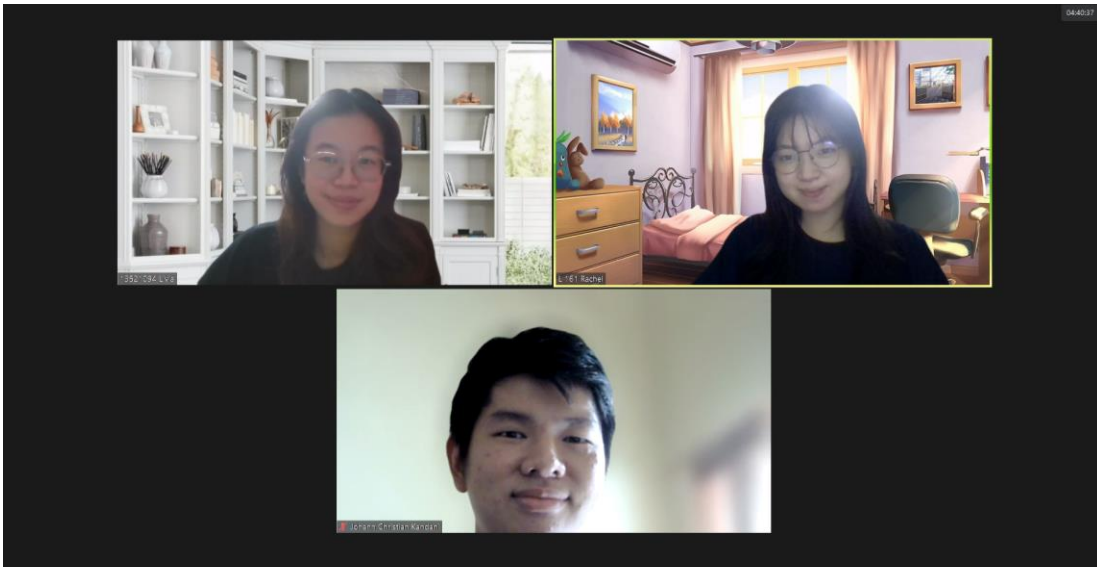
Tugas Besar 1 IF 2123 Aljabar Linear Dan Geometri

Sistem Persamaan Linear, Determinan, Dan Aplikasinya
Semester I Tahun 2022/2023



Kelompok JAR:

Rachel Gabriela Chen	13521044
Angela Livia Arumsari	13521094
Johann Christian Kandani	13521138

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2022

DAFTAR ISI

BAB I

Deskripsi Masalah

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

Di dalam Tugas Besar 1 ini, dibuat satu atau lebih library aljabar linear dalam Bahasa Java. Library tersebut berisi fungsi-fungsi seperti eliminasi Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung determinan, kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n peubah dan n persamaan). Selanjutnya, library tersebut digunakan di dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi.

BAB II

Teori Singkat

A. Eliminasi Gauss

Eliminasi Gauss adalah suatu metode menyelesaikan sistem persamaan linear dengan membuat matriks ke bentuk matriks eselon baris. Matriks eselon baris merupakan matriks dengan beberapa karakteristik, yaitu:

1. Jika suatu baris memiliki angka tidak 0, maka angka pertama yang tidak 0 adalah angka 1. Angka 1 ini disebut *leading one*.
2. Jika ada suatu baris yang seluruhnya 0, maka baris tersebut diletakkan di matriks paling bawah.
3. *Leading one* di baris yang lebih bawah berada di sebelah kanan dari *leading one* baris di atasnya.

Berikut merupakan contoh matriks yang berupa matriks eselon baris dengan tanda "*" bisa berisi angka berapa pun.

$$\begin{bmatrix}1 & * & * & * \\ 0 & 1 & * & * \\ 0 & 0 & 1 & * \\ 0 & 0 & 0 & 1\end{bmatrix}, \quad \begin{bmatrix}1 & * & * & * \\ 0 & 1 & * & * \\ 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0\end{bmatrix}, \quad \begin{bmatrix}1 & * & * & * \\ 0 & 1 & * & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0\end{bmatrix}$$

Gambar 2.1. Contoh matriks eselon baris

Cara untuk membuat matriks menjadi bentuk matriks eselon baris yaitu dengan melakukan Operasi Baris Elementer (OBE). Jenis operasi yang dapat dilakukan ada tiga, yaitu:

1. Menukar dua baris
2. Mengalikan suatu baris dengan bilangan bukan 0
3. Menambah kelipatan suatu baris ke baris lain

Setelah memperoleh matriks eselon baris melalui Operasi Baris Elementer, diperoleh beberapa persamaan. Dari persamaan-persamaan yang diperoleh, lakukan substitusi mundur untuk memperoleh solusi sistem persamaan linear.

B. Eliminasi Gauss Jordan

Eliminasi Gauss Jordan adalah suatu metode menyelesaikan sistem persamaan linear dengan membuat matriks ke bentuk matriks eselon baris tereduksi. Matriks eselon baris tereduksi memiliki karakteristik yang sama dengan matriks eselon baris dengan satu tambahan karakteristik, yaitu seluruh kolom yang mengandung *leading one* tidak memiliki angka lain selain 0 di kolom tersebut. Berikut merupakan contoh matriks yang berupa matriks eselon baris tereduksi dengan tanda "*" bisa berisi angka berapa pun.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 & 0 & * \\ 0 & 1 & 0 & * \\ 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 & * & * \\ 0 & 1 & * & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Gambar 2.2. Contoh matriks eselon baris tereduksi

Sama halnya seperti pada Eliminasi Gauss, membuat matriks menjadi bentuk matriks eselon baris tereduksi dapat dilakukan dengan memanfaatkan Operasi Baris Elementer (OBE). Dalam melakukan OBE pada Eliminasi Gauss Jordan, algoritma dapat dibagi menjadi dua bagian, yaitu: Jenis operasi yang dapat dilakukan ada tiga, yaitu:

1. Forward phase: tahap membuat 0 di bawah *leading one*, dengan kata lain membuat bentuk matriks eselon baris.
2. Backward phase: tahap membuat 0 di atas *leading one* yang menghasilkan matriks eselon baris tereduksi.

Setelah memperoleh matriks eselon baris tereduksi melalui Operasi Baris Elementer, diperoleh beberapa persamaan. Dari persamaan-persamaan yang diperoleh, dapat dicari solusi dari sistem persamaan linear.

C. Determinan

Untuk setiap matriks persegi, kita dapat mengasosiasikan suatu angka yang merupakan determinan matriks tersebut. Matriks persegi sendiri merupakan suatu matriks yang memiliki jumlah kolom dan baris yang sama. Determinan untuk suatu matriks A yang memiliki n baris dan kolom dilambangkan dengan

$$|A| = \det(A) = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}$$

Gambar 2.3. Lambang determinan matriks

Metode utama untuk mencari determinan yaitu metode reduksi baris dan ekspansi kofaktor. Metode reduksi baris memanfaatkan determinan matriks segitiga. Berikut merupakan contoh dari matriks segitiga.

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

Gambar 2.4. Contoh matriks segitiga atas

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

Gambar 2.5. Contoh matriks segitiga bawah

$$\det(A) = a_{11}a_{22}a_{33}a_{44}$$
$$\det(A) = a_{11}a_{22} \dots a_{nn}$$

Gambar 2.6. Rumus determinan matriks segitiga

Jika suatu matriks A adalah suatu matriks yang berukuran n x n, terdapat matriks B yang merupakan hasil manipulasi dari matriks A. Hubungan kedua determinan matriks tersebut diatur sebagai berikut:

1. Jika matriks A dikali sebuah barisnya dengan konstanta k sehingga menghasilkan matriks B, maka $\det(B) = k \det(A)$.
2. Jika matriks A dilakukan pertukaran antara dua baris sehingga menghasilkan matriks B, maka $\det(B) = -\det(A)$.

3. Jika sebuah matriks A ditambah kelipatan k baris lainnya sehingga menghasilkan matriks B, maka $\det(B) = \det(A)$.

Perhitungan determinan melalui reduksi baris dapat memanfaatkan properti determinan matriks segitiga dan aturan terkait determinan hasil manipulasi matriks. Pertama-tama dapat dilakukan Operasi Baris Elementer (OBE) untuk memperoleh matriks segitiga bawah atau atas. Setelah itu, determinan dari matriks dapat diperoleh dengan mempertimbangkan aturan determinan hasil manipulasi matriks.

Cara kedua untuk memperoleh determinan yaitu melalui ekspansi kofaktor. Untuk suatu matriks A yang berukuran n x n, didefinisikan M_{ij} yang merupakan minor entri dan C_{ij} yang merupakan kofaktor entri. Minor entri (M_{ij}) didapat dengan menghitung determinan dari submatriks yang elemen-elemennya tidak berada pada baris i dan kolom j. Sedangkan kofaktor entri dapat diperoleh melalui rumus berikut.

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

Gambar 2.7. Rumus kofaktor entri

Kofaktor entri berkoresponden dengan minor entri. Maka dari itu, jika sudah diketahui minor entri, kofaktor entri juga dapat diperoleh melalui pola tanda kofaktor entri.

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

Gambar 2.8. Pola kofaktor entri

Dengan memanfaatkan kofaktor entri, determinan suatu matriks A dapat diperoleh secara baris maupun kolos. Secara baris, determinan A dapat diperoleh sebagai berikut.

$$\det(A) = a_{11}C_{11} + a_{12}C_{12} + \cdots + a_{1n}C_{1n}$$
$$\det(A) = a_{21}C_{21} + a_{22}C_{22} + \cdots + a_{2n}C_{2n}$$
$$\dots$$
$$\det(A) = a_{n1}C_{n1} + a_{n2}C_{n2} + \cdots + a_{nn}C_{nn}$$

Gambar 2.9. Determinan kofaktor secara baris

Secara kolom, determinan A dapat diperoleh sebagai berikut.

$$\det(A) = a_{11}C_{11} + a_{21}C_{21} + \cdots + a_{n1}C_{n1}$$
$$\det(A) = a_{12}C_{12} + a_{22}C_{22} + \cdots + a_{n2}C_{n2}$$
$$\ldots$$
$$\det(A) = a_{1n}C_{1n} + a_{2n}C_{2n} + \cdots + a_{nn}C_{nn}$$

Gambar 2.10. Determinan kofaktor secara kolom

D. Matriks Balikan

Untuk suatu matriks persegi A berukuran n x n, didefinisikan matriks balikan A adalah A^{-1} sedemikian sehingga $AA^{-1} = A^{-1}A = I$ dengan I merupakan matriks identitas dari A. Matriks identitas merupakan matriks yang berelemen 1 pada diagonalnya dan berelemen 0 di seluruh tempat lainnya. Untuk I matriks identitas A didefinisikan $IA = A$. Suatu matriks A memiliki matriks balikan jika dan hanya jika $\det(A) \neq 0$. Untuk mendapatkan matriks balikan A, dapat digunakan dua metode, yaitu metode Eliminasi Gauss Jordan dan metode matriks adjoin.

Metode Elminasi Gauss Jordan dapat digunakan untuk menghitung matriks balikan. Untuk matriks A yang berukuran n x n, matriks balikannya dapat diperoleh dengan melakukan eliminasi secara simultan pada bentuk $[A \mid I]$ hingga menjadi $[I \mid A^{-1}]$. Dalam hal ini I adalah matriks identitas berukuran n x n.

Metode kedua yang dapat digunakan yaitu dengan memanfaatkan matriks adjoin. Matriks balikan A dapat diperoleh dengan rumus sebagai berikut.

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

Gambar 2.11. Rumus matriks balikan

E. Matriks Kofaktor

Misalkan A adalah suatu matriks berukuran n x n dan C_{ij} adalah kofaktor entri dari a_{ij}, maka matriks kofaktor dari A adalah

$$\begin{bmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \dots & \dots & \dots & \dots \\ C_{n1} & C_{n2} & \dots & C_{nn} \end{bmatrix}$$

Gambar 2.12. Matriks kofaktor

F. Matriks Adjoin

Matriks adjoin adalah matriks transpose dari matriks kofaktor. Matriks transpose adalah matriks yang diperoleh dengan menukar baris dan kolom suatu matriks.

G. Kaidah Cramer

Kaidah Cramer merupakan suatu kaidah yang dapat digunakan untuk mencari solusi dari sistem persamaan linear. Jika Ax = b adalah suatu persamaan linear dengan n buah persamaan dan n buah peubah sedemikian sehingga $\det(A) \neq 0$, maka sistem persamaan linear tersebut memiliki solusi yang unik yaitu sebagai berikut.

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

Gambar 2.13. Solusi SPL berdasarkan Kaidah Cramer

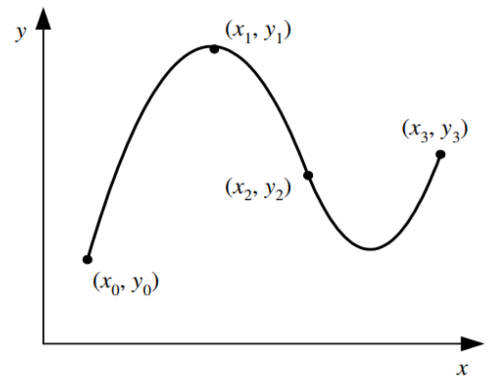
Dalam hal ini, A_j adalah matriks yang diperoleh dengan mengganti entri dari kolom j pada matriks A dengan matriks b.

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}$$

Gambar 2.14. Matriks b pada Kaidah Cramer

H. Interpolasi Polinom

Menginterpolasi titik-titik data artinya membuat sebuah kurva yang melalui titik-titik data tersebut. Bila fungsi yang didapat dari kurva tersebut berbentuk polinom, maka fungsi dapat disebut sebagai polinom interpolasi. Polinom interpolasi derajat n yang menginterpolasi titik-titik $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$ berbentuk $p_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$. Jika hanya ada dua titik, (x_0, y_0) dan (x_1, y_1), maka polinom yang menginterpolasi kedua titik tersebut adalah $p_1(x) = a_0 + a_1x$ yang berupa persamaan garis lurus. Jika tersedia tiga titik, $(x_0, y_0), (x_1, y_1)$ dan (x_2, y_2), maka polinom yang menginterpolasi ketiga titik tersebut adalah $p_2(x) = a_0 + a_1x + a_2x^2$, demikian seterusnya.



Gambar 2.15. Kurva polinom interpolasi

Dengan cara yang sama, kita dapat memeroleh polinom interpolasi berderajat n untuk n yang lebih tinggi asalkan tersedia (n+1) buah titik data. Dengan melakukan substitusi (x_i, y_i) ke dalam persamaan polinom $p_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$ untuk $i = 0, 1, 2, \ldots, n$ akan diperoleh n buah sistem persamaan lanjar dalam $a_0, a_1, a_2, \ldots, a_n$.

$$a_0 + a_1x_0 + a_2{x_0}^2 + \cdots + a_n{x_0}^n = y_0$$

$$a_0 + a_1x_1 + a_2{x_1}^2 + \cdots + a_n{x_1}^n = y_1$$

...

$$a_0 + a_1x_n + a_2{x_n}^2 + \cdots + a_n{x_n}^n = y_n$$

Gambar 2.16. Sistem persamaan lanjar hasil substitusi polinom

Solusi sistem persamaan lanjar ini, yaitu nilai $a_0, a_1, a_2, \ldots, a_n$ dapat diperoleh dengan menggunakan metode eliminasi Gauss. Sebagai contoh, diberikan tiga buah titik yaitu (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513). Tentukan polinom interpolasi kuadratik lalu estimasi nilai fungsi pada x = 9.2. Polinom kuadratik berbentuk $p_2(x) = a_0 + a_1x + a_2x^2$. Dengan melakukan substitusi ketiga buah titik data ke dalam polinom tersebut, diperoleh sisten persamaan lanjar sebagai berikut.

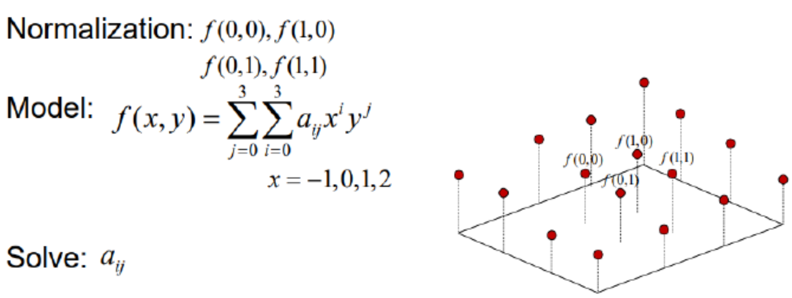
$$a_0 + 8.0a_1 + 64.00a_2 = 2.0794$$
$$a_0 + 9.0a_1 + 81.00a_2 = 2.1972$$
$$a_0 + 9.5a_1 + 90.25a_2 = 2.2513$$

Gambar 2.17. Sistem persamaan lanjar hasil substitusi titik

Penyelesaian sistem persamaan linear tersebut dengan metode eliminasi Gauss menghasilkan $a_0 = 0.6762,\ a_1 = 0.2266, a_2 = -0.0064$. Polinom interpolasi yang melalui ketiga buah titik tersebut adalah $p_2(x) = 0.6762 + 0.2266x - 0.0064x^2$. Dengan menggunakan polinom ini, maka nilai fungsi pada x = 9.2 dapat ditaksir sebagai berikut: $p_2(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(9.2)^2 = 2.2192$.

I. Interpolasi Bicubic

Interpolasi bicubic adalah suatu teknik pada data 2D yang umumnya digunakan dalam pembesaran citra. Interpolasi bicubic merupakan pengembangan dari interpolasi linear dan cubic. Diberikan sebuah matrix awal, misal M, kita akan mencari persamaan interpolasi $f(x,y)$ dengan pemodelan sebagai berikut:



Gambar 2.18. Pemodelan persamaan interpolasi

Melakukan substitusi nilai-nilai diketahui pada matriks 4 x 4 tersebut ke persamaan f(x,y) akan menghasilkan sebuah matriks persamaan:

$$y = Xa$$

$$\begin{bmatrix} f(-1,-1) \\ f(0,-1) \\ f(1,-1) \\ f(2,-1) \\ f(-1,0) \\ f(0,0) \\ f(1,0) \\ f(2,0) \\ f(-1,1) \\ f(0,1) \\ f(1,1) \\ f(2,1) \\ f(-1,2) \\ f(0,2) \\ f(1,2) \\ f(2,2) \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 2 & 4 & 8 & -1 & -2 & -4 & -8 & 1 & 2 & 4 & 8 & -1 & -2 & -4 & -8 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 \\ 1 & -1 & 1 & -1 & 2 & -2 & 2 & -2 & 4 & -4 & 4 & -4 & 8 & -8 & 8 & -8 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 8 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 1 & 2 & 4 & 8 & 2 & 4 & 8 & 16 & 4 & 8 & 16 & 32 & 8 & 16 & 32 & 64 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30} \\ a_{01} \\ a_{11} \\ a_{21} \\ a_{31} \\ a_{02} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{03} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}$$

Gambar 2.19. Hasil substitusi ke persamaan

Elemen pada matrix X adalah koefisien a_{ij} yang diperoleh dari persamaan f(x,y) di atas. Sebagai contoh, elemen pada baris 4 kolom ke 10 adalah koefisien dari a_{12} dan diperoleh dari $2^1 \times (-1)^2 \ = \ 2$, sesuai persamaan $x^i \times y^i$. Vektor a dapat dicari dari persamaan tersebut menggunakan metode inverse. Kemudian vektor a digunakan sebagai nilai variabel dalam f(x,y) sehingga terbentuk fungsi interpolasi bicubic sesuai model.

J. Regresi Linear Berganda

Regresi Linear merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Meskipun sudah ada rumus jadi untuk menghitung regresi linear sederhana, terdapat rumus umum dari regresi linear yang bisa digunakan untuk regresi linear berganda sebagai berikut.

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

Gambar 2.20. Persamaan umum regresi linear

Untuk mendapatkan nilai dari setiap β_i dapat digunakan Normal Estimation Equation for Multiple Linear Regression sebagai berikut.

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

Gambar 2.21. Normal Estimation Equation for Multiple Linear Regression

Sistem persamaan linear tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

BAB III

Impelementasi Pustaka dan Program

A. Implementasi Pustaka

Pustaka yang dibuat berisi Tipe Data Abstrak (Abstract Data Type) Matrix berserta fungsi-fungsi primitifnya, fungsi-fungsi seperti eliminasi Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung determinan, kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n peubah dan n persamaan), serta Utility yang memuat fungsi-fungsi yang dipakai berulang kali untuk memudahkan pembuatan pustaka dan program.

1. *Matrix.java*

Matrix.java memuat Abstract Data Type (ADT) Matrix beserta primitif-primitifnya.

- **Atribut**

  a) *public int* row : menyimpan jumlah baris matrix.
  b) *public int* col : menyimpan jumlah kolom matrix.
  c) *public double* mat[][] : menyimpan elemen-elemen matrix.

- **Metode**

  a) public Matrix()
  Konstruktor Matrix kosong dengan jumlah baris 0, jumlah kolom 0, dan matrix belum terdefinisi.

  b) public Matrix (int row, int col)
  Konstruktor Matrix dengan jumlah baris sesuai parameter row, jumlah kolom sesuai parameter col, dan membuat sebuah Matrix yang semua elemennya adalah 0 dengan ukuran sesuai parameter .

  c) public void displayMatrix()
  I.S. : Matrix terdefinisi.
  F.S. : Setiap elemen Matrix dicetak ke layar.

  d) public void inputMatrix()
  I.S. : Elemen Matrix sembarang.
  F.S. : Jumlah baris (m), jumlah kolom (n), elemen Matrix terdefinisi sesuai input pengguna.

  e) public void inputSquareMatrix()

I.S. : Elemen Matrix sembarang.
F.S. : Matriks berukuran n x n dengan elemen terdefinisi sesuai input pengguna.

f) public void readMatrix()
I.S. : Row dan col Matrix terdefinisi, elemen-elemen Matrix tidak terdefinisi.
F.S. : Elemen-elemen Matrix terdefinisi sesuai input pengguna.

g) public static Matrix CreateIdMat(int n)
I.S. : Matrix sembarang.
F.S. : Matrix merupakan Matrix identitas dengan ukuran n x n.

h) public Matrix transpose()
Mengembalikan hasil transpose dari Matrix

i) public void swapRow(int row1, int row2)
I.S. : Matrix terdefinisi.
F.S. : Elemen-elemen pada baris row1 dengan elemen-elemen pada baris row2 ditukar.

j) public void addRow (int row1, int row2, double k)
I.S. : Matrix terdefinisi.
F.S. : k kali elemen-elemen pada baris row2 ditambahkan ke elemen-elemen pada baris row1.

k) public void scalarMulti (double k)
I.S. : Matrix terdefinisi.
F.S. : Semua elemen Matrix telah dikalikan dengan sebuah konstanta k.

l) public void multRow (int row, double k)
I.S. : Matrix terdefinisi.
F.S. : Semua elemen Matrix pada baris row dikalikan dengan sebuah konstanta k.

m) public boolean isSquare()
Mengembalikan true jika Matrix adalah Matrix persegi, dan false jika sebaliknya.

n) public boolean isSingular()
Mengembalikan true jika Matrix adalah Matrix singular, dan false jika sebaliknya.

o) public static Matrix multiplyMat (Matrix m1, Matrix m2)
Mengembalikan Matrix hasil perkalian dari Matrix m1 dengan m2.

p) public static Matrix addMat (Matrix m1, Matrix m2)
Mengembalikan Matrix hasil penjumlahan m1 dengan m2.

q) public static Matrix subsMat (Matrix m1, Matrix m2)
Mengembalikan Matrix hasil pengurangan m1 dengan m2.

r) public static Matrix augMatrix (Matrix m1, Matrix m2)
Membentuk Augmented Matrix dari Matrix m1 dan m2

s) public void splitMatrix (Matrix m1, Matrix m2, int colNum)
I.S. : Matrix m1, m2 sembarang
F.S. : Matrix m1 merupakan memuat elemen-elemen pada colNum kolom pertama dari Matrix awal, dan m2 sisanya.

t) public void copyMatrix (Matrix m)
I.S. : Matrix m sembarang
F.S. : Matrix terduplikat ke Matrix m.

u) public void eliminasiGauss()
I.S. : Matrix terdefinisi
F.S. : Matrix dalam bentuk Matrix eselon baris.

v) public void eliminasiGaussJordan()
I.S. : Matrix terdefinisi
F.S. : Matrix dalam bentuk Matrix eselon baris tereduksi.

w) public Matrix getTempKofaktor(int p, int q)
Mengembalikan Matrix tanpa baris p dan kolom q.

x) public Matrix kofaktor()
Mengembalikan Matrix kofaktor.

y) public Matrix adjoint()
Mengembalikan Matrix adjoint.

2. *Utils.java*

Utils.java memuat fungsi dan prosedur yang dipakai berulang kali dalam pembuatan pustaka dan program.

- Atribut : --
- Metode

a) public static double toDouble (String str)
Mengembalikan nilai double dari str.

b) public static void matrixToFile (Matrix m)

I.S. : Matrix m terdefinisi.
F.S. : Menyimpan Matrix m ke dalam sebuah file jika pengguna ingin menyimpan file.

c) public static void stringToFile (String s)
I.S. : String s terdefinisi
F.S. : String s disimpan dalam sebuah file jika pengguna ingin menyimpan file.

d) public static Matrix readMatrixFromFile()
Mengembalikan Matrix dengan elemen-elemen yang dibaca dari sebuah file.

e) public static double setPrec (double num, int decPlaces)
Membuat presisi sebuah double sesuai dengan decPlaces.

3. *Determinant.java*

Determinant.java memuat fungsi yang dipakai berulang kali dalam perhitungan determinan matriks.

- Atribut : --
- Metode

a) public static double determinanEliminasiGauss(Matrix mdet)
Mengembalikan nilai determinan matriks mdet dengan metode reduksi baris bentuk segitiga menggunakan eliminasi Gauss.

b) public static double getKofaktor(Matrix m, int p, int q)
Mengembalikan nilai determinan matriks m tanpa baris p dan kolom q dengan metode ekspansi kofaktor.

c) public static double determinanKofaktor(Matrix m)
Mengembalikan nilai determinan matriks m dengan metode ekspansi kofaktor.

4. *Inverse.java*

Inverse.java memuat fungsi yang dipakai berulang kali dalam memperoleh matriks balikan.

- Atribut : --
- Metode

a) public static Matrix inversiGaussJordan(Matrix m)

Mengembalikan Matrix berupa matriks balikan dari Matrix m dengan matrix m harus berupa matriks persegi dengan dimensi lebih dari 0. Matriks balikan diperoleh melalui metode Gauss Jordan.

b) public static Matrix inversiKofaktor(Matrix m)
Mengembalikan Matrix berupa matriks balikan dari Matrix m dengan matrix m harus berupa matriks persegi dengan determinan ≠ 0. Matriks balikan diperoleh melalui matriks adjoint.

5. *SPL.java*

SPL.java memuat fungsi yang dipakai berulang kali dalam mencari solusi dari Sistem Persamaan Linear (SPL).

- Atribut : --
- Metode

a) public static String gaussMethod(Matrix m)
Mengembalikan string solusi persamaan linear yang didapatkan menggunakan metode eliminasi Gauss: $x_i = k + t_1 + t_2 + \cdots;\ t_j, k \in \mathbb{R}$ untuk setiap x_i yang memiliki nilai konstan atau berbentuk parametrik, dengan k adalah sebuah konstanta dan t_j adalah parameter untuk setiap persamaan dengan solusi banyak. Persamaan menjadi bentuk $x_i = k$ jika memiliki solusi tunggal. Jika persamaan tidak memiliki solusi, string yang dikembalikan adalah “Persamaan tidak memiliki solusi.”. Persamaan diselesaikan melalui substitusi balik yang nilainya disimpan dalam tabel *subs*.

b) public static String gaussjordanMethod(Matrix m)
Mengembalikan string solusi persamaan linear yang didapatkan menggunakan metode eliminasi Gauss-Jordan: $x_i = k + t_1 + t_2 + \cdots;\ t_j, k \in \mathbb{R}$ untuk setiap x_i yang memiliki nilai konstan atau berbentuk parametrik, dengan k adalah sebuah konstanta dan t_j adalah parameter untuk setiap persamaan dengan solusi banyak. Persamaan menjadi bentuk $x_i = k$ jika memiliki solusi tunggal. Jika persamaan tidak memiliki solusi, string yang dikembalikan adalah “Persamaan tidak memiliki solusi.”.

c) public static String inversMethod(Matrix m)
Mengembalikan string solusi persamaan linear yang didapatkan menggunakan metode inversi matriks $x = A^{-1}b$: $x_i = k;\ k \in \mathbb{R}$ dengan k adalah sebuah konstanta. Jika matriks A bukan matriks persegi, maka string yang dikembalikan adlaah "SPL tidak bisa diselesaikan dengan metode invers. Silakan gunakan metode lain.”, atau jika matriks tidak memiliki solusi unik, string yang dikembalikan adalah "SPL memiliki banyak solusi atau tidak memiliki solusi. Silakan gunakan metode lain.".

d) public static String cramersRule(Matrix m)

Mengembalikan string solusi persamaan linear yang didapatkan menggunakan kaidah Cramer: $x_i = k;\ k \in \mathbb{R}$ dengan k adalah sebuah konstanta. Jika matriks A bukan matriks persegi, maka string yang dikembalikan adlaah "SPL tidak bisa diselesaikan dengan kaidah Cramer. Silakan gunakan metode lain.", atau jika matriks tidak memiliki solusi unik, string yang dikembalikan adalah "SPL memiliki banyak solusi atau tidak memiliki solusi. Silakan gunakan metode lain.".

B. Program

1. *Main.java*

Main adalah program utama yang menampilkan menu pilihan-pilihan permasalahan yang ingin diselesaikan oleh pengguna, yaitu:

1. Sistem Persamaan Linear
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic
6. Regresi Linear Berganda
7. Keluar

Pengguna diminta untuk memasukkan pilihan permasalahan yang ingin diselesaikan. Program akan berhenti saat pengguna memilih pilihan keluar (7).

2. *SPLApp.java*

SPLApp.java memuat aplikasi dari pustaka untuk mencari solusi dari Sistem Persamaan Linear (SPL).

- Metode

a) public static boolean fromFile()
Mengembalikan true jika pengguna ingin memasukkan input dari file, dan false jika sebaliknya.

b) public static void menu()
Menampilkan pilihan-pilihan metode, yaitu:
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer

Prosedur menerima input pilihan metode, menampilkan pilihan input (keyboard/file) dan menerima pilihan input, serta menampilkan hasil penyelesaian SPL berdasarkan metode yang dipilih. Prosedur menerima input pilihan user untuk menyimpan atau tidak hasil yang didapat ke dalam file.

3. *DeterminantApp.java*

DeterminantApp.java memuat aplikasi dari pustaka untuk mencari nilai determinan matriks.

- Metode

a) public static boolean fromFile()

Mengembalikan true jika pengguna ingin memasukkan input dari file, dan false jika sebaliknya.

c) public static void menu()

Menampilkan pilihan-pilihan metode, yaitu:

5. Metode Reduksi Baris
6. Metode Ekspansi Kofaktor

Prosedur menerima input pilihan metode, menampilkan pilihan input (keyboard/file) dan menerima pilihan input, serta mencetak hasil determinan berdasarkan metode yang dipilih. Prosedur menerima input pilihan user untuk menyimpan atau tidak hasil yang didapat ke dalam file.

4. *InverseApp.java*

InverseApp.java memuat aplikasi dari pustaka untuk menyelesaikan permasalahan balikan matriks.

- Metode

a) public static boolean fromFile()

Mengembalikan true jika pengguna ingin memasukkan input dari file, dan false jika sebaliknya.

b) public static void menu()

Menampilkan pilihan-pilihan metode, yaitu:

1. Metode Eliminasi Gauss Jordan
2. Metode Kofaktor

dan menerima input pilihan metode, menampilkan pilihan input (keyboard/file) dan menerima pilihan input, mencetak hasil Matrix balikan berdasarkan metode yang dipilih user.

5. *InterpolasiApp.java*

InterpolasiApp.java memuat aplikasi dari pustaka untuk membuat polinom interpolasi dari titik-titik dan menghasilkan nilai taksiran dari sebuah nilai x.

- Metode

a) public static Matrix inputToMatrix(int n)
Mengembalikan Matrix berupa Augmented Matrix dari sistem persamaan lanjar yang dihasilkan dari input n pasang titik yang dimasukkan user.

b) public static void readFileInterpolasi(Matrix ret, double[] x)
I.S. : Matrix ret sembarang, double[] x memiliki panjang 1 dengan nilai sembarang
F.S. : Matrix ret berisi nilai titik-titik dari file yang dipilih user, double[] x berisi baris terakir dari file yang dipilih yang merupakan nilai x yang ingin ditaksir

c) public static Matrix fileToMatrix(Matrix inputMat)
Mengembalikan Matrix berupa Augmented Matrix dari sistem persamaan lanjar yang dihasilkan dari Matrix inputMat yaitu matriks hasil pembacaan titik dari file.

d) public static Matrix solusiMatrix(Matrix m)
Menghasilkan matriks eselon baris tereduksi yang merupakan solusi persamaan lanjar dari Augmented Matrix m.

e) public static String printPolinom(Matrix m)
Menghasilkan string polinom interpolasi dari Matrix m yang merupakan matriks solusi.

f) public static String printTaksiran(Matrix m, double x)
Menghasilkan string hasil taksiran dari substitusi nilai x ke polinom interpolasi yang dengan koefisien pada matriks solusi Matrix m.

g) public static void menu()
I.S. : ---
F.S. : Terdapat pilihan input (keyboard/file), hasil polinom interpolasi dan taksiran dicetak ke layar, pengguna dapat memilih untuk menyimpan hasil polinom interpolasi dan taksiran ke file.

6. *BicubicInterpolationApp.java*

BicubicInterpolationApp.java memuat aplikasi dari pustaka untuk membuat interpolasi bikubik dengan memanfaatkan persamaan y = Ax.

- Metode

a) public static Matrix getMatrixX()
Mengembalikan Matrix berukuran 16x16 yang merepresentasikan Matrix X untuk nilai masukan f(a,b) diaman a dan b pasti berada dalam rentang [0,1]

b) public static Matrix getMatrixY()
Mengembalikan Matrix berukuran 16x1 yang merepresentasikan Matrix y.

c) public static void readFile (Matrix inputMat, double[] point)
I.S. : inputMat dan point sembarang.
F.S. : inputMat terdefinisi yaitu nilai-nilai f(x,y) dan point terdefinisi yaitu point yang akan dicari interpolasi bikubiknya, keduanya dibaca dari sebuah file.

d) public static Matrix solve (double [] point, Matrix a)
I.S. : point terdefinisi, point[0] = a, point[1] = b
F.S. : Mengembalikan hasil interpolasi bikubik titik (a,b)

e) public static void menu()
I.S. : ---
F.S. : Terdapat pilihan input (keyboard/file), hasil f(a,b) dicetak ke layar, pengguna dapat memilih untuk menyimpan hasil f(a,b) ke file.

7. RLBApp.java

RLBApp.java memuat aplikasi dari pustaka untuk melakukan regresi linear berganda, kemudian menampilkan persamaan regresi dan juga menaksir nilai f(x_k).

- Metode

a) public static Matrix solve (Matrix m)
Membuat normal estimastion equation berdasarkan Matrix m dalam bentuk augmented matrix dan mengembalikan Matrix eselon tereduksi dari augmented matrix tersebut.

b) public static void readFile (Matrix ret, Matrix x)
Membaca semua nilai-nilai x_{1i}, x_{2i}, ..., x_{ni}, nilai y_i dari sebuah file, dan menyimpannya ke Matrix ret, serta membaca nilai-nilai x_k yang akan ditaksir nilai fungsinya dari file yang sama dan menyimpannya ke Matrix x.

c) public static void readKey (Matrix ret, Matrix x)
Membaca semua nilai-nilai x_{1i}, x_{2i}, ..., x_{ni}, nilai y_i dari keyboard, dan menyimpannya ke Matrix ret, serta membaca nilai-nilai x_k yang akan ditaksir nilai fungsinya dari keyboard dan menyimpannya ke Matrix x.

d) public static void output (Matrix m, Matrix x)

I.S. : Matrix m terdefinisi sebagai augmented matrix yang memuat koefisien persamaan regresi dan Matrix x terdefinisi sebagai matrix yang memuat nilai-nilai x_k yang akan ditaksir nilai fungsinya.

F.S. : Persamaan regresi dan hasil taksiran ditampilkan di layar dan atau tersimpan ke dalam sebuah file jika pengguna memilih untuk menyimpan hasil ke file.

e) public static void menu()

Menampilkan dan menerima pilihan input dari user (Keyboard/File) serta memanggil prosedur dan fungsi lain untuk menyelesaikan persoalan RLB.

BAB IV

Eksperimen

A. Solusi SPL Ax = b

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

a.

Output :

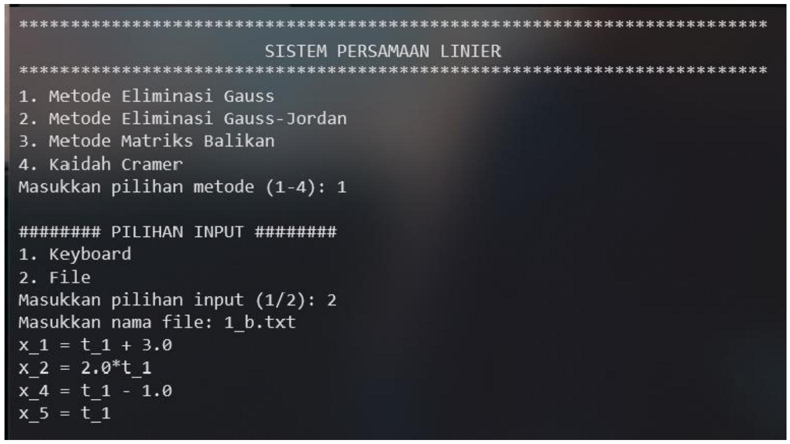
```
*************************************************************************
                        SISTEM PERSAMAAN LINIER
*************************************************************************
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
Masukkan pilihan metode (1-4): 1

######## PILIHAN INPUT ########
1. Keyboard
2. File
Masukkan pilihan input (1/2): 2
Masukkan nama file: 1_a.txt
SPL tidak memiliki solusi.
```

Dengan metode eliminasi Gauss, ditentukan bahwa SPL tersebut tidak memiliki solusi.

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

b.

Output :



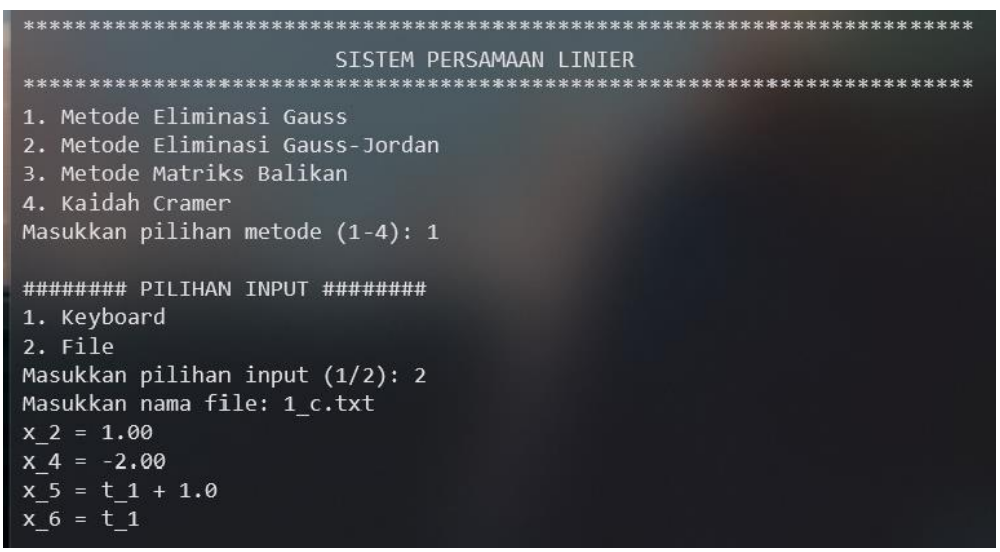
```
*************************************************************************
                        SISTEM PERSAMAAN LINIER
*************************************************************************
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
Masukkan pilihan metode (1-4): 1

######## PILIHAN INPUT ########
1. Keyboard
2. File
Masukkan pilihan input (1/2): 2
Masukkan nama file: 1_b.txt
x_1 = t_1 + 3.0
x_2 = 2.0*t_1
x_4 = t_1 - 1.0
x_5 = t_1
```

Dengan metode eliminasi Gauss, dapat dilihat bahwa variabel yang berpengaruh terhadap solusi SPL hanyalah x_1, x_2, x_4, dan x_5. Sedangkan, x_3 tidak dicetak ke layar karena nilai x_3 tidak berpengaruh terhadap variabel lain dalam solusi SPL.

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

c.

Output :



```
*************************************************************************
                        SISTEM PERSAMAAN LINIER
*************************************************************************
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
Masukkan pilihan metode (1-4): 1

######## PILIHAN INPUT ########
1. Keyboard
2. File
Masukkan pilihan input (1/2): 2
Masukkan nama file: 1_c.txt
x_2 = 1.00
x_4 = -2.00
x_5 = t_1 + 1.0
x_6 = t_1
```

Dengan metode eliminasi Gauss, dapat dilihat bahwa variabel yang berpengaruh terhadap solusi SPL hanyalah x_2, x_4, x_5, dan x_6. Sedangkan, x_1 dan x_3 tidak dicetak ke layar karena nilai x_1 dan x_3 tidak berpengaruh terhadap variabel lain dalam solusi SPL.

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

d.

Untuk n = 6:

```
***************************************************************************
                          SISTEM PERSAMAAN LINIER
***************************************************************************
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
Masukkan pilihan metode (1-4): 3

######## PILIHAN INPUT ########
1. Keyboard
2. File
Masukkan pilihan input (1/2): 2
Masukkan nama file: 1_d.txt
x_1 = 36.00
x_2 = -630.00
x_3 = 3360.00
x_4 = -7560.00
x_5 = 7560.00
x_6 = -2772.00
```

Untuk n = 10:

```
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
Masukkan pilihan metode (1-4): 3

######## PILIHAN INPUT ########
1. Keyboard
2. File
Masukkan pilihan input (1/2): 2
Masukkan nama file: 1_d2.txt
x_1 = 100.00
x_2 = -4949.80
x_3 = 79195.67
x_4 = -600560.53
x_5 = 2522331.25
x_6 = -6305780.06
x_7 = 9608745.47
x_8 = -8750772.98
x_9 = 4375365.28
x_10 = -923684.29
```

B. SPL berbentuk matriks augmented

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

a.

Output :

```
***************************************************************************
                          SISTEM PERSAMAAN LINIER
***************************************************************************
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
Masukkan pilihan metode (1-4): 1

######## PILIHAN INPUT ########
1. Keyboard
2. File
Masukkan pilihan input (1/2): 2
Masukkan nama file: 2_a.txt
x_1 = t_2 - 1.0
x_4 = t_2
x_2 = 2.0*t_1
x_3 = t_1
```

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

Output :

```
*************************************************************************
                         SISTEM PERSAMAAN LINIER
*************************************************************************
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
Masukkan pilihan metode (1-4): 2

######## PILIHAN INPUT ########
1. Keyboard
2. File
Masukkan pilihan input (1/2): 2
Masukkan nama file: 2_b.txt
x_1 = 0.00
x_2 = 2.00
x_3 = 1.00
x_4 = 1.00
```

C. SPL Berbentuk

a. $8x_1 + x_2 + 3x_3 + 2x_4 = 0$

$2x_1 + 9x_2 - x_3 - 2x_4 = 1$

$x_1 + 3x_2 + 2x_3 - x_4 = 2$

$x_1 + 6x_3 + 4x_4 = 3$

```
*************************************************************************
                         SISTEM PERSAMAAN LINIER
*************************************************************************
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
Masukkan pilihan metode (1-4): 4

######## PILIHAN INPUT ########
1. Keyboard
2. File
Masukkan pilihan input (1/2): 2
Masukkan nama file: 3_a.txt
x_1 = -0.22
x_2 = 0.18
x_3 = 0.71
x_4 = -0.26
```

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

b.

Output:

```
######## PILIHAN INPUT ########
1. Keyboard
2. File
Masukkan pilihan input (1/2): 2
Masukkan nama file: 3_b.txt
SPL tidak memiliki solusi.
```

D. Studi Kasus Interpolasi

a. Interpolasi nilai-nilai pada tabel :

x	0.4	0.7	0.11	0.14	0.17	0.2	0.23
$f(x)$	0.043	0.005	0. 058	0.072	0.1	0.13	0.147

```
Polinom yang melalui titik-titik tersebut yaitu:
f(x) =  - 4212.4345x^6 + 7102.3992x^5 - 4346.3140x^4 + 1220.8549x^3 - 163.9157x^2 + 10.2764x - 0.1846
```

```
Taksiran nilai fungsi dari nilai x yaitu:
f(0.2) = 0.1300
```

```
Taksiran nilai fungsi dari nilai x yaitu:
f(0.55) = 2.1376
```

```
Taksiran nilai fungsi dari nilai x yaitu:
f(0.85) = -66.2696
```

```
Taksiran nilai fungsi dari nilai x yaitu:
f(1.28) = -3485.1449
```

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:

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

Tanggal (desimal) adalah tanggal yang sudah diolah ke dalam bentuk desimal 3 angka di belakang koma dengan memanfaatkan perhitungan sebagai berikut: tanggal(desimal) = bulan + (tanggal / jumlah hari pada bulan tersebut).

Interpolasi Polinom perkiraan kasus Covid :

```
f(x) =  - 140994.5598x^9 + 9372906.0628x^8 - 275476226.8079x^7 + 4695835438.
3742x^6 - 51132198648.8689x^5 + 368553169420.9829x^4 - 1756821693899.8840x^3
 + 5334238927153.8950x^2 - 9347057986137.9340x + 7187117988941.8590
```

1. 16/07/2022 (Tanggal desimal 7.516)

```
f(7.516) = 53536.6680
```

2. 10/08/2022 (Tanggal desimal 8.323)

```
f(8.323) = 36292.8398
```

3. 05/09/2022 (Tanggal desimal 9.167)

```
f(9.167) = -667699.8594
```

4. **03/10/2022 (Tanggal decimal 10.097)**

```
f(10.097) = -332786332.3438
```

c. Sederhanakan fungsi

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

dengan polinom interpolasi derajat n di dalam selang [0, 2]. Pada eksperimen kali ini, digunakan n = 5.

```
Polinom yang melalui titik-titik tersebut yaitu:
f(x) =  0.2363x^5 - 1.4213x^4 + 3.2371x^3 - 3.5527x^2 + 2.0353x
```

E. STUDI KASUS INTERPOLASI BIKUBIK

Matriks input :

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

a. f(0, 0)

```
Hasil interpolasi bikubik f(0.00,0.00): 161.000000
```

b. f(0.5, 0.5)

```
Hasil interpolasi bikubik f(0.50,0.50): 97.726562
```

c. f(0.25, 0.75)

```
Hasil interpolasi bikubik f(0.25,0.75): 82.502075
```

d. f(0.1, 0.9)

```
Hasil interpolasi bikubik f(0.10,0.90): 74.696119
```

F. STUDI KASUS REGRESI LINEAR BERGANDA

Diberikan tabel data Nitrous Oxide:

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.

Dengan regresi linear berganda, didapatkan persamaan regresi:

```
f(x) = -3.5078 + -0.0026x1 + 0.0008x2 + 0.1542x3
```

Estimasi nilai Nitrous Oxide apabila Humidity bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30 :

```
f( 50.00  76.00  29.30  0.00 ) = 0.9384
```

BAB V

Kesimpulan, Saran dan Refleksi

KESIMPULAN

Operasi matriks, seperti penentuan determinan, penentuan matriks balikan, eliminasi Gauss, dan eliminasi Gauss Jordan dapat digunakan untuk menyelesaikan berbagai masalah. Salah satu contohnya adalah Sistem Persamaan Linear. Sistem Persamaan Linear sendiri sangat umum digunakan di dunia nyata untuk menyelesaikan berbagai persoalan, sehingga dibutuhkan penyelesaian SPL yang cepat. Hal tersebut dapat dicapai dengan membuat program komputer yang dapat menyelesaikan SPL dalam bentuk matriks.

Ada 3 macam penyelesaian SPL, yaitu banyak solusi, tidak ada solusi, dan satu solusi unik. Terdapat 4 metode penyelesaian SPL yang dibahas dalam laporan ini, yaitu metode Gauss, metode Gauss-Jordan, metode Invers, dan kaidah Cramer. Pada beberapa studi kasus yang dibahas pada BAB IV, ada beberapa SPL yang tidak dapat diselesaikan dengan metode Invers ataupun kaidah Cramer karena kedua metode ini hanya dapat menyelesaikan permasalahan matriks segiempat dengan determinan bukan 0. Sedangkan, metode Gauss dan metode Gauss-Jordan dapat menampilkan persamaan parametrik sebagai solusi SPL.

SPL kemudian dapat dimanfaatkan untuk melakukan Interpolasi Polinom, Interpolasi Bikubik, dan Regresi Linear Berganda. Interpolasi Polinom dapat digunakan untuk membuat sebuah persamaan polinom dari n buah titik yang diketahui dan memprediksi nilai-nilai yang tidak diketahui. Dalam BAB IV, interpolasi polinom digunakan untuk memprediksi jumlah kasus Covid-19 pada tanggal tertentu dan juga untuk menyederhakan sebuah fungsi Matematika. Interpolasi Bikubik digunakan untuk memprediksi nilai-nilai pada titik tertentu dengan mempertimbangkan 16 nilai yang berada di sekitar titik tersebut. Regresi Linear Berganda dapat digunakan untuk memprediksi sebuah nilai yang dipengaruhi oleh beberapa peubah. Dalam BAB IV, RLB digunakan untuk memprediksi Nitrous Oxide berdasarkan Humidity, Pressure, dan Temperature.

SARAN

Saran penulis untuk selanjutnya adalah masukan dan luaran untuk beberapa masalah sebaiknya lebih diperjelas dalam spek. Misalnya, apakah titik yang ditaksir nilainya juga dimasukkan ke dalam file masukan.

REFLEKSI

Secara umum, tim penulis dapat mengerjakan tugas besar ini dengan baik. Tim penulis juga berkesempatan untuk mempelajari beberapa materi baru, seperti interpolasi bikubik dan regresi linear berganda. Tim penulis juga berhasil membuat project Java dengan baik. Namun belum dapat mengerjakan soal bonus karena belum benar-benar bekerja dengan maksimal.

DAFTAR REFERENSI

Anton, H., Rorres, C., & Kaul, A. 2010. Elementary Linear Algebra: Applications Version (11th ed.). Wiley.

Munir, Rinaldi. 2022. “Sistem persamaan linear (Bagian 2: Metode Eliminasi Gauss Jordan)”. Diakses 30 September 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2022-2023/algeo22-23.htm

Munir, Rinaldi. 2022. “Determinan (Bagian 1: menghitung determinan dengan reduksi baris)”. Diakses 30 September 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2022-2023/algeo22-23.htm

Munir, Rinaldi. 2022. “Determinan (Bagian 2: menghitung determinan dengan ekspansi kofaktor)”. Diakses 30 September 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2022-2023/algeo22-23.htm

Weisstein, Eric W. "Gaussian Elimination." From MathWorld--A Wolfram Web Resource. Diakses 30 September 2022, dari https://mathworld.wolfram.com/GaussianElimination.html

Weisstein, Eric W. "Identity Matrix." From MathWorld--A Wolfram Web Resource. Diakses 30 September 2022, dari https://mathworld.wolfram.com/IdentityMatrix.html

REPOSITORY

Link repository dari Tugas Besar 1 IF 2123 Aljabar Linear Dan Geometri kelompok JAR adalah sebagai berikut.

https://github.com/chaerla/Algeo01-21044.git