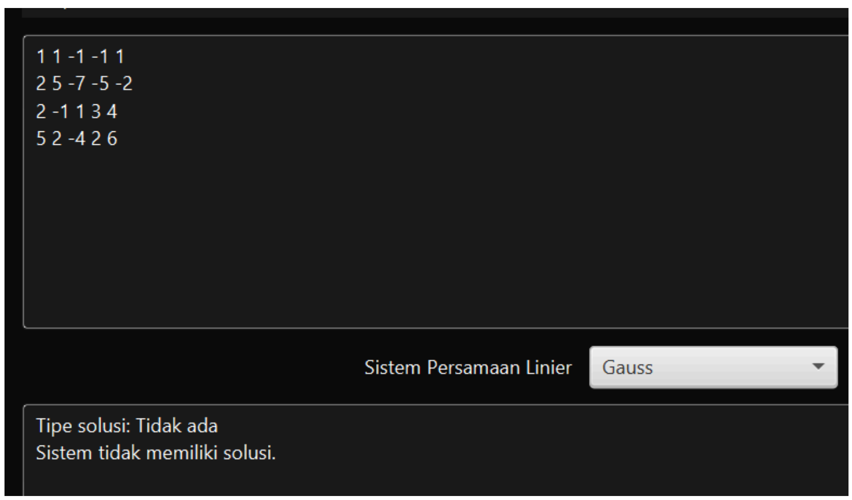
# BAB IV : Eksperimen

Untuk menguji program anda, tes dengan beberapa SPL, persoalan interpolasi polinom, dan matriks-matriks sebagai berikut :

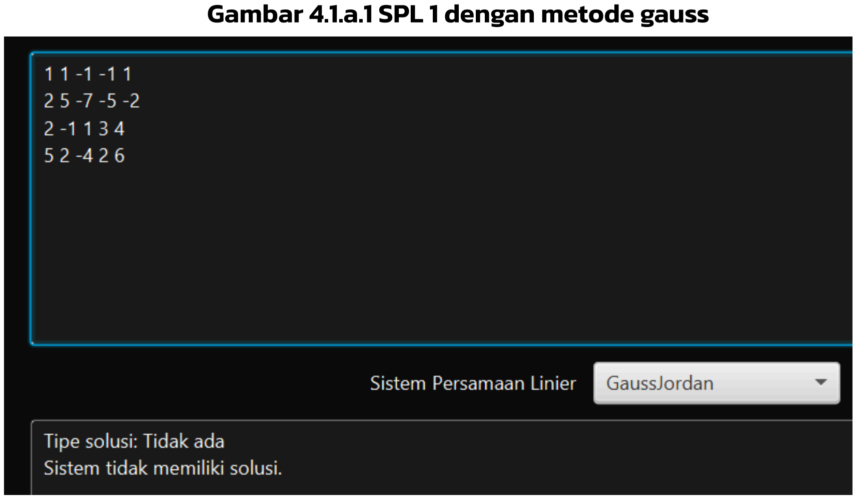
## 1. Temukan solusi SPL *Ax* = *b*, berikut:

a. A berukuran 4 x 4, b berukuran 4 x 1

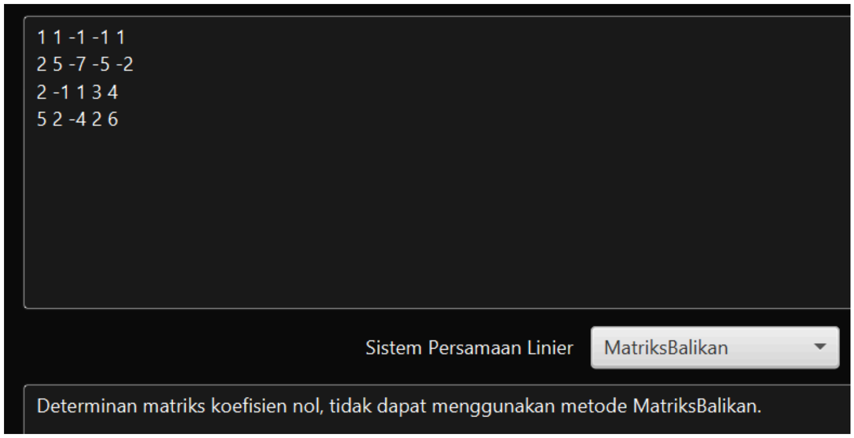
$$A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$$



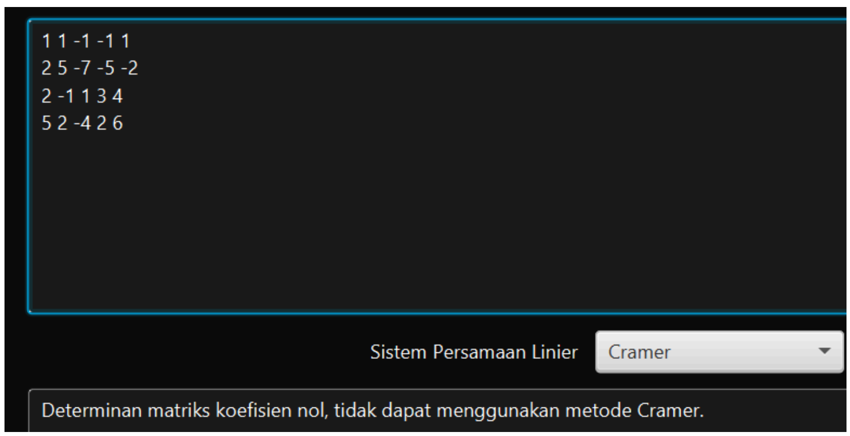

**Gambar 4.1.a.1 SPL 1 dengan metode gauss**



**Gambar 4.1.a.2 SPL 1 dengan metode gauss jordan**



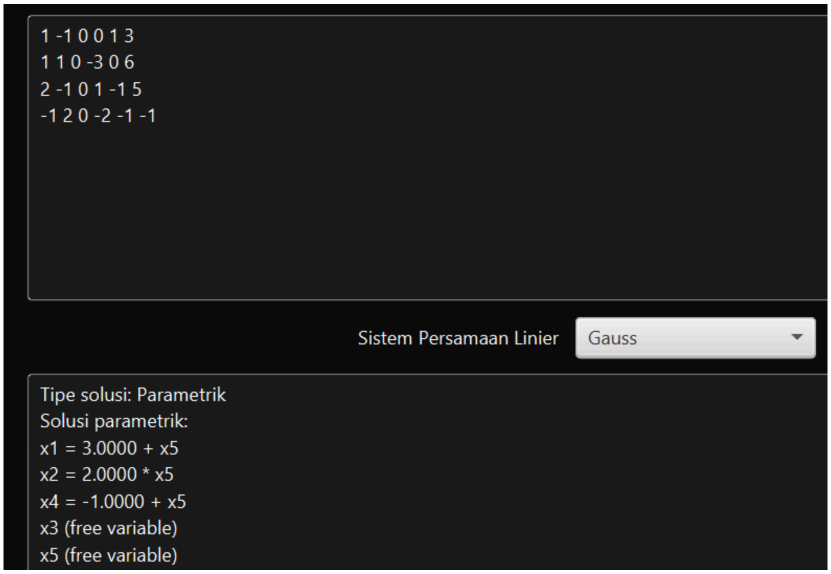

**Gambar 4.1.a.3 SPL 1 dengan metode matriks balikan**



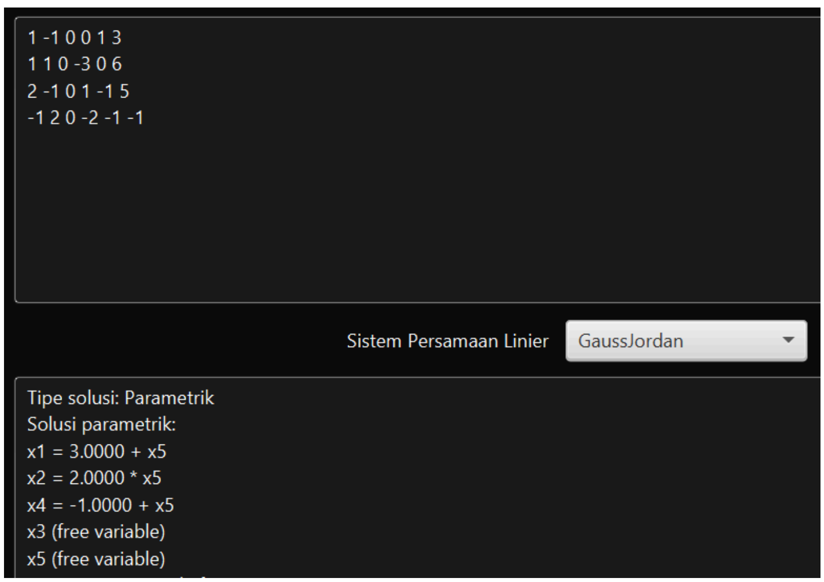

**Gambar 4.1.a.4 SPL 1 dengan metode cramer**

b. A berukuran 4 x 5, b berukuran 4 x 1

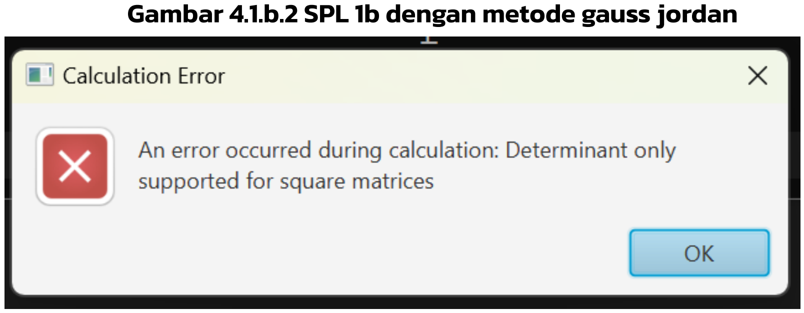
$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$$




**Gambar 4.1.b.1 SPL 1b dengan metode gauss**

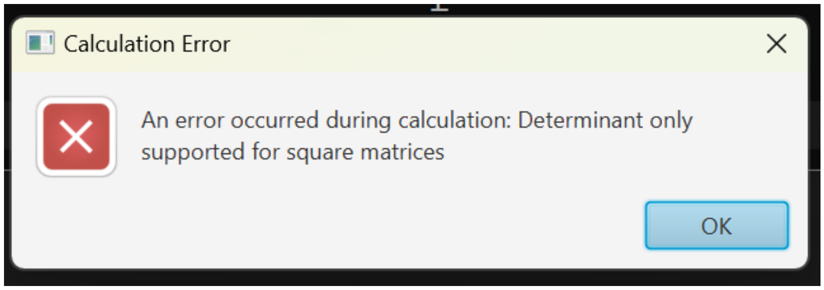



**Gambar 4.1.b.2 SPL 1b dengan metode gauss jordan**




**Gambar 4.1.b.3 SPL 1b dengan metode matriks balikan**

*Note: karena matrix koefisiennya tidak persegi determinannya tidak bisa dicari, karena kita mencari inversenya lewat 1/det * adjoin(matriks)*

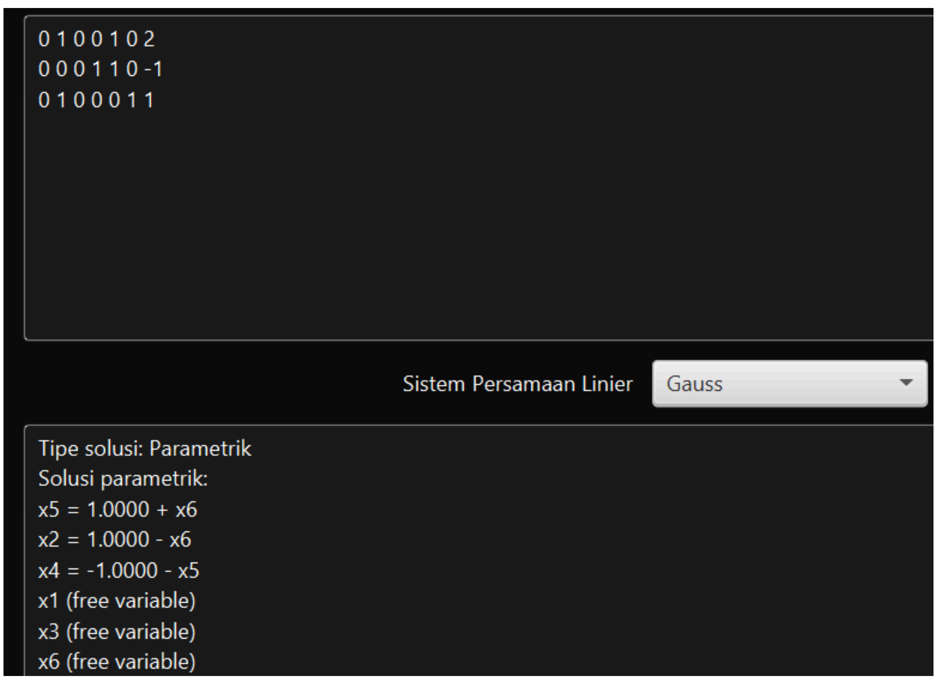



**Gambar 4.1.b.4 SPL 1b dengan metode cramer**

*Note: karena matrix koefisiennya tidak persegi determinannya tidak bisa dicari)*

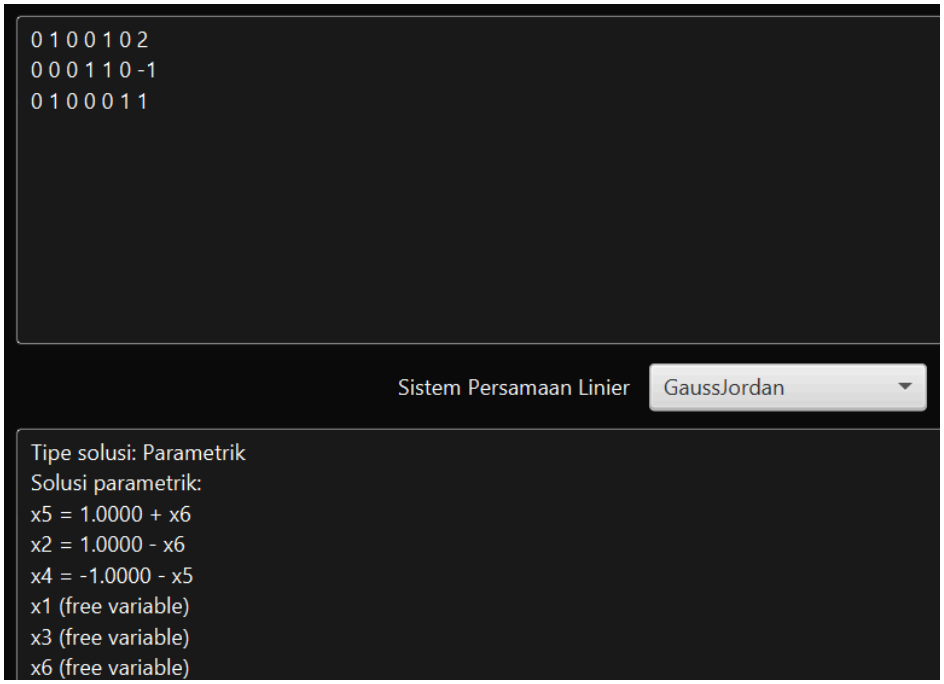
c. A berukuran 3 x 6, b berukuran 3 x 1

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

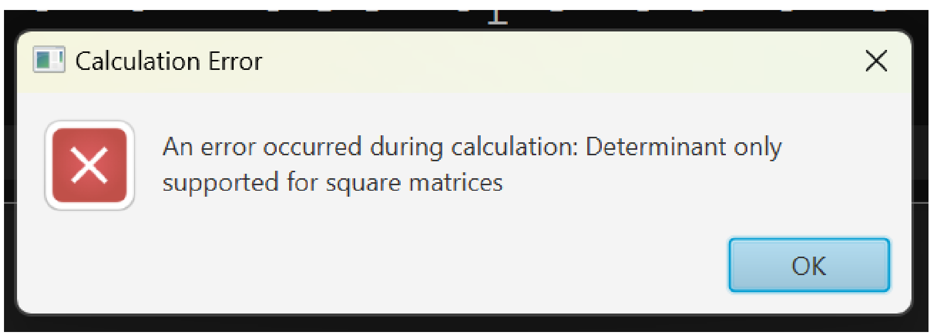



**Gambar 4.1.c.1 SPL 1c dengan metode gauss**

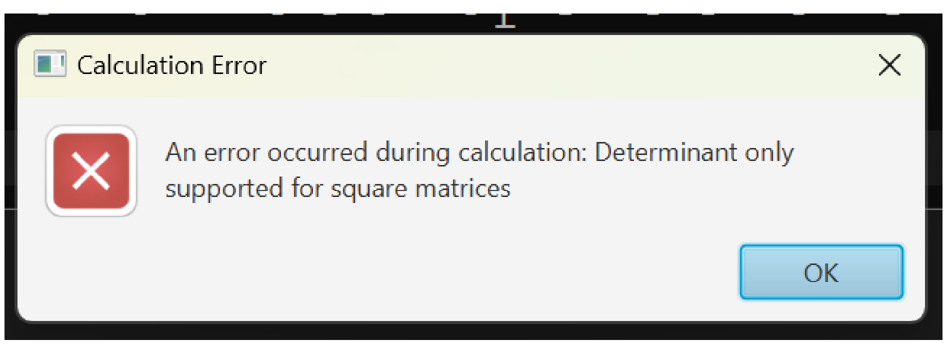
*Disclaimer: hasil di atas sudah benar tapi belum dalam bentuk yang paling sederhana. Karena x4 = −1 − x5 = −1 − (1 + x6) = −2 − x6*




**Gambar 4.1.c.2 SPL 1c dengan metode gauss jordan**




**Gambar 4.1.c.3 SPL 1c dengan metode matriks balikan**



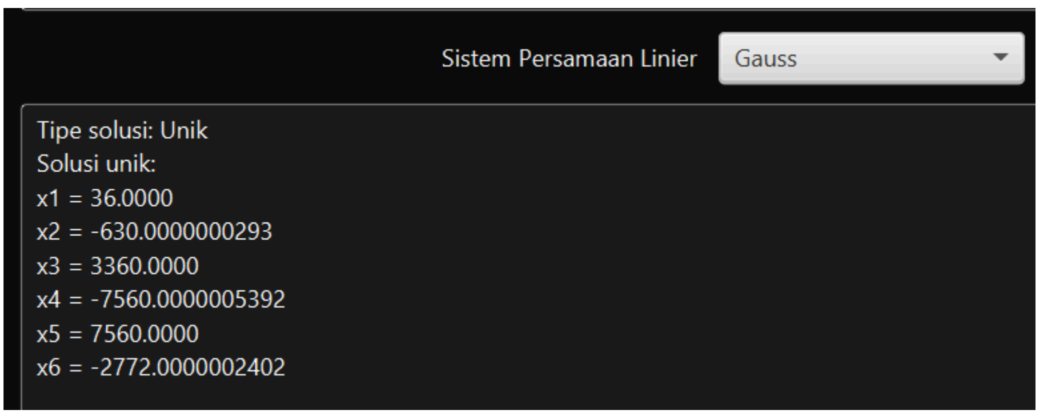

**Gambar 4.1.c.4 SPL 1c dengan metode cramer**

d. Hilbert berukuran n x n, b berukuran n x 1

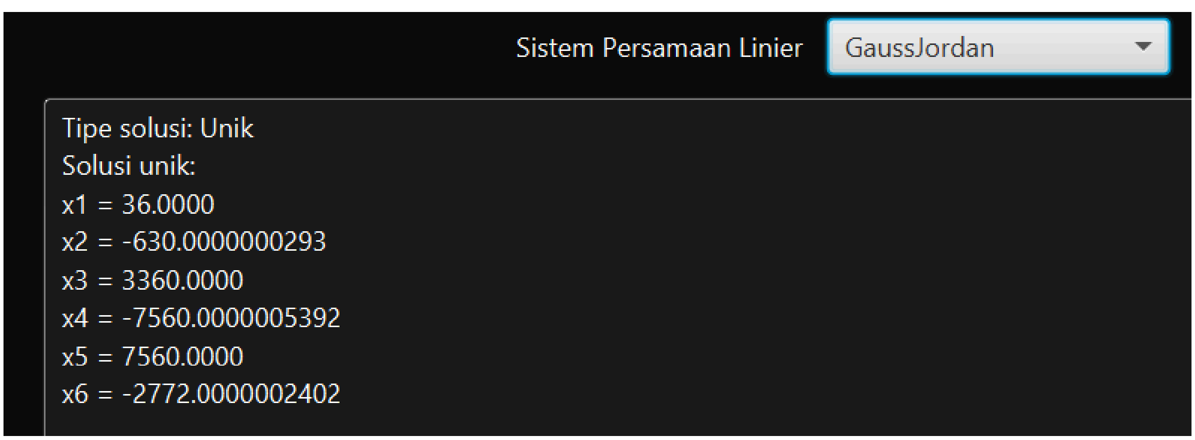
$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n+1} \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

**Gambar 4.1.d H matriks *Hilbert***

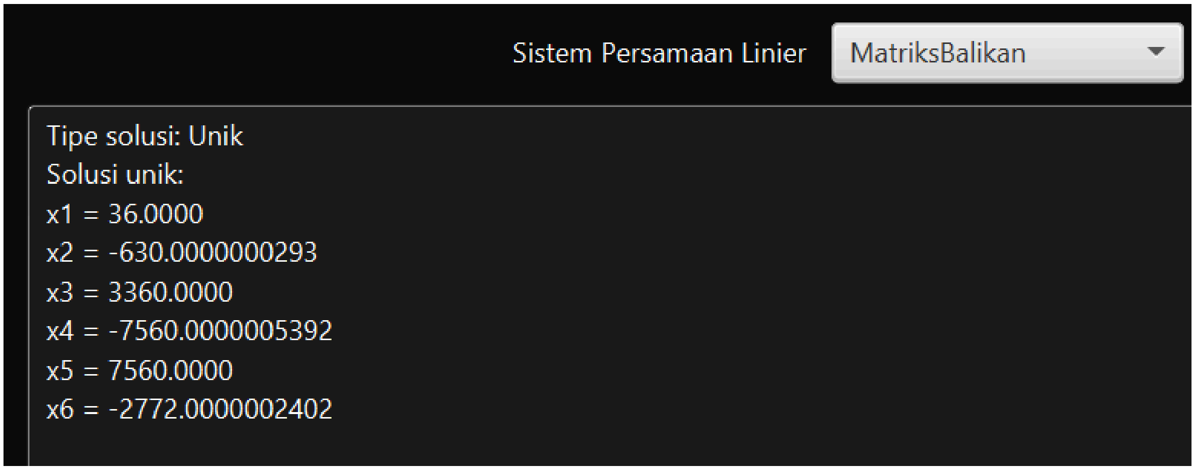
a. **Untuk n = 6:**



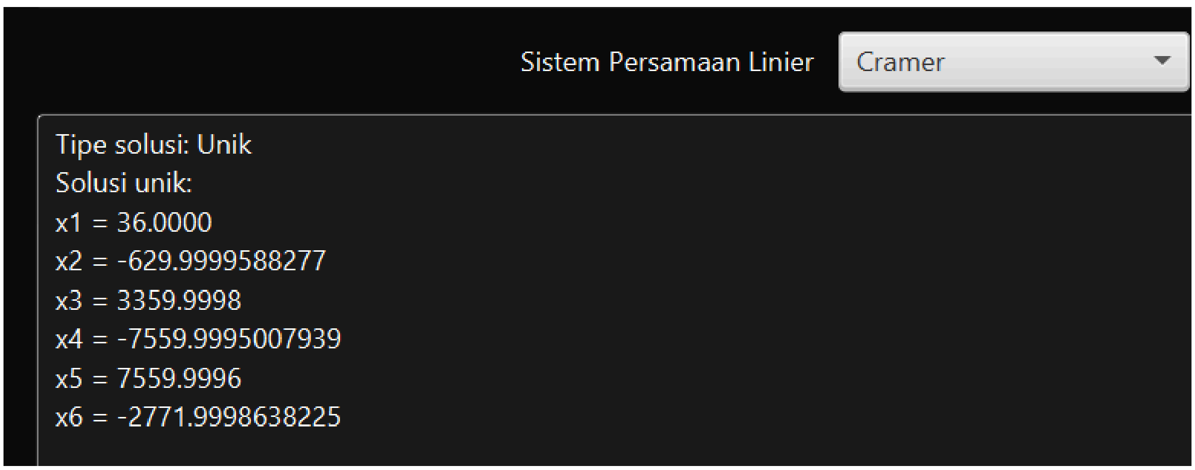

**Gambar 4.1.d.1 SPL 1d dengan metode gauss**




**Gambar 4.1.d.2 SPL 1d dengan metode gauss**




**Gambar 4.1.d.3 SPL 1d dengan metode matriks balikan**




**Gambar 4.1.d.4 SPL 1d dengan metode Cramer**

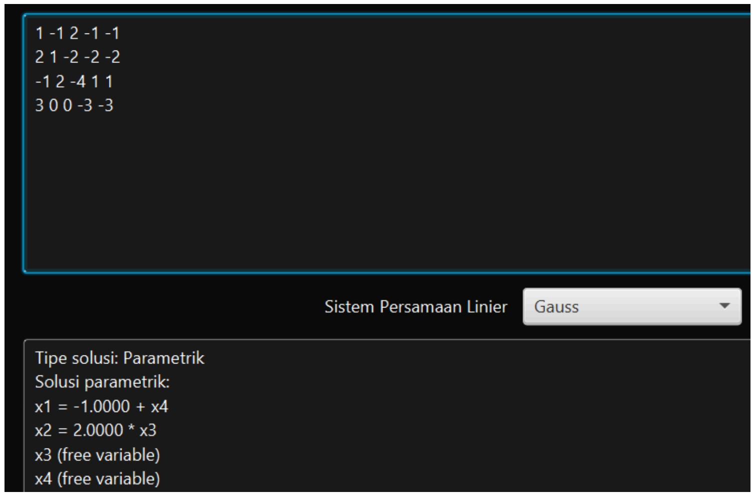
b. **Untuk n = 10:**

Masih error, hasilnya malah tidak ada solusi

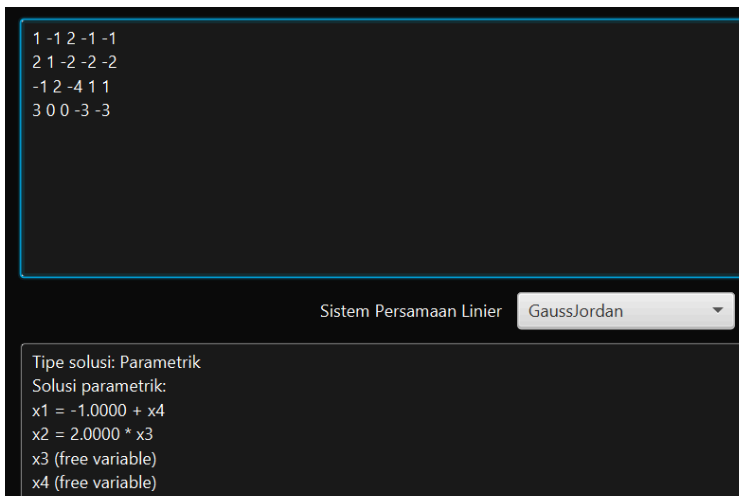
## 2. SPL berbentuk matriks *augmented*

### a. Matriks 4 x 5

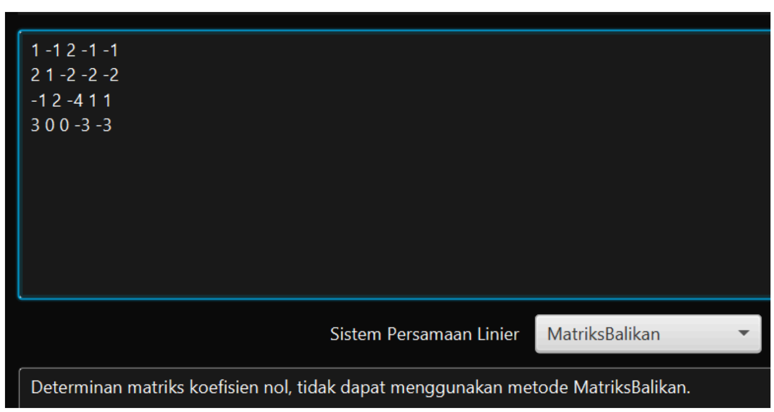
$$\begin{bmatrix} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{bmatrix}.$$



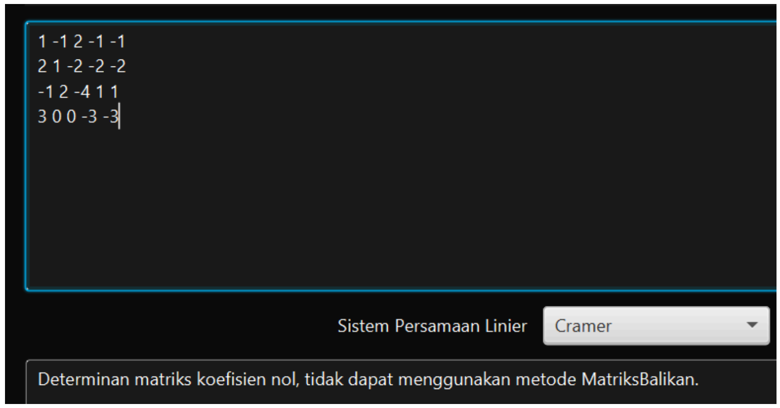

**Gambar 4.2.a.1 SPL 2a dengan metode Gauss**




**Gambar 4.2.a.2 SPL 2a dengan metode Gauss-Jordan**



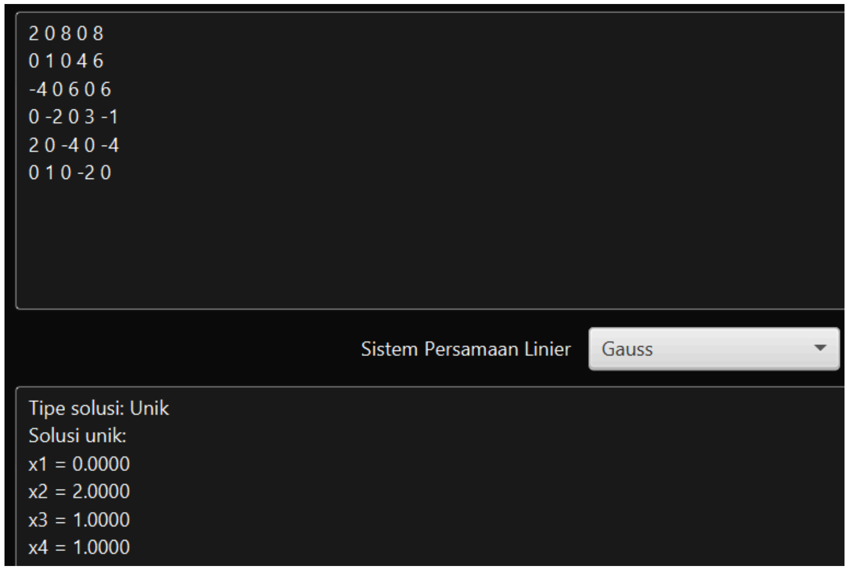

**Gambar 4.2.a.3 SPL 2a dengan metode Matriks Balikan**



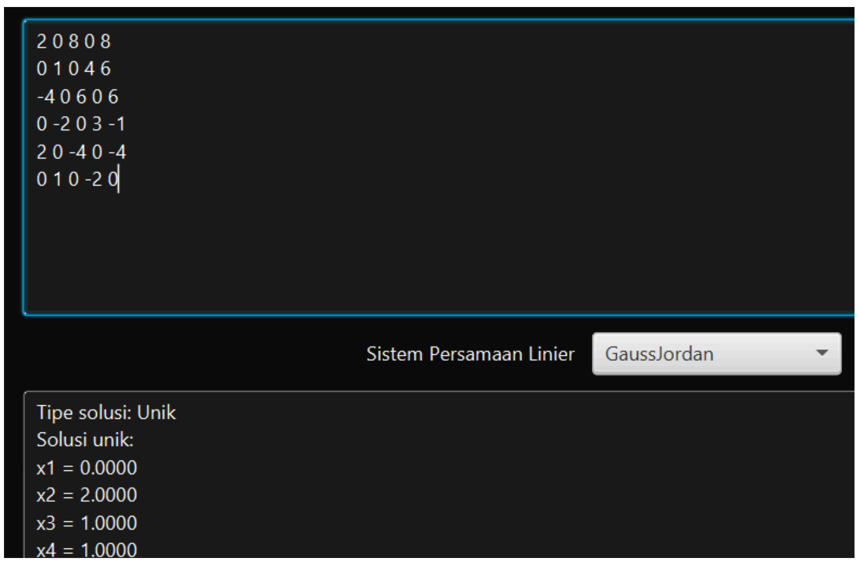

**Gambar 4.2.a.4 SPL 2a dengan metode Cramer**

## b. Matriks 6 x 5

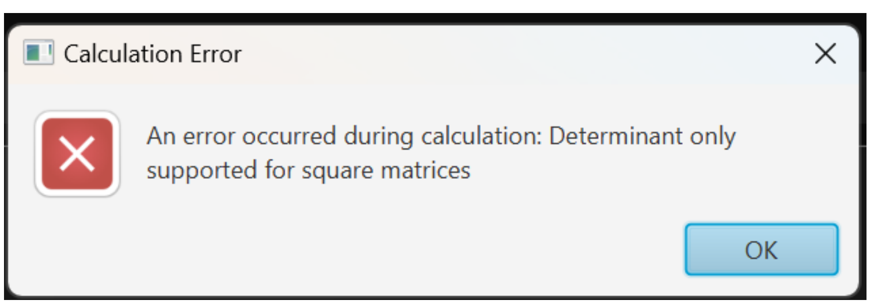
$$\begin{bmatrix} 2 & 0 & 8 & 0 & 8 \\ 0 & 1 & 0 & 4 & 6 \\ -4 & 0 & 6 & 0 & 6 \\ 0 & -2 & 0 & 3 & -1 \\ 2 & 0 & -4 & 0 & -4 \\ 0 & 1 & 0 & -2 & 0 \end{bmatrix}.$$



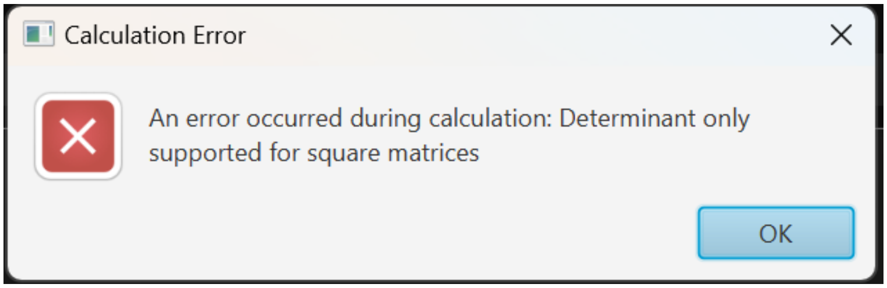

**Gambar 4.2.b.1 SPL 2b dengan metode Gauss**




**Gambar 4.2.b.2 SPL 2b dengan metode Gauss-Jordan**




**Gambar 4.2.c.2 SPL 2b dengan metode matriks Balikan**

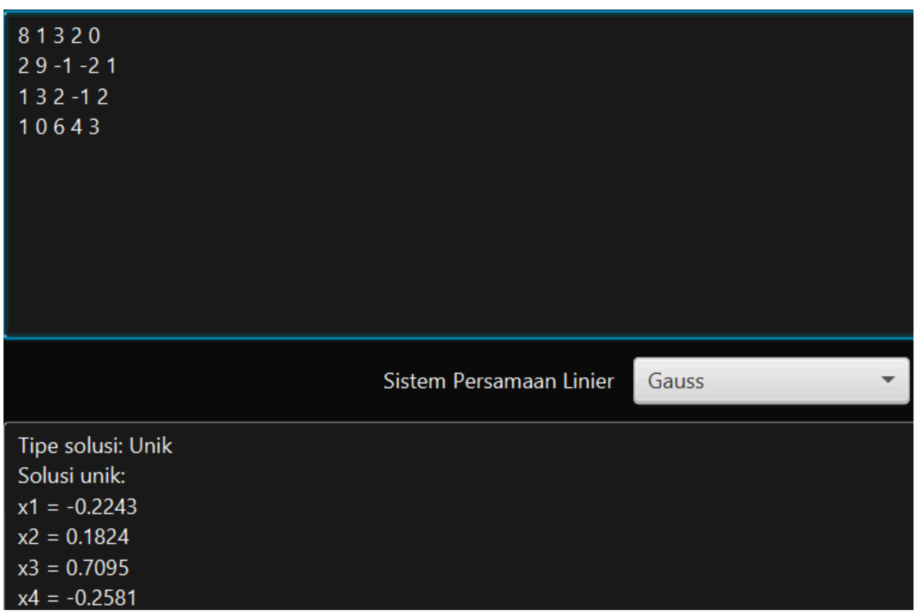



**Gambar 4.2.d.2 SPL 2b dengan metode matriks Cramer**

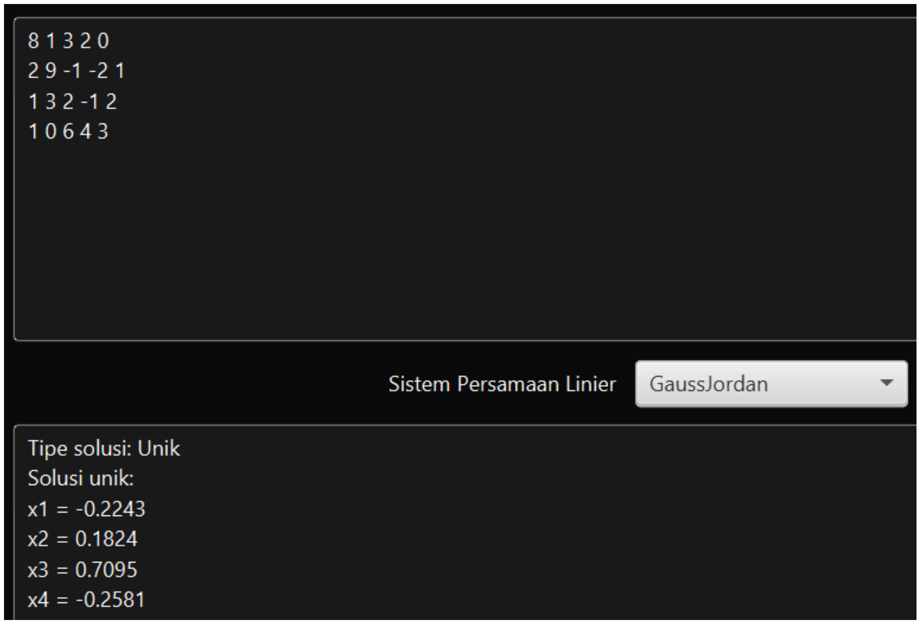
## 3. Solusi Peubah untuk SPL

### a. SPL 4 Peubah

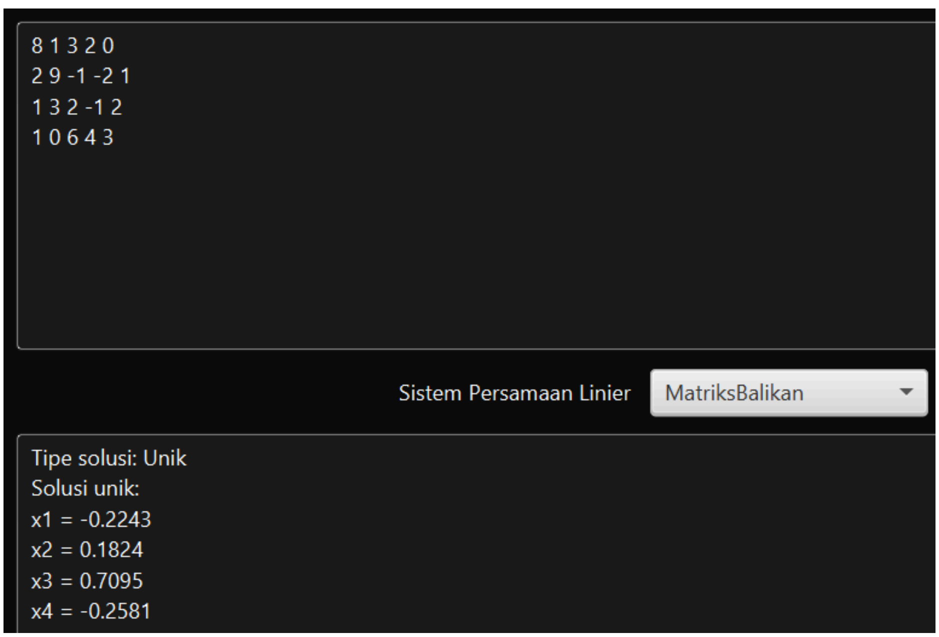
$$
\begin{aligned}
8x_1 + x_2 + 3x_3 + 2x_4 &= 0 \\
2x_1 + 9x_2 - x_3 - 2x_4 &= 1 \\
x_1 + 3x_2 + 2x_3 - x_4 &= 2 \\
x_1 + 6x_3 + 4x_4 &= 3
\end{aligned}
$$



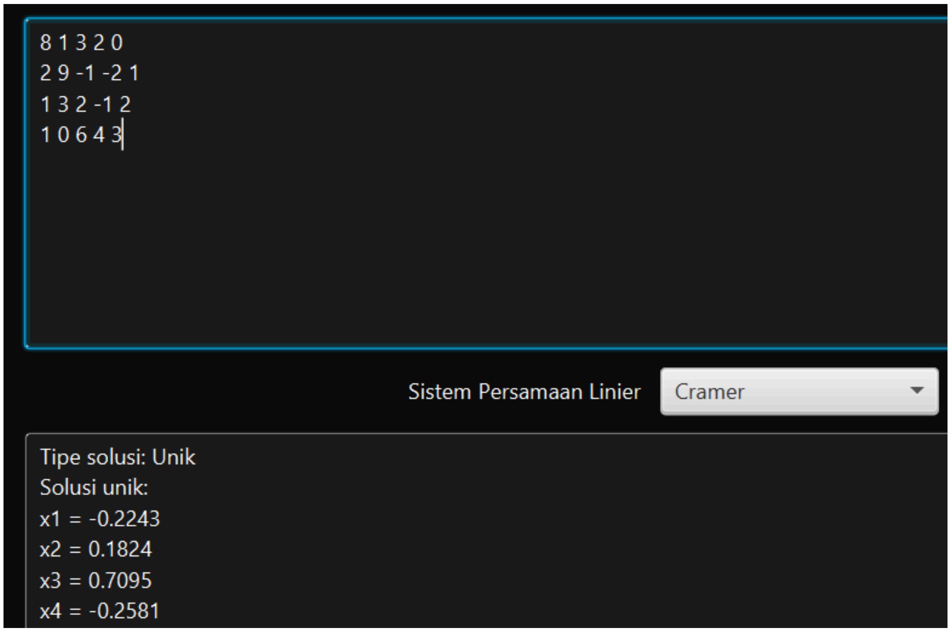

**Gambar 4.3.a.1 SPL 3a dengan metode Gauss**




**Gambar 4.3.a.2 SPL 3a dengan metode Gauss-Jordan**



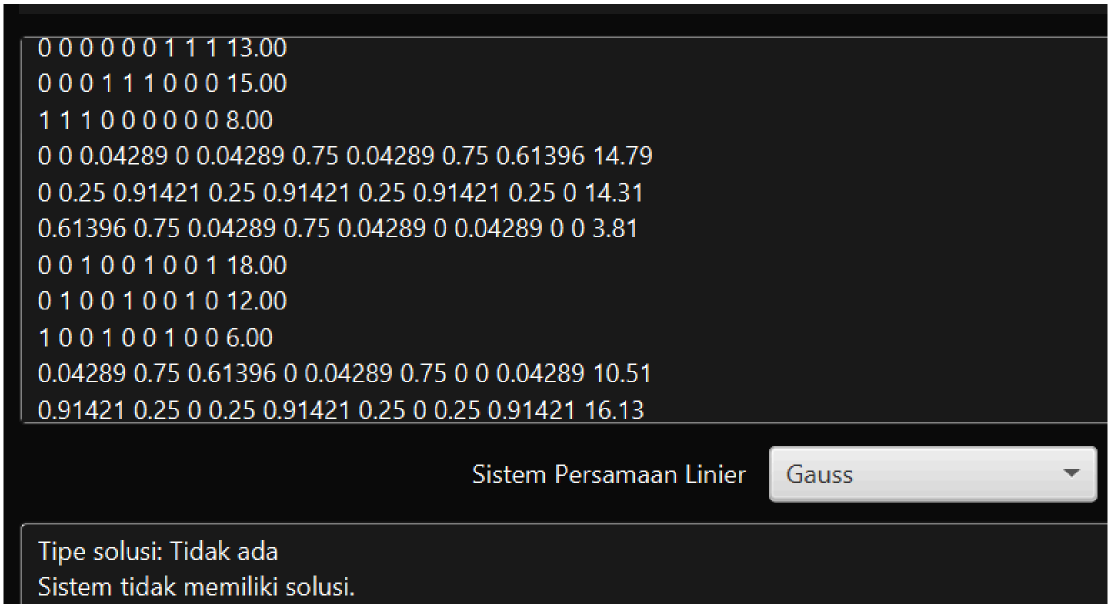

**Gambar 4.3.a.3 SPL 3a dengan metode Balikan**



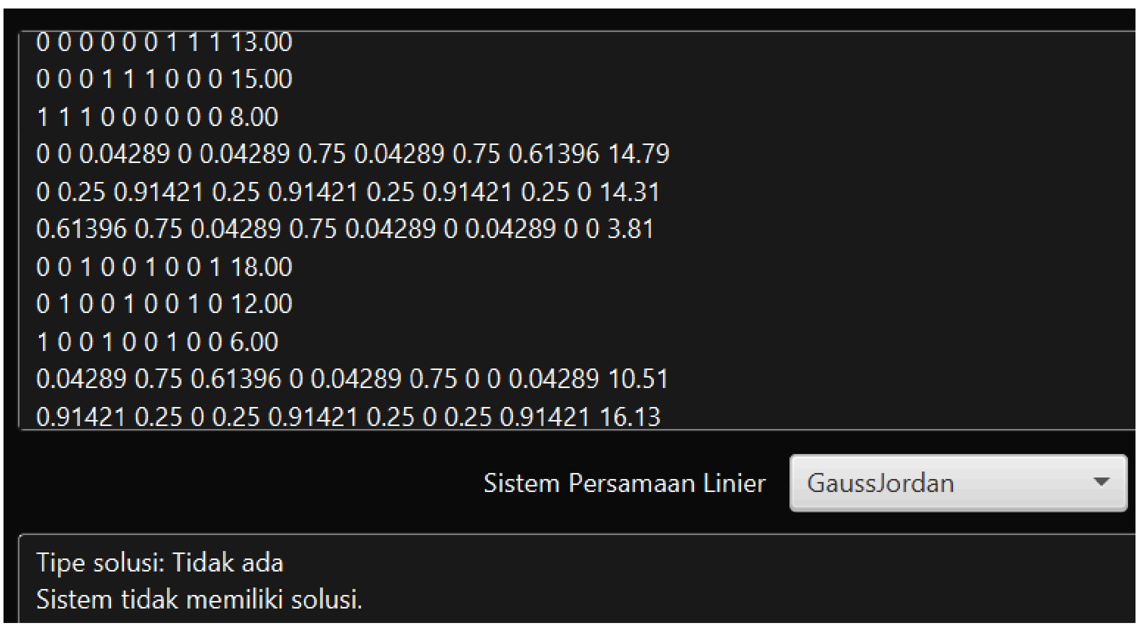

**Gambar 4.3.a.4 SPL 3a dengan metode Cramer**

## b. SPL 9 Peubah

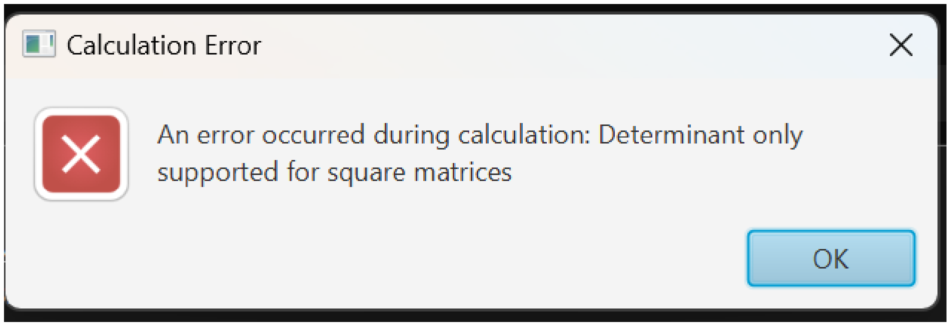
$$
\begin{aligned}
x_7 + x_8 + x_9 &= 13.00 \\
x_4 + x_5 + x_6 &= 15.00 \\
x_1 + x_2 + x_3 &= 8.00 \\
0.04289(x_3 + x_5 + x_7) + 0.75(x_6 + x_8) + 0.61396 x_9 &= 14.79 \\
0.91421(x_3 + x_5 + x_7) + 0.25(x_2 + x_4 + x_6 + x_8) &= 14.31 \\
0.04289(x_3 + x_5 + x_7) + 0.75(x_2 + x_4) + 0.61396 x_1 &= 3.81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



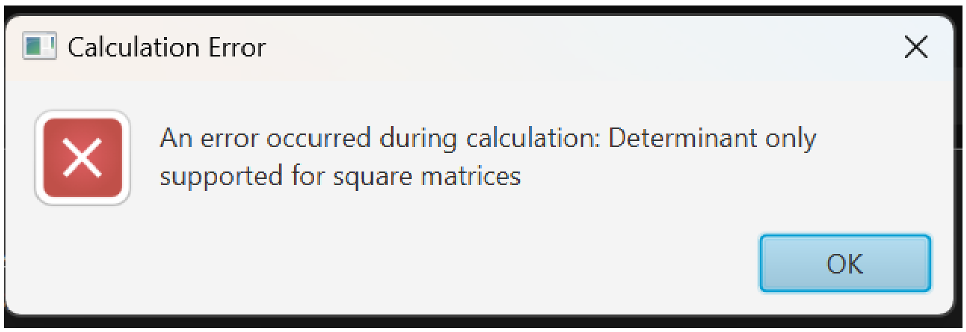

**Gambar 4.3.b.1 SPL 3b dengan metode Gauss**




**Gambar 4.3.b.2 SPL 3b dengan metode Gauss-Jordan**



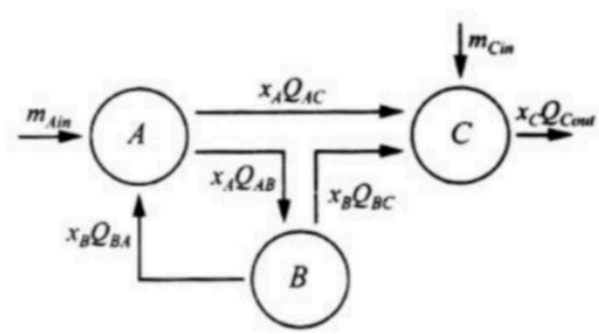

**Gambar 4.3.b.3 SPL 3b dengan metode matriks Balikan**




**Gambar 4.3.b.4 SPL 3b dengan metode matriks Cramer**

## 4. Sistem reaktor

Lihatlah sistem reaktor pada gambar berikut.



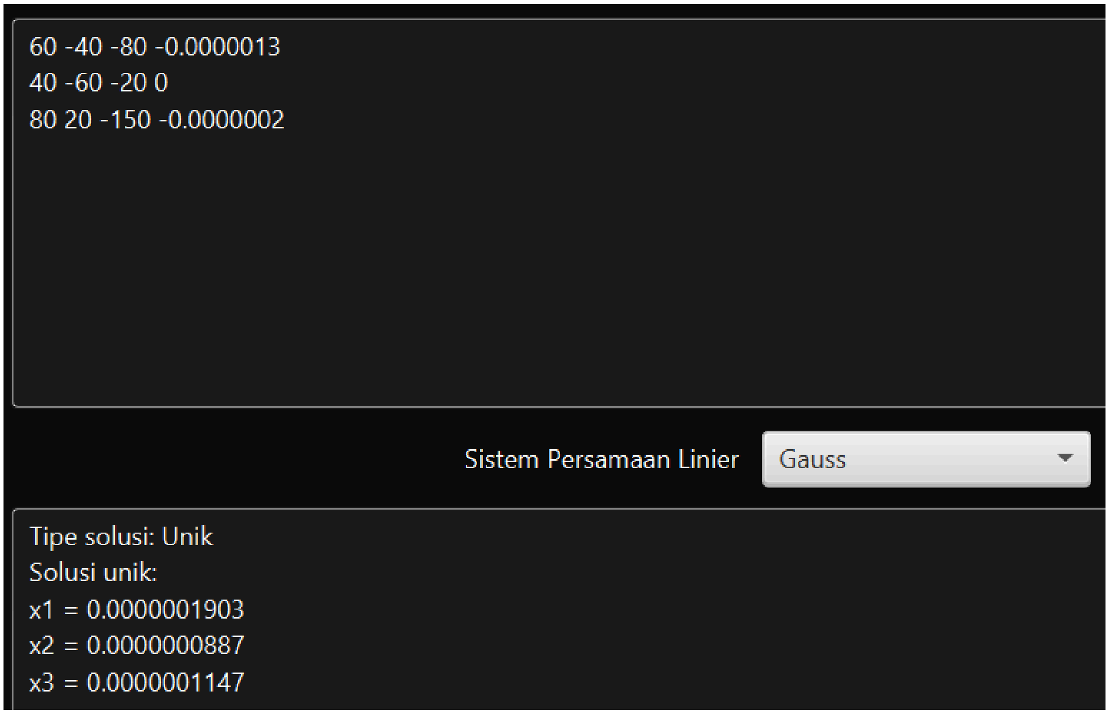

Dengan laju volume *Q* dalam $m^3/s$ dan input massa min dalam $mg/s$. Konservasi massa pada tiap inti reaktor adalah sebagai berikut:

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

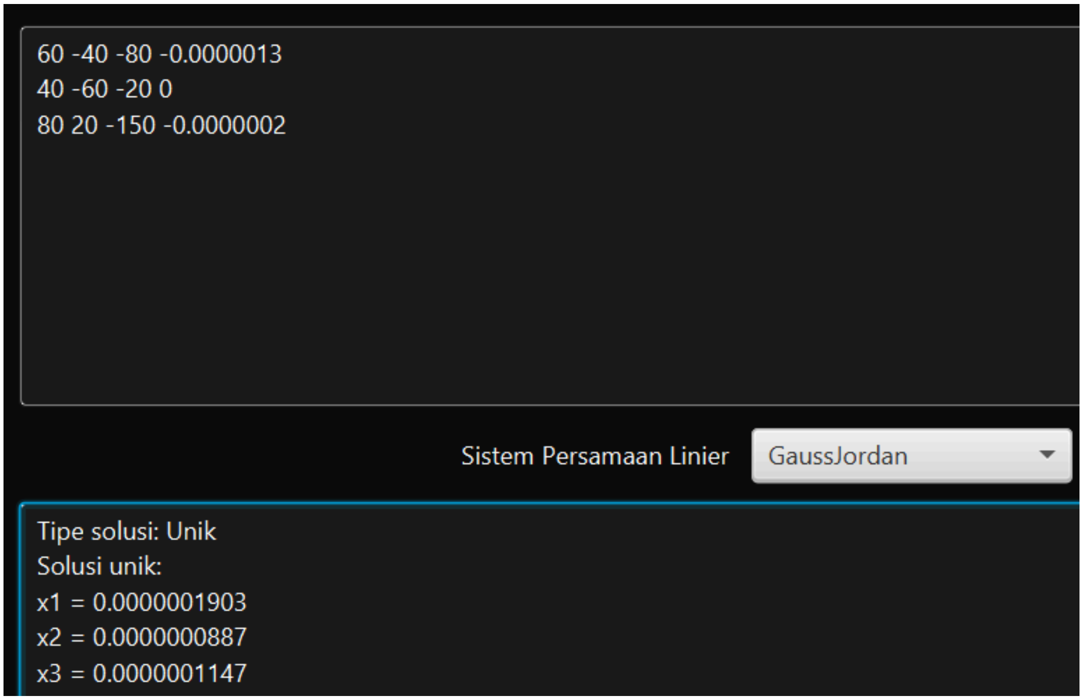
Tentukan solusi $x_A$, $x_B$, $x_C$ dengan menggunakan parameter berikut : $Q_{AB}$ = 40, $Q_{AC}$ = 80, $Q_{BA}$ = 60, $Q_{BC}$ = 20 dan $Q_{Cout}$ = 150 $m^3$/s dan $m_{Ain}$ = 1300 dan $m_{Cin}$ = 200 *mg/s*.

m_Ain = 1300 mg/s = 13e-7 m^3/s

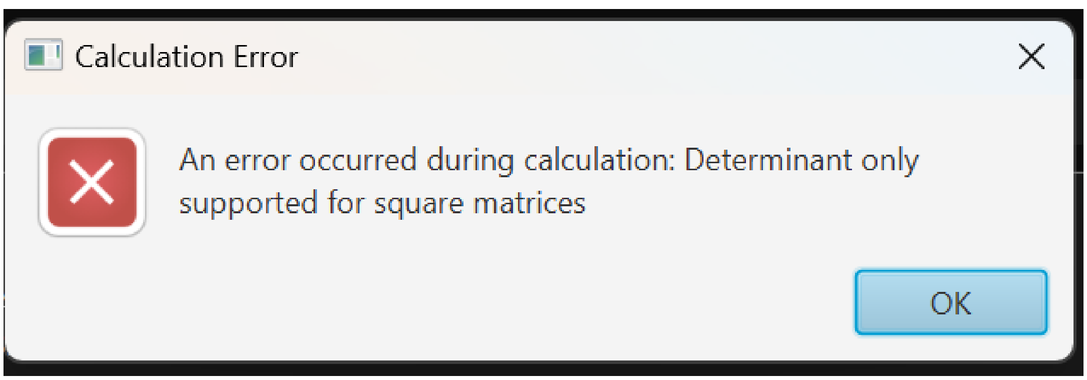
m_Cin = 200 mg/s= 2e-7 m^3/s



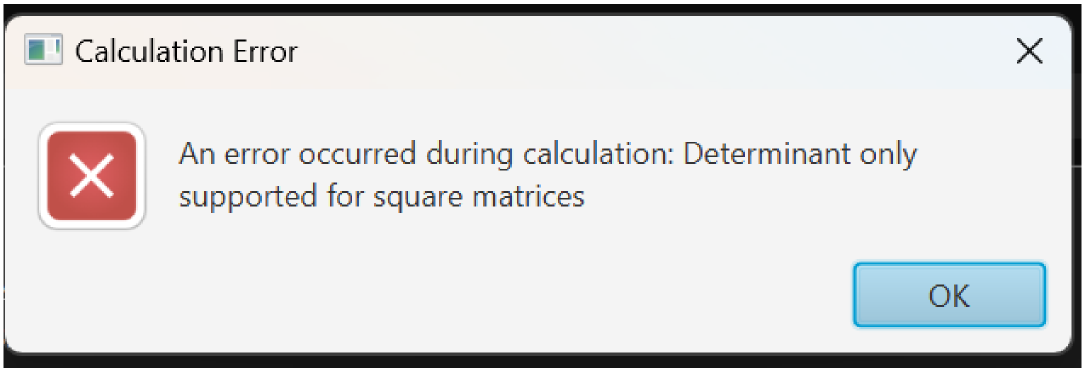

**Gambar 4.4.1 SPL Sistem Reaktor dengan metode Gauss**




**Gambar 4.4.2 SPL Sistem Reaktor dengan metode Gauss-Jordan**




**Gambar 4.4.3 SPL Sistem Reaktor dengan metode matriks Balikan**




**Gambar 4.4.4 SPL Sistem Reaktor dengan metode matriks Cramer**

## 5. Studi Kasus Interpolasi

### a. Aplikasi Teori Interpolasi

Gunakan tabel di bawah ini untuk mencari polinom interpolasi dari pasangan titik-titik yang terdapat dalam tabel. Program menerima masukan nilai $x$ yang akan dicari nilai fungsi $f(x)$.

| $x$ | 0.1 | 0.3 | 0.5 | 0.7 | 0.9 | 1.1 | 1.3 |
|---|---|---|---|---|---|---|---|
| $f(x)$ | 0.003 | 0.067 | 0.148 | 0.248 | 0.370 | 0.518 | 0.697 |

Lakukan pengujian pada nilai-nilai berikut:

$x$ = 0.2 $f(x)$ = ?
$x$ = 0.55 $f(x)$ = ?
$x$ = 0.85 $f(x)$ = ?
$x$ = 1.28 $f(x)$ = ?

```
Interpolation polynomial:
f(x) = 0.0000000000000591041 * x^6
       + 0.00000000000002550134 * x^5 + 0.0260 * x^4
       + 0.00000000000003440997 * x^3 + 0.1974 * x^2
       + 0.2400 * x - 0.02297656250000013800
```

**Gambar 4.5.a.1 persamaan polinomial soal a**

```
f(0.20) = 0.0330
```

**Gambar 4.5.a.2 nilai kodomain x = 0.2 soal a**

```
f(0.55) = 0.1711
```

**Gambar 4.5.a.3 nilai kodomain x = 0.55 soal a**

```
f(0.85) = 0.3372
```

**Gambar 4.5.a.4 nilai kodomain x = 0.85 soal a**

```
f(1.28) = 0.6775
```

**Gambar 4.5.a.5 nilai kodomain x = 1.28 soal a**

**b. Studi Kasus Aplikasi Interpolasi**

Jumlah kasus positif baru Covid-19 di Indonesia semakin fluktuatif dari hari ke hari. Di bawah ini diperlihatkan jumlah kasus baru Covid-19 di Indonesia mulai dari tanggal 17 Juni 2022 hingga 31 Agustus 2022:

| Tanggal | Tanggal (desimal) | Jumlah Kasus Baru |
|---|---|---|
| 17/06/2022 | 6,567 | 12.624 |
| 30/06/2022 | 7 | 21.807 |
| 08/07/2022 | 7,258 | 38.391 |
| 14/07/2022 | 7,451 | 54.517 |

| | | |
|---|---|---|
| 17/07/2022 | 7,548 | 51.952 |
| 26/07/2022 | 7,839 | 28.228 |
| 05/08/2022 | 8,161 | 35.764 |
| 15/08/2022 | 8,484 | 20.813 |
| 22/08/2022 | 8,709 | 12.408 |
| 31/08/2022 | 9 | 10.534 |

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

| |
|---|
| Tanggal (desimal) = bulan + (tanggal / jumlah hari pada bulan tersebut) |

Sebagai contoh, untuk tanggal 17/06/2022 (dibaca: 17 Juni 2022) diperoleh tanggal(desimal) sebagai berikut:

Tanggal (desimal) = 6 + (17/30) = 6,567

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

```
Interpolation polynomial:
f(x) = 140993.7122486359400000000 * x^9
       + 9372849.2391 * x^8 - 275474539.420669300000000000 * x^7
       + 4695806315.4288 * x^6 - 51131876760.132750000000000000 * x^5
       + 368550807175.5339 * x^4 - 1756810186361.380900000000000000 * x^3
       + 5334203055240.1950 * x^2 - 9346993079172.328000000000000000 * x
       + 7187066071657.8670
```

**Gambar 4.5.b.1 persamaan polinomial soal b**

a. **16/07/2022**

```
f(7.516) = 53537.9961
```

**Gambar 4.5.b.2 Hasil Interpolasi 16/07/2022**

b. **10/08/2022**

```
f(8.323) = 36294.8477
```

**Gambar 4.5.b.3 Hasil Interpolasi 10/08/2022**

c. **05/09/2022**

```
f(9.167) = -667693.2188
```

**Gambar 4.5.b.4 Hasil Interpolasi 05/09/2022**

d. **Tanggal (desimal) yang sudah diolah**

asumsi prediksi selalu dilakukan untuk tahun 2022. (Masukan = 18/6/2022 atau 6.600 (dalam decimal)

```
f(6.600) = 62619.8809
```

**Gambar 4.5.b.5 Hasil Interpolasi 6600 atau 18/6/2022**

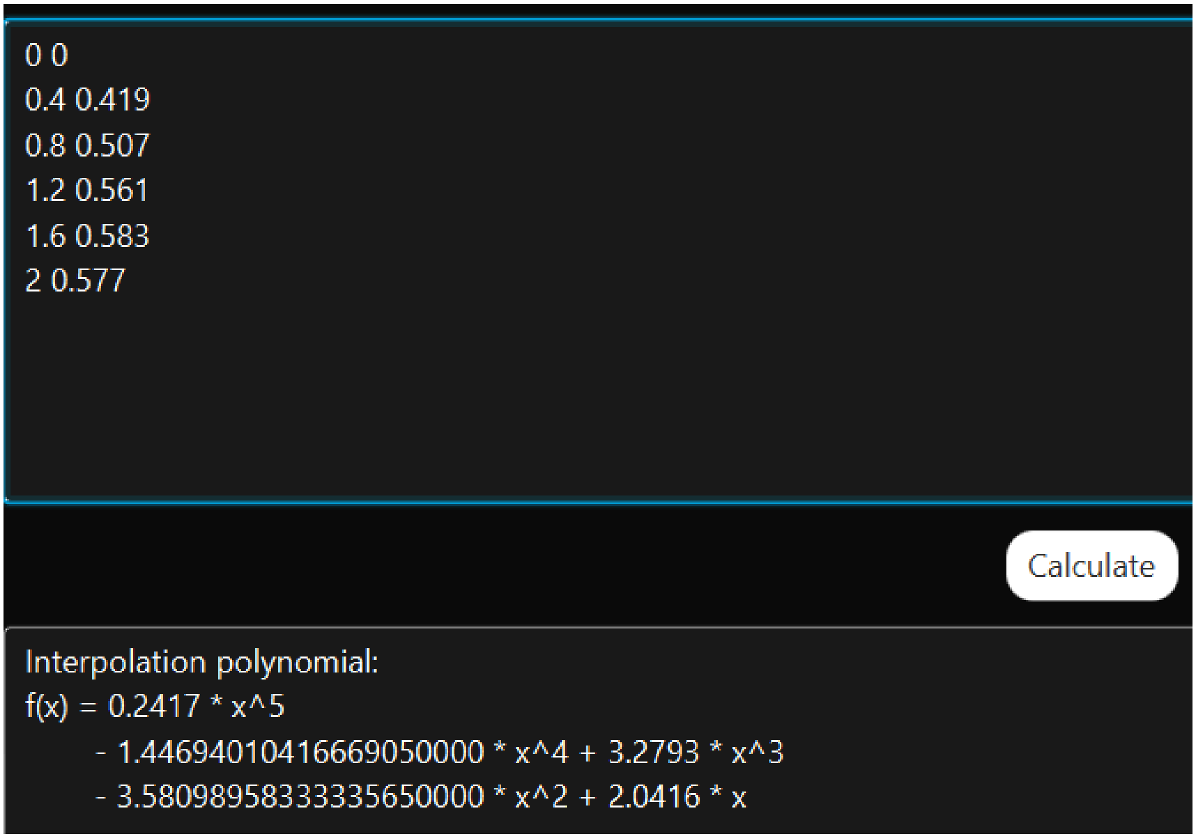
## c. Menyederhanakan fungsi f(x)

Sederhanakan fungsi $f(x)$ yang memenuhi kondisi

$$f(x) = \frac{x^2 + \sqrt{x}}{e^x + x}$$

dengan polinom interpolasi derajat $n$ di dalam selang [0, 2].

Sebagai contoh, jika $n$ = 5, maka titik-titik $x$ yang diambil di dalam selang [0, 2] berjarak $h$ = (2 – 0)/5 = 0.4.




**Gambar 4.5.c Hasil penyederhanaan fungsi f(x)**

## 6. Studi Kasus Regresi Linear dan Kuadratik Berganda

Diberikan sekumpulan data sesuai pada tabel berikut ini.

Table 12.1: Data for Example 12.1

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

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

Gunakan *Normal Estimation Equation for Multiple Linear Regression* untuk mendapatkan regresi linear berganda dari data pada tabel di atas, kemudian estimasi nilai Nitrous Oxide apabila Humidity bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30.

Dari data-data tersebut, apabila diterapkan *Normal Estimation Equation for Multiple Linear Regression*, maka diperoleh sistem persamaan linear sebagai berikut.

$$20b_0 + 863.1b_1 + 1530.4b_2 + 587.84b_3 = 19.42$$

$$863.1b_0 + 54876.89b_1 + 67000.09b_2 + 25283.395b_3 = 779.477$$

$$1530.4b_0 + 67000.09b_1 + 117912.32b_2 + 44976.867b_3 = 1483.437$$

$$587.84b_0 + 25283.395b_1 + 44976.867b_2 + 17278.5086b_3 = 571.1219$$

Silahkan terapkan model-model ini pada *Multiple Quadratic Equation* juga dan bandingkan hasilnya. Sistem persamaan linear tidak akan diberikan untuk kasus ini.

### Hasil Multiple Linear Regression


**Gambar 4.6.a Hasil model dan prediksi dengan Multiple Linear Regression**

### Hasil Multiple Quadratic Regression


**Gambar 4.6.b Hasil model dan prediksi dengan Multiple Quadratic Regression**

## 7. Studi Kasus Interpolasi *Bicubic Spline*

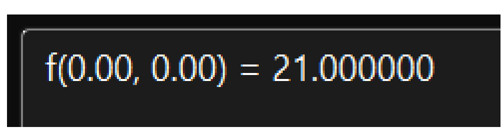
Diberikan matriks input dengan bentuk sebagai berikut. Format matriks masukan bukan mewakili nilai matriks, tetapi mengikuti format masukan pada bagian "Spesifikasi Tugas" nomor 7.

$$\begin{pmatrix} 21 & 98 & 125 & 153 \\ 51 & 101 & 161 & 59 \\ 0 & 42 & 72 & 210 \\ 16 & 12 & 81 & 96 \end{pmatrix}$$

**Gambar 4.7 Input Matriks Bicubic Interpolation**

Tentukan nilai:

a. ***f*(0, 0)**




**Gambar 4.7.a Hasil Prediksi Kasus Pertama**

b. *f*(0.5, 0.5)

```
f(0.50, 0.50) = 87.796875
```

**Gambar 4.7.b Hasil Prediksi Kasus Kedua**

c. *f*(0.25, 0.75)

```
f(0.25, 0.75) = 117.732178
```

**Gambar 4.7.c Hasil Prediksi Kasus Ketiga**

d. *f*(0.1, 0.9)

```
f(0.10, 0.90) = 128.575187
```

**Gambar 4.7.d Hasil Prediksi Kasus Keempat**

# BAB V : KESIMPULAN

Tugas besar ini berkesan kepada seluruh anggota kelompok, melalui eksplorasi dan pemanfaatan tools baru serta mendapatkan pengalaman baru dalam mengerjakan projek secara berkelompok, baik itu bekerja secara remote ataupun secara luring bersama. Projek ini bisa menjadikan pedoman dalam rancangan bagaimana sebuah proyek secara umum dikerjakan oleh banyak orang, bisa mencapai hingga 100 orang lebih. Proyek ini juga memaksakan kita untuk eksplorasi pada teknologi yang lebih kompleks, seperti regresi kuadratik, image scaling, dan banyak hal lainnya. Secara keseluruhan, tugas ini menarik untuk membuka rasa ingin tahu serta mengembangkan kemampuan riset dalam mencari jawaban untuk permasalahan kita.

Proses pengembangan proyek aplikasi ini mungkin dapat dilakukan menggunakan tools seperti gantt chart ataupun tools timeline lainnya sehingga proyek dapat dilaksanakan dan terselesaikan tepat waktu sesuai dengan hasil yang diharapkan. Selain itu, bisa dilakukan pembagian tugas lebih jelas pada awalnya untuk proyek dan juga penentuan kesepakatan dalam hal tipe data, semantics, dan lainnya.

~~Karena waktu deadline sudah dekat kami cukupkan disini 🙂~~

# Lampiran

Referensi:

https://docs.google.com/document/d/1_VVhIat1qDTyzMbPL2if3Rct3LGREfEYWxBFMCOeeUM/edit?tab=t.0

https://informatika.stei.itb.ac.id/~rinaldi.munir/

https://math.stackexchange.com/questions/3155866/multivariate-quadratic-regression

https://www.mssc.mu.edu/~daniel/pubs/RoweTalkMSCS_BiCubic.pdf

https://gist.github.com/joshbuchea/6f47e86d2510bce28f8e7f42ae84c716#file-semantic-commit-messages-md

https://github.com/jehna/readme-best-practices/blob/master/README-default.md

https://www.programcreek.com/2011/07/build-a-java-library-for-yourself/

https://developer.ibm.com/tutorials/j-javalibrary/

Github:

https://github.com/Farhanabd05/Algeo01-23006

Video:

https://drive.google.com/drive/folders/1KtBKULSiNxh9LJujLpRb61Xz-2B9Emwg?usp=drive_link

Komentar informal dari tim:
“AKHIRNYA KELAR DAN BAGUS YESSSSSSSSSSS” - Jovi
“easy” - William
“Gk usah lah” - Farhan