Laporan Tugas Besar 1

IF2313 Aljabar Linier dan Geometri

Sistem Persamaan Linier, Determinan, dan Penerapannya

SEMESTER 1 TAHUN 2025/2026

v1. 1

DEADLINE : JUMAT, 2025/10/10 23:59 WIB

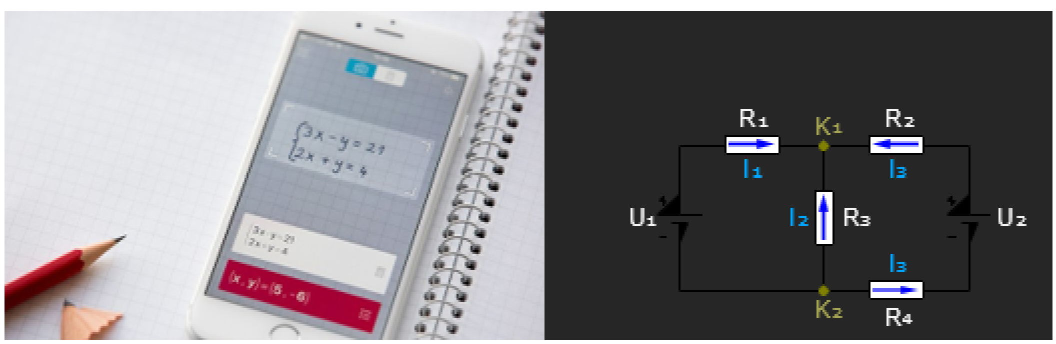
LABORATORIUM ILMU DAN REKAYASA KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

1 Pendahuluan

Aplikasi seperti Photomath telah menjadi solusi populer dalam membantu menyelesaikan berbagai permasalahan matematika, termasuk soal matriks dan sistem persamaan linear yang kompleks. Kemampuan aplikasi tersebut dalam memberikan langkah-langkah penyelesaian secara otomatis tidak terlepas dari penerapan algoritma dan konsep matematika yang kuat, khususnya aljabar linier.

Aljabar linier merupakan salah satu cabang matematika yang sangat penting dalam pengembangan ilmu komputer, teknik, fisika, ekonomi, dan berbagai bidang lainnya. Dalam kehidupan sehari-hari, konsep matriks, dan sistem persamaan linier sering kali diaplikasikan untuk menyelesaikan berbagai persoalan nyata, mulai dari pemodelan data, analisis statistik, hingga simulasi sistem dinamis.

Pada tugas besar ini, Anda akan mengimplementasikan berbagai metode penyelesaian sistem persamaan linier, perhitungan determinan, pencarian invers matriks, interpolasi polinomial, interpolasi kurva Bezier kubik, dan regresi polinomial menggunakan bahasa pemrograman Java dalam bentuk pustaka (*library*) yang dapat digunakan secara modular dan terdokumentasi dengan baik.



Gambar 1: Aplikasi Photomath (kiri) dan diagram rangkaian listrik (kanan)

Penyelesaian sistem persamaan linier, misalnya, sangat krusial dalam pemodelan sirkuit listrik, analisis struktur bangunan, hingga pemrosesan citra digital. Sementara itu, interpolasi dan regresi polinomial digunakan untuk memperkirakan nilai di antara data yang tersedia atau memprediksi tren data di masa depan. Melalui tugas besar ini, diharapkan Anda tidak hanya memahami teori di balik metode-metode tersebut, tetapi juga mampu mengimplementasikannya secara nyata yang dapat digunakan untuk menyelesaikan berbagai kasus uji (*test cases*) yang telah disediakan.

Secara formal, tujuan dari tugas besar ini adalah:

1. Mengimplementasikan berbagai metode penyelesaian sistem persamaan linier, perhitungan determinan, invers matriks, interpolasi, dan regresi polinomial secara mandiri dalam bahasa Java.
2. Membuat pustaka (*library*) yang dapat digunakan secara modular dan terdokumentasi dengan baik. Mengintegrasikan pustaka yang dibuat ke dalam sebuah program yang dapat menerima masukan dari pengguna dan menampilkan hasilnya dengan format yang jelas.
3. Menguji pustaka yang dibuat pada berbagai kasus uji dan menganalisis hasilnya.

Dengan demikian, tugas besar ini diharapkan dapat menjadi sarana pembelajaran yang efektif dalam memahami dan mengaplikasikan konsep-konsep aljabar linier secara komputasional, serta membekali Anda dengan keterampilan praktis yang sangat dibutuhkan di dunia profesional.

2 Dasar Teori

2.1 Sistem Persamaan Linear

Sistem Persamaan Linear (SPL) adalah himpunan berhingga dari persamaan-persamaan linear dengan variabel-variabel yang sama. Bentuk umum dari SPL dengan *m* persamaan dan *n* variabel adalah

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n & = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n & = b_2 \\ \ldots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots + a_{nn}x_n & = b_n \end{cases}$$

atau, dalam bentuk perkalian matriks (kiri) dan matriks *augmented* (kanan),

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix}}_{A} \cdot \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}_{b} \qquad \longrightarrow \qquad \left[\begin{array}{cccc|c} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ a_{21} & a_{22} & \ldots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} & b_n \end{array}\right].$$

Solusi dari SPL dapat berupa solusi tunggal, banyak solusi, atau tidak ada solusi. Metode penyelesaian yang umum digunakan antara lain eliminasi Gauss, eliminasi Gauss-Jordan, kaidah Cramer, dan metode matriks balikan.

2.2 Determinan Matriks

Determinan adalah sebuah nilai skalar khusus yang dapat dihitung dari elemen-elemen sebuah matriks persegi. Determinan dari matriks A dinotasikan sebagai $\det(A)$ atau $|A|$. Nilai determinan sangat krusial karena dapat mengindikasikan apakah matriks tersebut memiliki invers. Beberapa metode untuk menghitung determinan matriks mencakup metode ekspansi kofaktor dan metode reduksi baris.

2.3 Invers Matriks

Invers dari sebuah matriks persegi A adalah matriks A^{-1} yang memenuhi properti $AA^{-1} = A^{-1}A = I$, di mana I adalah matriks identitas. Sebuah matriks hanya memiliki invers jika dan hanya jika determinannya tidak nol ($\det(A) \neq 0$). Matriks yang memiliki invers disebut matriks *invertible* atau non-singular. Perhitungan invers dari suatu matriks dapat dilakukan dengan metode matriks adjoin maupun metode eliminasi Gauss-Jordan.

2.4 Interpolasi Polinomial

Interpolasi polinomial adalah metode untuk menentukan suatu polinomial $P_n(x)$ berderajat n yang melalui $n + 1$ titik data $(x_0,y_0),(x_1,y_1),\ldots,(x_n,y_n)$, x_i diurutkan dengan x_0 terkecil dan x_n terbesar. Secara umum, polinomial tersebut dapat dituliskan sebagai:

$$P_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$$

dengan koefisien $a_0,a_1,\ldots,a_n$ yang ditentukan sedemikian rupa sehingga $P_n(x_{\text{perkiraan}}) = y_{\text{perkiraan}}$ untuk semua nilai $x_0 < x_{\text{perkiraan}} < x_n$. Derajat persamaan polinom ditentukan oleh jumlah titik data yang diberikan dikurangi satu.

Koefisien polinomial dapat dicari dengan menyulihkan titik-titik data ke dalam persamaan polinomial dan menyelesaikan sistem persamaan linear yang dihasilkan dengan metode eliminasi Gauss yang dijelaskan sebelumnya. Kita dapat menyusun sistem persamaan linear sebagai berikut.

$$\begin{cases} a_0 + a_1x_0 + a_2x_0^2 + \cdots + a_nx_0^n & = y_0 \\ a_0 + a_1x_1 + a_2x_1^2 + \cdots + a_nx_1^n & = y_1 \\ \vdots & \\ a_0 + a_1x_n + a_2x_n^2 + \cdots + a_nx_n^n & = y_n \end{cases}$$

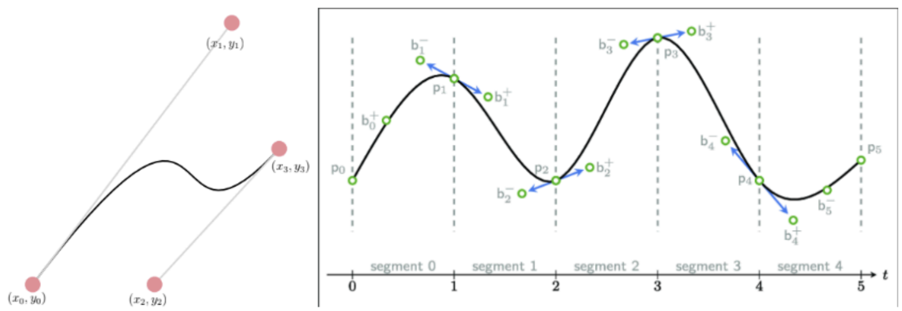
Persamaan-persamaan di atas dapat dipetakan menjadi sebuah permasalahan sistem persamaan linear, sehingga dapat diselesaikan dengan menggunakan metode eliminasi Gauss untuk menyelesaikan

2.5 Interpolasi Splina B'ezier Kubik

Ketika menggunakan *pen tool* di Adobe Photoshop/Illustrator, kita akan berhadapan dengan kurva Bezier, atau lebih spesifiknya kurva Bezier kubik. Kurva ini sangat berguna untuk menggambar bentuk-bentuk halus dan kompleks dengan kontrol yang presisi.

Kurva Bezier kubik adalah kurva parametrik yang didefinisikan oleh empat titik kontrol yang mempengaruhi bentuk kurva tersebut. Titik kontrol pertama (P_0) dan terakhir (P_3) adalah titik awal dan akhir kurva, sedangkan titik-titik kontrol lainnya (P_1 dan P_2) menentukan arah dan kelengkungan kurva. Secara matematis, kurva Bezier kubik dapat ditulis sebagai persamaan parametrik

$$B(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t) P_2 + t^3 P_3, \qquad 0 \le t \le 1.$$



Gambar 3: Kurva Bezier kubik dengan empat titik kontrol (kiri) dan kurva komposit yang tersusun atas banyak splina Bezier kubik (kanan, sumber).

Misalkan kita diberikan titik-titik interpolasi $S_0, S_1, ..., S_n$ dan kita perlu menemukan titik-titik kontrol $B_0, B_1, ..., B_n$ yang sesuai untuk membangun kurva halus[1] yang melewati semua titik S_i. Kita dapat membangun kurvanya dari **splina Bezier kubik**, yakni himpunan kurva Bezier kubik.

Masalah ini dapat direduksi menjadi penyelesaian sistem persamaan linear[2] yang didapatkan dari hubungan antara titik-titik S_i dan titik-titik kontrol B_i (silakan baca referensi). Sistem persamaan tersebut adalah

$$\begin{cases} 4\mathbf{b_1} + \mathbf{b_2} & = 6\mathbf{s_1} - \mathbf{s_0} \\ \mathbf{b_1} + 4\mathbf{b_2} + \mathbf{b_3} & = 6\mathbf{s_2} \\ \mathbf{b_2} + 4\mathbf{b_3} + \mathbf{b_4} & = 6\mathbf{s_3} \\ \dots & \\ \mathbf{b_{n-2}} + 4\mathbf{b_{n-1}} & = 6\mathbf{s_{n-1}} - \mathbf{s_n} \end{cases},$$

dengan $\mathbf{b_k}$ dan $\mathbf{s_k}$ adalah vektor posisi dari titik-titik kontrol B_k dan titik-titik interpolasi S_k, atau dapat dinyatakan dalam bentuk perkalian matriks,

$$\begin{bmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & 1 & 4 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & 4 \end{bmatrix} \begin{bmatrix} \mathbf{b_1} \\ \mathbf{b_2} \\ \mathbf{b_3} \\ \vdots \\ \mathbf{b_{n-1}} \end{bmatrix} = \begin{bmatrix} 6\mathbf{s_1} - \mathbf{s_0} \\ 6\mathbf{s_2} \\ 6\mathbf{s_3} \\ \vdots \\ 6\mathbf{s_{n-1}} - \mathbf{s_n} \end{bmatrix}.$$

Karena vektor posisi $\mathbf{b_k}$ dan $\mathbf{s_k}$ memiliki dua komponen (koordinat x dan y), kita dapat memisahkan sistem persamaan di atas menjadi dua sistem persamaan linear yang terpisah, satu untuk komponen x dan satu untuk komponen y. Dengan menyelesaikan kedua sistem persamaan tersebut, kita dapat menentukan posisi titik-titik kontrol B_k yang diperlukan untuk membentuk kurva Bezier kubik yang halus dan sesuai dengan titik-titik interpolasi yang diberikan.

2.6 Regresi Polinomial Berganda (*Multivariate Polynomial Regression*)

Diberikan n titik data $(x_1,y_1),(x_2,y_2),...,(x_n,y_n)$ dengan x_i adalah variabel dan y_i adalah hasil pengukuran ke-i ($1 \leq i \leq n$), regresi linear sederhana[3] bertujuan untuk menemukan garis lurus

$$y = \beta_0 + \beta_1 x + \epsilon$$

yang paling sesuai dengan data tersebut dengan ϵ adalah galatnya.[4] Namun, dalam banyak kasus, besaran yang kita amati bisa dipengaruhi oleh lebih dari satu variabel. Untuk mengatasi hal ini, kita dapat menggunakan **regresi linier berganda** (*multivariate linear regression*).

Misalnya, jika y dipengaruhi oleh tiga variabel independen[5] x_1, x_2, dan x_3, kita dapat memodelkan hubungan tersebut dengan persamaan linier berganda

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon,$$

dengan ϵ adalah galat (error) yang mengikuti distribusi normal dengan rata-rata nol dan varians konstan. Tugas kita adalah mengestimasi nilai koefisien $\beta_0,\beta_1,\beta_2,$ dan β_3 berdasarkan data yang diberikan. Secara umum, andaikata ada n variabel independen $x_1,x_2,...,x_n$, maka persamaan regresi linier berganda dapat dituliskan sebagai

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon,$$

sehingga jika kita memiliki m titik data dan x_{ij} adalah nilai dari variabel x_j pada titik data ke-i serta y_i dan ϵ_i adalah nilai hasil pengukuran dan galatnya pada titik data ke-i, kita dapat menuliskan sistem persamaan linear berikut untuk m titik data,

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}_{m\times 1} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1n} \\ 1 & x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}_{m\times(n+1)} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}_{(n+1)\times 1} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_m \end{bmatrix}_{m\times 1},$$

atau

$$Y = X\beta + \boldsymbol{\epsilon},$$

dengan matriks Y adalah vektor hasil pengukuran, X adalah matriks desain, $\boldsymbol{\beta}$ adalah vektor koefisien, dan $\boldsymbol{\epsilon}$ adalah vektor galat.

Tugas kita pada dasarnya adalah mengestimasi nilai vektor koefisien[6] $\hat{\beta}$ seakurat mungkin, atau dengan kata lain, dengan galat sekecil mungkin. Metode umum yang digunakan adalah metode kuadrat terkecil (*least squares*) yang meminimalkan jumlah kuadrat galat, yaitu $\| \epsilon \|^2 = \| \mathbf{y} - X\beta \|^2$. Dengan metode ini, kita dapat menemukan solusi untuk $\hat{\beta}$ dengan menyelesaikan persamaan normal berikut:[7]

$$X^T X\beta = X^T\mathbf{y},$$

atau

$$\hat{\beta} = (X^T X)^{-1} X^T\mathbf{y}.$$

Sekarang, bagaimana jika hubungan antara y dan variabel-variabel independennya tidak linier? Salah satu pendekatan yang dapat kita gunakan adalah **regresi polinomial berganda** (*multivariate polynomial regression*). Idenya masih mirip: mencari fungsi polinomial yang kurvanya paling sesuai dengan data yang diberikan.[8] Namun, kali ini akan terdapat variabel interaksi, yakni variabel yang merupakan hasil kali dari beberapa variabel independen. Misalnya, untuk tiga variabel independen x_1, x_2, dan x_3 dan derajat 3, fungsi polinomial yang kita hasilkan bisa saja berbentuk

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_1^2 + \beta_5 x_1 x_2 + \beta_6 x_1 x_3 + \beta_7 x_2^2 + \beta_8 x_2 x_3 + \beta_9 x_2^3 + \beta_{10} x_1^3$$
$$+ \beta_{11} x_1^2 x_2 + \beta_{12} x_1^2 x_3 + \beta_{13} x_1 x_2^2 + \beta_{14} x_1 x_2 x_3 + \beta_{15} x_1 x_3^2 + \beta_{16} x_2^3 + \beta_{17} x_2^2 x_3 + \beta_{18} x_2 x_3^2 + \beta_{19} x_3^3 + \epsilon.$$

Secara umum, jika kita memiliki k variabel independen dan ingin membuat polinomial dengan derajat d, jumlah total suku dalam polinomial tersebut adalah $p = \binom{d+k}{k}$. Lantas, bagaimana cara kita menemukan seluruh koefisien $\beta_0, \beta_1, \ldots, \beta_p$?

Kita bisa mengadopsi pendekatan serupa dengan yang kita gunakan pada regresi linier berganda. Kali ini, variabel-variabel seperti x_1^2, $x_1 x_2$, x_1^3, dan sebagainya akan dianggap sebagai variabel independen yang berbeda.[9] Dengan demikian, kita dapat menyusun sistem persamaan linear yang mirip dengan yang sebelumnya, tetapi dengan lebih banyak kolom pada matriks desain X untuk mengakomodasi semua suku dalam polinomial. Setelah itu, kita dapat menggunakan metode kuadrat terkecil untuk menemukan estimasi koefisien $\hat{\beta}$ (persamaannya sama persis seperti pada regresi linier berganda).

Misalnya, jika kita memiliki lima sampel data dan tiga variabel independen ($k = 3$) dan ingin membuat polinomial dengan derajat dua ($d = 2$), maka jumlah total suku dalam polinomial tersebut adalah $\binom{2+3}{3} = 10$ dan matriks desain X akan berbentuk seperti ini:

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} & x_{11}^2 & x_{11}x_{12} & x_{11}x_{13} & x_{12}^2 & x_{12}x_{13} & x_{13}^2 \\ 1 & x_{21} & x_{22} & x_{23} & x_{21}^2 & x_{21}x_{22} & x_{21}x_{23} & x_{22}^2 & x_{22}x_{23} & x_{23}^2 \\ 1 & x_{31} & x_{32} & x_{33} & x_{31}^2 & x_{31}x_{32} & x_{31}x_{33} & x_{32}^2 & x_{32}x_{33} & x_{33}^2 \\ 1 & x_{41} & x_{42} & x_{43} & x_{41}^2 & x_{41}x_{42} & x_{41}x_{43} & x_{42}^2 & x_{42}x_{43} & x_{43}^2 \\ 1 & x_{51} & x_{52} & x_{53} & x_{51}^2 & x_{51}x_{52} & x_{51}x_{53} & x_{52}^2 & x_{52}x_{53} & x_{53}^2 \end{bmatrix},$$

yang sebenarnya analog dengan matriks desain untuk regresi linier berganda

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} & x_{17} & x_{18} & x_{19} \\ 1 & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} & x_{27} & x_{28} & x_{29} \\ 1 & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} & x_{37} & x_{38} & x_{39} \\ 1 & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} & x_{47} & x_{48} & x_{49} \\ 1 & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} & x_{57} & x_{58} & x_{59} \end{bmatrix}$$

dengan x_{ij} adalah nilai dari suku ke-j pada titik data ke-i. Pada intinya, cukup gunakan persamaan (1) untuk menemukan estimasi koefisien $\hat{\beta}$.

3 Implementasi

Masing-masing file pada program yang sudah kami implementasikan berisi method-method yang menunjang kelima spesifikasi wajib pada program ini, yaitu penyelesaian SPL, perhitungan determinan, pembuatan matriks invers, perhitungan interpolasi, dan regresi polinomial berganda. Berikut merupakan penjelasan dari masing-masing file di dalam program ini.

3.1 Matriks

3.1.1 Kelas Matriks

Kelas Matrix mengenkapsulasi semua data dan operasi yang terkait dengan matriks, memastikan konsistensi dan kemudahan penggunaan di seluruh program.

A. Representasi Data dan Konstruktor

- Representasi Internal: Matriks direpresentasikan secara internal menggunakan array dua dimensi `private double[][] data`. Dimensi matriks (jumlah baris dan kolom) disimpan dalam variabel `private int rows` dan `private int cols` untuk efisiensi akses.
- Konstruktor: Terdapat dua konstruktor utama untuk membuat objek Matrix:
    1. `public Matrix(double[][] initialData)`: Membuat matriks dari array 2D yang sudah ada. Konstruktor ini melakukan deep copy untuk memastikan bahwa data internal matriks tidak terhubung dengan referensi data eksternal.
    2. `public Matrix(int rows, int cols)`: Membuat matriks baru dengan dimensi yang ditentukan, di mana semua elemennya diinisialisasi dengan nilai nol.

Kedua konstruktor dilengkapi dengan validasi input untuk mencegah pembuatan matriks yang tidak valid (misalnya, dimensi non-positif atau data input kosong).

B. Operasi Baris Elementer (OBE)

OBE adalah inti dari banyak algoritma aljabar linier. Kelas Matrix menyediakan tiga metode fundamental untuk OBE, yang memodifikasi matriks secara in-place (langsung pada objeknya).

- `public void swapRows(int row1, int row2)`: Menukar posisi dua baris pada matriks.
- `public void multiplyRow(int row, double scalar)`: Mengalikan semua elemen pada satu baris dengan sebuah nilai skalar.
- `public void addMultiplyOfRow(int targetRow, int sourceRow, double scalar)`: Menambahkan hasil perkalian sebuah baris (`sourceRow`) dengan skalar ke baris lain (`targetRow`). Ini adalah operasi utama yang digunakan untuk eliminasi pada algoritma Gauss dan Gauss-Jordan.

C. Metode Transformasi dan Operasi Utama

Kelas ini menyediakan metode tingkat tinggi yang memanfaatkan OBE untuk melakukan transformasi matriks.

- `public Matrix toEselonMatrixWithSteps()`: Mengimplementasikan algoritma eliminasi Gauss (forward elimination) untuk mengubah matriks menjadi bentuk eselon baris. Metode ini dirancang secara khusus untuk memenuhi spesifikasi tugas besar dengan menampilkan setiap langkah OBE (penukaran baris dan eliminasi) yang dilakukannya ke konsol.
- `public static Matrix multiply(Matrix a, Matrix b)`: Sebuah metode statis untuk melakukan perkalian matriks. Metode ini memeriksa kesesuaian dimensi kedua matriks sebelum melakukan perkalian.
- `public Matrix transpose()`: Menghasilkan matriks transpose baru dari matriks saat ini tanpa mengubah matriks aslinya.

D. Metode Pendukung dan Utilitas

Berbagai metode pendukung disediakan untuk memfasilitasi algoritma yang lebih kompleks di kelas lain.

- `public Matrix copy()`: Menghasilkan salinan (deep copy) dari objek matriks. Ini sangat penting untuk mencegah modifikasi yang tidak diinginkan pada matriks asli saat diproses oleh sebuah fungsi.
- `public static Matrix minorOf(Matrix A, int row, int col)`: Sebuah metode statis yang menghasilkan matriks minor dengan menghilangkan baris dan kolom tertentu. Metode ini merupakan komponen kunci untuk perhitungan determinan dengan metode ekspansi kofaktor dan untuk menemukan matriks adjoin.
- `public static boolean isZero(double x)`: Metode statis untuk membandingkan sebuah angka `double` dengan nol menggunakan toleransi `EPS` (epsilon). Ini penting untuk mengatasi masalah presisi pada komputasi floating-point.

E. Input/Output dan Tampilan

Untuk memudahkan interaksi dengan pengguna dan pengujian, kelas Matrix dilengkapi fungsionalitas I/O.

- `public static Matrix fromFile(String path)`: Metode statis yang mampu membaca dan mem-parsing data matriks dari sebuah file teks. File harus memiliki format di mana setiap baris berisi elemen-elemen matriks yang dipisahkan oleh spasi.
- `public void display()` dan `@Override public String toString()`: Menyediakan cara untuk menampilkan matriks ke konsol dengan format yang rapi dan mudah dibaca, di mana setiap elemen dibulatkan hingga tiga angka desimal.

3.2 Penyelesaian SPL (SPLSolver)

3.2.1 Penyelesaian SPL dengan Eliminasi Gauss-Jordan

Metode ini mengimplementasikan algoritma Eliminasi Gauss-Jordan untuk mengubah matriks augmented menjadi bentuk eselon tereduksi (RREF). Dari bentuk RREF inilah solusi SPL dapat langsung ditentukan.

Metode ini memiliki alur kerja implementasi sebagai berikut:

1. Inisialisasi, yaitu fungsi menerima sebuah `Matrix` augmented sebagai input dan membuat salinannya untuk dimodifikasi, Sebuah variabel `pivotRow` diinisialisasi untuk melacak baris yang sedang digunakan sebagai pivot.'
2. Iterasi Pivot, yaitu program melakukan iterasi untuk setiap kolom (kecuali kolom hasil) untuk melakukan pivot.
    - Pencarian Pivot: Pada kolom `j`, program mencari baris pertama yang memiliki elemen non-nol.
    - Penukaran Baris: Jika baris dengan elemen non-nol ditemukan (`i`) tetapi tidak berada di posisi `pivotRow`, maka baris `i` dengan `pivotRow` ditukar (`swapRows`) untuk memindahkan pivot kek posisi yang tepat.
    - Normalisasi Baris Pivot: Elemen pivot pada `matriks[pivotRow][j]` dibuat menjadi 1 dengan membagi seluruh baris `pivotRow` dengan nilai elemen pivot itu sendiri (`multiplyRow`).
    - Eliminasi Kolom: Setelah pivot menjadi 1, program melakukan iterasi ke semua baris lain (di atas dan di bawah `pivotRow`). Untuk setiap baris `k`, elemen pada kolom `j` di baris tersebut dibuat menjadi nol. Ini dilakukan dengan cara mengurangkan baris `k` dengan hasil perkalian baris `pivotRow` dan nilai `matriks[k][j] (addMulitplyOfRow).`
3. Iterasi Lanjutan: `pivotRow` diinkrementasi dan proses berlanjut ke kolom berikutnya hingga semua kolom telah diproses atau semua baris telah memiliki pivot.
4. Hasil: Fungsi mengembalikan matriks yang sudah dalam bentuk RREF.

3.2.2 Penyelesaian SPL dengan Eliminasi Gauss

Metode ini menggunakan Eliiminasi Gauss untuk emngubah matriks menjadi bentuk eselon baris (Row Echelon Form), kemudian melakukan substitusi balik untuk menemukan solusi.

Metode ini memiliki alur kerja implementasi sebagai berikut:

1. Transformasi Eselon Baris: Fungsi memanggil metode eksternal (`toEselonMatrixWithSteps`) untuk mengubah matriks augmented input menjadi bentuk eselon baris. Proses ini tidak harus menghasilkan pivot bernilai 1.
2. Substitusi Balik: Setelah matriks berada dalam bentuk eselon baris, proses substitusi balik dimulai dari baris paling bawah.

- Inisialisasi: Sebuah array `double[]` solution dibuat untuk menyimpan nilai dari setiap variabel x.
- Iterasi Baris (dari bawah ke atas): Program melakukan iterasi dari baris `i = rows - 1` hingga ke baris 0.
- Perhitungan Variabel: Untuk setiap baris `i`, nilai variabel x_i dihitung menggunakan rumus:

$$x_i = \frac{(b_i - \sum_{j=i+1}^{n-1} a_{ij} x_j)}{a_{ii}}$$

  Di mana b_i adalah elemen di kolom hasil, a_{ij} adalah koefisien, dan x_j adalah nilai variabel yang sudah ditemukan pada iterasi sebelumnya.

3. Hasil: Fungsi mengembalikan array `double[]` yang berisi solusi unik dari SPL.

3.2.3 Penyelesaian SPL dengan Kaidah Cramer

Metode ini diimplementasikan berdasarkan teori Kaidah Cramer yang menggunakan determinan untuk mencari solusi SPL.

Metode ini memiliki alur kerja sebagai berikut:

1. Input: Fungsi menerima dua matriks: matriks koefisien A dan matriks hasil (vektor kolom) B.
2. Hitung Determinan Utama: Determinan dari matriks A (det(A)) dihitung terlebih dahulu menggunakan metode reduksi baris. Jika det(A) bernilai nol, metode dihentikan karena Kaidah Cramer tidak dapat digunakan.
3. Iterasi untuk Setiap Variabel: Program melakukan iterasi sebanyak jumlah variabel (kolom pada A).
   - Bentuk Matrix A_i dibuat dengan menyalin matriks A dan mengganti kolom ke-i dengan kolom dari matriks B.
   - Hitung Determinan A_i: Determinan dari matriks A_i dihitung.
   - Hitung x_i: Nilai variabel x_i dihitung dengan rumus $x_i = det(A_i)/det(A)$.
4. Hasil: Fungsi mengembalikan array `double[]` yang berisi solusi dari SPL.

3.2.4 Penyelesaian SPL dengan Matriks Balikan

Metode ini menyelesaikan SPL dengan memanfaatkan properti matriks $A . X = B => X = A^{-1} . B$

Metode ini memiliki alur kerja sebagai berikut:

1. Input: Fungsi ini menerima matriks koefisien A dan matriks hasil B.
2. Cari Invers Matriks A: Program memanggil metode eksternal (`Invers.inversOBE(a)`) untuk mencari matriks balikan dari A. Jika matriks A tidak memiliki invers (singular), metode akan berhenti.

3. Perkalian Matriks: Jika invers A^{-1} ditemukan, solusi X didapatkan dengan mengalikan A^{-1} dengan matriks B.
4. Hasil: Fungsi mengembalikan sebuah Matrix (vektor kolom) yang merupakan solusi dari SPL.

3.2.5 Penentuan dan Penampilan Jenis Solusi

Fungsi ini adalah komponen kunci yang bertugas menerjemahkan matriks RREF hasil eliminasi menjadi solusi yang dapat dibaca oleh pengguna. Fungsi ini mampu membedakan antara solusi unik, banyak solusi, dan tidak ada solusi.

Alur Kerja Implementasinya sebagai berikut:

1. Deteksi Tidak Ada Solusi:
    - Program memeriksa setiap baris pada matriks RREF.
    - Jika ditemukan sebuah baris di mana semua elemen pada kolom variabel bernilai nol tetapi elemen pada kolom hasil (konstanta) tidak nol (contoh: 0 0 0 | 1), maka disimpulkan bahwa SPL tidak memiliki solusi dan proses dihentikan.
2. Identifikasi Variabel Pivot dan Variabel Bebas:
    - Program mengidentifikasi kolom mana yang memiliki pivot (elemen 1 utama). Kolom-kolom ini merepresentasikan variabel terikat (variabel pivot).
    - Kolom yang tidak memiliki pivot merepresentasikan variabel bebas.
3. Deteksi Solusi Unik:
    - Jika jumlah pivot sama dengan jumlah variabel, maka setiap variabel adalah variabel terikat.
    - Artinya, SPL memiliki solusi unik. Program akan menampilkan nilai untuk setiap variabel.
4. Deteksi Banyak Solusi:
    - Jika jumlah pivot lebih sedikit dari jumlah variabel, maka terdapat satu atau lebih variabel bebas.
    - Penamaan Parameter: variabel bebas akan direpresentasikan sebagai parameter.
    - Ekspresi Parametrik: Program menyusun persamaan untuk setiap variabel terikat. Nilainya diekspresikan dalam bentuk kombinasi linier dari variabel-variabel bebas dan konstanta dari baris pivotnya.

3.3 Determinan

3.3.1 Determinan Metode Ekspansi Kofaktor

Metode ini menghitung determinan matriks dengan membagi matriks menjadi minor dan menghitung kofaktor dari setiap elemen baris pertama. Proses ini dilakukan secara rekursif hingga mencapai matriks berukuran 1x1 atau 2x2, yang dapat dihitung secara langsung.

Alur Kerja:

1. Validasi Matriks Persegi – Pastikan input berbentuk matriks persegi.
2. Basis Rekursi:

   - Matriks 1x1 → determinan = elemen itu sendiri.
   - Matriks 2x2 → determinan dihitung langsung dari elemen-elemen matriks.
3. Rekursi:
   - Untuk matriks lebih besar, setiap elemen pada baris pertama digunakan untuk membentuk matriks minor.
   - Hitung determinan minor secara rekursif, kalikan dengan elemen dan tanda kofaktor.
4. Penjumlahan Kofaktor – Determinan diperoleh dengan menjumlahkan semua hasil perkalian elemen dan determinan minor.

3.3.2 Determinan Metode Reduksi Baris

Metode ini menghitung determinan dengan mengubah matriks menjadi bentuk segitiga atas melalui operasi baris elementer. Determinan diperoleh dari perkalian elemen diagonal, sambil memperhatikan tanda akibat pertukaran baris.

Alur Kerja:

1. Validasi Matriks Persegi – Pastikan input berbentuk matriks persegi.
2. Salin Matriks – Buat salinan untuk dimodifikasi tanpa mengubah matriks asli.
3. Iterasi Pivot:
   - Cari pivot terbesar di kolom saat ini untuk stabilitas numerik.
   - Jika pivot = 0 → matriks singular → determinan = 0.
4. Pertukaran Baris:
   - Jika pivot bukan baris saat ini, tukar baris dan ubah tanda determinan.
5. Eliminasi Kolom:
   - Buat semua elemen di bawah pivot menjadi nol dengan operasi baris.
6. Perkalian Diagonal:
   - Setelah matriks menjadi segitiga atas, determinan diperoleh dari perkalian elemen diagonal, dikalikan dengan tanda akibat pertukaran baris.

3.4 Invers

3.4.1 Invers Metode Adjoin (Kofaktor)

Metode Adjoin menghitung invers matriks dengan menggunakan matriks kofaktor dan transpose. Proses ini mengikuti langkah-langkah manual dari definisi invers, sehingga mudah dipahami untuk matriks kecil.

Alur Kerja:

1. Validasi Matriks Persegi – Pastikan input berbentuk matriks persegi.

2. Hitung Determinan Matriks – Jika determinan = 0, matriks singular → proses dihentikan.
3. Bentuk Matriks Kofaktor – Setiap elemen dikalikan dengan tanda tertentu dan hasil minor disimpan.
4. Transpos Matriks Kofaktor – Matriks adjoin dibuat dari transpose kofaktor.
5. Hitung Invers – Matriks invers diperoleh dengan mengalikan adjoin dengan faktor skala (1/determinan).

3.4.2 Invers Metode Augment (OBE)

Metode OBE (Operasi Baris Elementer) atau Gauss–Jordan mengubah matriks menjadi bentuk eselon tereduksi (RREF) dan secara bersamaan membentuk invers. Matriks invers diperoleh dari sisi kanan augmented matrix setelah sisi kiri menjadi identitas.

Alur Kerja:

1. Validasi Matriks Persegi – Hanya matriks persegi yang dapat diinvers.
2. Bentuk Matriks Augmentasi [A | I] – Tambahkan matriks identitas di sisi kanan A.
3. Pivoting untuk Setiap Kolom – Cari elemen pivot terbesar untuk stabilitas numerik; jika pivot ≈ 0 → matriks singular.
4. Normalisasi Baris Pivot – Pivot dibuat 1 dengan membagi seluruh baris.
5. Eliminasi Kolom Lain – Semua elemen di kolom pivot selain pivot dibuat 0 dengan operasi baris.
6. Iterasi Lanjutkan – Proses diulang untuk seluruh kolom sampai sisi kiri menjadi identitas.
7. Ambil Matriks Invers – Matriks invers adalah sisi kanan dari matriks augmentasi.

3.5 Interpolasi

3.5.1 Interpolasi Polinomial

Interpolasi polinomial digunakan untuk mencari polinomial Pn(x) derajat n−1 yang melewati semua titik data (x_i, y_i). Konsep utamanya adalah menyelesaikan sistem linear Vandermonde:

$$A \times c = Y$$

di mana:
A = Matriks Vandermonde dari x
c = Koefisien polinomial
Y = Matriks kolom dari y

Alur Kerja:

1. Validasi Input:
   Panjang array x dan y harus sama.
2. Bentuk Matriks Vandermonde (A):
   Elemen $A_{ij} = x_i^j$
3. Bentuk Matriks Kolom Y:
   Elemen $Y_i = y_i$
4. Hitung Invers Matriks A:
   Gunakan metode OBE untuk menghitung A^{-1}
   Jika A singular → interpolasi gagal.
5. Hitung Koefisien Polinomial:
   $c = A^{-1}.Y$

3.5.2 Interpolasi Splina Bezier Kubik

Interpolasi Bézier kubik membentuk kurva halus melalui titik data menggunakan titik kontrol. Titik kontrol dihitung sedemikian rupa sehingga kurva kubik spline melewati semua titik data.

Alur Kerja:

1. Validasi Titik:
   Minimal 3 titik dibutuhkan untuk spline kubik.
2. Bentuk Matriks Koefisien A:
   Matriks tridiagonal ukuran $(n - 1) \times (n - 1)$ dengan diagonal utama = 4 dan diagonal samping = 1.
3. Bentuk Matriks SX dan SY:
   Matriks kolom berisi koordinat x dan y titik kontrol yang dimodifikasi berdasarkan rumus spline kubik.
4. Hitung Invers Matriks A:
   Gunakan metode OBE untuk menghitung A^{-1}
   Jika A singular → interpolasi gagal.
5. Hitung Titik Kontrol Tengah (BX dan BY):
   $BX = A^{-1}.SX$
   $BY = A^{-1}.SY$
6. Gabungkan Titik Kontrol Penuh:
   Titik awal dan akhir = titik data asli.
   Titik tengah = hasil BX dan BY.

3.6 Regresi

Metode regresi polinomial berganda pada kelas `Regresi` digunakan untuk menentukan hubungan antara variabel independen (X) dan variabel dependen (Y) dengan membentuk polinomial hingga derajat tertentu.

Alur Kerja:

1. Persiapan Basis dan Matriks Desain:
    a. Program pertama-tama menghasilkan basis polinomial yang berisi semua kombinasi pangkat variabel hingga derajat polinomial yang diinginkan.
    b. Berdasarkan basis ini, setiap baris input X diubah menjadi baris matriks desain dengan menghitung hasil perkalian variabel sesuai pangkat yang ada di basis.
2. Membangun Sistem Linier:
    a. Matriks desain ditranspose dan dikalikan dengan matriks desain asli untuk membentuk matriks koefisien.
    b. Matriks desain ditranspose dikalikan dengan matriks hasil Y untuk membentuk matriks hasil.
3. Penyelesaian Koefisien Polinomial:
    a. Program mencoba menghitung invers dari matriks koefisien.
    b. Jika invers berhasil dihitung, matriks hasil dikalikan dengan invers untuk mendapatkan koefisien polinomial.
    c. Jika invers gagal (misal kasus singular), program menyelesaikan sistem linier menggunakan metode SPL berbasis Gauss.
4. Output dan Visualisasi:
    a. Koefisien regresi yang diperoleh dapat ditampilkan dalam bentuk persamaan polinomial yang mudah dibaca, menunjukkan kontribusi masing-masing variabel dan pangkatnya terhadap hasil regresi.

3.7 Main (App)

Pada file ini, terdapat sebuah program utama yang mengatur keberlangsungan program ini. Pada Main/App, kita bisa memilih opsi untuk menyelesaikan beberapa persoalan yang sudah dijelaskan sebelumnya, yaitu penyelesaian solusi SPL, determinan, matriks invers, interpolasi, dan regresi. Selain program tersebut, ada method yang lain untuk membantu program utama berjalan, yaitu membaca file, menulis file, membaca input dan mengeluarkan output, dan beberapa method lain yang diperlukan untuk membaca input dan mengeluarkan output yang benar.

Alur dari program utama adalah sebagai berikut.

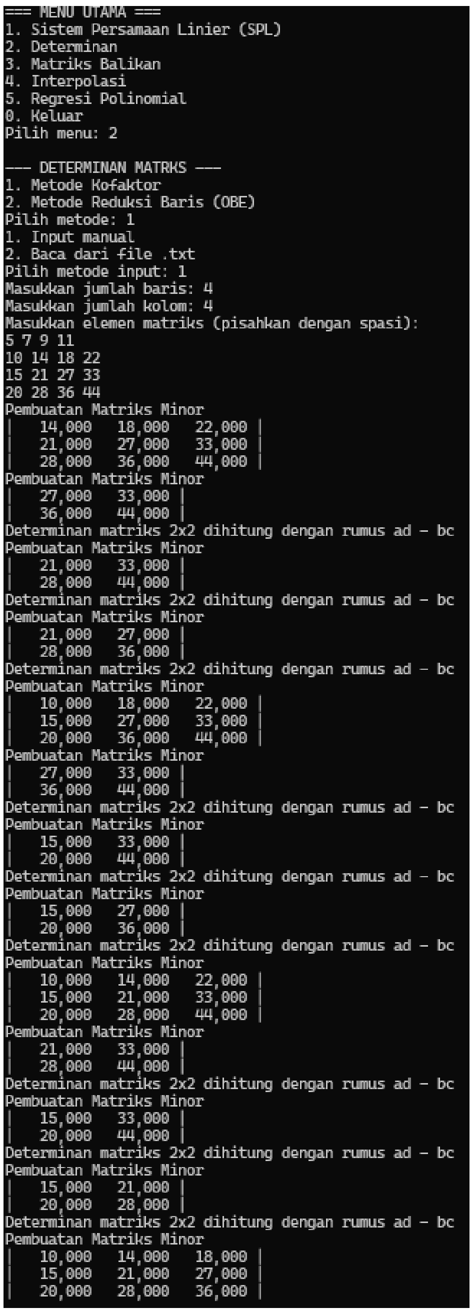
1. Pemilihan Jenis Penyelesaian Masalah
    a. Program akan meminta kita untuk memilih permasalahan yang ingin diselesaikan, seperti solusi SPL, determinan, invers, interpolasi, dan regresi.
    b. Lalu, program akan meminta kita untuk memilih metode yang tersedia. Umumnya masing-masing hal tersebut memiliki lebih dari satu metode penyelesaian, kecuali di regresi.
2. Menerima input dan mengolahnya.
    a. Menerima masukan yang sesuai dengan spesifikasinya masing-masing.
    b. Mengolah masukan sesuai dengan metode penyelesaian yang dipilih. Selanjutnya, mengembalikan data yang diminta.
3. Program akan mengulang kembali
    a. Program akan menyimpan jawaban pada file .txt
    b. Selanjutnya, program akan mengulang kembali agar kita bisa menggunakannya kembali. Untuk menghentikan program, dapat dilakukan dengan mengirim angka nol.

4 Eksperimen

Berikut merupakan hasil pengujian dari masing-masing *case* terhadap program yang kami sudah implementasikan.

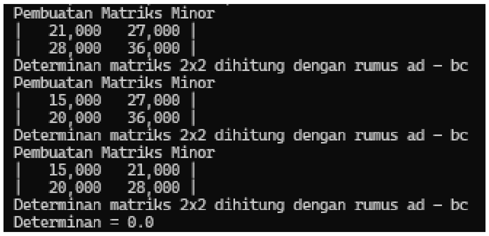
4.1 Determinan Matriks

4.1.1 Kasus Uji 1

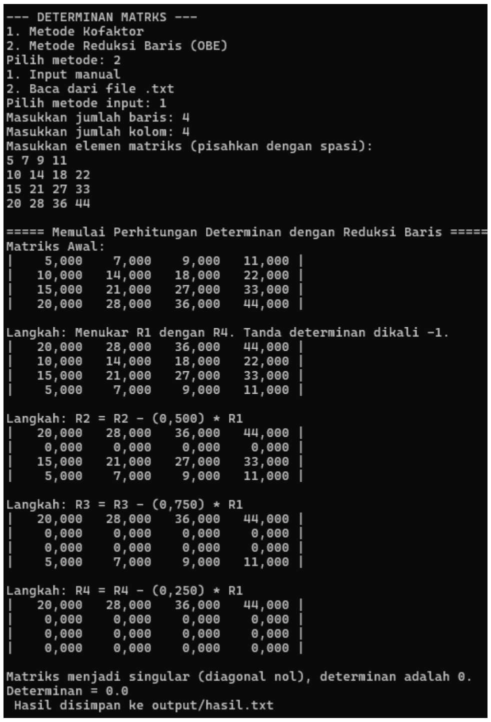


```
=== MENU UTAMA ===
1. Sistem Persamaan Linier (SPL)
2. Determinan
3. Matriks Balikan
4. Interpolasi
5. Regresi Polinomial
0. Keluar
Pilih menu: 2

--- DETERMINAN MATRKS ---
1. Metode Kofaktor
2. Metode Reduksi Baris (OBE)
Pilih metode: 1
1. Input manual
2. Baca dari file .txt
Pilih metode input: 1
Masukkan jumlah baris: 4
Masukkan jumlah kolom: 4
Masukkan elemen matriks (pisahkan dengan spasi):
5 7 9 11
10 14 18 22
15 21 27 33
20 28 36 44
Pembuatan Matriks Minor
|   14,000   18,000   22,000 |
|   21,000   27,000   33,000 |
|   28,000   36,000   44,000 |
Pembuatan Matriks Minor
|   27,000   33,000 |
|   36,000   44,000 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|   21,000   33,000 |
|   28,000   44,000 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|   21,000   27,000 |
|   28,000   36,000 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|   10,000   18,000   22,000 |
|   15,000   27,000   33,000 |
|   20,000   36,000   44,000 |
Pembuatan Matriks Minor
|   27,000   33,000 |
|   36,000   44,000 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|   15,000   33,000 |
|   20,000   44,000 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|   15,000   27,000 |
|   20,000   36,000 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|   10,000   14,000   22,000 |
|   15,000   21,000   33,000 |
|   20,000   28,000   44,000 |
Pembuatan Matriks Minor
|   21,000   33,000 |
|   28,000   44,000 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|   15,000   33,000 |
|   20,000   44,000 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|   15,000   21,000 |
|   20,000   28,000 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|   10,000   14,000   18,000 |
|   15,000   21,000   27,000 |
|   20,000   28,000   36,000 |
```



```
Pembuatan Matriks Minor
|   21,000   27,000 |
|   28,000   36,000 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|   15,000   27,000 |
|   20,000   36,000 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|   15,000   21,000 |
|   20,000   28,000 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Determinan = 0.0
```



```
--- DETERMINAN MATRKS ---
1. Metode Kofaktor
2. Metode Reduksi Baris (OBE)
Pilih metode: 2
1. Input manual
2. Baca dari file .txt
Pilih metode input: 1
Masukkan jumlah baris: 4
Masukkan jumlah kolom: 4
Masukkan elemen matriks (pisahkan dengan spasi):
5 7 9 11
10 14 18 22
15 21 27 33
20 28 36 44

===== Memulai Perhitungan Determinan dengan Reduksi Baris =====
Matriks Awal:
|    5,000    7,000    9,000   11,000 |
|   10,000   14,000   18,000   22,000 |
|   15,000   21,000   27,000   33,000 |
|   20,000   28,000   36,000   44,000 |

Langkah: Menukar R1 dengan R4. Tanda determinan dikali -1.
|   20,000   28,000   36,000   44,000 |
|   10,000   14,000   18,000   22,000 |
|   15,000   21,000   27,000   33,000 |
|    5,000    7,000    9,000   11,000 |

Langkah: R2 = R2 - (0,500) * R1
|   20,000   28,000   36,000   44,000 |
|    0,000    0,000    0,000    0,000 |
|   15,000   21,000   27,000   33,000 |
|    5,000    7,000    9,000   11,000 |

Langkah: R3 = R3 - (0,750) * R1
|   20,000   28,000   36,000   44,000 |
|    0,000    0,000    0,000    0,000 |
|    0,000    0,000    0,000    0,000 |
|    5,000    7,000    9,000   11,000 |

Langkah: R4 = R4 - (0,250) * R1
|   20,000   28,000   36,000   44,000 |
|    0,000    0,000    0,000    0,000 |
|    0,000    0,000    0,000    0,000 |
|    0,000    0,000    0,000    0,000 |

Matriks menjadi singular (diagonal nol), determinan adalah 0.
Determinan = 0.0
 Hasil disimpan ke output/hasil.txt
```

4.1.2 Kasus Uji 2

```
=== MENU UTAMA ===
1. Sistem Persamaan Linier (SPL)
2. Determinan
3. Matriks Balikan
4. Interpolasi
5. Regresi Polinomial
0. Keluar
Pilih menu: 2

--- DETERMINAN MATRKS ---
1. Metode Kofaktor
2. Metode Reduksi Baris (OBE)
Pilih metode: 1
1. Input manual
2. Baca dari file .txt
Pilih metode input: 1
Masukkan jumlah baris: 4
Masukkan jumlah kolom: 4
Masukkan elemen matriks (pisahkan dengan spasi):
1 1/2 1/3 1/4
1/2 1/3 1/4 1/5
1/3 1/4 1/5 1/6
1/4 1/5 1/6 1/7
Pembuatan Matriks Minor
|     0,333    0,250    0,200 |
|     0,250    0,200    0,167 |
|     0,200    0,167    0,143 |
Pembuatan Matriks Minor
|     0,200    0,167 |
|     0,167    0,143 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|     0,250    0,167 |
|     0,200    0,143 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|     0,250    0,200 |
|     0,200    0,167 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|     0,500    0,250    0,200 |
|     0,333    0,200    0,167 |
|     0,250    0,167    0,143 |
Pembuatan Matriks Minor
|     0,200    0,167 |
|     0,167    0,143 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|     0,333    0,167 |
|     0,250    0,143 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|     0,333    0,200 |
|     0,250    0,167 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|     0,500    0,333    0,200 |
|     0,333    0,250    0,167 |
|     0,250    0,200    0,143 |
Pembuatan Matriks Minor
|     0,250    0,167 |
|     0,200    0,143 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|     0,333    0,167 |
|     0,250    0,143 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|     0,333    0,250 |
|     0,250    0,200 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|     0,500    0,333    0,250 |
|     0,333    0,250    0,200 |
|     0,250    0,200    0,167 |
Pembuatan Matriks Minor
|     0,250    0,200 |
|     0,200    0,167 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|     0,333    0,200 |
|     0,250    0,167 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Pembuatan Matriks Minor
|     0,333    0,250 |
|     0,250    0,200 |
Determinan matriks 2x2 dihitung dengan rumus ad - bc
Determinan = 1.6534391534331948E-7
```

```
--- DETERMINAN MATRKS ---
1. Metode Kofaktor
2. Metode Reduksi Baris (OBE)
Pilih metode: 2
1. Input manual
2. Baca dari file .txt
Pilih metode input: 1
Masukkan jumlah baris: 4
Masukkan jumlah kolom: 4
Masukkan elemen matriks (pisahkan dengan spasi):
1 1/2 1/3 1/4
1/2 1/3 1/4 1/5
1/3 1/4 1/5 1/6
1/4 1/5 1/6 1/7

===== Memulai Perhitungan Determinan dengan Reduksi Baris =====
Matriks Awal:
|    1,000    0,500    0,333    0,250 |
|    0,500    0,333    0,250    0,200 |
|    0,333    0,250    0,200    0,167 |
|    0,250    0,200    0,167    0,143 |

Langkah: R2 = R2 - (0,500) * R1
|    1,000    0,500    0,333    0,250 |
|    0,000    0,083    0,083    0,075 |
|    0,333    0,250    0,200    0,167 |
|    0,250    0,200    0,167    0,143 |

Langkah: R3 = R3 - (0,333) * R1
|    1,000    0,500    0,333    0,250 |
|    0,000    0,083    0,083    0,075 |
|    0,000    0,083    0,089    0,083 |
|    0,250    0,200    0,167    0,143 |

Langkah: R4 = R4 - (0,250) * R1
|    1,000    0,500    0,333    0,250 |
|    0,000    0,083    0,083    0,075 |
|    0,000    0,083    0,089    0,083 |
|    0,000    0,075    0,083    0,080 |

Langkah: Menukar R2 dengan R3. Tanda determinan dikali -1.
|    1,000    0,500    0,333    0,250 |
|    0,000    0,083    0,089    0,083 |
|    0,000    0,083    0,083    0,075 |
|    0,000    0,075    0,083    0,080 |

Langkah: R3 = R3 - (1,000) * R2
|    1,000    0,500    0,333    0,250 |
|    0,000    0,083    0,089    0,083 |
|    0,000    0,000   -0,006   -0,008 |
|    0,000    0,075    0,083    0,080 |

Langkah: R4 = R4 - (0,900) * R2
|    1,000    0,500    0,333    0,250 |
|    0,000    0,083    0,089    0,083 |
|    0,000    0,000   -0,006   -0,008 |
|    0,000    0,000    0,003    0,005 |

Langkah: R4 = R4 - (-0,600) * R3
|    1,000    0,500    0,333    0,250 |
|    0,000    0,083    0,089    0,083 |
|    0,000    0,000   -0,006   -0,008 |
|    0,000    0,000    0,000    0,000 |

Matriks segitiga atas terbentuk. Determinan adalah perkalian diagonal:
Determinan = 1.6534391534393E-7
```

4.2 Invers Matriks

4.2.1 Kasus Uji 1

```
=== MENU UTAMA ===
1. Sistem Persamaan Linier (SPL)
2. Determinan
3. Matriks Balikan
4. Interpolasi
5. Regresi Polinomial
0. Keluar
Pilih menu: 3

--- MATRIKS BALIKAN ---
1. Metode OBE (Gauss?Jordan)
2. Metode Adjoin
Pilih metode: 1
1. Input manual
2. Baca dari file .txt
Pilih metode input: 1
Masukkan jumlah baris: 4
Masukkan jumlah kolom: 4
Masukkan elemen matriks (pisahkan dengan spasi):
5 2 -3 4
7 7 8 -9
12 13 2 14
17 18 19 4

===== Metode OBE (Gauss?Jordan) untuk Invers =====
|    5,000    2,000   -3,000    4,000 |
|    7,000    7,000    8,000   -9,000 |
|   12,000   13,000    2,000   14,000 |
|   17,000   18,000   19,000    4,000 |

 Matriks Augmentasi [A | I]:
|    5,000    2,000   -3,000    4,000    1,000    0,000    0,000    0,00
0 |
|    7,000    7,000    8,000   -9,000    0,000    1,000    0,000    0,00
0 |
|   12,000   13,000    2,000   14,000    0,000    0,000    1,000    0,00
0 |
|   17,000   18,000   19,000    4,000    0,000    0,000    0,000    1,00
0 |

Menukar baris R1 dengan R4 (pivoting)
```

```
Menukar baris R1 dengan R4 (pivoting)
|   17,000   18,000   19,000    4,000    0,000    0,000    0,000    1,00
0 |
|    7,000    7,000    8,000   -9,000    0,000    1,000    0,000    0,00
0 |
|   12,000   13,000    2,000   14,000    0,000    0,000    1,000    0,00
0 |
|    5,000    2,000   -3,000    4,000    1,000    0,000    0,000    0,00
0 |

Normalisasi pivot R1 dengan 17,0000000000
|    1,000    1,059    1,118    0,235    0,000    0,000    0,000    0,05
9 |
|    7,000    7,000    8,000   -9,000    0,000    1,000    0,000    0,00
0 |
|   12,000   13,000    2,000   14,000    0,000    0,000    1,000    0,00
0 |
|    5,000    2,000   -3,000    4,000    1,000    0,000    0,000    0,00
0 |
Eliminasi: R2 = R2 - (7,000000)*R1
|    1,000    1,059    1,118    0,235    0,000    0,000    0,000    0,05
9 |
|    0,000   -0,412    0,176  -10,647    0,000    1,000    0,000   -0,41
2 |
|   12,000   13,000    2,000   14,000    0,000    0,000    1,000    0,00
0 |
|    5,000    2,000   -3,000    4,000    1,000    0,000    0,000    0,00
0 |
Eliminasi: R3 = R3 - (12,000000)*R1
|    1,000    1,059    1,118    0,235    0,000    0,000    0,000    0,05
9 |
|    0,000   -0,412    0,176  -10,647    0,000    1,000    0,000   -0,41
2 |
|    0,000    0,294  -11,412   11,176    0,000    0,000    1,000   -0,70
6 |
|    5,000    2,000   -3,000    4,000    1,000    0,000    0,000    0,00
0 |
Eliminasi: R4 = R4 - (5,000000)*R1
|    1,000    1,059    1,118    0,235    0,000    0,000    0,000    0,05
9 |
|    0,000   -0,412    0,176  -10,647    0,000    1,000    0,000   -0,41
```

```
6 |
Eliminasi: R3 = R3 - (-0,938416)*R4
|    1,000    0,000    0,000    0,000    0,314   -0,041   -0,139    0,08
1 |
|    0,000    1,000    0,000    0,000   -0,303    0,162    0,231   -0,14
0 |
|   -0,000   -0,000    1,000    0,000    0,004   -0,095   -0,092    0,10
3 |
|   -0,000   -0,000   -0,000    1,000    0,012   -0,102   -0,010    0,04
6 |

===== Matriks telah direduksi menjadi [I | A??] =====
|    1,000    0,000    0,000    0,000    0,314   -0,041   -0,139    0,08
1 |
|    0,000    1,000    0,000    0,000   -0,303    0,162    0,231   -0,14
0 |
|   -0,000   -0,000    1,000    0,000    0,004   -0,095   -0,092    0,10
3 |
|   -0,000   -0,000   -0,000    1,000    0,012   -0,102   -0,010    0,04
6 |

 Matriks Invers (metode OBE):
|    0,314   -0,041   -0,139    0,081 |
|   -0,303    0,162    0,231   -0,140 |
|    0,004   -0,095   -0,092    0,103 |
|    0,012   -0,102   -0,010    0,046 |

 Invers Matriks:
|    0,314   -0,041   -0,139    0,081 |
|   -0,303    0,162    0,231   -0,140 |
|    0,004   -0,095   -0,092    0,103 |
|    0,012   -0,102   -0,010    0,046 |
 Hasil disimpan ke output/hasil.txt

=== MENU UTAMA ===
1. Sistem Persamaan Linier (SPL)
2. Determinan
3. Matriks Balikan
4. Interpolasi
5. Regresi Polinomial
0. Keluar
Pilih menu:
```

4.2.2 Kasus Uji 2

```
=== MENU UTAMA ===
1. Sistem Persamaan Linier (SPL)
2. Determinan
3. Matriks Balikan
4. Interpolasi
5. Regresi Polinomial
0. Keluar
Pilih menu: 3

--- MATRIKS BALIKAN ---
1. Metode OBE (Gauss?Jordan)
2. Metode Adjoin
Pilih metode: 1
1. Input manual
2. Baca dari file .txt
Pilih metode input: 1
Masukkan jumlah baris: 6
Masukkan jumlah kolom: 6
Masukkan elemen matriks (pisahkan dengan spasi):
5 7 9 11 13 15
10 14 18 22 26 30
15 21 27 33 39 45
20 28 36 44 52 60
25 35 45 55 65 75
30 42 54 66 78 90

===== Metode OBE (Gauss?Jordan) untuk Invers =====
|    5,000    7,000    9,000   11,000   13,000   15,000 |
|   10,000   14,000   18,000   22,000   26,000   30,000 |
|   15,000   21,000   27,000   33,000   39,000   45,000 |
|   20,000   28,000   36,000   44,000   52,000   60,000 |
|   25,000   35,000   45,000   55,000   65,000   75,000 |
|   30,000   42,000   54,000   66,000   78,000   90,000 |

 Matriks Augmentasi [A | I]:
|    5,000    7,000    9,000   11,000   13,000   15,000    1,000    0,00
0    0,000    0,000    0,000    0,000 |
|   10,000   14,000   18,000   22,000   26,000   30,000    0,000    1,00
0    0,000    0,000    0,000    0,000 |
|   15,000   21,000   27,000   33,000   39,000   45,000    0,000    0,00
0    1,000    0,000    0,000    0,000 |
```

```
 Matriks Augmentasi [A | I]:
|    5,000    7,000    9,000   11,000   13,000   15,000    1,000    0,00
0    0,000    0,000    0,000    0,000 |
|   10,000   14,000   18,000   22,000   26,000   30,000    0,000    1,00
0    0,000    0,000    0,000    0,000 |
|   15,000   21,000   27,000   33,000   39,000   45,000    0,000    0,00
0    1,000    0,000    0,000    0,000 |
|   20,000   28,000   36,000   44,000   52,000   60,000    0,000    0,00
0    0,000    1,000    0,000    0,000 |
|   25,000   35,000   45,000   55,000   65,000   75,000    0,000    0,00
0    0,000    0,000    1,000    0,000 |
|   30,000   42,000   54,000   66,000   78,000   90,000    0,000    0,00
0    0,000    0,000    0,000    1,000 |

Menukar baris R1 dengan R6 (pivoting)
|   30,000   42,000   54,000   66,000   78,000   90,000    0,000    0,00
0    0,000    0,000    0,000    1,000 |
|   10,000   14,000   18,000   22,000   26,000   30,000    0,000    1,00
0    0,000    0,000    0,000    0,000 |
|   15,000   21,000   27,000   33,000   39,000   45,000    0,000    0,00
0    1,000    0,000    0,000    0,000 |
|   20,000   28,000   36,000   44,000   52,000   60,000    0,000    0,00
0    0,000    1,000    0,000    0,000 |
|   25,000   35,000   45,000   55,000   65,000   75,000    0,000    0,00
0    0,000    0,000    1,000    0,000 |
|    5,000    7,000    9,000   11,000   13,000   15,000    1,000    0,00
0    0,000    0,000    0,000    0,000 |

Normalisasi pivot R1 dengan 30,0000000000
|    1,000    1,400    1,800    2,200    2,600    3,000    0,000    0,00
0    0,000    0,000    0,000    0,033 |
|   10,000   14,000   18,000   22,000   26,000   30,000    0,000    1,00
0    0,000    0,000    0,000    0,000 |
|   15,000   21,000   27,000   33,000   39,000   45,000    0,000    0,00
0    1,000    0,000    0,000    0,000 |
|   20,000   28,000   36,000   44,000   52,000   60,000    0,000    0,00
0    0,000    1,000    0,000    0,000 |
|   25,000   35,000   45,000   55,000   65,000   75,000    0,000    0,00
0    0,000    0,000    1,000    0,000 |
|    5,000    7,000    9,000   11,000   13,000   15,000    1,000    0,00
0    0,000    0,000    0,000    0,000 |
```

```
0    0,000    0,000    0,000    0,000 |
Eliminasi: R2 = R2 - (10,000000)*R1
|    1,000    1,400    1,800    2,200    2,600    3,000    0,000    0,00
0    0,000    0,000    0,000    0,033 |
|    0,000    0,000    0,000    0,000    0,000    0,000    0,000    1,00
0    0,000    0,000    0,000   -0,333 |
|   15,000   21,000   27,000   33,000   39,000   45,000    0,000    0,00
0    1,000    0,000    0,000    0,000 |
|   20,000   28,000   36,000   44,000   52,000   60,000    0,000    0,00
0    0,000    1,000    0,000    0,000 |
|   25,000   35,000   45,000   55,000   65,000   75,000    0,000    0,00
0    0,000    0,000    1,000    0,000 |
|    5,000    7,000    9,000   11,000   13,000   15,000    1,000    0,00
0    0,000    0,000    0,000    0,000 |
Eliminasi: R3 = R3 - (15,000000)*R1
|    1,000    1,400    1,800    2,200    2,600    3,000    0,000    0,00
0    0,000    0,000    0,000    0,033 |
|    0,000    0,000    0,000    0,000    0,000    0,000    0,000    1,00
0    0,000    0,000    0,000   -0,333 |
|    0,000    0,000    0,000    0,000    0,000    0,000    0,000    0,00
0    1,000    0,000    0,000   -0,500 |
|   20,000   28,000   36,000   44,000   52,000   60,000    0,000    0,00
0    0,000    1,000    0,000    0,000 |
|   25,000   35,000   45,000   55,000   65,000   75,000    0,000    0,00
0    0,000    0,000    1,000    0,000 |
|    5,000    7,000    9,000   11,000   13,000   15,000    1,000    0,00
0    0,000    0,000    0,000    0,000 |
Eliminasi: R4 = R4 - (20,000000)*R1
|    1,000    1,400    1,800    2,200    2,600    3,000    0,000    0,00
0    0,000    0,000    0,000    0,033 |
|    0,000    0,000    0,000    0,000    0,000    0,000    0,000    1,00
0    0,000    0,000    0,000   -0,333 |
|    0,000    0,000    0,000    0,000    0,000    0,000    0,000    0,00
0    1,000    0,000    0,000   -0,500 |
|    0,000    0,000    0,000    0,000    0,000    0,000    0,000    0,00
0    0,000    1,000    0,000   -0,667 |
|   25,000   35,000   45,000   55,000   65,000   75,000    0,000    0,00
0    0,000    0,000    1,000    0,000 |
|    5,000    7,000    9,000   11,000   13,000   15,000    1,000    0,00
0    0,000    0,000    0,000    0,000 |
Eliminasi: R5 = R5 - (25,000000)*R1
|    1,000    1,400    1,800    2,200    2,600    3,000    0,000    0,00
```

```
|    5,000    7,000    9,000   11,000   13,000   15,000    1,000    0,00
0    0,000    0,000    0,000    0,000 |
Eliminasi: R5 = R5 - (25,000000)*R1
|    1,000    1,400    1,800    2,200    2,600    3,000    0,000    0,00
0    0,000    0,000    0,000    0,033 |
|    0,000    0,000    0,000    0,000    0,000    0,000    0,000    1,00
0    0,000    0,000    0,000   -0,333 |
|    0,000    0,000    0,000    0,000    0,000    0,000    0,000    0,00
0    1,000    0,000    0,000   -0,500 |
|    0,000    0,000    0,000    0,000    0,000    0,000    0,000    0,00
0    0,000    1,000    0,000   -0,667 |
|    0,000    0,000    0,000   -0,000    0,000    0,000    0,000    0,00
0    0,000    0,000    1,000   -0,833 |
|    5,000    7,000    9,000   11,000   13,000   15,000    1,000    0,00
0    0,000    0,000    0,000    0,000 |
Eliminasi: R6 = R6 - (5,000000)*R1
|    1,000    1,400    1,800    2,200    2,600    3,000    0,000    0,00
0    0,000    0,000    0,000    0,033 |
|    0,000    0,000    0,000    0,000    0,000    0,000    0,000    1,00
0    0,000    0,000    0,000   -0,333 |
|    0,000    0,000    0,000    0,000    0,000    0,000    0,000    0,00
0    1,000    0,000    0,000   -0,500 |
|    0,000    0,000    0,000    0,000    0,000    0,000    0,000    0,00
0    0,000    1,000    0,000   -0,667 |
|    0,000    0,000    0,000   -0,000    0,000    0,000    0,000    0,00
0    0,000    0,000    1,000   -0,833 |
|    0,000    0,000    0,000    0,000    0,000    0,000    1,000    0,00
0    0,000    0,000    0,000   -0,167 |

Kolom 2 tidak memiliki pivot. Matriks singular.

 Matriks tidak memiliki invers unik (singular).
 Hasil disimpan ke output/hasil.txt

=== MENU UTAMA ===
1. Sistem Persamaan Linier (SPL)
2. Determinan
3. Matriks Balikan
4. Interpolasi
5. Regresi Polinomial
0. Keluar
Pilih menu:
```

4.3 Sistem Persamaan Linear

4.3.1 Kasus Uji 1

```
1. Input manual
2. Baca dari file .txt
Pilih metode input: 1
Masukkan jumlah baris: 4
Masukkan jumlah kolom: 5
Masukkan elemen matriks (pisahkan dengan spasi):
1 1 -1 -1 1
2 5 -7 -5 -2
2 -1 1 3 4
5 2 -4 2 6

===== Memulai Forward Elimination (Gauss) =====
Matriks Awal:
|    1.000    1.000   -1.000   -1.000    1.000 |
|    2.000    5.000   -7.000   -5.000   -2.000 |
|    2.000   -1.000    1.000    3.000    4.000 |
|    5.000    2.000   -4.000    2.000    6.000 |
------------------------------------


Langkah: Eliminasi di R2, K1 -> R2 = R2 - (2.000) * R1
|    1.000    1.000   -1.000   -1.000    1.000 |
|    0.000    3.000   -5.000   -3.000   -4.000 |
|    2.000   -1.000    1.000    3.000    4.000 |
|    5.000    2.000   -4.000    2.000    6.000 |

Langkah: Eliminasi di R3, K1 -> R3 = R3 - (2.000) * R1
|    1.000    1.000   -1.000   -1.000    1.000 |
|    0.000    3.000   -5.000   -3.000   -4.000 |
|    0.000   -3.000    3.000    5.000    2.000 |
|    5.000    2.000   -4.000    2.000    6.000 |

Langkah: Eliminasi di R4, K1 -> R4 = R4 - (5.000) * R1
|    1.000    1.000   -1.000   -1.000    1.000 |
```

```
Langkah: Eliminasi di R4, K1 -> R4 = R4 - (5.000) * R1
|    1.000    1.000   -1.000   -1.000    1.000 |
|    0.000    3.000   -5.000   -3.000   -4.000 |
|    0.000   -3.000    3.000    5.000    2.000 |
|    0.000   -3.000    1.000    7.000    1.000 |

Langkah: Eliminasi di R3, K2 -> R3 = R3 - (-1.000) * R2
|    1.000    1.000   -1.000   -1.000    1.000 |
|    0.000    3.000   -5.000   -3.000   -4.000 |
|    0.000    0.000   -2.000    2.000   -2.000 |
|    0.000   -3.000    1.000    7.000    1.000 |

Langkah: Eliminasi di R4, K2 -> R4 = R4 - (-1.000) * R2
|    1.000    1.000   -1.000   -1.000    1.000 |
|    0.000    3.000   -5.000   -3.000   -4.000 |
|    0.000    0.000   -2.000    2.000   -2.000 |
|    0.000    0.000   -4.000    4.000   -3.000 |

Langkah: Eliminasi di R4, K3 -> R4 = R4 - (2.000) * R3
|    1.000    1.000   -1.000   -1.000    1.000 |
|    0.000    3.000   -5.000   -3.000   -4.000 |
|    0.000    0.000   -2.000    2.000   -2.000 |
|    0.000    0.000    0.000    0.000    1.000 |

===== Forward Elimination Selesai, Matriks Eselon Baris Terbentuk =====
|    0.000    3.000   -5.000   -3.000   -4.000 |
|    0.000    0.000   -2.000    2.000   -2.000 |
|    0.000    0.000    0.000    0.000    1.000 |
```

```
===== Memulai Back Substitution =====
Menghitung x4:
  Persamaan dari baris 4: (0.00x4) = 1.00
  x4 = (1.000 - 0.000) / 0.000 = Infinity
Menghitung x3:
  Persamaan dari baris 3: (-2.00x3) + (2.00x4) = -2.00
  x3 = (-2.000 - Infinity) / -2.000 = Infinity
Menghitung x2:
  Persamaan dari baris 2: (3.00x2) + (-5.00x3) + (-3.00x4) = -4.00
  x2 = (-4.000 - -Infinity) / 3.000 = Infinity
Menghitung x1:
  Persamaan dari baris 1: (1.00x1) + (1.00x2) + (-1.00x3) + (-1.00x4) = 1.00
  x1 = (1.000 - NaN) / 1.000 = NaN
===== Back Substitution Selesai =====

Hasil SPL:
x1 = NaN
x2 = Infinity
x3 = Infinity
x4 = Infinity

 Hasil disimpan ke output/hasil.txt
```

4.3.2 Kasus Uji 2

```
=== MENU UTAMA ===
1. Sistem Persamaan Linier (SPL)
2. Determinan
3. Matriks Balikan
4. Interpolasi
5. Regresi Polinomial
0. Keluar
Pilih menu: 1

--- SISTEM PERSAMAAN LINEAR ---
1. Eliminasi Gauss
2. Eliminasi Gauss-Jordan
3. Kaidah Cramer
4. Metode Matriks Balikan
Pilih metode: 2
1. Input manual
2. Baca dari file .txt
Pilih metode input: 1
Masukkan jumlah baris: 4
Masukkan jumlah kolom: 6
Masukkan elemen matriks (pisahkan dengan spasi):
1 -1 0 0 1 3
1 1 0 -3 0 6
2 -1 0 1 -1 5
-1 2 0 -2 -1 -1

==== Memulai Eliminasi Gauss Jordan =====
Matriks Awal:
|    1.000   -1.000    0.000    0.000    1.000    3.000 |
|    1.000    1.000    0.000   -3.000    0.000    6.000 |
|    2.000   -1.000    0.000    1.000   -1.000    5.000 |
|   -1.000    2.000    0.000   -2.000   -1.000   -1.000 |
------------------------------------
```

```
Langkah: Eliminasi di R2, K1 -> R2 = R2 - (1.000) * R1
|    1.000   -1.000    0.000    0.000    1.000    3.000 |
|    0.000    2.000    0.000   -3.000   -1.000    3.000 |
|    2.000   -1.000    0.000    1.000   -1.000    5.000 |
|   -1.000    2.000    0.000   -2.000   -1.000   -1.000 |

Langkah: Eliminasi di R3, K1 -> R3 = R3 - (2.000) * R1
|    1.000   -1.000    0.000    0.000    1.000    3.000 |
|    0.000    2.000    0.000   -3.000   -1.000    3.000 |
|    0.000    1.000    0.000    1.000   -3.000   -1.000 |
|   -1.000    2.000    0.000   -2.000   -1.000   -1.000 |

Langkah: Eliminasi di R4, K1 -> R4 = R4 - (-1.000) * R1
|    1.000   -1.000    0.000    0.000    1.000    3.000 |
|    0.000    2.000    0.000   -3.000   -1.000    3.000 |
|    0.000    1.000    0.000    1.000   -3.000   -1.000 |
|    0.000    1.000    0.000   -2.000    0.000    2.000 |

Normalisasi baris R2 -> R2 = R2 / 2.000
|    1.000   -1.000    0.000    0.000    1.000    3.000 |
|    0.000    1.000    0.000   -1.500   -0.500    1.500 |
|    0.000    1.000    0.000    1.000   -3.000   -1.000 |
|    0.000    1.000    0.000   -2.000    0.000    2.000 |

Langkah: Eliminasi di R1, K2 -> R1 = R1 - (-1.000) * R2
|    1.000    0.000    0.000   -1.500    0.500    4.500 |
|    0.000    1.000    0.000   -1.500   -0.500    1.500 |
|    0.000    1.000    0.000    1.000   -3.000   -1.000 |
|    0.000    1.000    0.000   -2.000    0.000    2.000 |
```

```
Langkah: Eliminasi di R3, K2 -> R3 = R3 - (1.000) * R2
|    1.000    0.000    0.000   -1.500    0.500    4.500 |
|    0.000    1.000    0.000   -1.500   -0.500    1.500 |
|    0.000    0.000    0.000    2.500   -2.500   -2.500 |
|    0.000    1.000    0.000   -2.000    0.000    2.000 |

Langkah: Eliminasi di R4, K2 -> R4 = R4 - (1.000) * R2
|    1.000    0.000    0.000   -1.500    0.500    4.500 |
|    0.000    1.000    0.000   -1.500   -0.500    1.500 |
|    0.000    0.000    0.000    2.500   -2.500   -2.500 |
|    0.000    0.000    0.000   -0.500    0.500    0.500 |

Normalisasi baris R3 -> R3 = R3 / 2.500
|    1.000    0.000    0.000   -1.500    0.500    4.500 |
|    0.000    1.000    0.000   -1.500   -0.500    1.500 |
|    0.000    0.000    0.000    1.000   -1.000   -1.000 |
|    0.000    0.000    0.000   -0.500    0.500    0.500 |

Langkah: Eliminasi di R1, K4 -> R1 = R1 - (-1.500) * R3
|    1.000    0.000    0.000    0.000   -1.000    3.000 |
|    0.000    1.000    0.000   -1.500   -0.500    1.500 |
|    0.000    0.000    0.000    1.000   -1.000   -1.000 |
|    0.000    0.000    0.000   -0.500    0.500    0.500 |

Langkah: Eliminasi di R2, K4 -> R2 = R2 - (-1.500) * R3
|    1.000    0.000    0.000    0.000   -1.000    3.000 |
|    0.000    1.000    0.000    0.000   -2.000    0.000 |
|    0.000    0.000    0.000    1.000   -1.000   -1.000 |
|    0.000    0.000    0.000   -0.500    0.500    0.500 |
```

```
Langkah: Eliminasi di R4, K4 -> R4 = R4 - (-0.500) * R3
|    1.000    0.000    0.000    0.000   -1.000    3.000 |
|    0.000    1.000    0.000    0.000   -2.000    0.000 |
|    0.000    0.000    0.000    1.000   -1.000   -1.000 |
|    0.000    0.000    0.000    0.000    0.000    0.000 |

===== Proses Eliminasi Selssai =====

Hasil SPL:
Banyak Solusi (Solusi Parametrik):
x1 = 3.000 + 1.000s
x2 = 2.000s
x3 = t
x4 = -1.000 + 1.000s
x5 = s

 Hasil disimpan ke output/hasil.txt
```

4.3.3 Kasus Uji 3

```
=== MENU UTAMA ===
1. Sistem Persamaan Linier (SPL)
2. Determinan
3. Matriks Balikan
4. Interpolasi
5. Regresi Polinomial
0. Keluar
Pilih menu: 1

--- SISTEM PERSAMAAN LINEAR ---
1. Eliminasi Gauss
2. Eliminasi Gauss-Jordan
3. Kaidah Cramer
4. Metode Matriks Balikan
Pilih metode: 2
1. Input manual
2. Baca dari file .txt
Pilih metode input: 1
Masukkan jumlah baris: 3
Masukkan jumlah kolom: 7
Masukkan elemen matriks (pisahkan dengan spasi):
1 -1 0 0 1 3
1 1 0 -3 0 6
2 -1 0 1 -1 5
-1 2 0 -2 -1 -1
```

```
==== Memulai Eliminasi Gauss Jordan =====
Matriks Awal:
|    1.000   -1.000    0.000    0.000    1.000    3.000    1.000 |
|    1.000    0.000   -3.000    0.000    6.000    2.000   -1.000 |
|    0.000    1.000   -1.000    5.000   -1.000    2.000    0.000 |
------------------------------------

Langkah: Eliminasi di R2, K1 -> R2 = R2 - (1.000) * R1
|    1.000   -1.000    0.000    0.000    1.000    3.000    1.000 |
|    0.000    1.000   -3.000    0.000    5.000   -1.000   -2.000 |
|    0.000    1.000   -1.000    5.000   -1.000    2.000    0.000 |

Langkah: Eliminasi di R1, K2 -> R1 = R1 - (-1.000) * R2
|    1.000    0.000   -3.000    0.000    6.000    2.000   -1.000 |
|    0.000    1.000   -3.000    0.000    5.000   -1.000   -2.000 |
|    0.000    1.000   -1.000    5.000   -1.000    2.000    0.000 |

Langkah: Eliminasi di R3, K2 -> R3 = R3 - (1.000) * R2
|    1.000    0.000   -3.000    0.000    6.000    2.000   -1.000 |
|    0.000    1.000   -3.000    0.000    5.000   -1.000   -2.000 |
|    0.000    0.000    2.000    5.000   -6.000    3.000    2.000 |

Normalisasi baris R3 -> R3 = R3 / 2.000
|    1.000    0.000   -3.000    0.000    6.000    2.000   -1.000 |
|    0.000    1.000   -3.000    0.000    5.000   -1.000   -2.000 |
|    0.000    0.000    1.000    2.500   -3.000    1.500    1.000 |

Langkah: Eliminasi di R1, K3 -> R1 = R1 - (-3.000) * R3
|    1.000    0.000    0.000    7.500   -3.000    6.500    2.000 |
|    0.000    1.000   -3.000    0.000    5.000   -1.000   -2.000 |
|    0.000    0.000    1.000    2.500   -3.000    1.500    1.000 |
```

```
Langkah: Eliminasi di R2, K3 -> R2 = R2 - (-3.000) * R3
|    1.000    0.000    0.000    7.500   -3.000    6.500    2.000 |
|    0.000    1.000    0.000    7.500   -4.000    3.500    1.000 |
|    0.000    0.000    1.000    2.500   -3.000    1.500    1.000 |

===== Proses Eliminasi Selssai =====

Hasil SPL:
Banyak Solusi (Solusi Parametrik):
x1 = 2.000 - 7.500t + 3.000s - 6.500r
x2 = 1.000 - 7.500t + 4.000s - 3.500r
x3 = 1.000 - 2.500t + 3.000s - 1.500r
x4 = t
x5 = s
x6 = r

 Hasil disimpan ke output/hasil.txt
```

4.4 Sistem Persamaan Linear Berbentuk Matriks Augmented

4.4.1 Kasus Uji 1

```
=== MENU UTAMA ===
1. Sistem Persamaan Linier (SPL)
2. Determinan
3. Matriks Balikan
4. Interpolasi
5. Regresi Polinomial
0. Keluar
Pilih menu: 1

--- SISTEM PERSAMAAN LINEAR ---
1. Eliminasi Gauss
2. Eliminasi Gauss-Jordan
3. Kaidah Cramer
4. Metode Matriks Balikan
Pilih metode: 2
1. Input manual
2. Baca dari file .txt
Pilih metode input: 1
Masukkan jumlah baris: 4
Masukkan jumlah kolom: 5
Masukkan elemen matriks (pisahkan dengan spasi):
1 -1 2 -1 -1
2 1 -2 -2 -2
-1 2 -4 1 1
3 0 0 -3 -3
```

```
==== Memulai Eliminasi Gauss Jordan =====
Matriks Awal:
|    1.000   -1.000    2.000   -1.000   -1.000 |
|    2.000    1.000   -2.000   -2.000   -2.000 |
|   -1.000    2.000   -4.000    1.000    1.000 |
|    3.000    0.000    0.000   -3.000   -3.000 |
----------------------------------------

Langkah: Eliminasi di R2, K1 -> R2 = R2 - (2.000) * R1
|    1.000   -1.000    2.000   -1.000   -1.000 |
|    0.000    3.000   -6.000    0.000    0.000 |
|   -1.000    2.000   -4.000    1.000    1.000 |
|    3.000    0.000    0.000   -3.000   -3.000 |

Langkah: Eliminasi di R3, K1 -> R3 = R3 - (-1.000) * R1
|    1.000   -1.000    2.000   -1.000   -1.000 |
|    0.000    3.000   -6.000    0.000    0.000 |
|    0.000    1.000   -2.000    0.000    0.000 |
|    3.000    0.000    0.000   -3.000   -3.000 |

Langkah: Eliminasi di R4, K1 -> R4 = R4 - (3.000) * R1
|    1.000   -1.000    2.000   -1.000   -1.000 |
|    0.000    3.000   -6.000    0.000    0.000 |
|    0.000    1.000   -2.000    0.000    0.000 |
|    0.000    3.000   -6.000    0.000    0.000 |

Normalisasi baris R2 -> R2 = R2 / 3.000
|    1.000   -1.000    2.000   -1.000   -1.000 |
|    0.000    1.000   -2.000    0.000    0.000 |
|    0.000    1.000   -2.000    0.000    0.000 |
|    0.000    3.000   -6.000    0.000    0.000 |
```

```
Langkah: Eliminasi di R1, K2 -> R1 = R1 - (-1.000) * R2
|    1.000    0.000    0.000   -1.000   -1.000 |
|    0.000    1.000   -2.000    0.000    0.000 |
|    0.000    1.000   -2.000    0.000    0.000 |
|    0.000    3.000   -6.000    0.000    0.000 |

Langkah: Eliminasi di R3, K2 -> R3 = R3 - (1.000) * R2
|    1.000    0.000    0.000   -1.000   -1.000 |
|    0.000    1.000   -2.000    0.000    0.000 |
|    0.000    0.000    0.000    0.000    0.000 |
|    0.000    3.000   -6.000    0.000    0.000 |

Langkah: Eliminasi di R4, K2 -> R4 = R4 - (3.000) * R2
|    1.000    0.000    0.000   -1.000   -1.000 |
|    0.000    1.000   -2.000    0.000    0.000 |
|    0.000    0.000    0.000    0.000    0.000 |
|    0.000    0.000    0.000    0.000    0.000 |

===== Proses Eliminasi Selssai =====

Hasil SPL:
Banyak Solusi (Solusi Parametrik):
x1 = -1.000 + 1.000s
x2 = 2.000t
x3 = t
x4 = s

 Hasil disimpan ke output/hasil.txt
```

4.4.2 Kasus Uji 2

```
=== MENU UTAMA ===
1. Sistem Persamaan Linier (SPL)
2. Determinan
3. Matriks Balikan
4. Interpolasi
5. Regresi Polinomial
0. Keluar
Pilih menu: 1

--- SISTEM PERSAMAAN LINEAR ---
1. Eliminasi Gauss
2. Eliminasi Gauss-Jordan
3. Kaidah Cramer
4. Metode Matriks Balikan
Pilih metode: 2
1. Input manual
2. Baca dari file .txt
Pilih metode input: 1
Masukkan jumlah baris: 6
Masukkan jumlah kolom: 5
Masukkan elemen matriks (pisahkan dengan spasi):
2 0 8 0 8
0 1 0 4 4
-4 0 6 0 6
0 -2 0 3 -1
2 0 -4 0 -4
0 1 0 -2 0
```

```
==== Memulai Eliminasi Gauss Jordan =====
Matriks Awal:
|    2.000    0.000    8.000    0.000    8.000 |
|    0.000    1.000    0.000    4.000    4.000 |
|   -4.000    0.000    6.000    0.000    6.000 |
|    0.000   -2.000    0.000    3.000   -1.000 |
|    2.000    0.000   -4.000    0.000   -4.000 |
|    0.000    1.000    0.000   -2.000    0.000 |
------------------------------------

Normalisasi baris R1 -> R1 = R1 / 2.000
|    1.000    0.000    4.000    0.000    4.000 |
|    0.000    1.000    0.000    4.000    4.000 |
|   -4.000    0.000    6.000    0.000    6.000 |
|    0.000   -2.000    0.000    3.000   -1.000 |
|    2.000    0.000   -4.000    0.000   -4.000 |
|    0.000    1.000    0.000   -2.000    0.000 |

Langkah: Eliminasi di R3, K1 -> R3 = R3 - (-4.000) * R1
|    1.000    0.000    4.000    0.000    4.000 |
|    0.000    1.000    0.000    4.000    4.000 |
|    0.000    0.000   22.000    0.000   22.000 |
|    0.000   -2.000    0.000    3.000   -1.000 |
|    2.000    0.000   -4.000    0.000   -4.000 |
|    0.000    1.000    0.000   -2.000    0.000 |

Langkah: Eliminasi di R5, K1 -> R5 = R5 - (2.000) * R1
|    1.000    0.000    4.000    0.000    4.000 |
|    0.000    1.000    0.000    4.000    4.000 |
|    0.000    0.000   22.000    0.000   22.000 |
|    0.000   -2.000    0.000    3.000   -1.000 |
|    0.000    0.000  -12.000    0.000  -12.000 |
|    0.000    1.000    0.000   -2.000    0.000 |
```

```
Langkah: Eliminasi di R4, K2 -> R4 = R4 - (-2.000) * R2
|    1.000    0.000    4.000    0.000    4.000 |
|    0.000    1.000    0.000    4.000    4.000 |
|    0.000    0.000   22.000    0.000   22.000 |
|    0.000    0.000    0.000   11.000    7.000 |
|    0.000    0.000  -12.000    0.000  -12.000 |
|    0.000    1.000    0.000   -2.000    0.000 |

Langkah: Eliminasi di R6, K2 -> R6 = R6 - (1.000) * R2
|    1.000    0.000    4.000    0.000    4.000 |
|    0.000    1.000    0.000    4.000    4.000 |
|    0.000    0.000   22.000    0.000   22.000 |
|    0.000    0.000    0.000   11.000    7.000 |
|    0.000    0.000  -12.000    0.000  -12.000 |
|    0.000    0.000    0.000   -6.000   -4.000 |

Normalisasi baris R3 -> R3 = R3 / 22.000
|    1.000    0.000    4.000    0.000    4.000 |
|    0.000    1.000    0.000    4.000    4.000 |
|    0.000    0.000    1.000    0.000    1.000 |
|    0.000    0.000    0.000   11.000    7.000 |
|    0.000    0.000  -12.000    0.000  -12.000 |
|    0.000    0.000    0.000   -6.000   -4.000 |

Langkah: Eliminasi di R1, K3 -> R1 = R1 - (4.000) * R3
|    1.000    0.000    0.000    0.000    0.000 |
|    0.000    1.000    0.000    4.000    4.000 |
|    0.000    0.000    1.000    0.000    1.000 |
|    0.000    0.000    0.000   11.000    7.000 |
|    0.000    0.000  -12.000    0.000  -12.000 |
|    0.000    0.000    0.000   -6.000   -4.000 |
```

```
Langkah: Eliminasi di R5, K3 -> R5 = R5 - (-12.000) * R3
|    1.000    0.000    0.000    0.000    0.000 |
|    0.000    1.000    0.000    4.000    4.000 |
|    0.000    0.000    1.000    0.000    1.000 |
|    0.000    0.000    0.000   11.000    7.000 |
|    0.000    0.000    0.000    0.000    0.000 |
|    0.000    0.000    0.000   -6.000   -4.000 |

Normalisasi baris R4 -> R4 = R4 / 11.000
|    1.000    0.000    0.000    0.000    0.000 |
|    0.000    1.000    0.000    4.000    4.000 |
|    0.000    0.000    1.000    0.000    1.000 |
|    0.000    0.000    0.000    1.000    0.636 |
|    0.000    0.000    0.000    0.000    0.000 |
|    0.000    0.000    0.000   -6.000   -4.000 |

Langkah: Eliminasi di R2, K4 -> R2 = R2 - (4.000) * R4
|    1.000    0.000    0.000    0.000    0.000 |
|    0.000    1.000    0.000    0.000    1.455 |
|    0.000    0.000    1.000    0.000    1.000 |
|    0.000    0.000    0.000    1.000    0.636 |
|    0.000    0.000    0.000    0.000    0.000 |
|    0.000    0.000    0.000   -6.000   -4.000 |

Langkah: Eliminasi di R6, K4 -> R6 = R6 - (-6.000) * R4
|    1.000    0.000    0.000    0.000    0.000 |
|    0.000    1.000    0.000    0.000    1.455 |
|    0.000    0.000    1.000    0.000    1.000 |
|    0.000    0.000    0.000    1.000    0.636 |
|    0.000    0.000    0.000    0.000    0.000 |
|    0.000    0.000    0.000    0.000   -0.182 |
```

```
===== Proses Eliminasi Selssai =====

Hasil SPL:
Sistem persamaan linear tidak memiliki solusi.

 Hasil disimpan ke output/hasil.txt
```

4.5 Sistem Persamaan Linier Berbentuk Umum

4.5.1 Kasus Uji 1

4.5.2 Kasus Uji 2

4.6 Aplikasi Sistem Persamaan Linier

4.6.1 Input

```
=== MENU UTAMA ===
1. Sistem Persamaan Linier (SPL)
2. Determinan
3. Matriks Balikan
4. Interpolasi
5. Regresi Polinomial
0. Keluar
Pilih menu: 1

--- SISTEM PERSAMAAN LINEAR ---
1. Eliminasi Gauss
2. Eliminasi Gauss-Jordan
3. Kaidah Cramer
4. Metode Matriks Balikan
Pilih metode: 2
1. Input manual
2. Baca dari file .txt
Pilih metode input: 1
Masukkan jumlah baris: 3
Masukkan jumlah kolom: 4
Masukkan elemen matriks (pisahkan dengan spasi):
-120 60 0 -1300
40 -80 0 0
80 20 -250 -200
```

4.6.2 Output

```
==== Memulai Eliminasi Gauss Jordan =====
Matriks Awal:
| -120.000   60.000    0.000 -1300.000 |
|   40.000  -80.000    0.000    0.000 |
|   80.000   20.000 -250.000 -200.000 |
------------------------------------

Normalisasi baris R1 -> R1 = R1 / -120.000
|    1.000   -0.500   -0.000   10.833 |
|   40.000  -80.000    0.000    0.000 |
|   80.000   20.000 -250.000 -200.000 |

Langkah: Eliminasi di R2, K1 -> R2 = R2 - (40.000) * R1
|    1.000   -0.500   -0.000   10.833 |
|    0.000  -60.000    0.000 -433.333 |
|   80.000   20.000 -250.000 -200.000 |

Langkah: Eliminasi di R3, K1 -> R3 = R3 - (80.000) * R1
|    1.000   -0.500   -0.000   10.833 |
|    0.000  -60.000    0.000 -433.333 |
|    0.000   60.000 -250.000 -1066.667 |
```

```
Normalisasi baris R2 -> R2 = R2 / -60.000
|     1.000   -0.500   -0.000   10.833 |
|    -0.000    1.000   -0.000    7.222 |
|     0.000   60.000 -250.000 -1066.667 |

Langkah: Eliminasi di R1, K2 -> R1 = R1 - (-0.500) * R2
|     1.000    0.000   -0.000   14.444 |
|    -0.000    1.000   -0.000    7.222 |
|     0.000   60.000 -250.000 -1066.667 |

Langkah: Eliminasi di R3, K2 -> R3 = R3 - (60.000) * R2
|     1.000    0.000   -0.000   14.444 |
|    -0.000    1.000   -0.000    7.222 |
|     0.000    0.000 -250.000 -1500.000 |

Normalisasi baris R3 -> R3 = R3 / -250.000
|     1.000    0.000   -0.000   14.444 |
|    -0.000    1.000   -0.000    7.222 |
|    -0.000   -0.000    1.000    6.000 |
```

```
===== Proses Eliminasi Selssai =====

Hasil SPL:
Solusi Unik:
x1 = 14.444444
x2 = 7.222222
x3 = 6.000000

 Hasil disimpan ke output/hasil.txt
```

4.7 Studi Kasus Interpolasi Polinomial

4.7.1 Studi Kasus 1

```
=== MENU UTAMA ===
1. Sistem Persamaan Linier (SPL)
2. Determinan
3. Matriks Balikan
4. Interpolasi
5. Regresi Polinomial
0. Keluar
Pilih menu: Input tidak valid, coba lagi: 4

--- INTERPOLASI ---
1. Interpolasi Polinomial
2. Interpolasi Splina Bézier Kubik
Pilih metode: 1
Masukkan jumlah titik data: 7
x1 y1: 0,1 0,003
x2 y2: 0,3 0,067
x3 y3: 0,5 0,148
x4 y4: 0,7 0,248
x5 y5: 0,9 0,370
x6 y6: 1,1 0,518
x7 y7: 1,3 0,697
=== Langkah-langkah Interpolasi Polinomial ===
1. Membentuk matriks Vandermonde berdasarkan titik x:
|    1,000    0,100    0,010    0,001    0,000    0,000    0,000 |
|    1,000    0,300    0,090    0,027    0,008    0,002    0,001 |
|    1,000    0,500    0,250    0,125    0,063    0,031    0,016 |
|    1,000    0,700    0,490    0,343    0,240    0,168    0,118 |
|    1,000    0,900    0,810    0,729    0,656    0,590    0,531 |
|    1,000    1,100    1,210    1,331    1,464    1,611    1,772 |
|    1,000    1,300    1,690    2,197    2,856    3,713    4,827 |
2. Membentuk matriks Y berdasarkan titik y:
|    0,003 |
|    0,067 |
|    0,148 |
|    0,248 |
|    0,370 |
|    0,518 |
|    0,697 |
3. Menghitung invers dari matriks Vandermonde...
```

```
===== Metode OBE (Gauss?Jordan) untuk Invers =====
|     1,000    0,100    0,010    0,001    0,000    0,000    0,000 |
|     1,000    0,300    0,090    0,027    0,008    0,002    0,001 |
|     1,000    0,500    0,250    0,125    0,063    0,031    0,016 |
|     1,000    0,700    0,490    0,343    0,240    0,168    0,118 |
|     1,000    0,900    0,810    0,729    0,656    0,590    0,531 |
|     1,000    1,100    1,210    1,331    1,464    1,611    1,772 |
|     1,000    1,300    1,690    2,197    2,856    3,713    4,827 |

 Matriks Augmentasi [A | I]:
|     1,000    0,100    0,010    0,001    0,000    0,000    0,000    1,00
0    0,000    0,000    0,000    0,000    0,000    0,000 |
|     1,000    0,300    0,090    0,027    0,008    0,002    0,001    0,00
0    1,000    0,000    0,000    0,000    0,000    0,000 |
|     1,000    0,500    0,250    0,125    0,063    0,031    0,016    0,00
0    0,000    1,000    0,000    0,000    0,000    0,000 |
|     1,000    0,700    0,490    0,343    0,240    0,168    0,118    0,00
0    0,000    0,000    1,000    0,000    0,000    0,000 |
|     1,000    0,900    0,810    0,729    0,656    0,590    0,531    0,00
0    0,000    0,000    0,000    1,000    0,000    0,000 |
|     1,000    1,100    1,210    1,331    1,464    1,611    1,772    0,00
0    0,000    0,000    0,000    0,000    1,000    0,000 |
|     1,000    1,300    1,690    2,197    2,856    3,713    4,827    0,00
0    0,000    0,000    0,000    0,000    0,000    1,000 |

Normalisasi pivot R1 dengan 1,0000000000
|     1,000    0,100    0,010    0,001    0,000    0,000    0,000    1,00
0    0,000    0,000    0,000    0,000    0,000    0,000 |
|     1,000    0,300    0,090    0,027    0,008    0,002    0,001    0,00
0    1,000    0,000    0,000    0,000    0,000    0,000 |
|     1,000    0,500    0,250    0,125    0,063    0,031    0,016    0,00
0    0,000    1,000    0,000    0,000    0,000    0,000 |
|     1,000    0,700    0,490    0,343    0,240    0,168    0,118    0,00
0    0,000    0,000    1,000    0,000    0,000    0,000 |
|     1,000    0,900    0,810    0,729    0,656    0,590    0,531    0,00
0    0,000    0,000    0,000    1,000    0,000    0,000 |
|     1,000    1,100    1,210    1,331    1,464    1,611    1,772    0,00
0    0,000    0,000    0,000    0,000    1,000    0,000 |
|     1,000    1,300    1,690    2,197    2,856    3,713    4,827    0,00
0    0,000    0,000    0,000    0,000    0,000    1,000 |
Eliminasi: R2 = R2 - (1,000000)*R1
```

```
|  200,738 -1087,240 2451,172 -2947,049 1995,443 -722,656  109,592 |
| -104,167  598,958 -1432,292 1822,917 -1302,083  494,792  -78,125 |
|   21,701 -130,208  325,521 -434,028  325,521 -130,208   21,701 |
4. Mengalikan A?? dengan Y untuk mendapatkan koefisien polinomial:
|   -0,023 |
|    0,240 |
|    0,197 |
|    0,000 |
|    0,026 |
|    0,000 |
|   -0,000 |
=== Selesai ===


Koefisien Polinomial:
|   -0,023 |
|    0,240 |
|    0,197 |
|    0,000 |
|    0,026 |
|    0,000 |
|   -0,000 |
Pn(x) = -0,0230 + (0,2400)x^1 + (0,1974)x^2 + (0,0000)x^3 + (0,0260)x^4
+ (0,0000)x^5 + (-0,0000)x^6

Masukkan nilai x yang ingin dihitung: 0,2
P(0,2000) = 0,032961

[!] Terjadi kesalahan dalam pemrosesan: Index 4 out of bounds for length
 4
Pastikan input Anda (misal: dimensi matriks) sesuai untuk operasi yang d
ipilih.
Kembali ke menu utama...

=== MENU UTAMA ===
1. Sistem Persamaan Linier (SPL)
2. Determinan
3. Matriks Balikan
4. Interpolasi
5. Regresi Polinomial
0. Keluar
Pilih menu:
```

4.7.2 Studi Kasus 2

```
=== MENU UTAMA ===
1. Sistem Persamaan Linier (SPL)
2. Determinan
3. Matriks Balikan
4. Interpolasi
5. Regresi Polinomial
0. Keluar
Pilih menu: Input tidak valid, coba lagi: 4

--- INTERPOLASI ---
1. Interpolasi Polinomial
2. Interpolasi Splina Bézier Kubik
Pilih metode: 1
Masukkan jumlah titik data: 10
x1 y1: 6,567 12,624
x2 y2: 6,967 21,807
x3 y3: 7,258 38,391
x4 y4: 7,451 19,541
x5 y5: 7,548 19,582
x6 y6: 8,032 28,935
x7 y7: 8,258 25,854
x8 y8: 8,484 20,935
x9 y9: 8,709 12,408
x10 y10: 8,997 10,534
=== Langkah-langkah Interpolasi Polinomial ===
1. Membentuk matriks Vandermonde berdasarkan titik x:
|    1,000    6,567   43,125  283,205 1859,808 12213,358 80205,121 52670
7,029 3458885,058 22714498,179 |
|    1,000    6,967   48,539  338,172 2356,043 16414,553 114360,189 7967
47,435 5550939,376 38673394,634 |
|    1,000    7,258   52,679  382,341 2775,031 20141,176 146184,654 1061
008,216 7700797,634 55892389,230 |
|    1,000    7,451   55,517  413,660 3082,182 22965,337 171114,724 1274
975,806 9499844,733 70783343,108 |
|    1,000    7,548   56,972  430,027 3245,843 24499,626 184923,178 1395
800,149 10535499,527 79521950,430 |
|    1,000    8,032   64,513  518,169 4161,930 33428,624 268498,707 2156
581,616 17321663,536 139127601,521 |
|    1,000    8,258   68,195  563,151 4650,499 38403,817 317138,722 2618
931,563 21627136,849 178596896,096 |
|    1,000    8,484   71,978  610,664 5180,869 43954,495 372909,939 3163
```

```
|    1,000    8,997   80,946  728,271 6552,256 58950,651 530379,003 4771
819,893 42932063,578 386259776,010 |
2. Membentuk matriks Y berdasarkan titik y:
|   12,624 |
|   21,807 |
|   38,391 |
|   19,541 |
|   19,582 |
|   28,935 |
|   25,854 |
|   20,935 |
|   12,408 |
|   10,534 |
3. Menghitung invers dari matriks Vandermonde...

===== Metode OBE (Gauss?Jordan) untuk Invers =====
|    1,000    6,567   43,125  283,205 1859,808 12213,358 80205,121 52670
7,029 3458885,058 22714498,179 |
|    1,000    6,967   48,539  338,172 2356,043 16414,553 114360,189 7967
47,435 5550939,376 38673394,634 |
|    1,000    7,258   52,679  382,341 2775,031 20141,176 146184,654 1061
008,216 7700797,634 55892389,230 |
|    1,000    7,451   55,517  413,660 3082,182 22965,337 171114,724 1274
975,806 9499844,733 70783343,108 |
|    1,000    7,548   56,972  430,027 3245,843 24499,626 184923,178 1395
800,149 10535499,527 79521950,430 |
|    1,000    8,032   64,513  518,169 4161,930 33428,624 268498,707 2156
581,616 17321663,536 139127601,521 |
|    1,000    8,258   68,195  563,151 4650,499 38403,817 317138,722 2618
931,563 21627136,849 178596896,096 |
|    1,000    8,484   71,978  610,664 5180,869 43954,495 372909,939 3163
767,926 26841407,085 227722497,709 |
|    1,000    8,709   75,847  660,549 5752,719 50100,430 436324,644 3799
951,327 33093776,108 288213696,127 |
|    1,000    8,997   80,946  728,271 6552,256 58950,651 530379,003 4771
819,893 42932063,578 386259776,010 |

 Matriks Augmentasi [A | I]:
|    1,000    6,567   43,125  283,205 1859,808 12213,358 80205,121 52670
7,029 3458885,058 22714498,179    1,000    0,000    0,000    0,000    0,
000    0,000    0,000    0,000    0,000    0,000 |
|    1,000    6,967   48,539  338,172 2356,043 16414,553 114360,189 7967
```

```
 Matriks Augmentasi [A | I]:
|    1,000    6,567   43,125  283,205 1859,808 12213,358 80205,121 52670
7,029 3458885,058 22714498,179    1,000    0,000    0,000    0,000    0,
000    0,000    0,000    0,000    0,000    0,000 |
|    1,000    6,967   48,539  338,172 2356,043 16414,553 114360,189 7967
47,435 5550939,376 38673394,634    0,000    1,000    0,000    0,000    0
,000    0,000    0,000    0,000    0,000    0,000 |
|    1,000    7,258   52,679  382,341 2775,031 20141,176 146184,654 1061
008,216 7700797,634 55892389,230    0,000    0,000    1,000    0,000
0,000    0,000    0,000    0,000    0,000    0,000 |
|    1,000    7,451   55,517  413,660 3082,182 22965,337 171114,724 1274
975,806 9499844,733 70783343,108    0,000    0,000    0,000    1,000
0,000    0,000    0,000    0,000    0,000    0,000 |
|    1,000    7,548   56,972  430,027 3245,843 24499,626 184923,178 1395
800,149 10535499,527 79521950,430    0,000    0,000    0,000    0,000
 1,000    0,000    0,000    0,000    0,000    0,000 |
|    1,000    8,032   64,513  518,169 4161,930 33428,624 268498,707 2156
581,616 17321663,536 139127601,521    0,000    0,000    0,000    0,000
  0,000    1,000    0,000    0,000    0,000    0,000 |
|    1,000    8,258   68,195  563,151 4650,499 38403,817 317138,722 2618
931,563 21627136,849 178596896,096    0,000    0,000    0,000    0,000
  0,000    0,000    1,000    0,000    0,000    0,000 |
|    1,000    8,484   71,978  610,664 5180,869 43954,495 372909,939 3163
767,926 26841407,085 227722497,709    0,000    0,000    0,000    0,000
  0,000    0,000    0,000    1,000    0,000    0,000 |
|    1,000    8,709   75,847  660,549 5752,719 50100,430 436324,644 3799
951,327 33093776,108 288213696,127    0,000    0,000    0,000    0,000
  0,000    0,000    0,000    0,000    1,000    0,000 |
|    1,000    8,997   80,946  728,271 6552,256 58950,651 530379,003 4771
819,893 42932063,578 386259776,010    0,000    0,000    0,000    0,000
  0,000    0,000    0,000    0,000    0,000    1,000 |

Normalisasi pivot R1 dengan 1,0000000000
|    1,000    6,567   43,125  283,205 1859,808 12213,358 80205,121 52670
7,029 3458885,058 22714498,179    1,000    0,000    0,000    0,000    0,
000    0,000    0,000    0,000    0,000    0,000 |
|    1,000    6,967   48,539  338,172 2356,043 16414,553 114360,189 7967
47,435 5550939,376 38673394,634    0,000    1,000    0,000    0,000    0
,000    0,000    0,000    0,000    0,000    0,000 |
|    1,000    7,258   52,679  382,341 2775,031 20141,176 146184,654 1061
```

```
|    0,000    0,000    0,000    1,000    0,000    0,000    0,000    0,00
0 1612367,721 94618007,934 -35868,989 356704,424 -965414,279    0,000 10
47042,635 -737492,923    0,000 707813,652 -439879,921 67095,401 |
|   -0,000   -0,000   -0,000   -0,000    1,000    0,000    0,000 17161,2
03 814285,868 23215173,708  172,287 -731,722   -0,000   -0,000 2182,075
-4164,373   -0,000 6415,596 -4748,850  874,988 |
|    0,000    0,000    0,000    0,000    0,000    1,000    0,000 -1308,3
06 -55193,340 -1475545,760   -8,497   36,405    0,000    0,000 -109,941
 211,991    0,000 -329,693  245,175  -45,440 |
|   -0,000   -0,000   -0,000   -0,000   -0,000   -0,000    1,000   55,30
4 1750,227 41601,194    0,174   -0,753   -0,000   -0,000    2,302   -4,4
84   -0,000    7,042   -5,262    0,981 |
|   -0,000   -0,000   -0,000   -0,000   -0,000   -0,000   -0,000    1,00
0   62,562 2204,302   -0,252    2,588   -7,162   -0,000    7,938   -5,79
4   -0,000    5,744   -3,626    0,564 |
|    0,000    0,000    0,000    0,000    0,000    0,000    0,000   -0,02
6   -1,677  -60,030    0,003   -0,015    0,000    0,000    0,086   -0,52
4    1,000   -0,822    0,308   -0,035 |
|    0,000    0,000    0,000    0,000    0,000    0,000    0,000   -0,04
8   -3,040 -107,439    0,010   -0,075    0,000    1,000   -1,150    0,37
4    0,000   -0,330    0,203   -0,031 |
Eliminasi: R5 = R5 - (17161,202734)*R8
|    1,000    0,000    0,000    0,000    0,000    0,000    0,000    0,00
0 -13382958,521 -837264622,139 514153,189 -4971618,264 13205889,302    0
,000 -14075221,301 9653846,884    0,000 -9060227,198 5572535,770 -839357
,382 |
|    0,000    1,000    0,000    0,000    0,000    0,000    0,000    0,00
0 13843567,412 852698282,339 -453555,639 4429134,127 -11840909,971    0,
000 12694894,837 -8784185,882    0,000 8304139,117 -5124469,601 774953,0
12 |
|    0,000    0,000    1,000    0,000    0,000    0,000    0,000    0,00
0 -6254940,128 -377477989,036 171239,547 -1687909,413 4540752,872    0,0
00 -4896602,326 3418375,882    0,000 -3255777,972 2016086,493 -306165,08
2 |
|    0,000    0,000    0,000    1,000    0,000    0,000    0,000    0,00
0 1612367,721 94618007,934 -35868,989 356704,424 -965414,279    0,000 10
47042,635 -737492,923    0,000 707813,652 -439879,921 67095,401 |
|    0,000    0,000    0,000    0,000    1,000    0,000    0,000    0,00
0 -259353,320 -14613294,606 4501,961 -45147,702 122908,038    0,000 -134
051,072 95266,161    0,000 -92152,284 57483,588 -8808,689 |
|    0,000    0,000    0,000    0,000    0,000    1,000    0,000 -1308,3
```

```
|   -0,169    4,141  -37,098  132,500 -115,236   44,114  -46,951   24,59
6   -6,390    0,494 |
4. Mengalikan A?? dengan Y untuk mendapatkan koefisien polinomial:
| 58441207363,319 |
| -67106748430,923 |
| 34202664981,913 |
| -10155636357,231 |
| 1936033074,604 |
| -245741394,288 |
| 20768775,105 |
| -1126998,119 |
| 35630,706 |
| -500,060 |
=== Selesai ===


Koefisien Polinomial:
| 58441207363,319 |
| -67106748430,923 |
| 34202664981,913 |
| -10155636357,231 |
| 1936033074,604 |
| -245741394,288 |
| 20768775,105 |
| -1126998,119 |
| 35630,706 |
| -500,060 |
Pn(x) = 58441207363,3188 + (-67106748430,9232)x^1 + (34202664981,9125)x^
2 + (-10155636357,2308)x^3 + (1936033074,6045)x^4 + (-245741394,2881)x^5
 + (20768775,1046)x^6 + (-1126998,1191)x^7 + (35630,7060)x^8 + (-500,059
9)x^9

Masukkan nilai x yang ingin dihitung: 1
P(1,0000) = 17091456145,025314

[!] Terjadi kesalahan dalam pemrosesan: Index 4 out of bounds for length
 4
Pastikan input Anda (misal: dimensi matriks) sesuai untuk operasi yang d
ipilih.
Kembali ke menu utama...

=== MENU UTAMA ===
```

4.8 Studi Kasus Interpolasi Splina Bezier Kubik

```
=== MENU UTAMA ===
1. Sistem Persamaan Linier (SPL)
2. Determinan
3. Matriks Balikan
4. Interpolasi
5. Regresi Polinomial
0. Keluar
Pilih menu: 4

--- INTERPOLASI ---
1. Interpolasi Polinomial
2. Interpolasi Splina Bézier Kubik
Pilih metode: 2
Masukkan jumlah titik sampel: 6
x1 y1: 0 0
x2 y2: 3 11
x3 y3: 6 -4
x4 y4: 8 0
x5 y5: 11 -10
x6 y6: 17 0
=== Langkah-langkah Interpolasi Bézier Kubik ===
1. Membentuk matriks koefisien A:
|    4,000    1,000    0,000    0,000 |
|    1,000    4,000    1,000    0,000 |
|    0,000    1,000    4,000    1,000 |
|    0,000    0,000    1,000    4,000 |
2. Membentuk matriks SX dan SY:
SX:
|   18,000 |
|   36,000 |
|   48,000 |
|   49,000 |
SY:
|   66,000 |
|  -24,000 |
|    0,000 |
|  -60,000 |
3. Menghitung invers dari matriks A...

===== Metode OBE (Gauss?Jordan) untuk Invers =====
|    4,000    1,000    0,000    0,000 |
```

```
===== Metode OBE (Gauss?Jordan) untuk Invers =====
|    4,000    1,000    0,000    0,000 |
|    1,000    4,000    1,000    0,000 |
|    0,000    1,000    4,000    1,000 |
|    0,000    0,000    1,000    4,000 |

 Matriks Augmentasi [A | I]:
|    4,000    1,000    0,000    0,000    1,000    0,000    0,000    0,00
0 |
|    1,000    4,000    1,000    0,000    0,000    1,000    0,000    0,00
0 |
|    0,000    1,000    4,000    1,000    0,000    0,000    1,000    0,00
0 |
|    0,000    0,000    1,000    4,000    0,000    0,000    0,000    1,00
0 |

Normalisasi pivot R1 dengan 4,0000000000
|    1,000    0,250    0,000    0,000    0,250    0,000    0,000    0,00
0 |
|    1,000    4,000    1,000    0,000    0,000    1,000    0,000    0,00
0 |
|    0,000    1,000    4,000    1,000    0,000    0,000    1,000    0,00
0 |
|    0,000    0,000    1,000    4,000    0,000    0,000    0,000    1,00
0 |
Eliminasi: R2 = R2 - (1,000000)*R1
|    1,000    0,250    0,000    0,000    0,250    0,000    0,000    0,00
0 |
|    0,000    3,750    1,000    0,000   -0,250    1,000    0,000    0,00
0 |
|    0,000    1,000    4,000    1,000    0,000    0,000    1,000    0,00
0 |
|    0,000    0,000    1,000    4,000    0,000    0,000    0,000    1,00
0 |

Normalisasi pivot R2 dengan 3,7500000000
|    1,000    0,250    0,000    0,000    0,250    0,000    0,000    0,00
0 |
|    0,000    1,000    0,267    0,000   -0,067    0,267    0,000    0,00
0 |
|    0,000    1,000    4,000    1,000    0,000    0,000    1,000    0,00
```

```
0 |
|    0,000    0,000    1,000    4,000    0,000    0,000    0,000    1,00
0 |
Eliminasi: R1 = R1 - (0,250000)*R2
|    1,000    0,000   -0,067    0,000    0,267   -0,067    0,000    0,00
0 |
|    0,000    1,000    0,267    0,000   -0,067    0,267    0,000    0,00
0 |
|    0,000    1,000    4,000    1,000    0,000    0,000    1,000    0,00
0 |
|    0,000    0,000    1,000    4,000    0,000    0,000    0,000    1,00
0 |
Eliminasi: R3 = R3 - (1,000000)*R2
|    1,000    0,000   -0,067    0,000    0,267   -0,067    0,000    0,00
0 |
|    0,000    1,000    0,267    0,000   -0,067    0,267    0,000    0,00
0 |
|    0,000    0,000    3,733    1,000    0,067   -0,267    1,000    0,00
0 |
|    0,000    0,000    1,000    4,000    0,000    0,000    0,000    1,00
0 |

Normalisasi pivot R3 dengan 3,7333333333
|    1,000    0,000   -0,067    0,000    0,267   -0,067    0,000    0,00
0 |
|    0,000    1,000    0,267    0,000   -0,067    0,267    0,000    0,00
0 |
|    0,000    0,000    1,000    0,268    0,018   -0,071    0,268    0,00
0 |
|    0,000    0,000    1,000    4,000    0,000    0,000    0,000    1,00
0 |
Eliminasi: R1 = R1 - (-0,066667)*R3
|    1,000    0,000    0,000    0,018    0,268   -0,071    0,018    0,00
0 |
|    0,000    1,000    0,267    0,000   -0,067    0,267    0,000    0,00
0 |
|    0,000    0,000    1,000    0,268    0,018   -0,071    0,268    0,00
0 |
|    0,000    0,000    1,000    4,000    0,000    0,000    0,000    1,00
0 |
Eliminasi: R2 = R2 - (0,266667)*R3
|    1,000    0,000    0,000    0,018    0,268   -0,071    0,018    0,00
```

```
|    0,000    1,000    0,000   -0,071   -0,071    0,286   -0,071    0,00
0 |
|    0,000    0,000    1,000    0,268    0,018   -0,071    0,268    0,00
0 |
|    0,000    0,000    0,000    3,732   -0,018    0,071   -0,268    1,00
0 |

Normalisasi pivot R4 dengan 3,7321428571
|    1,000    0,000    0,000    0,018    0,268   -0,071    0,018    0,00
0 |
|    0,000    1,000    0,000   -0,071   -0,071    0,286   -0,071    0,00
0 |
|    0,000    0,000    1,000    0,268    0,018   -0,071    0,268    0,00
0 |
|    0,000    0,000    0,000    1,000   -0,005    0,019   -0,072    0,26
8 |
Eliminasi: R1 = R1 - (0,017857)*R4
|    1,000    0,000    0,000    0,000    0,268   -0,072    0,019   -0,00
5 |
|    0,000    1,000    0,000   -0,071   -0,071    0,286   -0,071    0,00
0 |
|    0,000    0,000    1,000    0,268    0,018   -0,071    0,268    0,00
0 |
|    0,000    0,000    0,000    1,000   -0,005    0,019   -0,072    0,26
8 |
Eliminasi: R2 = R2 - (-0,071429)*R4
|    1,000    0,000    0,000    0,000    0,268   -0,072    0,019   -0,00
5 |
|    0,000    1,000    0,000    0,000   -0,072    0,287   -0,077    0,01
9 |
|    0,000    0,000    1,000    0,268    0,018   -0,071    0,268    0,00
0 |
|    0,000    0,000    0,000    1,000   -0,005    0,019   -0,072    0,26
8 |
Eliminasi: R3 = R3 - (0,267857)*R4
|    1,000    0,000    0,000    0,000    0,268   -0,072    0,019   -0,00
5 |
|    0,000    1,000    0,000    0,000   -0,072    0,287   -0,077    0,01
9 |
|    0,000    0,000    1,000    0,000    0,019   -0,077    0,287   -0,07
2 |
|    0,000    0,000    0,000    1,000   -0,005    0,019   -0,072    0,26
```

```
 Matriks Invers (metode OBE):
|    0,268   -0,072    0,019   -0,005 |
|   -0,072    0,287   -0,077    0,019 |
|    0,019   -0,077    0,287   -0,072 |
|   -0,005    0,019   -0,072    0,268 |
4. Menghitung titik kontrol BX dan BY:
BX:
|    2,923 |
|    6,306 |
|    7,852 |
|   10,287 |
BY:
|   19,694 |
|  -12,775 |
|    7,407 |
|  -16,852 |
5. Menggabungkan hasil menjadi titik kontrol Bézier penuh:
|    0,000    0,000 |
|    2,923   19,694 |
|    6,306  -12,775 |
|    7,852    7,407 |
|   10,287  -16,852 |
|   17,000    0,000 |
=== Selesai ===


Titik Kontrol Bézier Kubik:
|    0,000    0,000 |
|    2,923   19,694 |
|    6,306  -12,775 |
|    7,852    7,407 |
|   10,287  -16,852 |
|   17,000    0,000 |
```

4.9 Studi Kasus Regresi Polinomial Berganda

```
[!] Terjadi kesalahan dalam pemrosesan: Cannot invoke "algeo.lib.Matrix.getCols()" because "a" is null
Pastikan input Anda (misal: dimensi matriks) sesuai untuk operasi yang dipilih.
Kembali ke menu utama...

=== MENU UTAMA ===
1. Sistem Persamaan Linier (SPL)
2. Determinan
3. Matriks Balikan
4. Interpolasi
5. Regresi Polinomial
0. Keluar
Pilih menu:
```

5 Penutup

Melalui pengerjaan Tugas Besar 1 mata kuliah Aljabar Linier dan Geometri, kami telah berhasil mengimplementasikan berbagai konsep dasar aljabar linier ke dalam bentuk program komputasional menggunakan bahasa Java. Program yang dikembangkan tidak hanya mampu menyelesaikan sistem persamaan linear dengan berbagai metode (Eliminasi Gauss, Gauss-Jordan, Kaidah Cramer, dan Matriks Balikan), tetapi juga mendukung operasi fundamental seperti perhitungan determinan dan invers matriks, serta metode lanjutan seperti interpolasi polinomial, interpolasi Bezier kubik, dan regresi polinomial berganda.

Melalui proses pengembangan ini, kami memahami bahwa aljabar linier tidak hanya bersifat teoritis, tetapi juga sangat aplikatif dalam pemecahan berbagai persoalan nyata di bidang teknik, sains, dan analisis data. Proyek ini memperkuat keterampilan kami dalam:

1. Pemrograman matematis, terutama dalam penerapan algoritma numerik secara efisien dan stabil;
2. Perancangan pustaka modular, yang dapat digunakan kembali di konteks lain tanpa perlu modifikasi besar;
3. Kolaborasi tim, baik dalam pembagian tugas, penyusunan dokumentasi, maupun penyelarasan gaya kode agar konsisten dan mudah dipahami.

Implementasi yang dilakukan juga memperlihatkan hubungan yang kuat antara teori aljabar linier dengan dunia komputasi modern, misalnya pada pengolahan citra, optimasi sistem, dan pemodelan data. Dengan demikian, Tugas Besar ini telah menjadi wadah untuk melatih kemampuan menyelesaikan masalah, menguji pengetahuan tentang aljabar linier dan geometri, dan membuat sebuah program nyata dengan bahasa Java.

Untuk bagian saran terhadap asisten, Kami menghargai bimbingan para asisten dalam memberikan penjelasan konsep dan bimbingan dalam bentuk asistensi. Secara keseluruhan, bimbingan yang diberikan sudah sangat membantu dalam memahami korelasi antara teori dan praktik nyata di mata kuliah aljabar linier.

Pesan dari anggota Funkynaris:

1. Brain : Ga lagi2 deadliner :)
2. Dongun : Comeback stronger👍
3. Rama: Tobat

Daftar Pustaka

[1] Oracle, Java Platform, Standard Edition 25 Documentation. Oracle Corporation, 2025. [Online]. Available: https://docs.oracle.com/en/java/javase/25/

[2] H. Anton, Elementary Linear Algebra, 10th ed. Hoboken, NJ, USA: John Wiley & Sons, 2010. [Online]. Available: https://archive.org/details/ElementryLinearAlgebraByHowardAnton10thEdition

[3] G. Muntingh, Topics in Polynomial Interpolation Theory, Ph.D. dissertation, Centre of Mathematics for Applications, Univ. of Oslo, Oslo, Norway, Dec. 2010. [Online]. Available: https://www.researchgate.net/publication/265819309_Topics_in_Polynomial_Interpolation_Theory

[4] L. P. Quan and T. A. Nhan, "A closed-form solution to the inverse problem in interpolation by a Bézier-spline curve," Arab. J. Math., vol. 9, pp. 155–165, 2020. [Online]. Available: https://doi.org/10.1007/s40065-019-0241-0

[5] M. Rosenfeld, Matrix Formulation of OLS: Notes for SOC Methods Project 3. Stanford, CA, USA: Stanford Univ., Dept. of Sociology. [Online]. Available: https://web.stanford.edu/~mrosenfe/soc_meth_proj3/matrix_OLS_NYU_notes.pdf

[6] A. B. Kashlak, Applied Regression Analysis: Course Notes for STAT 378/502. Edmonton, AB, Canada: Univ. of Alberta, Dept. of Mathematical & Statistical Sciences, 2022. [Online]. Available: https://sites.ualberta.ca/~kashlak/data/stat378.pdf

[7] S. Weisberg, Applied Linear Regression, 4th ed. Hoboken, NJ, USA: Wiley-Blackwell, 2013. [Online]. Available: https://www.stat.purdue.edu/~qfsong/teaching/525/book/Weisberg-Applied-Linear-Regression-Wiley.pdf

Lampiran

GitHub **:** https://github.com/IRK-23/algeo1-funkynaris