Tugas Besar 1 IF2123 ALJABAR LINEAR DAN GEOMETRI

SISTEM PERSAMAAN LINEAR, DETERMINAN, DAN APLIKASINYA
SEMESTER 1 TAHUN 2023/2024

Oleh :

Justin Aditya Putra Prabakti 13522130
Muhammad Dzaki Arta 13522149
Axel Santadi Warih 13522155

Kelompok “Java adalah koentji”

Bab 1
Deskripsi masalah

SPESIFIKASI TUGAS

1. Buatlah pustaka (*library* atau *package*) dalam **Bahasa Java** untuk menemukan solusi SPL dengan metode eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah *Cramer* (kaidah *Cramer* khusus untuk SPL dengan n peubah dan n persamaan), menghitung determinan matriks dengan reduksi baris dan dengan ekspansi kofaktor, dan menghitung balikan matriks.
2. Gunakan pustaka di atas untuk membuat program penyelesaian berbagai persoalan dalam bentuk SPL, menyelesaikan persoalan interpolasi dan regresi linier, menghitung matriks balikan, menghitung determinan matriks dengan berbagai metode (reduksi baris dan ekspansi kofaktor).

Spesifikasi program adalah sebagai berikut:

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

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

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

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

   Luaran (*output*) disesuaikan dengan persoalan (determinan atau invers) dan penghitungan balikan/invers dilakukan dengan metode matriks balikan dan adjoin.

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

   8.0 2.0794
   9.0 2.1972
   9.5 2.2513

8.3

4. Untuk persoalan regresi, masukannya jika dari *keyboard* adalah n (jumlah peubah x), m (jumlah sampel), semua nilai-nilai x_{1i}, x_{2i}, …, x_{ni}, nilai y_i, dan nilai-nilai x_k yang akan ditaksir nilai fungsinya. Jika masukannya dari *file*, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.
5. Untuk persoalan SPL, luaran program adalah solusi SPL. Jika solusinya tunggal, tuliskan nilainya. Jika solusinya tidak ada, tuliskan solusi tidak ada, jika solusinya banyak, maka tuliskan solusinya dalam bentuk parametrik (misalnya $x_4 = -2$, $x_3 = 2s - t$, $x_2 = s$, dan $x_1 = t$).
6. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan. Contoh luaran untuk interpolasi adalah

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

   dan untuk regresi adalah

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

7. Untuk persoalan *bicubic spline interpolation*, masukan dari *file text* (.txt) yang berisi matriks berukuran 4 x 4 yang berisi konfigurasi nilai fungsi dan turunan berarah disekitarnya, diikuti dengan nilai a dan b untuk mencari nilai $f(a, b)$.

   Misalnya jika nilai dari $f(0, 0)$, $f(1, 0)$, $f(0, 1)$, $f(1, 1)$, $f_x(0, 0)$, $f_x(1, 0)$, $f_x(0, 1)$, $f_x(1, 1)$, $f_y(0, 0)$, $f_y(1, 0)$, $f_y(0, 1)$, $f_y(1, 1)$, $f_{xy}(0, 0)$, $f_{xy}(1, 0)$, $f_{xy}(0, 1)$, $f_{xy}(1, 1)$ berturut-turut adalah 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 serta nilai a dan b yang dicari berturut-turut adalah 0.5 dan 0.5 maka isi *file text* ditulis sebagai berikut:

   1 2 3 4

   5 6 7 8

   9 10 11 12

   13 14 15 16

   0.5 0.5

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

8. Luaran program harus dapat ditampilkan **pada layar komputer dan dapat disimpan ke dalam *file***.
9. Bahasa program yang digunakan adalah Java. Anda bebas untuk menggunakan versi java apapun dengan catatan di atas java versi 8 (8/9/11/15/17/19/20).
10. Program **tidak harus** berbasis GUI, cukup *text-based* saja, namun boleh menggunakan GUI (memakai kakas *Eclipse* misalnya).
11. Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan dirancang masing-masing. Misalnya, menu:

    MENU

    1. Sistem Persamaaan Linier
    2. Determinan

3. Matriks balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic Spline
6. Regresi linier berganda
7. Keluar

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

1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer

Begitu juga untuk pilihan menu nomor 2 dan 3.

Bab 2
Teori singkat

2.1 Eliminasi Gauss

Eliminasi gauss ditemukan oleh Carl Friedrich Gauss, metode ini dapat dimanfaatkan untuk memecahkan sistem persamaan linear dengan merepresentasikan (mengubah) menjadi bentuk matriks, matriks tersebut lalu diubah kebentuk Eselon Baris melalui Operasi Baris Elementer. Kemudian sistem diselesaikan dengan substitusi balik.

$$\begin{bmatrix} 1 & 4 & -3 & 7 \\ 0 & 1 & 6 & 2 \\ 0 & 0 & 1 & 5 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 & 2 & 6 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Gambar 2.1 Contoh Matrix Eselon Baris

Matrix Eselon Baris memiliki kriteria sebagai berikut :

1. Jika didalam baris terdapat elemen-elemen yang tidak semuanya nol, maka bilangan tak nol pertama di dalam baris tersebut adalah 1.
2. Kalau ada baris-baris yang semua elemennya bernilai nol semua, maka baris baris tersebut harus dikelompokan dan di letakan di bagian bawah matrix
3. Jika terdapat 2 baris berurutan yang memiliki kriteria pertama, maka angka 1(pertama/utama) dari baris yang lebih rendah berada lebih kekanan dari angka 1 (pertama/utama) dari baris diatasnya.

Langkah langkah penyelesaian SPL ini dinamakan “Eliminasi Gauss” dan operasi yang dilakukan dinamakan Operasi Baris Elementer (OBE) dengan tujuan untuk membentuk Matrix Eselon Baris. Dalam metode ini terdapat 3 jenis operasi yang digunakan yaitu :

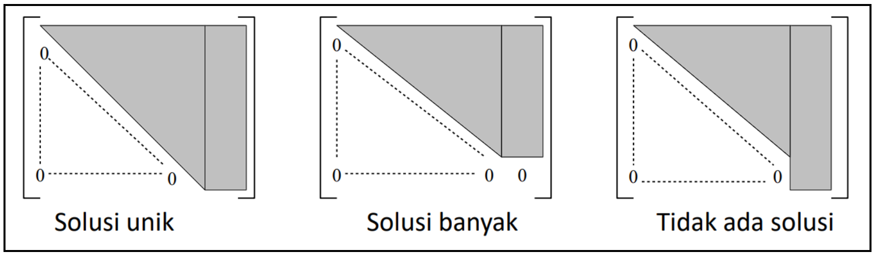
1. Mengganti urutan 2 baris
2. Mengalikan baris dengan angka yang bukan nol
3. Menambah suatu baris dengan kelipatan baris yang lainnya.

Solusi sebuah SPL diperoleh dengan menerapkan OBE pada matriks augmented sampai terbentuk matriks eselon baris atau matriks eselon baris tereduksi. Jika berakhir pada matriks eselon baris maka lakukan metode Eliminasi Gauss. Ada 3 langkah dalam menyelesaika SPL menggunakan Metode Eliminasi Gauss yaitu.

1. Nyatakan SPL dalam bentuk matriks augmented.
2. Terapkan OBE pada matriks augmented sampai terbentuk matriks eselon baris.
3. Selesaikan persamaan yang berkoresponden pada matriks eselon baris dengan teknik penyulihan mundur (backward substitution).

Ada tiga kemungkinan solusi yang dapat terjadi pada SPL:

1. Mempuntai solusi yang unik (Tunggal).
2. Mempunyai banyak solusi (Tidak Hingga Banyaknya).
3. Tidak memiliki Solusi sama sekali.



Gambar 2.2 Tiga kemungkinan Solusi SPL.

$$\begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix} \quad \begin{bmatrix} 1 & 3 & -2 & 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Ciri: Matriks augmented akhir berbentuk segitiga sampai barus terakhir

Ciri: ada satu atau lebih baris yang semuanya bernilai 0

Ciri: ada baris yang semuanya bernilai 0 kecuali nilai pada bagian kolom b

Gambar 2.3 Contoh matrix Tiga Kemungkinan Solusi SPL

2.2 Eliminasi Gauss Jordan

Metode Gauss Jordan merupakan pengembangan dari metode eliminasi Gauss. Operasi baris elementer (OBE) diterapkan pada matriks augmented sehingga menghasilkan matriks eselon baris tereduksi.

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

Gambar 2.4 Operasi Baris Elementer (OBE)

Tidak diperlukan lagi substitusi secara mundur untuk memperoleh nilai nilai variabel. Nilai variabel langsung diperoleh dari matriks augmented akhir (jika solusinya unik).Metode eliminasi Gauss-Jordan terdiri dari dua fase yaitu :

1. Fase maju (forward phase) atau fase eliminasi Gauss.
   - Menghasilkan nilai 0 dibawah 1 utama.

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

Gambar 2.5 Operasi Forward phase

2. Fase mundur (Backward phase).
   - Menghasilkan nilai-nilai 0 di atas satu utama.

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

Gambar 2.5 Operasi Backward phase

2.3 Determinan

Determinan adalah suatu nilai yang dapat dihitung hanya di matriks persegi. Matriks persegi adalah matriks yang memiliki jumlah kolom dan baris yang sama. Dalam menghitung determinan, ada 2 cara yang dapat dilakukan, yaitu dengan menggunakan metode ekspansi kofaktor dan dengan reduksi baris. Metode reduksi baris didapat dengan membuat matrix segitiga atas ataupun bawah.

Untuk metode reduksi baris didapat dengan memanipulasi matrix A sehingga menjadi matrix segitiga atas atau segitiga bawah. Berikut adalah aturan dalam metode reduksi baris.

- $A \xrightarrow{\text{Kalikan sebuah baris dengan k}} B$, maka det(B) = k det(A)
- $A \xrightarrow{\text{Pertukarkan dua baris}} B$, maka det(B) = –det(A)
- $A \xrightarrow{\text{Sebuah baris ditambahkan dengan k kali baris yang lain}} B$, maka det(B) = det(A)

Gambar 2.6 Aturan metode reduksi baris

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

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

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

p menyatakan banyaknya operasi pertukaran baris di dalam OBE

Gambar 2.7 Determinan metode reduksi baris

Jika ada sebuah matriks dengan ukuran $i \times j$, maka minor entri matriks dapat dinyatakan sebagai Mij didefinisikan menjadi determinan submatriks selain baris i dan kolom j. Lalu ada yang namanya kofaktor yang dapat dihitung dengan cara $(-1)i+jMij$ yang dapat dinyatakan sebagai Cij.

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

$$M_{11} = \begin{vmatrix} 2 & -4 \\ 5 & 3 \end{vmatrix} = (2)(3) - (-4)(5) = 26 \qquad C_{11} = (-1)^{1+1} M_{11} = M_{11} = 26$$

Gambar 2.8 Contoh Kofaktor dalam matrix.

Cara mengingat tanda positif dan negative untuk Cij adalah dengan memperhatikan pola berikut:

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

Gambar 2.9 Posisi negatif pada kofaktor

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1k} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2k} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & a_{k3} & \cdots & a_{kk} \end{bmatrix} \sim \begin{bmatrix} a'_{11} & a'_{12} & a'_{13} & \cdots & a'_{1k} \\ 0 & a'_{22} & a'_{23} & \cdots & a'_{2k} \\ 0 & 0 & a_{33} & \cdots & a'_{3k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a'_{kk} \end{bmatrix}$$

$$\det(A) = (-1)^p a'_{11} a'_{22} a'_{33} \dots a'_{kk}$$

Gambar 2.10 Metode Determinan dengan Expansi Kofaktor

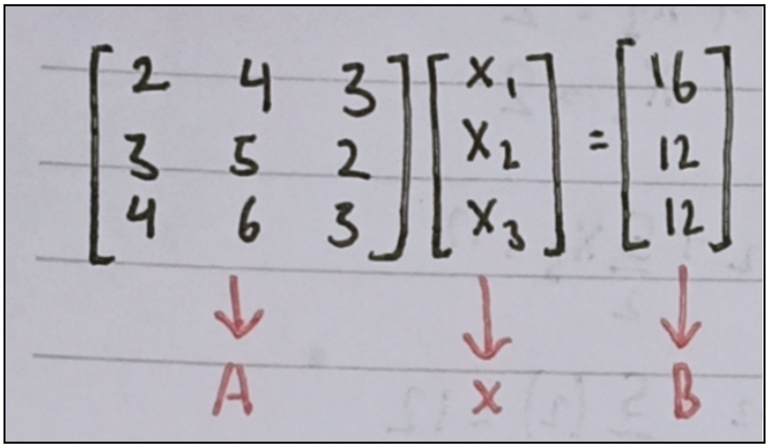
2.4 Matrix Balikan

Matrix balikan, atau yang biasa disebut matrix *inverse* (dan sering dilambangkan dengan A^-1), merupakan matrix yang jika dikalikan dengan matrix awalnya akan menghasilkan matrix identitas. Dengan kata lain, jika A adalah sebuah matrix, maka A * A^-1 = matrix identitas.

Ada beberapa syarat yang harus terpenuhi untuk mencari matrix balikan, yaitu:

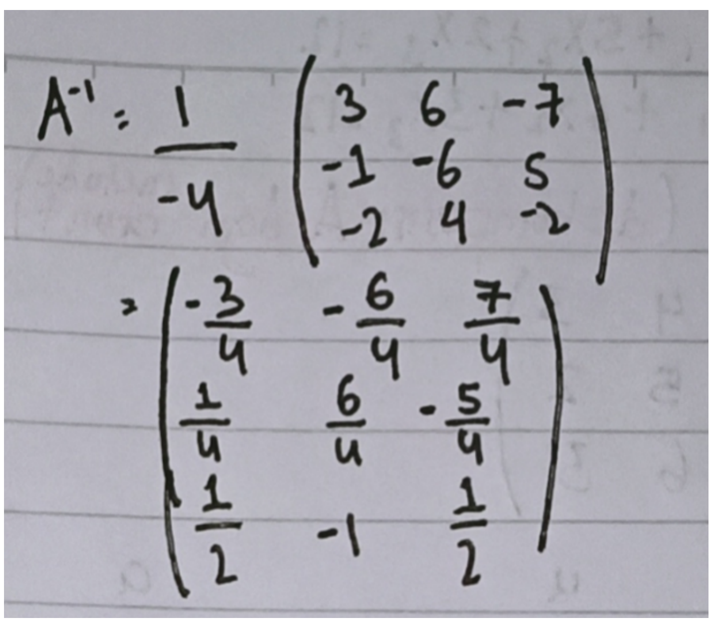
1. Matrix yang ingin di balikan adalah matrix persegi (besarnya n x n).
2. Determinan dari matrix yang ingin dibalik tidak sama dengan nol.

Matrix balikan mempunyai banyak peran dalam konsep aljabar linear, dan salah satu pengaplikasiannya ada pada pencarian Sistem Persamaan Linear. Aplikasi Matrix balikan pada SPL ini terdiri dari beberapa tahapan, yang pertama adalah ubah bentuk augmented matrix menjadi A * X = B.



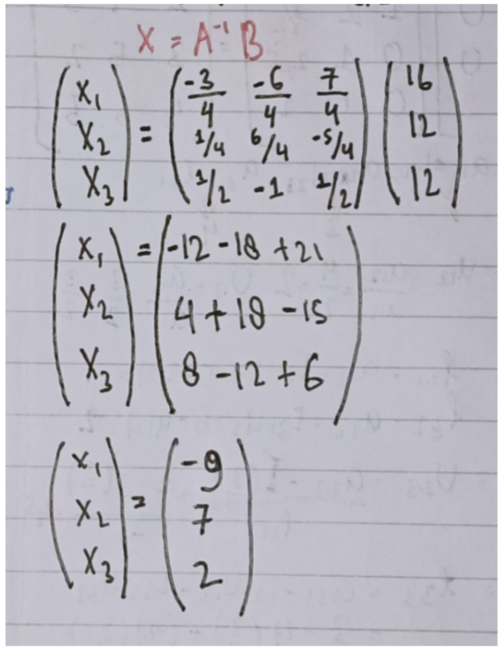
Gambar 2.11

Langkah selanjutnya adalah mencari matrix balikan dari matrix A:



Gambar 2.12

Langkah berikutnya, kita hanya perlu mengubah persamaa A * X = B menjadi X = A^-1 * B, yang menjadikan:



Gambar 2.13

2.5 Matrix Kofaktor

Matriks kofaktor adalah matriks yang terbentuk dari kofaktor-kofaktor matriks. Susunan elemen pada matriks sama saja dengan susunan pada matriksnya. Secara umum matriks kofaktor dapat ditulis sebagai berikut

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} & \cdots & C_{1n} \\ C_{21} & C_{22} & C_{23} & \cdots & C_{2n} \\ C_{31} & C_{32} & C_{33} & \cdots & C_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & C_{n3} & \cdots & C_{nn} \end{bmatrix}$$

Gambar 2.14 Matrix Cofaktor

2.6 Matrix Adjoin

Matrix Adjoint merupakan transpose dari matrix Cofaktor. Adjoint dari matrix dapat digunakan untuk menentukan balikan dari matrix, rumusnya adalah sebagai berikut.

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

Gambar 2.15 rumus menentukan Matrix Invers menggunakan Matrix Adjoin.

2.7 Kaidah *Crammer*

Kaidah cramer adalah salah satu metode untuk mencari SPL dalam Matrix dengan variabel jumlah persamaan dalam matrix tersebut. Perlu diketahui bahwa untuk kaidah ini bekerja perlu diasumsikan bahwa matrix koefisien dalam SPL bersifat non-singular, atau determinannya tidak sama dengan nol.

Untuk mencari nilai variabel suatu Xi dalam SPL matrix, kaidah cramer menggunakan rumus sebagai berikut:

$$Xi = \frac{|Ai|}{|A|}$$

Dimana $|Ai|$ adalah determinan matrix yang diperoleh dengan mengganti kolom ke-i matrix A dengan kolom matrix B.

2.8 Interpolasi Polinom

Interpolasi polinomial merupakan teknik interpolasi dengan mengasumsikan pola data yang kita miliki mengikuti pola polinomial baik berderajat satu (linier) maupun berderajat tinggi. Interpolasi dengan metode ini dilakukan dengan terlebih dahulu membentuk persamaan polinomial. Persamaan polinomial yang terbentuk selanjutnya digunakan untuk melakukan interpolasi dari nilai yang diketahui atau ekstrapolasi (prediksi) dari nilai diluar rentang data yang diketahui.

Metode untuk mencari persamaan interpolasi polinom adalah sebagai berikut :

1. Diberikan n + 1 buah titik yang berbeda.

$$(x_0, y_0), (x_1, y_1), ..., (x_n, y_n).$$

2. Akan terdapat suatu polinom $p_n(x)$ yang memenuhi:

$$y_i = p_n(X_i), \text{ untuk i = 1,2,3…}$$

Polinom tersebut diasumsikan memiliki bentuk :

$$p_n(x) = a_0 + a_1x + a_2x^2 + \; .... + a_nx^n$$

3. Buat matriks augmented dari persamaan $p_n(x)$tersebut

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n & y_0 \\ 1 & x_1 & x_1^2 & \cdots & x_1^n & y_1 \\ 1 & x_2 & x_2^2 & \cdots & x_2^n & y_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n & y_n \end{bmatrix}$$

Gambar 2.15 Matrix Augmented dari persamaan Polinom

4. Lakukan Eliminasi Gauss atau Eliminasi Gauss-Jordan pada matriks augmented agar mendapat nilai koefisien dari setiap variable(contoh dibawah ini menggunakan Eliminasi Gauss Jordan):

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & k_0 \\ 0 & 1 & 0 & \cdots & 0 & k_1 \\ 0 & 0 & 1 & \cdots & 0 & k_2 \\ \vdots & \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & 0 & 0 & 0 & 1 & k_n \end{bmatrix}$$

Sehingga didapat solusi :

$$a_0 = k_0$$
$$a_1 = k_1$$
$$a_2 = k_2$$
$$\vdots$$
$$a_n = k_n$$

2.9 Interpolasi *Bicubic Spline*

Interpolasi Bicubic adalah jenis interpolasi spline yang menggunakan polinominal kubik berbeda di antara dua titik data terdekat untuk menghasilkan interpolasi yang lebih halus. Interpolasi *bicubic spline* digunakan ketika terdapat serangkaian titik data dua dimensi yang membentuk kurva atau permukaan dalam ruang dua dimensi. Setelah itu akan dibangun fungsi S(x) yang mengestimasikan nilai Y pada setiap titik X dalam interval data.

Ada beberapa langkah yang harus dilakukan dalam membentuk *Spline*, yaitu:

1. Pembagian antara interval data menjadi segmen - segmen kecil yang dinamakan segmen spline. Pada setiap segmen ini, akan dibangin polinomial kubik yang menghubungkan dua titik data terdekat.
2. Dilakukannya polinomial kubik pada setiap segmen spline yang biasanya berbentuk seperti ini:

   $$Si(x) = ai(x - xi)^3 + bi(x - xi)^2 + ci(x - xi) + di$$

   Dalam persamaan tersebut, $Si(x)$ adalah polinomial kubik pada segmen spline ke-*i*, *x* adalah variabel yang ingin diinterpolasi, *xi* adalah titik awal segmen spline ke-*i*, dan *ai, bi, ci,* dan *di* adalah koefisien yang perlu ditentukan.

2.10 Regresi Linear Berganda

Analisis regresi digunakan untuk mengukur seberapa besar pengaruh antara variabel bebas dan variabel terikat. Apabila hanya terdapat satu variabel bebas dan satu variabel terikat, maka regresi tersebut dinamakan regresi linear sederhana (Juliandi, Irfan, & Manurung, 2014). Sebaliknya, apabila terdapat lebih dari satu variabel bebas atau variabel terikat, maka disebut regresi linear berganda. Regresi linear berganda merupakan model regresi yang melibatkan lebih dari satu variabel independen. Analisis regresi linear berganda dilakukan untuk mengetahui arah dan seberapa besar pengaruh variabel independen terhadap variabel dependen (Ghozali, 2018).

Metode untuk mencari persamaan Regresi Linear Berganda adalah sebagai berikut,

1. Misal terdapat k peubah ($x1, x2, x3, \ldots, xk$) pada suatu data.
2. Untuk melakukan regresi, dibutuhkan k buah titik data yang berbeda pada data tersebut
3. Akan terdapat sebuah fungsi $f(x_1, x_2, x_3, \ldots, x_k)$ yang memenuhi

   $$y_i = f(x_{1i}, x_{2i}, x_{3i}, \ldots, x_{ki})$$

   Fungsi tersebut memiliki bentuk :

   $$f(x_1, x_2, x_3, \ldots, x_k) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta x + \epsilon_i$$

4. Buatlah matriks augmented dari persamaan tersebut menggunakan rumus Normal Estimation Equation for Mutiple Linear Regression:

   $$\begin{bmatrix} k & \sum_{i=1}^{k} x_{1i} & \sum_{i=1}^{k} x_{2i} & \cdots & \sum_{i=1}^{k} x_{ki} & \sum_{i=1}^{k} y_i \\ \sum_{i=1}^{k} x_{1i} & \sum_{i=1}^{k} {x_{1i}}^2 & \sum_{i=1}^{k} x_{1i}x_{2i} & \cdots & \sum_{i=1}^{k} x_{1i}x_{ki} & \sum_{i=1}^{k} x_{1i}y_i \\ \sum_{i=1}^{k} x_{2i} & \sum_{i=1}^{k} x_{2i}x_{1i} & \sum_{i=1}^{k} {x_{2i}}^2 & \cdots & \sum_{i=1}^{k} x_{2i}x_{ki} & \sum_{i=1}^{k} x_{2i}y_i \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \sum_{i=1}^{k} x_{ki} & \sum_{i=1}^{k} x_{ki}x_{1i} & \sum_{i=1}^{k} x_{ki}x_{2i} & \cdots & \sum_{i=1}^{k} {x_{ki}}^2 & \sum_{i=1}^{k} x_{ki}y_i \end{bmatrix}$$

5. Lakukan Eliminasi Gauss atau Eliminasi Gauss-Jordan pada matriks augmented agar mendapat nilai koefisien dari setiap variabel (contoh dibawah merupakan hasil dari metode Eliminasi Gauss-Jordan):

   $$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & n_0 \\ 0 & 1 & 0 & \cdots & 0 & n_1 \\ 0 & 0 & 1 & \cdots & 0 & n_2 \\ \vdots & \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & 0 & 0 & 0 & 1 & n_k \end{bmatrix}$$

Bab 3
Implementasi pustaka

3.1 Class Matrix

- **Atribut**

Nama	**Tipe**	**Deskripsi**
mem	Public double[][]	Berisi elemen elemen matrix
numRows	Public int	Jumlah baris
numCols	Public int	Jumlah kolom

- **Metode**

Nama	**Tipe**	**Parameter**	**Deskripsi**
Matrix	Public Konstruktor	(int baris, int kolom)	Membuat Matrix dengan besar baris “baris” dan dengan besar kolom “Kolom”
getELMT	Public Double	(int row,int col)	Menghasilkan elemen pada pada matrix pada index baris row dan pada index kolom col
setElement	Public Double	(int row,int col, double value)	Menginput elemen dengan nilai value pada index baris row dan pada index kolom col
getLastIdxRow	Public Integer	-	Menghasilkan index terakhir pada baris
getLastIdxCol	Public Integer	-	Menghasilkan index terakhir pada kolom
readMatrix	Public Void	-	Membaca masukan matrix dengan menggunakan keyboard

printMatrix	Public Void	-	Menampilkan matrix pada terminal
copyMatrix	Public Matrix	(Matrix m)	Menghasilkan matrix dengan elemen-elemen yang sama pada matrix m
addMatrix	Public Matrix	(Matrix m1, Matrix m2)	Menghasilkan matrix baru dengan setiap elemennya merupakan penjumlahan elemen matrix m1 dengan matrix m2 pada index baris dan kolom yang sama
subtractMatrix	Public Matrix	(Matrix m1,Matrix m2)	Menghasilkan matrix baru dengan elemen-elemennya merupakan pengurangan elemen matrix m1 dengan matrix m2 pada index baris dan kolom yang sama
multiplyMatrix	Public Matrix	(Matrix m1, Matrix m2)	Menghasilkan matrix baru dengan elemen-elemennya merupakan perkalian elemen matrix m1 dengan matrix m2 pada index baris dan kolom yang sama
multiplyByConst	Public Matrix	(Matrix m, double x)	Menghasilkan matrix baru dengan elemen-elemennya merupakan perkalian elemen pada matrix m dengan x
divideRowByX	Public Matrix	(Matrix m, int row, double x)	Menghasilkan matrix baru dengan elemen-elemennya

			merupakan pembagian elemen matrix m dengan x
isMatrixEqual	Public boolean	(Matrix m1, Matrix m2)	Menghasilkan true jika matrix m1 sama dengan m2, menghasilkan false jika sebaliknya
isMatrixNotEqual	Public boolean	(Matrix m1, Matrix m2)	Menghasilkan true jika matrix m1 tidak sama dengan matrix m2, menghasilkan false jika sebaliknya
isBarisNol	Public boolean	(Int baris)	Menghasilkan true jika setiap elemen pada baris “baris” bernilai nol
countElmt	Public Int	(Matrix m)	Menghasilkan banyak elemen pada matrix m
createIdentity	Public Matrix	(Int n)	Menghasilkan matrix identitas dengan besar baris dan besar kolom n
createTranspose	Public Matrix	(Matrix m)	Menghasilkan matrix transpose dari matrix m
swapRows	Public Void	(Matrix matrix, int row1, int row2)	Menukar baris row1 dengan row2 pada matrix
hapusMinNol	Public Void	(Matrix matrix)	Mengubah elemen -0.0 pada matrix dengan 0.0

3.2 Class SPL

- **Atribut**

Class ini tidak memiliki atribut apapun

- **Metode**

Nama	Tipe	Parameter	Deskripsi
copyMatrix	Public Matrix	(Matrix m)	Menghasilkan matrix dengan elemen-elemen yang sama pada matrix m
swapRows	Public Void	(Matrix matrix, int row1, int row2)	Menukar baris row1 dengan row2 pada matrix
divideRowByX	Public Matrix	(Matrix m, int row, double x)	Menghasilkan matrix baru dengan elemen-elemennya merupakan pembagian elemen matrix m dengan x
hapusMinNol	Public Void	(Matrix matrix)	Mengubah elemen -0.0 pada matrix dengan 0.0
getLastIdxRow	Public Integer	-	Menghasilkan index terakhir pada baris
getLastIdxCol	Public Integer	-	Menghasilkan index terakhir pada kolom
Gauss	Public Matrix	(Matrix matrix)	Menerima matrix sembarang dan menghasilkan matrix eselon baris dengan menggunakan metode gauss
GaussJordan	Public Matrix	(Matrix matrix)	Menerima matrix sembarang dan menghasilkan matrix eselon baris tereduksi dengan metode gauss

			jordan
cekKondisi	Public Int	(Matrix matrix)	Menerima matrix eselon baris dan menghasilkan -1 jika SPL tidak memiliki solusi, menghasilkan 0 jika SPL memiliki solusi banyaj, menghasilkan 1 jika SPL memiliki solusi homogen
SPLGauss	Public void	(Matrix matrix)	Menerima matrix augmented, melakukan operasi gauss pada matrix tersebut, lalu meng-output penyelesaian SPL dari matrix tersebut.
SPLInverse	Public Matrix	(Matrix matrix)	Menerima matrix augmented, melakukan operasi inverse dari matrix segmen A, lalu mengalikan A^-1 dengan B, sehingga didapatkanlah matrix solusi dari X.
resolveParametric	Public String[]	(Matrix m)	Menerima matrix eselon baris dan menghasilan nama variabel parametrik untuk SPL dengan solusi banyak
SPLGaussJordan	Public void	(Matrix matrix)	Menerima matrix Augmented dan melakukan operasi penyelesaian SPL dengan menggunakan metode Gauss Jordan

3.3 Class ReduksiBaris

- **Atribut**

Class ini tidak memiliki atribut apapun

- **Metode**

Nama	**Tipe**	**Parameter**	**Derkripsi**
getLastIdxRow	Public Integer	-	Menghasilkan index terakhir pada baris
reduksibaris	Public Matrix	(Matrix matrix)	Menerima matrix dan menghasilkan matrix tereduksi
determinan	Public double	(Matrix matrix)	Menghasilkan determinan dari matrix dengan menggunakan metode reduksi baris

3.4 Class Cofaktor

- **Atribut**

Class ini tidak memiliki atribut apapun

- **Metode**

Nama	**Tipe**	**Parameter**	**Deskripsi**
getELMT	Public Double	(int row,int col)	Menghasilkan elemen pada pada matrix pada index baris row dan pada index kolom col
power	Public Int	(int num,int pow)	Menghasilkan num pangkat pow
minorMatrix	Public Matrix	(Matrix m, int row, int col)	

CofaktorExpansion			

3.5 Class Cramer

- **Atribut**

Class ini tidak memiliki atribut apapun

- **Metode**

Nama	Tipe	Parameter	Deskripsi
determinant	Private double	(Matrix m)	Memanggil Cofactor.CofactorExpansion untuk mendapatkan determinant matrix m (dilakukan untuk memudahkan pengetikan)
replaceRow	Public Matrix	(Matrix master, Matrix replacement, int targetRowidx)	Menukar baris
augmentedCramer			

3.6 Class Inverse

- **Atribut**

  Class ini tidak memiliki atribut apapun.

- **Metode**

Nama	Tipe	Parameter	Deskripsi
getLastIdxRow	Public Integer	-	Menghasilkan index terakhir pada baris
getLastIdxCol	Public Integer	-	Menghasilkan index terakhir pada kolom
getELMT	Public Double	(int row,int col)	Menghasilkan elemen

			pada pada matrix pada index baris row dan pada index kolom col
setElement	Public Double	(int row,int col, double value)	Menginput elemen dengan nilai value pada index baris row dan pada index kolom col
multiplyMatrix	Public Matrix	(Matrix m1, Matrix m2)	Menghasilkan matrix baru dengan elemen-elemennya merupakan perkalian elemen matrix m1 dengan matrix m2 pada index baris dan kolom yang sama
findInverse	Public Matrix	(Matrix matrix)	
determinan	Public Double	(Matrix matrix)	Menerima matrix dan menghasilkan determinan matrix dengan metode inverse

3.7 Class InterpolasiPolinom

- **Atribut**
  Class ini tidak memiliki atribut apapun.
- **Metode**

Nama	Tipe	Parameter	Deskripsi
getLastIdxRow	Public Integer	-	Menghasilkan index terakhir pada baris
getLastIdxCol	Public Integer	-	Menghasilkan index terakhir pada kolom
getELMT	Public Double	(int row,int col)	Menghasilkan elemen pada pada matrix pada index baris row dan pada index kolom

			col
GaussJordan	Public Matrix	(Matrix matrix)	Menerima matrix sembarang dan menghasilkan matrix eselon baris tereduksi dengan metode gauss jordan
cekKondisi	Public Int	(Matrix matrix)	Menerima matrix eselon baris dan menghasilkan -1 jika SPL tidak memiliki solusi, menghasilkan 0 jika SPL memiliki solusi banyaj, menghasilkan 1 jika SPL memiliki solusi homogen
resolveParametric	Public String[]	(Matrix m)	Menerima matrix eselon baris dan menghasilan nama variabel parametrik untuk SPL dengan solusi banyak
tampilkanPersamaan	Public void	(ArrayList<double> koefisienList)	Menerima ArrayList dari koefisien koefisien dari persamaan polinom dan menampilkan persamaan polinom tersebut
hitungNilai	Public double	(ArrayList<double> koefisienList,double x)	Menerima ArrayList dari koefisien koefisien polinom dan menghitung nilai dari F(x)
inputPolinom	Public void	-	Sebagai pemilihan cara input untuk memasukkan Matrix ke dalam class.
interpolasiPolinom	Public void	-	Menerima matrix

			augmented dan mencari hasil interpolasi polinom dari matrix tersebut.

3.8 Class BicubicSplineInterpolation

- **Atribut**
  Class ini tidak memiliki atribut apapun

- **Metode**

Nama	**Tipe**	**Parameter**	**Deskripsi**
BicubicSpline	Public double	(Matrix input, double a, double b)	Menerima input Matrix, dan mencari interpolasi bicubic spline dari titik (a, b)

3.9 Class LinearRegresion

- **Atribut**
  Class ini tidak memiliki atribut apapun

- **Metode**

Nama	**Tipe**	**Parameter**	**Deskripsi**
linearRegresion	Public double	-	Akan menerima input matrix (bisa dari input keyboard, maupun baca file), lalu mencari regresi linearnya.

Bab 4
Eksperimen

Contoh-contoh persoalan matrix, dan hasilnya

1. Hasil solusi SPL Ax = b

a.

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

```
0.00 0.00 0.00 0.00 1.00
Matrix tidak memiliki solusi.
```

b.

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

```
penyelesaian dari Matrix augmented tersebut adalah:
x1 = 1.00s  - 1.00r  + 3.00
x2 = s
x3 = Free
x4 = 1.00r  - 1.00
x5 = r
```

c.

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

```
0.00 0.00 0.00 0.00 1.00 -1.00 1.00
penyelesaian dari Matrix augmented tersebut adalah: X1 = 1.00, X2 = -2.00, X3 = 1.00.
Save to file? (Y/N)
```

d.

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

H adalah matriks *Hilbert*. Cobakan untuk $n = 6$ dan $n = 10$.

d.

2. SPL berbentuk matriks *augmented*

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

```
penyelesaian dari Matrix augmented tersebut adalah:
x1 = 1.00  - 2.00r  + 1.00s  - 1.00
x2 = 2.00r
x3 = r
x4 = s
```

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

```
0.00 0.00 0.00 0.00 0.00
penyelesaian dari Matrix augmented tersebut adalah:
x1 = -4.00  + 4.00
x2 = -4.00  + 6.00
x3 =  + 1.00
x4 =  + 1.00
```

3. Studi kasus linear berganda

Table 12.1: Data for Example 12.1

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

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

```
Hasil gauss jordan
1.00 0.00 0.00 0.00 -3.51
0.00 1.00 0.00 0.00 -0.00
0.00 0.00 1.00 0.00 0.00
0.00 0.00 0.00 1.00 0.15
```

```
Tafsirkan nilai Xk?(Y/N)
Y
Y
x1 : 0.5
x2 : 76
x3 : 29.30
Tafsiran nilai y : 1.068
```

4. Studi kasus Bicubic Spline Interpolation

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

f(0, 0)

```
Masukan elemen-elemen matrix:
21 98 125 153
51 101 161 59
0 42 72 210
16 12 81 96
Masukkan titik X yang ingin ditaksir:0
Masukkan titik Y yang ingin ditaksir:0
Hasil interpolasi f(0.0, 0.0) adalah: 21.0
```

f(0.5, 0.5)

```
Masukan elemen-elemen matrix:
21 98 125 153
51 101 161 59
0 42 72 210
16 12 81 96
Masukkan titik X yang ingin ditaksir:0.5
Masukkan titik Y yang ingin ditaksir:0.5
Hasil interpolasi f(0.5, 0.5) adalah: 218.03125
```

f(0.25, 0.75)

```
Masukan elemen-elemen matrix:
21 98 125 153
51 101 161 59
0 42 72 210
16 12 81 9
Masukkan titik X yang ingin ditaksir:0.25
Masukkan titik Y yang ingin ditaksir:0.75
Hasil interpolasi f(0.25, 0.75) adalah: 141.622802734375
```

f(0.1, 0.9)

```
Masukan elemen-elemen matrix:
21 98 125 153
51 101 161 59
0 42 72 210
16 12 81 9
Masukkan titik X yang ingin ditaksir:0.1
Masukkan titik Y yang ingin ditaksir:0.9
Hasil interpolasi f(0.1, 0.9) adalah: 105.98625100000002
```

Bab 5
Kesimpulan

Sistem persamaan linear dapat diatasi menggunakan berbagai metode, seperti Gauss, Gauss-Jordan, Cramer, dan Matriks Balikan. Setiap metode memiliki karakteristiknya sendiri dalam menyelesaikan SPL. Contohnya, metode Cramer dan Matriks Balikan hanya berlaku untuk matriks persegi dengan determinan non-nol.

SPL selalu memiliki solusi, tetapi ada tiga jenis solusi: solusi tunggal, solusi banyak, dan tidak ada solusi. Misalnya, pada kasus SPL, kami mendapatkan solusi tunggal dengan nilai $x1$=1.00, $x2$=−2.00, $x3$=-1.00.

Solusi banyak dapat ditemukan dalam kasus SPL tertentu, seperti pada kasus 1 bagian b dengan solusi $x1$=1s - 1r + 3, $x2$=s, $x3$=$Free$, $x4$=1r−1, dan $x5$=r.

Namun, ada juga kasus di mana SPL tidak memiliki solusi, seperti yang terjadi dalam kasus 1 bagian a, di mana ada baris dengan nilai variabel nol, tetapi nilai persamaannya bukan nol.

Selain itu, SPL memiliki berbagai aplikasi yang digunakan untuk menyelesaikan berbagai masalah, seperti Interpolasi Polinom, Regresi Linear Berganda, dan Bicubic Interpolation. Aplikasi ini memanfaatkan SPL untuk perhitungannya.

Misalnya, Interpolasi Polinom digunakan untuk memprediksi nilai dari fungsi yang telah diberikan, seperti dalam kasus 4 bagian b di mana kami memprediksi jumlah kasus positif.

Regresi Linear Berganda digunakan untuk mengestimasi nilai dalam suatu persoalan, seperti dalam kasus nomor 6 di mana kami mengestimasi nilai Nitrous Oxide.

Bicubic Interpolation adalah aplikasi lain yang menggunakan SPL untuk perbesaran citra, dan ini terlihat dalam kasus nomor 5.

Meskipun kami merasa bahwa pengerjaan tugas ini kurang maksimal, tetapi kami telah berusaha sebaik mungkin dan telah memberikan hasil yang baik secara keseluruhan dalam waktu yang telah diberikan.

Referensi

https://accounting.binus.ac.id/2021/08/12/memahami-analisis-regresi-linear-berganda/

https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-03-Sistem-Persamaan-Linier-2023.pdf

https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-04-Tiga-Kemungkinan-Solusi-SPL-2023.pdf

https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-05-Sistem-Persamaan-Linier-2023.pdf

https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-08-Determinan-bagian1-2023.pdf

https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-09-Determinan-bagian2-2023.pdf

https://clnazalia.wordpress.com/2018/10/27/sistem-persamaan-linier-metode-invers/

Howard Anton, Elementary Linear Algebra, 10th edition, John Wiley and Sons, 2010