Laporan Tugas Besar 1
IF2123 Aljabar Linear dan Geometri

Disusun oleh:

Kelompok 1 – Raka Gnarly

Muhammad Luqman Hakim	13523044
Zulfaqqar Nayaka Athadiansyah	13523094
Farrel Athalla Putra	13523118

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

2024

Daftar Isi

Daftar Tabel

Daftar Gambar

1 Deskripsi Masalah

1.1 Gambaran Umum

Sistem persamaan linear (SPL) adalah himpunan persamaan-persamaan linear yang menggunakan peubah yang sama. SPL kerap ditemui di beragam bidang keilmuan untuk beragam keperluan. Atas dasar itu, diperlukanlah suatu metode untuk menyelesaikan SPL.

Dalam mata kuliah IF2123 Aljabar Linear dan Geometri, metode-metode yang telah dipelajari adalah (1) metode eliminasi Gauss, (2) metode eliminasi Gauss-Jordan, (3) metode matriks balikan, dan (4) metode Cramer. Diberikan sebuah SPL, terdapat tiga kemungkinan yang kita miliki: SPL mempunyai solusi unik/tunggal, tak berhingga, atau bahkan tidak ada sama sekali.

1.2 Persoalan

Pada tugas besar ini, kita akan mengimplementasikan keempat[1] metode tersebut dalam sebuah pustaka (*library*) Java untuk kemudian diterapkan dalam beberapa persoalan, mencakup:

1. Menyelesaikan persoalan dalam bentuk SPL

   Diberikan masukan berupa SPL dalam bentuk matriks *augmented*, program harus mampu menentukan solusi dari SPL yang diberikan.

2. Menghitung determinan dan balikan dari suatu matriks

   Hal ini diimplementasikan dalam beberapa metode, misalnya metode ekspansi kofaktor Laplace dan reduksi baris untuk determinan serta metode Gauss-Jordan dan matriks adjoin untuk balikan matriks.

3. Interpolasi polinom dan *bicubic spline*

   Diberikan $n + 1$ buah titik berbeda, $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$, kita dapat menentukan polinom $p_n(x)$ yang melewati semua titik-titik tersebut sedemikian rupa sehingga $y_i = p_n(x_i)$ untuk $0 \leq i \leq n$.

4. Regresi berganda (linear dan kuadratik)

   Regresi adalah metode untuk menentukan *best-fit line* dari sekumpulan titik yang diberikan. Bedanya dengan interpolasi adalah *best-fit line* ini tidak harus melalui seluruh titik yang diberikan.

Sebagai tambahan, dalam tugas besar ini kami juga membuat tampilan antarmuka (*graphical interface*, GUI) dan video penjelasan.

[1] Secara implisit, ini berarti kita juga mengimplementasikan fungsi untuk menghitung determinan dengan metode reduksi baris maupun ekspansi kofaktor, *adjoin*, dan balikan dari suatu matriks.

2 Teori Singkat

2.1 Sistem Persamaan Linear

Sistem persamaan linear (SPL) adalah kumpulan persamaan linear yang melibatkan peubah-peubah yang sama, misalnya sebagai berikut:

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

yang terdiri atas n persamaan dengan n buah peubah (x_i untuk $i \in [1, n]$). Persamaan tersebut dapat dinyatakan kembali dalam bentuk perkalian matriks (kiri) atau, lebih sederhana lagi, dalam wujud matriks *augmented* (kanan):

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

Bentuk ini lebih mudah dimanipulasi untuk mendapatkan solusi dari SPL-nya dengan metode-metode yang akan kita bahas di subbab selanjutnya. Pada matriks *augmented*, ada tiga macam operasi baris elementer (OBE) yang dapat diberlakukan:

1. Mengalikan suatu baris dengan konstanta tidak nol;
2. Menukarkan dua baris; dan
3. Menjumlahkan salah satu baris dengan kelipatan dari baris lainnya.

Secara umum, ada tiga kemungkinan solusi dari sebuah SPL: solusi unik/tunggal, tak berhingga, atau tidak ada solusi.

2.2 Eliminasi Gauss

Metode eliminasi Gauss adalah metode untuk menyelesaikan SPL dengan cara menerapkan operasi baris elementer pada matriks *augmented* dari suatu SPL. Tujuan utamanya adalah membuat koefisien peubah pada posisi tertentu menjadi 1 dan mengeliminasi peubah lainnya di bawahnya, sedemikian sehingga terbentuk matriks segitiga atas (*upper triangular matrix*), di mana semua elemen di bawah diagonal utama adalah nol. Selanjutnya, dilakukan penyulihan mundur untuk mendapatkan nilai dari tiap peubah.

Sebagai ilustrasi, diberikan SPL dalam bentuk matriks *augmented* (kiri), kita dapat melakukan OBE hingga didapatkan matriks segitiga atas (kanan),

$$\left[\begin{array}{ccc|c} 7 & 1 & 5 & 27 \\ 4 & 3 & 5 & 21 \\ 6 & 1 & 2 & 9 \end{array}\right] \xrightarrow{OBE} \left[\begin{array}{ccc|c} 1 & 0 & \frac{10}{209} & \frac{60}{209} \\ 0 & 1 & \frac{335}{209} & \frac{1389}{209} \\ 0 & 0 & 1 & 6 \end{array}\right].$$

Dari sini didapatkan $x_3 = 6$. Melakukan penyulihan mundur memberikan kita $x_1 = 0$ dan $x_2 = -3$. Untuk n yang cukup besar, banyaknya operasi yang dibutuhkan kurang lebih sebanyak $n^3/3$.

2.3 Eliminasi Gauss-Jordan

Eliminasi Gauss-Jordan sebenarnya hanyalah kelanjutan dari eliminasi Gauss. Ketimbang membiarkannya dalam bentuk matriks eselon dan melakukan penyulihan mundur secara manual ke dalam SPL, kita dapat melakukannya dengan OBE hingga didapatkan matriks eselon tereduksi. Misalnya, untuk matriks pada bagian sebelumnya, kita dapatkan

$$\left[\begin{array}{ccc|c} 1 & 0 & \frac{10}{209} & \frac{60}{209} \\ 0 & 1 & \frac{335}{209} & \frac{1389}{209} \\ 0 & 0 & 1 & 6 \end{array}\right] \begin{array}{c} \xrightarrow{R1-10R3/209} \\ \xrightarrow{R2-335R3/209} \end{array} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & 6 \end{array}\right]$$

sehingga dapat langsung kita simpulkan $(x_1, x_2, x_3) = (0, -3, 6)$.

2.4 Matriks Kofaktor

Diberikan suatu matriks persegi A. Matriks kofaktor C dari A didefinisikan sebagai berikut.

$$C_{i,j} = (-1)^{i+j} M_{i,j}$$

dengan $M_{i,j}$ adalah minor dari entri $A_{i,j}$, yang setara dengan determinan dari matriks yang didapatkan dengan menghapus baris ke-i dan kolom ke-j. Perhatikan contoh berikut sebagai gambaran.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \implies M_{2,3} = \det\left(\begin{bmatrix} 1 & 2 & \square \\ \square & \square & \square \\ 7 & 8 & \square \end{bmatrix}\right) = \begin{vmatrix} 1 & 2 \\ 7 & 8 \end{vmatrix} = -6$$

Akibat ekspresi $(-1)^{i+j}$, tanda dari minor entri $A_{i,j}$ dapat dinyatakan dalam matriks "papan catur" berikut.

$$\begin{bmatrix} + & - & + & \dots \\ - & + & - & \dots \\ + & - & + & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

2.5 Matriks Adjoin

Matriks *adjoin* pada dasarnya hanyalah transpos dari matriks kofaktor. Matriks ini nantinya berguna dalam perhitungan determinan.

$$\text{adj}(A) = C_A^T.$$

2.6 Determinan

Determinan pada dasarnya hanyalah suatu bilangan, tetapi mempunyai makna yang besar dalam aljabar linear. Determinan dapat menentukan keberadaan dari balikan suatu matriks. Jika $\det(M) = 0$ maka kita katakan M adalah matriks yang *singular* dan **tidak** mempunyai balikan. Sebaliknya, jika $\det(M) \neq 0$ maka M mempunyai balikan.

Untuk matriks M berukuran 2×2, determinannya dapat dihitung sebagai berikut:

$$\det(M) = \begin{vmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{vmatrix} = m_{11}m_{22} - m_{12}m_{21}$$

Secara umum, ada setidaknya dua metode yang dapat digunakan untuk menghitung determinan dari suatu matriks, yakni metode ekspansi kofaktor Laplace dan metode reduksi baris.

2.6.1 Metode Ekspansi Kofaktor Laplace

Metode ekspansi kofaktor Laplace, seperti namanya, didasari atas matriks kofaktor. Langkah pertamanya adalah menentukan baris atau kolom untuk dijadikan *pivot*, kemudian menjumlahkan perkalian antara entri dengan kofaktor dari baris atau kolom tersebut. Umumnya, baris atau kolom yang banyak mengandung nol dipilih sebagai *pivot* karena dapat menyederhanakan perhitungan. Perhatikan contoh berikut dengan baris pertama sebagai *pivot*:

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a\begin{vmatrix} e & f \\ h & i \end{vmatrix} - b\begin{vmatrix} d & f \\ g & i \end{vmatrix} + c\begin{vmatrix} d & e \\ g & h \end{vmatrix} = \dots$$

2.6.2 Metode Reduksi Baris

Metode ini didasari atas metode ekspansi kofaktor Laplace dan beberapa identitas yang berlaku untuk determinan:

1. Menukarkan baris atau kolom dari suatu matriks persegi akan mengubah tanda determinannya.

$$\det(M) = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} \implies \begin{vmatrix} d & e & f \\ a & b & c \\ g & h & i \end{vmatrix} = \begin{vmatrix} b & a & c \\ e & d & f \\ h & g & i \end{vmatrix} = -\det(M)$$

2. Mengalikan satu baris atau kolom matriks persegi dengan suatu konstanta akan mengalikan determinannya dengan konstanta tersebut.

$$\det(M) = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} \implies \begin{vmatrix} 2\cdot a & b & c \\ 2\cdot d & e & f \\ 2\cdot g & h & i \end{vmatrix} = \begin{vmatrix} a & b & c \\ 2\cdot d & 2\cdot e & 2\cdot f \\ g & h & i \end{vmatrix} = 2\det(M)$$

3. Menjumlahkan satu baris dengan kelipatan dari baris lainnya (berlaku untuk kolom juga) tidak mengubah nilai determinannya.

$$\det(M) = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} \implies \begin{vmatrix} a+3d & b+3e & c+3f \\ d & e & f \\ g & h & i \end{vmatrix} = \begin{vmatrix} a+5c & b & c \\ d+5f & e & f \\ g+5i & h & i \end{vmatrix} = \det(M)$$

Metode reduksi baris melibatkan operasi baris dan kolom elementer untuk mendapatkan matriks segitiga atas.

$$\det(M) = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} \xrightarrow{OBE/OKE} \begin{vmatrix} a'_{11} & a'_{12} & \dots & a'_{1n} \\ 0 & a'_{22} & \dots & a'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a'_{3n} \end{vmatrix}$$

Dengan demikian, kita pada dasarnya bisa menerapkan metode ekspansi kofaktor Laplace pada kolom 1. Akibatnya, determinannya kemudian dapat dihitung dengan persamaan berikut:

$$\det(M) = \frac{(-1)^p a'_{11} \cdot a'_{22} \cdot \ldots \cdot a'_{nn}}{k_1 k_2 \ldots k_n}$$

di mana p adalah banyaknya operasi pertukaran baris dan $k_1, k_2, \ldots, k_n$ adalah konstanta-konstanta yang dikalikan pada baris atau kolom.

2.7 Matriks Balikan

Balikan atau invers dari suatu matriks A adalah invers perkalian (*multiplicative inverse*) dari matriks tersebut, sehingga memenuhi hubungan

$$A \cdot A^{-1} = I.$$

Matriks balikan dapat dihitung dengan menerapkan metode Gauss-Jordan pada matriks *augmented*

$$\begin{bmatrix} A | I \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} I | A^{-1} \end{bmatrix}$$

atau menggunakan formula

$$A^{-1} = \frac{1}{\det(A)} \operatorname{adj}(A)$$

2.8 Kaidah Cramer

Kaidah Cramer adalah kaidah untuk menyelesaikan SPL dengan melibatkan perhitungan determinan matriks koefisien dan determinan dari matriks yang didapatkan dengan mengganti salah satu kolom matriks koefisien dengan matriks yang berada di ruas kanan. Diberikan persamaan matriks $A\mathbf{x} = \mathbf{b}$ sebagai berikut,

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

solusinya adalah

$$x_i = \frac{\det(A_i)}{\det(A)}$$

dengan A_i adalah matriks A yang kolom ke-i-nya diganti dengan matriks kolom $\mathbf{y}$. Dengan metode yang naif ini, kurang lebih sekitar n^4 operasi dibutuhkan. Dengan demikian, metode Cramer lebih boros dibandingkan metode eliminasi Gauss sehingga cenderung digunakan untuk n yang kecil saja.

2.9 Interpolasi Polinom

Interpolasi adalah metode numerik untuk menentukan suatu fungsi yang melalui sejumlah titik yang diberikan. Lebih lanjut lagi, *interpolasi polinom* adalah bentuk spesifik dari interpolasi, di mana fungsinya berupa fungsi polinom dengan bentuk umum

$$f(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n.$$

Dengan mensubstitusikan nilai (x_i, y_i) dari titik yang kita punya akan didapatkan suatu SPL:

$$\begin{aligned} a_0 + a_1x_1 + a_2x_1^2 + \ldots + a_nx_1^n &= y_1 \\ a_0 + a_1x_2 + a_2x_2^2 + \ldots + a_nx_2^n &= y_2 \\ \ldots \\ a_0 + a_1x_n + a_2x_n^2 + \ldots + a_nx_n^n &= y_n \end{aligned} \longrightarrow \left[\begin{array}{ccccc|c} 1 & x_1 & x_1^2 & \ldots & x_1^n & y_1 \\ 1 & x_2 & x_2^2 & \ldots & x_2^n & y_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \\ 1 & x_n & x_n^2 & \ldots & x_n^n & y_n \end{array}\right]$$

$a_0, a_1, a_2, \ldots, a_n$ akan didapatkan dengan menyelesaikan SPL di atas. Matriks koefisien dari SPL ini disebut matriks Vandermonde.

2.10 Interpolasi *Bicubic Spline*

Pada interpolasi polinom, kita bisa membentuk polinom derajat-n dari $n + 1$ titik yang diketahui. Akan tetapi, untuk n yang besar, polinom yang terbentuk akan menghasilkan osilasi meskipun kurva aslinya justru mulus. Berangkat dari hal ini, kita bisa mengambil subset dari titik-titik yang ada dan menerapkan interpolasi secara sepotong (*piecewise*) dengan polinom berderajat lebih rendah. Kurva sepotong yang dihasilkan ini disebut *spline*, sementara jika polinomnya berderajat tiga, akan didapatkan interpolasi *cubic spline*. Jika ide ini dikembangkan ke $\mathbb{R}^3$, didapatkan interpolasi *bicubic spline*[2].

Pada interpolasi *bicubic spline*, kita membutuhkan 16 buah titik, dengan 4 titik di grid pusat sebagai referensi utama dan 12 titik sisanya untuk mengaproksimasi turunan berarahnya. Secara umum, interpolasi *bicubic spline* memodelkan fungsi objektif sebagai $f(x, y) = \sum_{j=0}^{3}\sum_{i=0}^{3} a_{ij}x^iy^j$ sehingga didapatkan sistem persamaan berikut,

$$f_x(x, y) = \sum_{j=0}^{3}\sum_{i=0}^{3} a_{ij}ix^{i-1}y^j \qquad f_y(x, y) = \sum_{j=0}^{3}\sum_{i=0}^{3} a_{ij}jx^iy^{j-1} \qquad f_{xy}(x, y) = \sum_{j=0}^{3}\sum_{i=0}^{3} a_{ij}ijx^{i-1}y^{j-1}$$

Dengan mensubstitusikan titik-titik (0, 0), (1, 0), (0, 1), dan (1, 1), akan didapatkan SPL yang dapat direpresentasikan dalam persamaan matriks $y = Ax$ berikut:

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

2.11 Regresi Linear Berganda

Berbeda dengan regresi linear biasa, regresi linear berganda melibatkan lebih dari satu peubah (akibatnya, "berganda"). Untuk data dengan n peubah $x_1, x_2, \ldots, x_n$, persamaan umum regresi linear berganda yang berlaku adalah

$$y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \ldots + \beta_nx_n + \varepsilon_i$$

[2] Pada kasus ini berarti kita bukan lagi memprediksi persamaan garis, melainkan persamaan permukaan.

dengan ε_i adalah galat dari y_i dengan $y(x_i)$, yang dapat kita tuliskan sebagai

$$\varepsilon_i = (y_i - \beta_0 - \beta_1 x_{1,i} - \beta_2 x_{2,i} - \ldots - \beta_n x_{n,i})$$

sehingga jumlah kuadrat dari ε_i adalah

$$S_\varepsilon = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{1,i} - \beta_2 x_{2,i} - \ldots - \beta_n x_{n,i})^2.$$

Menurunkan terhadap tiap peubah dan mengaturnya menjadi sama dengan nol akan memberikan SPL berikut:

$$\begin{bmatrix} n & \sum_{i=1}^n x_{1,i} & \sum_{i=1}^n x_{2,i} & \ldots & \sum_{i=1}^n x_{n,i} \\ \sum_{i=1}^n x_{1,i} & \sum_{i=1}^n x_{2,i} & \sum_{i=1}^n x_{1,i}x_{2,i} & \ldots & \sum_{i=1}^n x_{1,i}x_{n,i} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^n x_{1,i} & \sum_{i=1}^n x_{2,i} & \sum_{i=1}^n x_{1,i}x_{2,i} & \ldots & \sum_{i=1}^n x_{1,i}x_{n,i} \end{bmatrix} \cdot \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_{1,i}y_i \\ \vdots \\ \sum_{i=1}^n x_{n,i}y_i \end{bmatrix}.$$

Dengan demikian, persoalan regresi linear berganda direduksi dengan menyelesaikan SPL di atas untuk mendapatkan β_i dan menyusun persamaan garisnya.

2.12 Regresi Kuadratik Berganda

Analog dengan regresi linear berganda, regresi kuadratik berganda adalah regresi kuadratik yang melibatkan lebih dari satu peubah. Tidak hanya peubah independen berpangkat satu dan dua, tapi kita juga akan berurusan dengan peubah "interaksi", yakni perkalian antara dua peubah. Sebagai contoh, untuk $n = 2$, regresi kuadratik berganda menghasilkan model yang dapat dinyatakan dalam persamaan

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2 + \varepsilon_2$$

dan dengan penalaran yang sama seperti pada regresi linear berganda, kita bisa dapatkan SPL berikut (contoh untuk $n = 2$):

$$\begin{bmatrix} N & \sum x_{1,i} & \sum x_{2,i} & \sum x_{1,i}^2 & \sum x_{1,i}x_{2,i} & \sum x_{2,i}^2 \\ \sum x_{1,i} & \sum x_{1,i}^2 & \sum x_{1,i}x_{2,i} & \sum x_{1,i}^3 & \sum x_{1,i}^2 x_{2,i} & \sum x_{1,i}x_{2,i}^2 \\ \sum x_{2,i} & \sum x_{1,i}x_{2,i} & \sum x_{2,i}^2 & \sum x_{1,i}^2 x_{2,i} & \sum x_{1,i}x_{2,i}^2 & \sum x_{2,i}^3 \\ \sum x_{1,i}^2 & \sum x_{1,i}^3 & \sum x_{1,i}^2 x_{2,i} & \sum x_{1,i}^4 & \sum x_{1,i}^3 x_{2,i} & \sum x_{1,i}^2 x_{2,i}^2 \\ \sum x_{1,i}x_{2,i}^2 & \sum x_{1,i}^2 x_{2,i} & \sum x_{1,i}x_{2,i}^2 & \sum x_{1,i}^3 x_{2,i} & \sum x_{1,i}^2 x_{2,i}^2 & \sum x_{1,i}x_{2,i}^3 \\ \sum x_{2,i}^2 & \sum x_{1,i}x_{2,i}^2 & \sum x_{2,i}^3 & \sum x_{1,i}^2 x_{2,i}^2 & \sum x_{1,i}x_{2,i}^3 & \sum x_{2,i}^4 \end{bmatrix} \cdot \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum y_i x_{1,i} \\ \sum y_i x_{2,i} \\ \sum y_i x_{1,i}^2 \\ \sum y_i x_{1,i} x_{2,i} \\ \sum y_i x_{2,i}^2 \end{bmatrix}$$

3 Implementasi Program dan Pustaka

3.1 Struktur Program

Program ini tersusun atas `Main.java` dan lima folder yang memuat kode sumber tiap fitur. Struktur program disajikan melalui *directory tree* berikut.

```
/src
├── Main.java
├── Matrix
│   ├── MatrixADT.java
│   ├── DeterminanMK.java
│   ├── DeterminanReduksi.java
│   ├── InverseAdjoin.java
│   ├── InverseGaussJ.java
│   ├── SPLGauss.java
│   ├── SPLGaussJ.java
│   ├── SPLBalikan.java
│   ├── SPLCramer.java
│   ├── Regresi.java
│   ├── InterpolasiBicubicSpline.java
│   └── InterpolasiPolinom.java
└── ImageResizing
    └── ImageResizing.java
```

Gambar 1: *Directory tree* program

3.2 Garis Besar Program

3.3 Struktur *Class* dalam Pustaka

Bagian ini menjelaskan struktur dari seluruh *class* yang ada pada program, mencakup metode serta atribut dan konstruktor jika ada. Pada program ini, hanya `MatrixADT.java` yang mempunyai atribut dan konstruktor karena di kelas tersebut ADT untuk matriks didefinisikan, sementara *class* lainnya lebih berperan sebagai *utility*.

3.3.1 MatrixADT.java

Tabel 1: Daftar atribut MatrixADT.java

Atribut		
Nama	Tipe	Deskripsi
`nRows`	`int`	Panjang baris dari matriks
`nCols`	`int`	Panjang kolom dari matriks
`matrix`	`double[][]`	*array of array of double* yang menampung konten matriks

Tabel 2: Daftar konstruktor MatrixADT.java

Konstruktor	
Nama	Deskripsi
`MatrixADT(int nRows, int nCols)`	Membuat objek matriks bertipe `double[][]` dengan dimensi `nRows` × `nCols`

Tabel 3: Daftar metode MatrixADT.java

Metode	
Nama	Deskripsi
`getRows()`	Mengambil `nRows` dari matriks
`getCols()`	Mengambil `nCols` dari matriks
`getElmt(int row, int col)`	Mengambil elemen pada baris `row` kolom `col` dari matriks
`setElmt(int row, int col, double value)`	Mengubah elemen pada baris `row` kolom `col` dari matriks menjadi `value`
`printMatrix()`	Mencetak matriks dengan spasi sebagai pemisah antarkolom dan baris baru sebag[.5em]ai pemisah antarbaris
`readMatrix(int row, int col)`	Membaca matriks berukuran `row` × `col` dengan spasi sebagai pemisah antarkolom dan baris baru sebagai pemisah antarbaris
`matrixMinor(MatrixADT m, int i, int j)`	Mengembalikan matriks minor dari elemen pada baris `i` kolom `j`
`copyMatrix()`	Membuat salinan dari suatu matriks

3.3.2 txtIO.java

Tabel 4: Daftar metode txtIO.java

Metode	
Nama	Deskripsi
`readTXT(String filename)`	Membaca matriks dari file `filename.txt` dengan pemisah antarbaris enter dan pemisah antarkolom spasi lalu menyimpannya dalam `inputMatrix`
`writeTXT(String filename, MatrixADT outputMatrix)`	Menuliskan `outputMatrix` ke dalam `filename.txt` dengan pemisah antarbaris enter dan pemisah antarkolom spasi.

3.3.3 DeterminanMK.java

Tabel 5: Daftar metode DeterminanMK.java

Metode	
Nama	Deskripsi
detMK(MatrixADT m)	Menghitung determinan menggunakan metode ekspansi kofaktor Laplace

3.3.4 DeterminanReduksi.java

Tabel 6: Daftar metode DeterminanReduksi.java

Metode	
Nama	Deskripsi
`detRB(MatrixADT m)`	Menghitung determinan dengan reduksi baris dan kolom
`swapRows(MatrixADT m, int row1, int row2)`	Menukar `row1` dan `row2` pada matriks `m`

3.3.5 InverseAdjoin.java

Tabel 7: Daftar metode InverseAdjoin.java

Metode	
Nama	Deskripsi
`inverseAdj(MatrixADT m)`	Menghitung $\mathtt{m}^{-1}$
`cofactor(MatrixADT m, int i, int j)`	Menghitung kofaktor dari entri $\mathtt{m}_{\mathtt{ij}}$
`transpose(MatrixADT m)`	Menukar baris dengan kolom untuk menghasilkan transpos dari matriks `m`

3.3.6 InverseGaussJ.java

Tabel 8: Daftar metode InverseGaussJ.java

Metode	
Nama	Deskripsi
`inverseGaussJ(MatrixADT m)`	Menghitung $\mathtt{m}^{-1}$
`convertMatrix(MatrixADT m)`	Membentuk matriks *augmented* $[\mathtt{m}\vert I]$
`OBEGaussJ(MatrixADT m)`	Menerapkan OBE untuk mentransformasi $[\mathtt{m}\vert I]$ menjadi $[I\vert\mathtt{m}^{-1}]$
`swapRows(MatrixADT m, int row1, int row2)`	Menukar `row1` dan `row2` pada matriks `m`

3.3.7 SPLGauss.java

Tabel 9: Daftar metode SPLGauss.java

Metode	
Nama	Deskripsi
`gaussReduction()`	Menerapkan reduksi OBE untuk mendapatkan matriks eselon baris
`swapRows(int row1, int row2)`	Menukarkan `row1` dengan `row2`
`round(double value)`	Membulatkan bilangan yang harga mutlaknya di bawah 10^{-9} menjadi nol
`detectSolution()`	Memeriksa ada tidaknya solusi SPL
`printUniqueSolution(int[] pivotRow)`	Mencetak solusi unik dari SPL
`printParametricSolution(boolean[] isPivotColumn, int[] pivotRow)`	Mencetak solusi parametrik dari SPL

3.3.8 SPLGaussJ.java

Tabel 10: Daftar metode SPLGaussJ.java

Metode	
Nama	Deskripsi
`gaussReduction()`	Menerapkan reduksi OBE untuk mendapatkan matriks eselon baris
`swapRows(int row1, int row2)`	Menukarkan `row1` dengan `row2`
`round(double value)`	Membulatkan bilangan yang harga mutlaknya di bawah 10^{-9} menjadi nol
`detectSolution()`	Memeriksa ada tidaknya solusi SPL
`printUniqueSolution(int[] pivotRow)`	Mencetak solusi unik dari SPL
`printParametricSolution(boolean[] isPivotColumn, int[] pivotRow)`	Mencetak solusi parametrik dari SPL

3.3.9 SPLBalikan.java

Tabel 11: Daftar metode SPLBalikan.java

Metode	
Nama	Deskripsi
`balikan(MatrixADT m)`	Menghitung balikan dari `m` dengan metode balikan

3.3.10 SPLCramer.java

Tabel 12: Daftar metode SPLCramer.java

Metode	
Nama	Deskripsi
`cramer(MatrixADT m)`	Menghitung solusi SPL dengan kaidah Cramer
`computeDeterminant(MatrixADT m)`	Menghitung det(`m`) dengan reduksi baris
`swapRows(MatrixADT m, int row1, int row2)`	Menukarkan `row1` dengan `row2`

3.3.11 Regresi.java

Tabel 13: Daftar metode Regresi.java

Metode	
Nama	Deskripsi
`regresiLinearBerganda(MatrixADT data)`	Menerapkan regresi linear berganda pada `data` dan menghasilkan koefisien regresinya
`regresiKuadratikBerganda(MatrixADT data)`	Menerapkan regresi kuadratik berganda pada `data` dan menghasilkan koefisien regresinya
`printLinearSolution(MatrixADT coefs)`	Mencetak solusi regresi linear berganda berupa persamaan linear
`printQuadraticSolution(MatrixADT coefs, int nVars)`	Mencetak solusi regresi kuadratik berganda berupa persamaan kuadratik
`predict(MatrixADT coefs, MatrixADT queries)`	Memprediksi hasil menggunakan koefisien `coefs` yang diperoleh dari regresi linier dan `queries`
`multipleQuadraticEquation(MatrixADT coefs, MatrixADT queries)`	Menghitung hasil dari persamaan kuadratik ganda untuk `queries` dengan koefisien dari `coefs`

3.3.12 InterpolasiPolinom.java

Tabel 14: Daftar metode InterpolasiPolinom.java

Metode	
Nama	Deskripsi
`scanPoints(int n)`	Memindai `n+1` titik sampel
`SubstitutePoints(MatrixADT points)`	Mensubstitusikan titik-titik sampel ke dalam matriks Vandermonde
`PolynomialCoefficients(MatrixADT points)`	Memberikan koefisien dari polinom hasil interpolasi
`displaySolution(MatrixADT coefs)`	Menampilkan solusi interpolasi polinom

3.3.13 BicubicSplineInterpolation.java

Tabel 15: Daftar metode InterpolasiBicubicSpline.java

Metode	
Nama	Deskripsi
`bicubicSplineMatrixGenerator(MatrixADT coordinates)`	Menghasilkan matriks spline bicubic berdasarkan koordinat yang diberikan.
`bicubicSplineCoefs(MatrixADT values)`	Menghasilkan matriks koefisien A
`bicubicSplineCoefs(MatrixADT coordinates, MatrixADT values)`	Menghasilkan matriks koefisien untuk koordinat tertentu (selain *unit* square)
`bicubicEquation(MatrixADT coefs, double x, double y)`	Mengevaluasi nilai $f(x, y)$ berdasarkan matriks koefisien `coefs`
`bicubicSplineInterpolation(MatrixADT values, MatrixADT queries)`	Melakukan interpolasi bicubic berdasarkan nilai yang diberikan dan titik kueri.
`bicubicSplineInterpolation(MatrixADT coordinates, MatrixADT values, double x, double y)`	Menghitung interpolasi bicubic untuk koordinat dan nilai tertentu pada titik (x,y).
`printEquation(MatrixADT coefs)`	Mencetak persamaan spline bicubic dalam bentuk polinom dengan koefisien dari `coefs`

4 Eksperimen

4.1 Determinan

Akan dihitung determinan dari kedua matriks berikut dengan metode matriks kofaktor (MK) dan reduksi baris (RB).

$$(a)\begin{vmatrix} 1 & 0.5 & 0.333 & 0.25 & 0.2 & 0.166 \\ 0.5 & 0.333 & 0.25 & 0.2 & 0.166 & 0.1428 \\ 0.333 & 0.25 & 0.2 & 0.166 & 0.1428 & 0.125 \\ 0.25 & 0.2 & 0.166 & 0.1428 & 0.125 & 0.111 \\ 0.2 & 0.166 & 0.1428 & 0.125 & 0.111 & 0.1 \\ 0.166 & 0.1428 & 0.125 & 0.111 & 0.1 & 0.0909 \end{vmatrix} \qquad (b)\begin{vmatrix} 8 & 1 & 3 & 2 \\ 2 & 9 & -1 & -2 \\ 1 & 3 & 2 & -1 \\ 1 & 0 & 6 & 4 \end{vmatrix}$$

```
\\ KELUARAN
\\ Matriks A
Determinan matriks dengan MK adalah : -9.099246e-13
Determinan matriks dengan RB adalah : -9.099246e-13

\\ Matriks B
Determinan matriks dengan MK adalah : 740.000000
Determinan matriks dengan RB adalah : 740.000000
```

4.2 Invers

Akan dihitung invers dari kedua matriks berikut dengan metode adjoin maupun metode Gauss-Jordan.

$$(a)\begin{bmatrix} 1 & 0.5 & 0.333 & 0.25 & 0.2 & 0.166 \\ 0.5 & 0.333 & 0.25 & 0.2 & 0.166 & 0.1428 \\ 0.333 & 0.25 & 0.2 & 0.166 & 0.1428 & 0.125 \\ 0.25 & 0.2 & 0.166 & 0.1428 & 0.125 & 0.111 \\ 0.2 & 0.166 & 0.1428 & 0.125 & 0.111 & 0.1 \\ 0.166 & 0.1428 & 0.125 & 0.111 & 0.1 & 0.0909 \end{bmatrix} \qquad (b)\begin{bmatrix} 8 & 1 & 3 & 2 \\ 2 & 9 & -1 & -2 \\ 1 & 3 & 2 & -1 \\ 1 & 0 & 6 & 4 \end{bmatrix}$$

```
\\ KELUARAN
\\ Matriks A
InverseAdjMatrix adalah :
[ 6.473229 -20.287200 5.661845 -22.537032 104.428235 -75.098873 ]
[ -20.287200 126.688593 -138.063131 262.713415 -813.216318 601.703831 ]
[ 5.661845 -138.063131 518.917279 -887.920819 962.532452 -481.662764 ]
[ -22.537032 262.713415 -887.920819 698.764686 736.443324 -813.987221 ]
[ 104.428235 -813.216318 962.532452 736.443324 -797.683989 -258.538651 ]
[ -75.098873 601.703831 -481.662764 -813.987221 -258.538651 1143.645743 ]

InverseGaussJMatrix adalah :
[ 6.473229 -20.287200 5.661845 -22.537032 104.428235 -75.098873 ]
[ -20.287200 126.688593 -138.063131 262.713415 -813.216318 601.703831 ]
[ 5.661845 -138.063131 518.917279 -887.920819 962.532452 -481.662764 ]
[ -22.537032 262.713415 -887.920819 698.764686 736.443324 -813.987221 ]
[ 104.428235 -813.216318 962.532452 736.443324 -797.683989 -258.538651 ]
[ -75.098873 601.703831 -481.662764 -813.987221 -258.538651 1143.645743 ]

\\ Matriks B
InverseAdjMatrix adalah :
[ 0.137838 -0.018919 0.010811 -0.075676 ]
[ -0.033784 0.141892 -0.081081 0.067568 ]
[ -0.020270 -0.114865 0.351351 0.040541 ]
[ -0.004054 0.177027 -0.529730 0.208108 ]
InverseGaussJMatrix adalah :
[ 0.137838 -0.018919 0.010811 -0.075676 ]
[ -0.033784 0.141892 -0.081081 0.067568 ]
[ -0.020270 -0.114865 0.351351 0.040541 ]
[ -0.004054 0.177027 -0.529730 0.208108 ]
```

4.3 SPL-1

Tentukan solusi dari SPL $A\mathbf{x} = \mathbf{b}$ berikut.

(a)

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

```
// Masukan
1 1 -1 -1 1
2 5 -7 -5 -2
2 -1 1 3 4
5 2 -4 2 6

// Keluaran
Metode Reduksi Gauss:
Solusi tidak dapat ditentukan

Metode Reduksi Gauss-Jordan:
Solusi tidak dapat ditentukan

Metode Cramer:
Solusi tidak dapat ditentukan

Metode Matriks Balikan:
Solusi tidak dapat ditentukan
```

Studi Kasus 1: spl-1a.txt

Keluaran yang diperoleh menunjukkan bahwa solusi tidak dapat ditentukan oleh keempat metode penyelesaian SPL (Metode Reduksi Gauss, Reduksi Gauss-Jordan, Cramer, dan Matriks Balikan). Hal ini disebabkan oleh determinan dari matriks koefisien yang bernilai nol. Ketika determinan dari matriks koefisien sama dengan nol, ini berarti matriks tersebut tidak dapat di-invers, yang menyebabkan sistem persamaan tidak memiliki solusi tunggal atau konsisten. Dalam kasus ini, sistem tidak memiliki solusi tak hingga (parametrik) sehingga tidak memiliki solusi sama sekali (inkonsisten). Oleh karena itu, keluaran program yang menyatakan bahwa solusi tidak dapat ditentukan adalah benar dan sesuai dengan sifat matematis dari SPL tersebut.

(b)

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

```
// Masukan
1 -1 0 0 1 3
1 1 0 -3 0 6
2 -1 0 1 -1 5
-1 2 0 -2 -1 -1

// Keluaran
Metode Reduksi Gauss:
Parametric Solution:
X1 = 3,000000 + 1,000000a2 + -1,000000a5
X2 = 1,500000 + 1,500000a4 + 0,500000a5
X3 = a3
X4 = -1,000000 + 1,000000a5
X5 = a5

Metode Reduksi Gauss-Jordan:
Parametric Solution:
X1 = 3,000000 + 1,000000a5
X2 = 0,000000 + 2,000000a5
X3 = a3
X4 = -1,000000 + 1,000000a5
X5 = a5

Metode Cramer:
Solusi tidak dapat ditentukan

Metode Matriks Balikan:
Solusi tidak dapat ditentukan
```

Studi Kasus 2: spl-1b.txt

Keluaran dari program menunjukkan bahwa solusi dari sistem persamaan linear ini bervariasi bergantung pada metode yang digunakan. Berikut adalah penjelasan mengenai masing-masing metode:

Metode Reduksi Gauss: Hasil keluaran menunjukkan bahwa sistem memiliki solusi parametrik, di mana variabel X_1, X_2, X_3, X_4, dan X_5 dinyatakan dalam bentuk parameter bebas (a_2, a_3, a_4, dan a_5). Ini menunjukkan bahwa sistem memiliki banyak solusi, karena beberapa variabel bergantung pada nilai parameter bebas. Artinya, ada kebebasan dalam menentukan nilai beberapa variabel, sehingga menghasilkan solusi parametrik.

Metode Reduksi Gauss-Jordan: Hasil dari metode ini juga menghasilkan solusi parametrik, namun dalam bentuk yang lebih sederhana. Variabel X_1, X_2, X_3, X_4, dan X_5 juga dinyatakan dalam bentuk parameter bebas (a_3 dan a_5), tetapi dengan lebih sedikit ketergantungan pada parameter. Solusi ini menunjukkan bahwa sistem masih memiliki banyak solusi, tetapi lebih sederhana dibandingkan hasil dari metode Gauss.

Metode Cramer: Solusi tidak dapat ditentukan menggunakan metode Cramer karena dimensi dari matriks bukan $n \times (n+1)$. Dalam metode Cramer, kita memerlukan determinan dari matriks koefisien yang berbentuk $n \times n$.

Metode Matriks Balikan: Solusi juga tidak dapat ditentukan menggunakan metode ini karena dimensi dari matriks bukan $n \times (n+1)$. Dalam metode Matriks Balikan, kita memerlukan determinan dari matriks koefisien yang bisa di-invers, yang juga mengharuskan matriks berbentuk $n \times n$.

(c)

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

```
// Masukan
0 1 0 0 1 0 2
0 0 0 1 1 0 -1
0 1 0 0 0 1 1

// Keluaran
Metode Reduksi Gauss:
Parametric Solution:
X1 = a1
X2 = 2,000000 + -1,000000a5
X3 = a3
X4 = -1,000000 + -1,000000a5
X5 = 1,000000 + 1,000000a6
X6 = a6

Metode Reduksi Gauss-Jordan:
Parametric Solution:
X1 = a1
X2 = 1,000000 + -1,000000a6
X3 = a3
X4 = -2,000000 + -1,000000a6
X5 = 1,000000 + 1,000000a6
X6 = a6

Metode Cramer:
Solusi tidak dapat ditentukan

Metode Matriks Balikan:
Solusi tidak dapat ditentukan
```

Studi Kasus 3: spl-1c.txt

Keluaran dari program menunjukkan hasil sebagai berikut:

Metode Reduksi Gauss: Sistem menghasilkan solusi parametrik. Variabel X_1, X_2, X_3, X_4, X_5, dan X_6 dinyatakan dalam bentuk parameter bebas (a_1, a_3, a_5, dan a_6), yang berarti terdapat banyak solusi karena beberapa variabel bergantung pada nilai parameter bebas.

Metode Reduksi Gauss-Jordan: Solusi parametrik yang lebih sederhana dibandingkan metode Gauss, tetapi masih bergantung pada parameter bebas a_1, a_3, dan a_6. Sistem masih memiliki banyak solusi yang tergantung pada nilai parameter tersebut.

Metode Cramer: Solusi tidak dapat ditentukan menggunakan metode Cramer karena dimensi dari matriks bukan $n \times (n + 1)$.

Metode Matriks Balikan: Solusi juga tidak dapat ditentukan menggunakan metode ini karena dimensi dari matriks bukan $n \times (n + 1)$.

(d) Uji untuk $n = 6$ dan $n = 11$.

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

```
// Masukan
1.000000 0.500000 0.333333 0.250000 0.200000 0.166667 1
0.500000 0.333333 0.250000 0.200000 0.166667 0.142857 0
0.333333 0.250000 0.200000 0.166667 0.142857 0.125000 0
0.250000 0.200000 0.166667 0.142857 0.125000 0.111111 0
0.200000 0.166667 0.142857 0.125000 0.111111 0.100000 0
0.166667 0.142857 0.125000 0.111111 0.100000 0.090909 0

// Keluaran
Metode Reduksi Gauss:
Unique Solution:
X1 = 11,540412
X2 = 46,616694
X3 = -1130,944969
X4 = 3969,618023
X5 = -5045,250153
X6 = 2158,663954

Metode Reduksi Gauss-Jordan:
Unique Solution:
X1 = 11,540412
X2 = 46,616694
X3 = -1130,944969
X4 = 3969,618023
X5 = -5045,250153
X6 = 2158,663954

Metode Cramer:
X1 = 11,540412
X2 = 46,616694
X3 = -1130,944969
X4 = 3969,618023
X5 = -5045,250153
X6 = 2158,663954

Metode Matriks Balikan:
X1: 11,540412
X2: 46,616694
X3: -1130,944969
X4: 3969,618023
X5: -5045,250153
X6: 2158,663954
```

Studi Kasus 4: spl-1d1.txt

```
// Masukan
1.000000 0.500000 0.333333 0.250000 0.200000 0.166667 0.142857 0.125000 0.111111 0.100000 1
0.500000 0.333333 0.250000 0.200000 0.166667 0.142857 0.125000 0.111111 0.100000 0.090909 0
0.333333 0.250000 0.200000 0.166667 0.142857 0.125000 0.111111 0.100000 0.090909 0.083333 0
0.250000 0.200000 0.166667 0.142857 0.125000 0.111111 0.100000 0.090909 0.083333 0.076923 0
0.200000 0.166667 0.142857 0.125000 0.111111 0.100000 0.090909 0.083333 0.076923 0.071429 0
```

```
0.166667 0.142857 0.125000 0.111111 0.100000 0.090909 0.083333 0.076923 0.071429 0.066667 0
0.142857 0.125000 0.111111 0.100000 0.090909 0.083333 0.076923 0.071429 0.066667 0.062500 0
0.125000 0.111111 0.100000 0.090909 0.083333 0.076923 0.071429 0.066667 0.062500 0.058824 0
0.111111 0.100000 0.090909 0.083333 0.076923 0.071429 0.066667 0.062500 0.058824 0.055556 0
0.100000 0.090909 0.083333 0.076923 0.071429 0.066667 0.062500 0.058824 0.055556 0.052632 0

// Keluaran
Metode Reduksi Gauss:
Unique Solution:
X1 = 19,525627
X2 = -159,545670
X3 = 20,759112
X4 = 1983,940776
X5 = -5084,976735
X6 = 5660,147295
X7 = -4622,220507
X8 = 3257,960172
X9 = -623,836012
X10 = -456,355282

Metode Reduksi Gauss-Jordan:
Unique Solution:
X1 = 19,525627
X2 = -159,545670
X3 = 20,759112
X4 = 1983,940776
X5 = -5084,976735
X6 = 5660,147295
X7 = -4622,220507
X8 = 3257,960172
X9 = -623,836012
X10 = -456,355282

Metode Cramer:
X1 = 19,525627
X2 = -159,545670
X3 = 20,759112
X4 = 1983,940776
X5 = -5084,976735
X6 = 5660,147295
X7 = -4622,220507
X8 = 3257,960172
X9 = -623,836012
X10 = -456,355282

Metode Matriks Balikan:
X1: 19,525627
X2: -159,545670
X3: 20,759112
X4: 1983,940776
X5: -5084,976735
X6: 5660,147295
X7: -4622,220507
X8: 3257,960172
X9: -623,836012
X10: -456,355282
```

Studi Kasus 5: spl-1d2.txt

Keluaran program menunjukkan bahwa sistem persamaan linear yang diberikan memiliki solusi unik, dan keempat metode yang digunakan (Metode Reduksi Gauss, Reduksi Gauss-Jordan, Metode Cramer, dan Matriks Balikan) menghasilkan solusi yang sama. Dapat disimpulkan bahwa sistem ini memiliki determinan matriks koefisien tidak nol, memungkinkan metode eliminasi, Cramer, dan matriks balikan untuk memberikan solusi yang sama.

4.4 SPL-2

Selesaikan SPL berbentuk matriks *augmented* berikut.

$$(a)\begin{bmatrix} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{bmatrix} \qquad (b)\begin{bmatrix} 2 & 0 & 8 & 0 & 8 \\ 0 & 1 & 0 & 4 & 6 \\ -4 & 0 & 6 & 0 & 6 \\ 0 & -2 & 0 & 3 & -1 \\ 2 & 0 & -4 & 0 & -4 \\ 0 & 1 & 0 & -2 & 0 \end{bmatrix}$$

```
// Masukan
1 -1 2 -1 -1
2 1 -2 -2 -2
-1 2 -4 1 1
3 0 0 -3 -3

// Keluaran
Metode Reduksi Gauss:
Parametric Solution:
X1 = -1,000000 + 1,000000a2 + -2,000000a3 + 1,000000a4
X2 = 0,000000 + 2,000000a3
X3 = a3
X4 = a4

Metode Reduksi Gauss-Jordan:
Parametric Solution:
X1 = -1,000000 + 1,000000a4
X2 = 0,000000 + 2,000000a3
X3 = a3
X4 = a4

Metode Cramer:
Solusi tidak dapat ditentukan

Metode Matriks Balikan:
Solusi tidak dapat ditentukan
```

Studi Kasus 6: spl-2a.txt

```
// Masukan
2 0 8 0 8
0 1 0 4 6
-4 0 6 0 6
0 -2 0 3 -1
2 0 -4 0 -4
0 1 0 -2 0

// Keluaran
Metode Reduksi Gauss:
Unique Solution:
X1 = 0,000000
X2 = 2,000000
X3 = 1,000000
X4 = 1,000000
Metode Reduksi Gauss-Jordan:
Unique Solution:
X1 = 0,000000
X2 = 2,000000
X3 = 1,000000
X4 = 1,000000
Metode Cramer:
Cramer's Rule only applies to systems with n equations and n unknowns.
Metode Matriks Balikan:
Solusi tidak dapat ditentukan
```

Studi Kasus 7: spl-2b.txt

Di sini kaidah Cramer dan metode Matriks Balikan tidak dapat digunakan karena banyaknya persamaan tidak sama dengan banyaknya peubah, atau dengan kata lain matriks koefisien bukan matriks persegi sehingga tidak memiliki determinan.

4.5 SPL-3

(a)

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

```
// Masukan
8 1 3 2 0
2 9 -1 -2 1
1 3 2 -1 2
1 0 6 4 3

// Keluaran
Metode Reduksi Gauss:
Unique Solution:
X1 = -0,224324
X2 = 0,182432
X3 = 0,709459
X4 = -0,258108

Metode Reduksi Gauss-Jordan:
Unique Solution:
X1 = -0,224324
X2 = 0,182432
X3 = 0,709459
X4 = -0,258108

Metode Cramer:
X1 = -0,224324
X2 = 0,182432
X3 = 0,709459
X4 = -0,258108

Metode Matriks Balikan:
X1: -0,224324
X2: 0,182432
X3: 0,709459
X4: -0,258108
```

Studi Kasus 8: spl-3a.txt

Keluaran program menunjukkan bahwa sistem persamaan linear yang diberikan memiliki solusi unik, dan keempat metode yang digunakan (Metode Reduksi Gauss, Reduksi Gauss-Jordan, Metode Cramer, dan Matriks Balikan) menghasilkan solusi yang sama. Dapat disimpulkan bahwa sistem ini memiliki determinan matriks koefisien tidak nol, memungkinkan metode eliminasi, Cramer, dan matriks balikan untuk memberikan solusi yang sama.

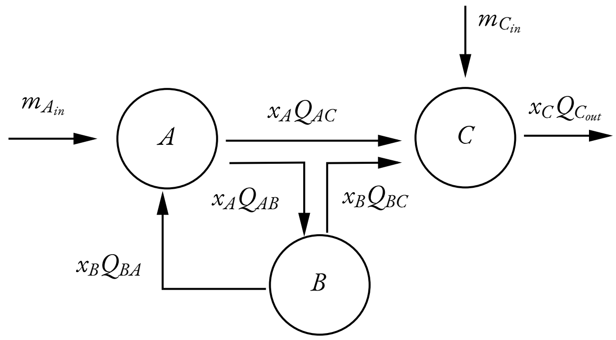
(b)

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

```
// Masukan
0 0 0 0 0 0 1 1 1 13.00
0 0 0 1 1 1 0 0 0 15.00
1 1 1 0 0 0 0 0 0 8.00
0 0 0.04289 0 0.04289 0.75 0.04289 0.75 0.61396 14.79
0 0.25 0.91421 0.25 0.91421 0.25 0.91421 0.25 0 14.31
0.61396 0.75 0.04289 0.75 0.04289 0 0.04289 0 0 3.81
0 0 1 0 0 1 0 0 1 18.00
0 1 0 0 1 0 0 1 0 12.00
1 0 0 1 0 0 1 0 0 6.00
0.04289 0.75 0.61396 0 0.04289 0.75 0 0 0.04289 10.51
0.91421 0.25 0 0.25 0.91421 0.25 0 0.25 0.91421 16.13
0.04289 0 0 0.75 0.04289 0 0.61396 0.75 0.04289 7.04

// Keluaran
Metode Reduksi Gauss:
No Solution

Metode Reduksi Gauss-Jordan:
No Solution

Metode Cramer:
Cramer's Rule only applies to systems with n equations and n unknowns.

Metode Matriks Balikan:
Solusi tidak dapat ditentukan
```

Studi Kasus 9: spl-3b.txt

Untuk metode Gauss dan Gauss-Jordan, tidak ada solusi karena setelah reduksi, terdapat baris di mana kolom paling akhir memiliki nilai, sedangkan kolom-kolom lainnya pada baris yang sama bernilai 0 semua. Ini menunjukkan adanya ketidakkonsistenan, di mana persamaan tersebut tidak mungkin dipenuhi, sehingga sistem persamaan tidak memiliki solusi. Metode Cramer dan Matriks Balikan tidak dapat ditentukan karena banyaknya persamaan tidak sama dengan banyaknya peubah, atau dengan kata lain matriks koefisien bukan matriks persegi sehingga tidak memiliki determinan.



Gambar 2: Model rangkaian reaktor

4.6 SPL-4

Lihatlah sistem reaktor pada gambar berikut.

Dengan laju volume Q dalam m^3/s dan input massa m_{in} dalam mg/s. Konservasi massa pada tiap inti reaktor adalah sebagai berikut:

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

Tentukan solusi x_A, x_B, dan x_C dengan menggunakan parameter berikut: $Q_{AB} = 40$ m^3/s, $Q_{AC} = 80$ m^3/s, $Q_{BA} = 60$ m^3/s, $Q_{BC} = 20$ m^3/s, $Q_{C\ out} = 150$ m^3/s, $m_{A_{in}} = 1300$ mg/s dan $m_{C_{in}} = 200$ mg/s.

Jawab. Dengan mensubstitusikan nilai yang sesuai, didapatkan persamaan berikut:

$$-120x_A + 60x_B = -1300$$
$$40x_A - 80x_B = 0$$
$$80x_A + 20x_B - 150x_C = -200$$

```
// Masukan
-120 60 0 -1300
40 -80 0 0
80 20 -150 -200

// Keluaran
Unique Solution:
X1 = 14,444444
X2 = 7,222222
X3 = 10,000000

Metode Reduksi Gauss-Jordan:
Unique Solution:
X1 = 14,444444
X2 = 7,222222
X3 = 10,000000

Metode Cramer:
X1 = 14,444444
X2 = 7,222222
X3 = 10,000000

Metode Matriks Balikan:
X1: 14,444444
X2: 7,222222
X3: 10,000000
```

Studi Kasus 10: spl-4.txt

Keluaran program menunjukkan bahwa sistem persamaan linear yang diberikan memiliki solusi unik, dan keempat metode yang digunakan (Metode Reduksi Gauss, Reduksi Gauss-Jordan, Metode Cramer, dan Matriks Balikan) menghasilkan solusi yang sama. Dapat disimpulkan bahwa sistem ini memiliki determinan matriks koefisien tidak nol, memungkinkan metode eliminasi, Cramer, dan matriks balikan untuk memberikan solusi yang sama.

4.7 Interpolasi Polinom

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

Tabel 16: Tabel data interpolasi polinom

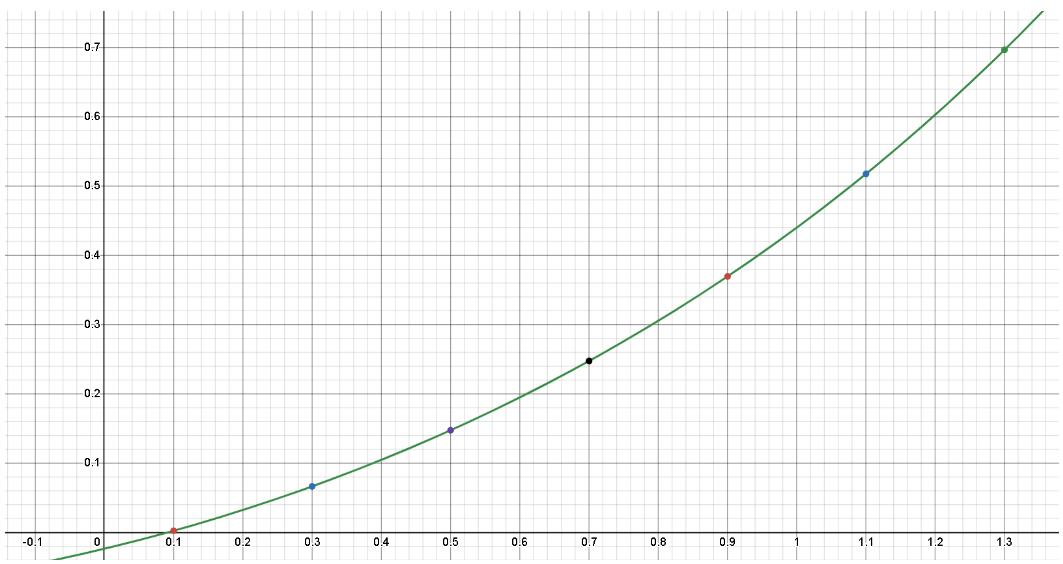
x	0.1	0.3	0.5	0.7	0.9	1.1	1.3
$f(x)$	0.003	0.067	0.148	0.248	0.370	0.518	0.697

Tentukan nilai $f(0.2)$, $f(0.55)$, $f(0.85)$, dan $f(1.28)$.

```
// Masukan
(0.1,0.003)
(0.3,0.067)
(0.5,0.148)
(0.7,0.248)
(0.9,0.370)
(1.1,0.518)
(1.3,0.697)

// Keluaran
>> Y = -0,02298 + 0,24000 X + 0,19740 X^2 + 0,02604 X^4
>> f(0,200000) = 0,03296
>> f(0,550000) = 0,17112
>> f(0,850000) = 0,33724
>> f(1,280000) = 0,67754
```

Studi Kasus 11: interpol-a.txt

Keluaran menunjukkan hasil estimasi nilai fungsi pada titik-titik yang tidak ada di antara data awal, menggunakan polinom interpolasi yang dihasilkan dari data yang diberikan.



Gambar 3: Plot kurva hasil estimasi menggunakan interpolasi polinom berdasarkan titik-titik yang dilewati

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

Tabel 17: Tabel kasus COVID-19 17/06/2022–31/08/2022

Tanggal	Tanggal (desimal)	Jumlah Kasus Baru
17/06/2022	6, 567	12.624
30/06/2022	7	21.807
08/07/2022	7, 258	38.391
14/07/2022	7, 451	54.517
17/07/2022	7, 548	51.952
26/07/2022	7, 839	28.228
05/08/2022	8, 161	35.764
15/08/2022	8, 484	20.813
22/08/2022	8, 709	12.408
31/08/2022	9	10.534

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

$$\text{Tanggal (desimal} = \text{bulan} + (\text{tanggal/jumlah hari pada bulan tersebut})$$

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

$$\text{Tanggal (desimal)} = 6 + (17/30) = 6.567$$

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

(i) 16/07/2022 (Tanggal (desimal) = 7, 5161)
(ii) 10/08/2022 (Tanggal (desimal) = 8, 3225)
(iii) 05/09/2022 (Tanggal (desimal) = 9, 1667)
(iv) Masukan user lainnya berupa tanggal (desimal) yang sudah diolah dengan asumsi prediksi selalu dilakukan untuk tahun 2022 (kita pilih 11/10/2022 = 10, 3548).

```
// Masukan
6.567 12.624
7.000 21.807
7.258 38.391
7.451 54.517
7.548 51.952
7.839 28.228
8.161 35.764
8.484 20.813
8.709 12.408
9.000 10.534

// Keluaran
>> Y = 7187066071652,80600 - 9346993079163,87900 X + 5334203055235,44800 X^2 -
    1756810186359,80440 X^3 + 368550807175,24060 X^4 - 51131876760,09328 X^5 +
    4695806315,42557 X^6 - 275474539,42049 X^7 + 9372849,23910 X^8 - 140993,71225 X^9
>> f(7,516100) = 53609,86523
>> f(8,322500) = 36433,16797
>> f(9,166700) = -664914,04688
>> f(10,354800) = -932130,96454
>> f(10,354800) = -932131180,43750
```

Studi Kasus 12: interpol-b.txt

Keluaran ini menunjukkan bahwa interpolasi polinomial dengan derajat tinggi dapat menghasilkan prediksi yang tidak realistis di luar rentang data, terutama pada tanggal yang lebih jauh dari data asli. Hal ini menunjukkan urgensi penggunaan metode lainnya, misalnya interpolasi *bicubic spline*.

(c) Sederhanakan fungsi f(x) yang memenuhi kondisi

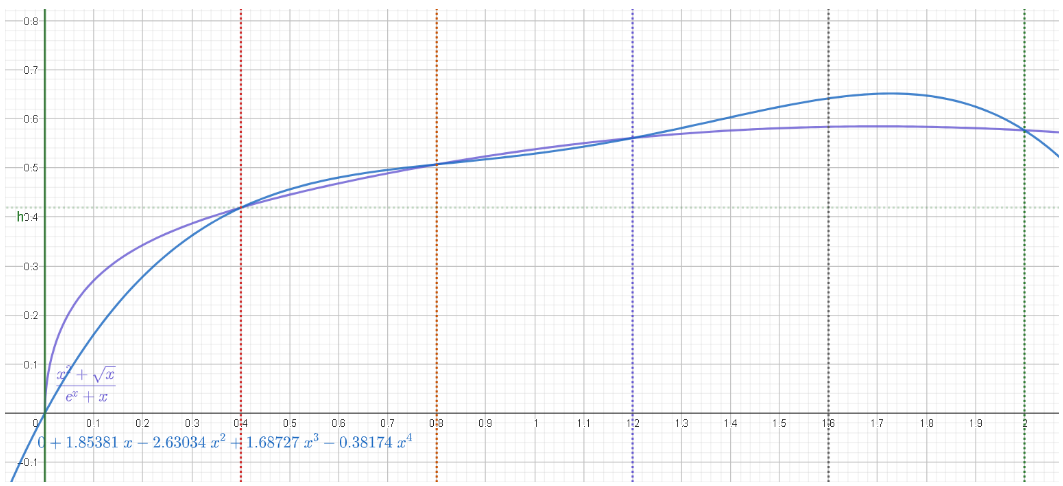
$$f(x) = \frac{x^2 + \sqrt{x}}{e^x + x}$$

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

```
// Masukan
0 0
0.4 0.4188842
0.8 0.507157968
1.2 0.5609246748
2 0.5766515297

// Keluaran
>> Y = 0.00000 + 1,85381 X - 2,63034 X^2 + 1,68727 X^3 - 0,38174 X^4
>> f(0,500000) = 0,45637
```

Studi Kasus 13: interpol-c.txt

Keluaran ini menunjukkan bagaimana polinom interpolasi digunakan untuk memperkirakan nilai $f(x)$ di antara titik-titik data yang diberikan.



Gambar 4: Plot fungsi asli $f(x) = \dfrac{x^2 + \sqrt{x}}{e^x + x}$ dan fungsi hasil estimasi $g(x)$ menggunakan interpolasi polinom

4.8 Regresi Linear Berganda dan Kuadratik Berganda

Diberikan sekumpulan data sesuai pada tabel berikut ini.

Tabel 18: Data untuk studi kasus regresi linear berganda dan kuadratik berganda

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

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

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

$$
\begin{aligned}
20b_0 + 863.1b_1 + 1530.4b_2 + 587.84b_3 &= 19.42 \\
863.1b_0 + 54876.89b_1 + 67000.09b_2 + 25283.395b_3 &= 779.477 \\
1530.4b_0 + 67000.09b_1 + 117912.32b_2 + 44976.867b_3 &= 1483.4 \\
587.84b_0 + 25283.395b_1 + 44976.867b_2 + 17278.5086b_3 &= 571.1219
\end{aligned}
$$

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

```
// Masukan
72.40 76.30 29.18 0.90
41.60 70.30 29.35 0.91
34.30 77.10 29.24 0.96
35.10 68.00 29.27 0.89
10.70 79.00 29.78 1.00
12.90 67.40 29.39 1.10
8.30 66.80 29.69 1.15
20.10 76.90 29.48 1.03
72.20 77.70 29.09 0.77
24.00 67.70 29.60 1.07
123.20 76.80 29.38 1.07
147.40 86.60 29.35 0.94
131.50 76.90 29.63 1.10
110.60 86.30 29.56 1.10
111.20 86.00 29.48 1.10
173.30 76.30 29.40 0.91
175.40 77.90 29.28 0.87
196.60 78.70 29.29 0.78
1107.40 86.80 29.03 0.82
154.90 70.90 29.37 0.95

Query:
50 76 29.3

// Keluaran
>> Model Regresi Linear Berganda:
>> Y = 0.00000 + 1,85381 X - 2,63034 X^2 + 1,68727 X^3 - 0,38174 X^4
>> Tes nilai (x = 0,5):
```

```
>> f(0,500000) = 0,45637

>> Model Regresi Kuadratik Berganda:
>> Y = −793,440 − 0,138X1 + 0,916X2 + 51,483X3 + 0,000X1^2 − 0,000X1X2 + 0,005X1X3 + 0,001X2^2 −
   0,034X2X3 − 0,829X3^2
>> Tes nilai (x = 0,5):
>> f(0,500000) = 0,9659864
```

Studi Kasus 14: interpol-c.txt

Keluaran menunjukkan dua model regresi yang dihasilkan dari data yang diberikan: regresi linear berganda dan regresi kuadratik berganda. Hasil dari kedua model tersebut dibandingkan dengan nilai input $x = 0.5$, di mana model kuadratik memberikan hasil yang sedikit lebih mendekati realitas daripada model linear.

4.9 Interpolasi *Bicubic Spline*

Diberikan matriks berikut,

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

tentukan nilai $f(0, 0)$, $f(0.5, 0.5)$, $f(0.25, 0.75)$, dan $f(0.1, 0.9)$.

```
// Masukan
21 98 125 153
51 101 161 59
0 42 72 210
16 12 81 96

// Keluaran
Masukkan titik: 0 0
>> f(0, 0) = 21
Masukkan titik: 0.25 0.74
>> f(0.25, 0.75) = 87.796875
Masukkan titik: 0.25 0.75
>> f(0.25, 0.75) = 117.732177
Masukkan titik: 0.1 0.9
>> f(0.1, 0.9) = 128.575187
```

Studi Kasus 15: interbicspl.txt

5 Penutup

5.1 Kesimpulan

Secara keseluruhan, kami berhasil menyelesaikan seluruh komponen nonbonus dengan baik. Untuk tugas bonus, kami tidak mengerjakan GUI karena dirasa cukup sukar dan penggunaan *command line interface* dirasa sudah cukup memfasilitasi penggunaan program ini. Kami juga berusaha untuk terus mengoptimasi kode-kode yang sudah kami buat sebelumnya supaya program dapat bekerja secara efisien.

Beberapa kesimpulan yang didapatkan selama pengerjaan, baik secara teoretis maupun praktis, adalah bahwa eliminasi Gauss ternyata lebih efisien dibandingkan kaidah Cramer. Dalam menghitung balikan matriks pun ternyata metode Gauss-Jordan juga lebih optimal dibandingkan cara adjoin.

5.2 Saran

Hemat kami, untuk tugas besar selanjutnya ada baiknya dicantumkan banyak referensi yang bisa digunakan untuk mengembangkan wawasan dan menyelesaikan persoalan terkait tugas besar tersebut. Referensi yang diberikan bisa menjadi awalan bagi kami untuk bereksplorasi dan mencari referensi lebih banyak lagi. Selain itu, nilai untuk bonus GUI terlalu kecil sehingga sebagian peserta kelas mungkin merasa kurang *worth it* untuk mengerjakannya.

5.3 Komentar

Tugas besar ini tidak hanya memperkaya khazanah keinformatikaan kami, tetapi juga mempererat persahabatan di antara kami bertiga.

Komentar 5.1: Muhammad Luqman Hakim

Tugas besar yang mantap.

Komentar 5.2: Zulfaqqar Nayaka Athadiansyah

Selagi menelusuri referensi untuk tugas besar ini, saya menemukan banyak buku-buku yang menarik.

Komentar 5.3: Farrel Athalla Putra

Saya senang mengerjakan tugas besar ini. Akan tetapi, menurut saya untuk regresi kuadratik berganda dan interpolasi *bicubic spline* perlu diberi referensi lebih banyak karena mencari pembahasan mendalam tentang topik ini relatif susah. [a]

[a] Terima kasih sudah mengerjakan bagian itu, Luqman!

5.4 Refleksi

Kami berhasil membagi tugas dengan bijak dan sigap dalam mengerjakan bagian masing-masing. Laporan ini sudah dikerjakan dengan progres lebih dari 50% hanya beberapa hari setelah rilis tugas besar. *Class* `MatrixADT`, balikan, SPL, dan determinan juga dikerjakan dengan cepat. Akan tetapi, masih ada sejumlah *blunder* yang kami lakukan, yakni menunda-nunda pekerjaan selama berhari-hari dan bersantai sejenak, tidak merencanakan struktur folder program dari awal, serta beberapa kesalahan minor lainnya. Masih ada penggunaan maupun pembuatan fungsi yang redundan. Terlepas dari itu, kami bersyukur bisa menyelesaikan tugas besar ini.

6 Daftar Pustaka

Craig, Bruce A., McCabe, George P., dan Moore, David S. (2014). *Introduction to the Practice of Statistics* (edisi ke-8). New York, NY: W. H. Freeman and Co.

Laboratorium Ilmu Rekayasa Komputasi. (2024). *Spesifikasi Tugas Besar 1 IF2123 Aljabar Linear dan Geometri 2024-2025*. Program Studi Teknik Informatika ITB.

Press, William H *et al*. (1992). *Numerical Recipes in C: The Art of Scientific Computing* (edisi ke-2). New York, NY: Cambridge University Press.

Strang, Gilbert. (2006). *Linear Algebra and Its Applications* (edisi ke-4). Florence, KY: Cengage Learning.

Süli, Endre dan Mayers, David F. (2003). *An Introduction to Numerical Analysis*. New York, NY: Cambridge University Press.

Steven C. Chapra, Raymond P. Canale. (2006). *Numerical methods for engineers* (edisi ke-6). New York, NY: McGraw-Hill

7 Lampiran

- Tautan repositori GitHub
- Tautan video penjelasan