LAPORAN TUGAS BESAR I

IF2123 ALJABAR LINEAR DAN GEOMETRI

Sistem Persamaan Linear, Determinan, dan Aplikasinya

Disusun oleh:

Agil Fadillah Sabri (13522006)

Bastian H. Suryapratama (13522034)

Haikal Assyauqi (13522052)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2023

DAFTAR ISI

BAB I
DESKRIPSI MASALAH

1. Tujuan

Tujuan tugas besar ini adalah sebagai berikut:

a. Membuat satu atau lebih *library* dalam bahasa pemrograman Java untuk:
   - Membuat fungsi untuk menemukan solusi Sistem Persamaan Linear dengan menggunakan Metode Eliminasi Gauss, Metode Eliminasi Gauss-Jordan, Metode Matriks Balikan, dan Kaidah Cramer.
   - Menghitung determinan sebuah matriks dengan Metode Reduksi Baris dan Ekspansi Kofaktor.
   - Mencari balikan sebuah matriks menggunakan Metode Reduksi Baris dan Metode Matriks Adjoin.

b. Menggunakan library yang telah dibuat untuk pengaplikasian metode numerik seperti:
   a. Interpolasi Polinomial.
   b. Regresi Linier Berganda.
   c. Bicubic Spline Interpolation.

2. Spesifikasi Tugas

a. 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

b. 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.

c. 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

d. 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.

e. 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 $x_1 = t$).

f. 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

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

g. 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 $f(0.5, 0.5)$.

h. Luaran program harus dapat ditampilkan pada layar komputer dan dapat disimpan ke dalam *file*.

i. 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).

j. Program tidak harus berbasis GUI, cukup *text-based* saja, namun boleh menggunakan GUI (memakai kakas Eclipse misalnya).

k. 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 submenu 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 II
TEORI SINGKAT

1. Eliminasi Gauss

Metode eliminasi Gauss sangat erat kaitannya dengan matriks eselon baris yang merupakan matriks dengan 1 utama pada setiap baris, kecuali baris yang seluruhnya adalah 0. Berikut contoh matriks eselon baris.

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

Gambar 1 Bentuk Matriks Eselon

Matriks eselon baris memiliki sifat-sifat yang antara lain adalah:

a. Jika sebuah baris tidak terdiri dari seluruhnya nol, maka bilangan tidak nol pertama di dalam baris tersebut adalah 1 (disebut 1 utama).
b. Jika ada baris yang seluruhnya nol, maka semua baris itu dikumpulkan pada bagian bawah matriks.
c. Di dalam dua baris berurutan yang tidak seluruhnya nol, maka 1 utama pada baris yang lebih rendah terdapat lebih jauh ke kanan daripada 1 utama pada baris yang lebih tinggi.

Mencari SPL dengan Eliminasi Gauss:

a. Nyatakan SPL dalam bentuk matriks *augmented*.
b. Terapkan OBE (operasi baris elementer pada matriks *augmented* sampai terbentuk matriks eselon baris).
c. Selesaikan persamaan yang berkorespondensi pada matriks eselon baris dengan teknik penyulihan mundur (*backward substitution*).

Contoh penyelesaian SPL dengan eliminasi Gauss:

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

Penyelesaian:

$$\begin{bmatrix} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{bmatrix} \underset{\sim}{R1/2} \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{bmatrix} \underset{R3+2R1}{\overset{R2-4R1}{\sim}} \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & -2 & -1 & -7 \\ 0 & 6 & -2 & 6 \end{bmatrix}$$

$$\underset{\sim}{R2/(-2)} \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 6 & -2 & 6 \end{bmatrix} \underset{\sim}{R3-6R2} \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & -5 & -15 \end{bmatrix} \underset{\sim}{R3/(-5)} \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

Gambar 2 Penyelesaian SPL dengan Eliminasi Gauss

2. Eliminasi Gauss-Jordan

Metode eliminasi Gauss-Jordan sangat erat kaitannya dengan matriks eselon baris tereduksi yang memiliki bentuk mirip dengan matriks eselon baris, yang hanya berbeda pada bagian atas dan bawah dari *leading one* memiliki nilai nol pada matriks eselon baris tereduksi.

Bentuk dari matriks eselon baris tereduksi yaitu:

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

Gambar 3 Matriks Eselon Tereduksi

Dalam penyelesaian SPL melalui eliminasi Gauss-Jordan, terdapat 2 tahapan yaitu fase maju dan fase mundur. Contoh penyelesaian SPL dengan eliminasi Gauss-Jordan:

$$\begin{aligned} 2x_1 - x_2 + 3x_3 + 4x_4 &= 9 \\ x_1 \qquad - 2x_3 + 7x_4 &= 11 \\ 3x_1 - 3x_2 + x_3 + 5x_4 &= 8 \\ 2x_1 + x_2 + 4x_3 + 4x_4 &= 10 \end{aligned}$$

Fase maju:

$$\begin{bmatrix} 2 & -1 & 3 & 4 & 9 \\ 1 & 0 & -2 & 7 & 11 \\ 3 & -3 & 1 & 5 & 8 \\ 2 & 1 & 4 & 4 & 10 \end{bmatrix} \text{R1} \leftrightarrow \text{R2} \begin{bmatrix} 1 & 0 & -2 & 7 & 11 \\ 2 & -1 & 3 & 4 & 9 \\ 3 & -3 & 1 & 5 & 8 \\ 2 & 1 & 4 & 4 & 10 \end{bmatrix} \begin{matrix} \text{R2} - 2\text{R1} \\ \text{R3} - 3\text{R1} \\ \text{R2} - 2\text{R1} \end{matrix} \begin{bmatrix} 1 & 0 & -2 & 7 & 11 \\ \mathbf{0} & -1 & 7 & -10 & -13 \\ \mathbf{0} & -3 & 7 & -16 & -25 \\ \mathbf{0} & 1 & 8 & -10 & -12 \end{bmatrix} (-1)*\text{R2}$$

$$\ldots \begin{bmatrix} 1 & 0 & -2 & 7 & 11 \\ 0 & 1 & -7 & 10 & 13 \\ 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & \mathbf{0} & -5 & -10 \end{bmatrix} (-1/5)*\text{R4} \underbrace{\begin{bmatrix} 1 & 0 & -2 & 7 & 11 \\ 0 & 1 & -7 & 10 & 13 \\ 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}}_{\text{Matriks eselon baris}}$$

Fase mundur:

$$\begin{bmatrix} 1 & 0 & -2 & 7 & 11 \\ 0 & 1 & -7 & 10 & 13 \\ 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix} \begin{matrix} \text{R1} + 2\text{R3} \\ \text{R2} + 7\text{R3} \end{matrix} \begin{bmatrix} 1 & 0 & \mathbf{0} & 5 & 9 \\ 0 & 1 & \mathbf{0} & 3 & 6 \\ 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix} \begin{matrix} \text{R1} - 5\text{R4} \\ \text{R2} - 3\text{R4} \\ \text{R3} + \text{R4} \end{matrix} \begin{bmatrix} 1 & 0 & 0 & \mathbf{0} & -1 \\ 0 & 1 & 0 & \mathbf{0} & 0 \\ 0 & 0 & 1 & \mathbf{0} & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

Gambar 4 SPL Menggunakan Eliminasi Gauss-Jordan

3. Kaidah Cramer

Kaidah Cramer digunakan untuk mencari solusi dari suatu Sistem Persamaan Linear yang memiliki banyak variabel dan berlaku ketika sistem memiliki solusi tunggal. Pencarian solusi dilakukan dengan menggunakan determinan matriks koefisien (dari sistem persamaan) dan determinan matriks lain yang diperoleh melalui penggantian salah satu kolom matriks koefisien dengan vektor yang terdapat pada sebelah kanan persamaan. Misal $Ax = b$ merupakan SPL yang terdiri dari k persamaan linear dengan k variabel. Jika $\det(A) \neq 0$, maka setiap variabel akan memiliki solusi unik, yaitu:

$$x_1 = \frac{\det(A_1)}{\det(A)}, x_2 = \frac{\det(A_2)}{\det(A)}, \ldots, x_k = \frac{\det(A_k)}{\det(A)}$$

Gambar 5 SPL Menggunakan Kaidah Cramer

4. Matriks Transpose

Transpose matriks adalah matriks baru yang elemen baris dan kolomnya merupakan elemen kolom dan baris matriks sebelumnya. Artinya, transpose matriks dibentuk oleh pembalikan elemen baris menjadi kolom dan elemen kolom menjadi baris. Jika matriks yang akan dijadikan transpose bukan matriks persegi, maka ordo pada transposenya merupakan kebalikan dari ordo matriks sebelumnya. Misalnya, matriks ordo 2 x 3 memiliki transpose matriks yang ordonya 3 x 2.

5. Matriks Kofaktor

Matriks kofaktor merupakan matriks yang terbentuk oleh kofaktor-kofaktor. Susunan elemen pada matriks ini juga sesuai dengan susunan pada matriksnya. Berikut contoh penulisan matriks kofaktor.

6. Matriks Adjoin

Adjoint dari suatu matriks persegi dapat didefinisikan sebagai transpose dari matriks kofaktornya, dengan penulisan adj(Matriks). Adjoint dari suatu matriks akan digunakan dalam mencari matriks balikan dari suatu matriks (jika menggunakan metode kofaktor).

7. Matriks Balikan

Balikan atau *invers* dari suatu matriks akan menghasilkan matriks identitas jika dilakukan perkalian dengan matriks awal ($A = I$). Umumnya, penggunaan matriks balikan bertujuan untuk mencari solusi dari Sistem Persamaan Linear yang memiliki solusi tunggal.

$$Ax = B$$

$$A^{-1}Ax = A^{-1}B$$

$$Ix = A^{-1}B$$

$$x = A^{-1}B$$

Gambar 6 Penggunaan Matriks Balikan dalam SPL

Matriks balikan dapat dicari menggunakan 2 cara:

a. Metode Kofaktor

Pada metode kofaktor determinan harus dicari terlebih dahulu. Apabila suatu matriks tidak memiliki determinan, maka matriks tersebut juga tidak memiliki matriks balikan. Namun, jika matriks memiliki determinan maka dapat dilanjutkan pada langkah selanjutnya yaitu mencari matriks kofaktor yang kemudian akan ditranspose sehingga akan menghasilkan matriks adjoint. Berikut rumus dari metode kofaktor dalam mencari matriks balikan.

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

Gambar 7 Persamaan Matriks Balikan

b. Metode Gauss-Jordan

Disebut juga metode reduksi baris, didapat dengan menggabungkan matriks utama dengan matriks identitas, lalu matriks utama dibentuk menjadi matriks identitas, dan matriks identitas yang telah dioperasi Gauss-Jordan menjadi matriks balikan.

8. Sistem Persamaan Linear

Sistem persamaan linear adalah sekumpulan persamaan linear yang terdiri dari beberapa variabel. Contohnya adalah:

$$
\begin{aligned}
3x + 2y - z &= 1 \\
2x - 2y + 4z &= -2 \\
-x + \tfrac{1}{2}y - z &= 0
\end{aligned}
$$

Gambar 8 Contoh Sistem Persamaan Linear

Sistem persamaan linear dapat diselesaikan dengan beberapa cara, di antaranya:

a. Eliminasi Gauss
b. Eliminasi Gauss-Jordan
c. Kaidah Cramer
d. Matriks Balikan

9. Determinan

Determinan merupakan sesuatu yang hanya dimiliki oleh matriks persegi, terdapat 2 cara dalam mencari matriks, antara lain:

a. Ekspansi Kofaktor

$$\text{Contoh: } A = \begin{bmatrix} 3 & 1 & -4 \\ 2 & 5 & 6 \\ 1 & 4 & 8 \end{bmatrix}$$

$$M_{11} = \begin{vmatrix} 5 & 6 \\ 4 & 8 \end{vmatrix} = 16,\ C_{11} = (-1)^{1+1}.M_{11} = 16$$

$$M_{32} = \begin{vmatrix} 3 & -4 \\ 2 & 6 \end{vmatrix} = 26,\ C_{32} = (-1)^{3+2}.M_{32} = -26$$

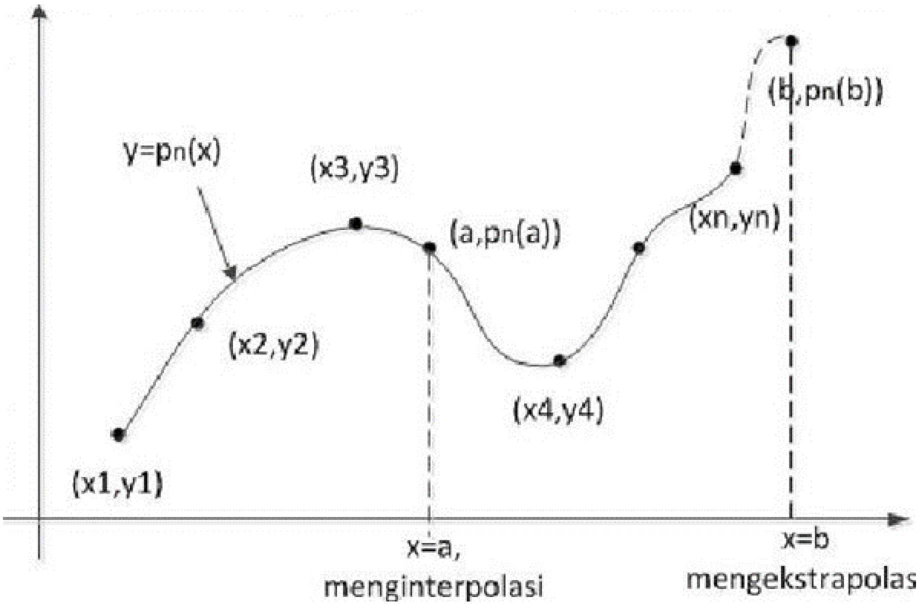
Gambar 9 Ekspansi Kofaktor

b. Reduksi Baris

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22} & a_{23} & a_{24} & a_{25} \\ 0 & 0 & a_{33} & a_{34} & a_{35} \\ 0 & 0 & 0 & a_{44} & a_{45} \\ 0 & 0 & 0 & 0 & a_{55} \end{bmatrix},\ \det(A) = a_{11}a_{22}a_{33}a_{44}a_{55}$$

Gambar 10 Reduksi Baris

10.Interpolasi Polinomial



Gambar 11 Grafik Interpolasi Polinom

Interpolasi Polinom merupakan suatu metode analisis numerik yang mengaproksimasi suatu nilai fungsi polinom berdasarkan beberapa titik data yang diketahui. Interpolasi Polinom dapat digunakan untuk memodelkan bagaimana data berubah-ubah terhadap variabel independennya. Selain membuat prediksi, Interpolasi Polinom juga dapat digunakan untuk menganalisis hubungan antara variabel-variabel data (berbanding terbalik, tidak berhubungan, berbanding lurus atau linear, berbanding kuadratik, dll).

Berikut langkah-langkahnya:

a. Misal diketahui (n+1) buah titik yang berbeda

$$(x_0, y_0)\ , (x_1, y_1)\ , (x_2, y_2)\ , \dots , (x_{n+!}, y_{n+1})$$

b. Akan terdapat suatu polinom Pn (x) yang memenuhi:

$$y_i = p_n(x_i), \forall i = 0, 1, 2, \dots, (n+1)$$

Polinomnya akan memiliki bentuk:

$$p_n(x) = a_0 + a_1 x + a_1 x^2 + \cdots + a_n x^n$$

c. Buat matriks a*ugmented* dari persamaan yang didapat sehingga diperoleh:

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

Gambar 12 Matriks Interpolasi Polinomial

d. Lakukan eliminasi hingga mendapat solusi akhir.

11. Regresi Linear Berganda

Regresi Linear Berganda merupakan metode statistika yang digunakan dalam pemodelan hubungan antara beberapa variabel independen terhadap suatu variabel dependen. Sesuai dengan namanya, Regresi Linear Berganda hanya dapat digunakan untuk membuat hampiran linear dari beberapa titik data yang berbeda. Tidak seperti Interpolasi Polinom yang dapat digunakan untuk membuat polinomial derajat n. Walau demikian, Regresi Linear Berganda dapat digunakan untuk memodelkan banyak variabel independen secara bersamaan, tidak seperti Interpolasi Polinom yang hanya dapat memodelkan satu variabel independen saja. Namun, keduanya memili persamaan dimana Regresi Linear Berganda juga dapat digunakan untuk memperkirakan hubungan antara variabel data independen (hubungan yang dapat ditinjau hanya yang merupakan hubungan berbanding terbalik, berhubungan, atau berbanding lurus / linear).

Berikut langkah-langkah pencarian persamaan Regresi Linear Berganda:

a. Misal terdapat k peubah $x_1, x_2, x_3, \dots, x_k$.
b. Dibutuhkan k buah titik data yang berbeda untuk regresi.
c. Akan terdapat fungsi $f(x_1, x_2, x_3, \dots, x_k)$ yang memenuhi $y_i = f(x_{1i}, x_{2i}, x_{3i}, \dots, x_{ki})$ fungsi tersebut memiliki bentuk:

$$f(x_1, x_2, x_3, \dots, x_k) = \beta_0 + \beta_1 x_1 + \beta_3 x_3 + \cdots + \beta_k x_k + \epsilon_i$$

d. Buat matriks *augmented*.

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

Gambar 13 Matriks Persamaan Regresi Linear Berganda

e. Lakukan eliminasi Gauss atau Gauss-Jordan.

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

Solusi:

$$\beta_0 = n_0$$
$$\beta_1 = n_1$$
$$\beta_2 = n_2$$
$$\vdots$$
$$\beta_k = n_k$$

Gambar 14 Solusi Regresi Linear Berganda

12. *Bicubic Spline Interpolation*

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

Persamaan polinomial yang digunakan antara lain:

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

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

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

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

Gambar 15 Empat Persamaan Bicubic Spline Interpolation

Persamaan tersebut dapat diubah ke dalam bentuk persamaan matriks sebagai berikut:

$$y = Xa$$

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

Gambar 16 Persamaan Matriks dari Rumus Sebelumnya

BAB III
IMPLEMENTASI

1. Garis Besar Program

Program Sistem Persamaan Linear, Determinan, dan Aplikasinya merupakan sebuah program berupa kalkulator matriks yang memiliki enam kemampuan, yaitu:

a. Mencari solusi dari sistem persamaan linear.
b. Mencari balikan/*inverse* sebuah matriks.
c. Mencari determinan sebuah matriks.
d. Mencari fungsi interpolasi polinom dari beberapa titik yang diketahui serta mencari tafsiran nilai fungsi yang ingin diketahui.
e. Mencari persamaan regresi linear berganda dari sekumpulan data yang diketahui dan persamaan yang diperoleh dapat digunakan untuk mencari nilai variabel terikat dari sekumpulan variabel bebas yang diketahui.
f. Melakukan *bicubic spline interpolation* dari sekumpulan empat buah titik yang diketahui nilai fungsi dan turunannya untuk mengaproksimasi nilai fungsi di sekitar titik-titik yang diketahui tersebut.

Di dalam program, pengguna akan diminta untuk memasukkan serangkaian instruksi tertentu untuk dapat menggunakan enam fungsi yang telah disebutkan diatas. Pengguna dapat memilih untuk memasukkan data langsung melalui terminal atau melalui pembacaan sebuah *file.txt*. Selain itu pengguna juga dapat memilih untuk melihat hasil perhitungan program langsung melalui terminal atau disimpan di dalam sebuah *file.txt*.

2. *Class* matriks

a. Atribut

Scanner scan = new Scanner(System.in);
{ Objek Scanner }

double[][] matrix;
{ Matriks }

int baris;
{ Baris sebuah matriks }

int kolom;
{ Kolom sebuah matriks }

b. Method

{ *Konstruktor* }
public matriks(int m, int n)
{ Membuat sebuah matriks berukuran m x n. }

{ *Getter* dan *Setter* }
public double getelmt(int i, int j)
{ Mendapatkan elemen sebuah matriks pada baris ke-i dan kolom ke-j. }

public int getbaris()
{ Mendapatkan banyak baris sebuah matriks. }

public int getkolom()
{ Mendapatkan banyak kolom sebuah matriks. }

public void setelmt (int i, int j, double val)
{ Memasukkan nilai ke elemen matriks pada baris i dan kolom j. }

{ *Method* }
public matriks UbahUkuran (int row, int col)
{ Mengembalikan sebuah matriks berukuran row x col. }

public matriks Identitas (int m, int n)
{ Membentuk sebuah matriks identitas berukuran mxn dengan syarat m = n. }

public void Gabung2Matriks (matriks mat1, matriks mat2)
{ Menggabungkan dua buah matriks menjadi satu. }

public void KurangiBaris()
{ Mengurangi baris sebuah matriks sebanyak satu. }

public void KurangiKolom()
{ Mengurangi kolom sebuah matriks sebanyak satu. }

public boolean IsRowEmpty (int baris)
{ Cek apakah baris suatu matriks kosong atau tidak. }

public boolean OnlyOne (int baris)
{ Cek apakah elemen di baris suatu matriks hanya satu. }

public void BacaMatriks()
{ Membaca matriks dari terminal dari masukan pengguna. }

public void BacaMatriksFile()
{ Membaca matriks dari sebuah *file.txt*. }

public matriks Multiply(matriks m1, matriks m2)
{ Mengembalikan sebuah matriks hasil perkalian matriks m1 dengan matriks m2. }

public boolean CekNol()
{ Cek apakah semua matriks bernilai nol atau tidak. }

public void CekSwap()
{ Melakukan pengecekan apakah *leading one* di satu baris lebih ke kanan di banding baris di bawahnya. }

public int HitungJumlahSwap()
{ Menghitung banyaknya percobaan *swap*. }

public void Swap (int baris1, int baris2)
{ Melkukan pertukaran baris antara baris1 dengan baris2. }

public void LeadingOne(int baris)
{ Cek apakah terdapat *leading one* dalam suatu baris. }

public int GetIdxLeadingOne(int baris)
{ Mendapatkan indeks *leading one* dalam suatu baris. }

public void BagiBaris(double pembagi, int baris)
{ Membagi sebuah baris pada matriks dengan konstanta (double pembagi). }

public int CekJumlahIndeks(int baris)
{ Mengitung banyak indeks yang bukan nol di suatu baris. }

public void CekBawah(int baris, int kolom)
{ Melakukan *forward step.* }

public void CekAtas(int baris, int kolom)
{ Melakukan *backward step*. }

public void KurangBaris(int baris1, int baris2, double koefisien)
{ Mengurangkan sebuah baris (baris1) dengan k (koefisien) kali baris lainnya (baris2). }

public void SolusiParametrik()
{ Melakukan penyelesaian untuk SPL dengan solusi parametrik (solusi banyak). }

public void allzero()
{ Mengubah semua nilai di matriks menjadi nol. }

public matriks SolusiUnik()
{ Melakukan penyelesaian untuk SPL dengan solusi tunggal/unik. }

public matriks BuatKofaktor()
{ Mengembealikan }

public void Transpose()
{ Mentranspose sebuah matriks. }

public double DeterminanBarisReduksi()
{ Mencari determinan sebuah matriks dengan metode reduksi baris. }

public double DeterminanKofaktor()
{ Mencari determinan sebuah matriks dengan metode ekspansi kofaktor. }

3. *Class* **SistemPersamaanLinear**

a. Atribut

Scanner scan = new Scanner(System.in);
{ Objek Scanner }

double[][] matrix;
{ Matriks }

int baris;
{ Baris sebuah matriks }

int kolom;
{ Kolom sebuah matriks }

b. Method

public static matriks Gauss(matriks m)
{ Menyelesaikan sebuah sistem persamaan linear menggunakan metode Gauss. }

public static matriks GaussJordan(matriks m)
{ Menyelesaikan sebuah sistem persamaan linear menggunakan metode Gauss-Jordan. }

public static matriks GaussBalikan(matriks m)
{ Melakukan proses eliminasi Gauss untuk mendapatkan matriks balikan. }

public static matriks SPLBalikan(matriks m)
{ Menyelesaikan sebuah sistem persamaan linear menggunakan metode matriks balikan. }

public static matriks Cramer(double determinanutama, matriks matriksutama)
{ Menyelesaikan sebuah sistem persamaan linear menggunakan metode Cramer. }

4. *Class* Determinan

a. Atribut

-

b. Method

public static double DeterminanBarisReduksi(matriks m)
{ Mengembalikan nilai determinan sebuah matriks menggunakan metode reduksi baris. }

public static double DeterminanKofaktor(matriks m)
{ Mengembalikan nilai determinan sebuah matriks menggunakan metode ekspansi kofaktor. }

5. *Class* Balikan

a. Atribut

-

b. Method

public static matriks BalikanAdjoin(matriks m)
{ Mengembalikan balikan sebuah matriks dengan menggunakan metode matriks adjoin. }

public static matriks BalikanReduksi (matriks m, matriks identitas)
{ Mengembalikan balikan sebuah matriks dengan menggunakan metode reduksi baris. }

6. *Class* InterpolasiPolinom

a. Atribut

-

b. Method

public static matriks dipangkat(matriks m)
{ Mengisi matriks sesuai ketentuan interpolasi polinom. }

public static double CariNilai(matriks m, double angka)
{ Mendapatkan nilai dari *f* (angka). }

7. *Class* MultipleLinearRegression

a. Atribut

-

b. Method

public static matriks findEquation(matriks sampleData)
{ Mencari matriks berisi koefisien-koefisien persamaan regresi. }

public static double predict(matriks equation, matriks independentVariables)
{ Mencari taksiran nilai y berdasarkan nilai-nilai x yang diberikan. }

private static matriks findNormalEstimationEquation(matriks sampleData)
{ Membuat matriks berisi koefisien-koefisien normal estimation equation. }

private static double calculateElementOfNEEMatrix(matriks sampleData, int indexRow, int indexColumn)
{ Menghitung masing-masing koefisien dari Normal Estimation Equation. }

8. *Class* BicubicSplineInterpolation

a. Atribut

-

b. Method

public static matriks getX()
{ Mendapatkan matriks X berukuran 16x16 pada persamaan y = Xa }

public static matriks getInversX(matriks X)
{ menginversekan matriks X }

public static matriks Matriks1Kolom(matriks M)
{ Mengubah matriks 4x4 menjadi matriks 16x1 agar sesuai dengan persamaan y = Xa }

public static matriks Matriks4Kolom(matriks M)
{ Mengubah matriks 16x1 menjadi matriks 4x4 }

public static double nilaiFungsi(matriks A, double a, double b)
{ Menghitung nilai taksiran f(a,b) dengan rumus persamaan 1
f(x,y) = sigma i[0..3] (sigma j [0..3] (a_ij * x^i * y^j)) }

9. *Class* Landing

a. Atribut

-

b. Method

public static void main(String[] args)
{ *Main Function* }

10. *Class* Output

a. Atribut

-

b. Method

public static void Pemisah()
{ Memberikan pemisah
"=="
Setiap kali berganti menu. }

public static void Pembukaan()
{ Menampilkan kalimat pembuka ketika pertama kali menjalankan program. }

public static void Menu()
{ Menampilkan pesan pilihan menu ke layar. }

public static void MetodeInput()
{ Menampilkan pesan pilihan metode input ke layar. }

public static void MetodeOutput()
{ Menampilkan pesan pilihan metode output ke layar. }

public static int mintaInputan()
{ Meminta metode input dari pengguna (1. Terminal, 2. file.txt). }

public static int mintaOutputan()
{ Meminta metode outpu dari pengguna (1. Terminal, 2. file.txt). }

public static void tulisHasilSPL(matriks HasilSPL, int metode)
{ Menampilkan hasil Sistem Persamaan Linear ke layar atau disimpan ke dalam file.txt tergantung metode output dari pengguna. Berlaku untuk solusi unik. }

public static void tulisHasilSPLparametrik(double[] angka, char[] nonangka, String[] kalimat , int metode, int panjang)
{ Menampilkan hasil Sistem Persamaan Linear ke layar atau disimpan ke dalam file.txt tergantung metode output dari pengguna. Berlaku untuk solusi parametrik. }

public static void tulisString(String kalimat, int metode)
{ Menampilkan suatu kalimat ke layar atau disimpan ke dalam file.txt tergantung metode output dari pengguna. Digunakan untuk menampilkan SPL yang tidak memiliki solusi dan matriks yang tidak memiliki invers. }

public static void tulisMatriks(matriks m, int metode)
{ Menampilkan sebuah matriks ke layar atau disimpan ke dalam file.txt tergantung metode output dari pengguna. }

public static void tulisHasilDeterminan(double determinan, int metode)
{ Menampilkan determinan sebuah matriks ke layar atau disimpan ke dalam file.txt tergantung metode output dari pengguna. }

public static void tulisHasilInterpolasi(matriks hasil, double hasilpersamaan, int metode, double angka)
{ Menampilkan persamaan yang diperoleh dari hasil interpolasi n buah titik serta nilai perkiraan untuk sebuah x0 ke layar atau disimpan ke dalam file.txt tergantung metode output dari pengguna. }

public static void tulisHasilBCS(int metode, matriks A, double a, double b)
{ Menampilkan nilai perkiraan sebuah titik (x,y) dengan menggunakan persamaan yang diperoleh dari metode Bicubic Spline Interpolation. }

public static String regressionEquationMatrixToString(matriks regressionEquation)
{Membentuk format persamaan regresi dengan string berdasarkan koefisien-koefisien persamaan regresi di dalam matriks}

public static String approximationValueToString(matriks independentVariables, double approximationValue)
{Membentuk format hasil taksiran dengan string berdasarkan nilai-nilai x dan nilai taksiran yang telah ditemukan}

BAB IV
EKSPERIMEN

1. Studi Kasus Sistem Persamaan Linear $A\mathrm{x} = b$

Dalam menu Sistem Persamaan Linear, program menerima inputan melalui terminal dengan format, meminta masukan banyak baris dan kolom terlebih dahulu, diikuti dengan meminta masukan elemen-elemen matriks A, dan diakhiri dengan meminta masukan elemen-elemen matriks b.

a. 1a

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

Hasil :

```
Tidak ada solusi.
```

Keempat metode penyelesaian SPL menghasilkan keluaran yang sama.

b. 1b

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

Hasil :

Dengan metode Eliminasi Gauss

```
Solusi SPL:
x1 = 3 + 1*(1.5 + 1.5*(-1 + 1p) + 0.5p) - 1p
x2 = 1.5 + 1.5*(-1 + 1p) + 0.5p
x3 = s
x4 = -1 + 1p
x5 = p
```

Dengan metode Eliminasi Gauss Jordan

```
Solusi SPL:
x1 = 3 + 1p
x2 = 2p
x3 = q
x4 = -1 + 1p
x5 = p
```

Karena matriks berukuran 4x5, maka hanya dapat menggunakan metode eliminasi Gauss dan eliminiasi Gauss-Jordan. Kedua hasil pada *test-case* di atas adalah sama. pada eliminasi Gauss dihasilkan bentuk yang lebih rumit karena tidak dilakukan proses *parsing* antara koefisien dengan variabel.

c. 1c

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

Hasil :

Dengan metode Eliminasi Gauss

```
Solusi SPL:
x1 = r
x2 = 2 + -1*(1 + 1p)
x3 = s
x4 = -2 + -1p
x5 = 1 + 1p
x6 = p
```

Dengan metode Eliminasi Gauss Jordan

```
Solusi SPL:
x1 = q
x2 = 1 + -1p
x3 = r
x4 = -2 + -1p
x5 = 1 + 1p
x6 = p
```

Kedua hasil pada *test-case* di atas adalah sama. pada eliminasi Gauss dihasilkan bentuk yang lebih rumit karena tidak dilakukan proses *parsing* antara koefisien dengan variabel.

d. 1d

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

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

Hasil:

n = 6

```
Solusi SPL:
X1 = 36.000000000980094
X2 = -630.0000000292657
X3 = 3360.000000203484
X4 = -7560.000000539233
X5 = 7560.000000603351
X6 = -2772.000000240222
```

n = 10

```
Solusi SPL:
X1 = 99.99035437047132
X2 = -4949.159598717291
X3 = 79181.9852859748
X4 = -600435.4050103598
X5 = 2521731.741596236
X6 = -6304125.909429149
X7 = 9606023.151161605
X8 = -8748135.347572234
X9 = 4373977.470228069
X10 = -923378.5098546298
```

Keempat metode penyelesaian SPL menghasilkan keluaran yang sama.

2. Studi Kasus Sistem Persamaan Linear berbentuk Matriks Augmented

a. 2a

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

Hasil :

```
Solusi SPL:
x1 = -1 + 1*(2p) - 2p + 1r
x2 = 2p
x3 = p
x4 = r
```

Dengan metode Eliminasi Gauss Jordan

```
Solusi SPL:
x1 = -1 + 1q
x2 = 2p
x3 = p
x4 = q
```

Kedua hasil pada test-case di atas adalah sama. pada eliminasi Gauss dihasilkan bentuk yang lebih rumit karena tidak dilakukan proses parsing antara koefisien dengan variabel.

b. 2b

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

Hasil:

```
Solusi SPL:
X1 = 0.0
X2 = 2.0
X3 = 1.0
X4 = 1.0
```

Baik eliminasi Gauss maupun Gauss-Jordan menghasilkan keluaran yang sama. Untuk metode Cramer matriks balikan tidak dapat dilakukan karena banyak kolom dan baris tidak sama.

3. Studi Kasus Sistem Persamaan Linear berbentuk Persamaan

a. 3a

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

Hasil :

```
Solusi SPL:
X1 = -0.2243
X2 = 0.1824
X3 = 0.7095
X4 = -0.2581
```

Keempat metode penyelesaian SPL menghasilkan keluaran yang sama.

b. 3b

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

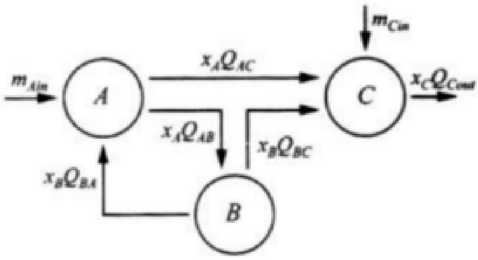
Hasil:

```
Tidak ada solusi.
```

Baik eliminasi Gauss maupun Gauss-Jordan menghasilkan keluaran yang sama. Untuk metode Cramer matriks balikan tidak dapat dilakukan karena banyak kolom dan baris tidak sama.

4. Studi Kasus Sistem Persamaan Linear berbentuk Cerita

Lihatlah sistem reaktor pada gambar berikut.



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

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

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

Hasil:

```
Solusi SPL:
X1 = 14.4444
X2 = 7.2222
X3 = 10
```

Keempat metode penyelesaian SPL menghasilkan keluaran yang sama.

5. Studi Kasus Interpolasi Polinom

Pada menu interpolasi polinom, program menerima masukan dari terminal dengan format, meminta masukan jumlah titik (n), diikuti dengan meminta masukan pasangan titik-titik (x,y) sebanyak n kali, dan diakhiri dengan meminta masukan nilai x yang henda ditaksir nilainya.

a. 5a

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

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

Lakukan pengujian pada nilai-nilai berikut:

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

Hasil:

x = 0.2

```
f(x) = -0x^6 + 0x^5 + 0.026x^4 + 0x^3 + 0.1974x^2 + 0.24x + -0.023
f(0.2) = 0.033
```

x = 0.55

```
f(x) = -0x^6 + 0x^5 + 0.026x^4 + 0x^3 + 0.1974x^2 + 0.24x + -0.023
f(0.55) = 0.1711
```

x = 0.85

```
f(x) = -0x^6 + 0x^5 + 0.026x^4 + 0x^3 + 0.1974x^2 + 0.24x + -0.023
f(0.85) = 0.3372
```

x = 1.28

```
f(x) = -0x^6 + 0x^5 + 0.026x^4 + 0x^3 + 0.1974x^2 + 0.24x + -0.023
f(1.28) = 0.6775
```

b. 5b

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)

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

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

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

a. 16/07/2022
b. 10/08/2022
c. 05/09/2022
d. Masukan user lainnya berupa **tanggal (desimal) yang sudah diolah** dengan asumsi prediksi selalu dilakukan untuk tahun 2022.

Hasil:

Tanggal 16 Juli 2022 (7,516)

```
f(x) = -140993.7122x^9 + 9372849.2391x^8 + -275474539.4207x^7 + 4695806315.4288x^6 + -51131876760.1328x^5 + 368550807175.5339x^4 + -1756810186361.3809x^3 + 5334203055240.195x^2
+ -9346993079172.328x + 7187066071657.867
f(7.516) = 53537.9961
```

Tanggal 10 Agustus 2022 (8,323)

```
f(x) = -140993.7122x^9 + 9372849.2391x^8 + -275474539.4207x^7 + 4695806315.4288x^6 + -51131876760.1328x^5 + 368550807175.5339x^4 + -1756810186361.3809x^3 + 5334203055
240.195x^2 + -9346993079172.328x + 7187066071657.867
f(8.323) = 36294.8477
```

Tanggal 5 September 2022 (9,167)

```
f(x) = -140993.7122x^9 + 9372849.2391x^8 + -275474539.4207x^7 + 4695806315.4288x^6 + -51131876760.1328x^5 + 368550807175.5339x^4 + -1756810186361.3809x^3 + 5334203055
240.195x^2 + -9346993079172.328x + 7187066071657.867
f(9.167) = -667693.2188
```

Tanggal 26 Juli 2022 (7,839)

```
f(x) = -140993.7122x^9 + 9372849.2391x^8 + -275474539.4207x^7 + 4695806315.4288x^6 + -51131876760.1328x^5 + 368550807175.5339x^4 + -1756810186361.3809x^3 + 5334203055
240.195x^2 + -9346993079172.328x + 7187066071657.867
f(7.839) = 28228.0781
```

c. 5c

Sederhanakan fungsi $f(x)$ yang memenuhi kondisi

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

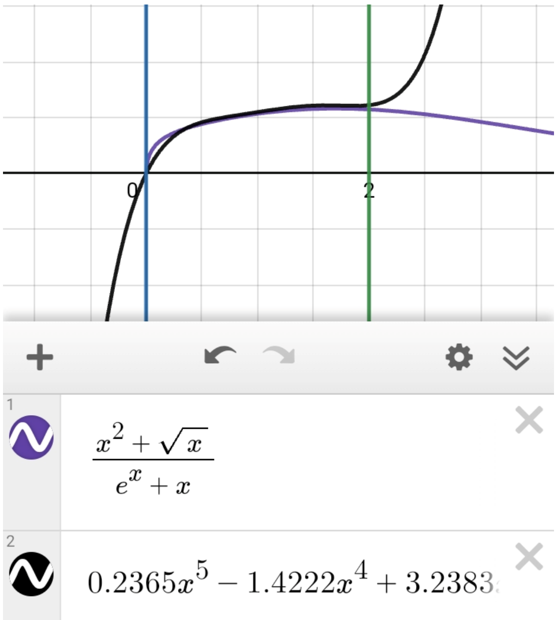
dengan polinom interpolasi derajat n di dalam selang [0, 2].
Sebagai contoh, jika n = 5, maka titik-titik x yang diambil di dalam selang [0, 2] berjarak $h = (2 - 0)/5 = 0.4$.

Untuk selang x [0,2] :

x = 0	y = 0,
x = 0.4	y = 0.4188
x = 0.8	y = 0.5071
x = 1.2	y = 0.5609
x = 1.6	y = 0.5836
x = 2	y = 0.5766

Hasil:

```
f(x) = 0.2365x^5 + -1.4222x^4 + 3.2382x^3 + -3.5528x^2 + 2.035x + 0
```



Dengan menggunakan aplikasi pembuat grafik, ditampilkan grafik fungsi awal $f(x)$ berwana ungu, sedangkan grafik fungsi hasil interpolasi polinomial berwarna hitam. Dapat dilihat dari gambar disamping bahwa untuk selang x [0,2], grafik fungsi interpolasi polinom mampu menghampiri nilai grafik fungsi $f(x)$ dengan cukup dekat.

6. Studi Kasus Regresi Linear Berganda

Pada menu regresi linear berganda, program menerima masukan dari terminal dengan format, meminta masukan banyaknya peubah x, kemudian meminta masukan banyaknya sampel yang diikuti dengan meminta masukan nilai sampel (dengan format x_{1i}, x_{2i}, x_{31}, …, x_{ni}, y_i), serta diakhiri dengan meminta masukan nilai-nilai x yang hendak dicari nilai y-nya.

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:

Dengan menggunakan 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, diperoleh hasil:

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

Taksiran nilai y:
f(50, 76, 29.3) = 0.9384
```

7. Studi Kasus *Bicubic Spline Interpolation*

Pada menu *bicubic spline interpolation*, program menerima masukan dari terminal dengan format, meminta masukan nilai fungsi pada titik (0,0), (1,0), (0,1), dan (0,0), serta nilai turunan fungsi terhadap x, terhadap y, dan terhadap xy di keempat titik tersebut, diikuti dengan meminta masukan nilai x dan y yang hendak dicari nilai tafsirannya.

Berikut adalah contoh masukan nilai fungsi pada titik (0,0), (1,0), (0,1), dan (0,0), serta nilai turunan fungsi terhadap x, terhadap y, dan terhadap xy di keempat titik tersebut

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

Tentukan:

$f(0, 0)$ = ?
$f(0.5, 0.5)$ = ?
$f(0.25, 0.75)$ = ?
$f(0.1, 0.9)$ = ?

Hasil:

$f(0, 0)$ (sesuai dengan matriks di atas)

```
Nilai f(0.0,0.0) = 21.0
```

$f(0.5, 0.5)$

```
Nilai f(0.5,0.5) = 87.796875
```

$f(0.25, 0.75)$

```
Nilai f(0.25,0.75) = 117.732177734375
```

$f(0.1, 0.9)$

```
Nilai f(0.1,0.9) = 128.57518699999997
```

BAB V
KESIMPULAN

1. Simpulan

a. *Library* telah diimplementasikan dalam program sederhana yang terdapat di bab III.
b. Program telah digunakan dalam menyelesaikan permasalahan mengenai sistem persamaan linear, pencarian invers serta determinan sebuah matriks, interpolasi polinom, dan regresi linear, *spline bicubic interpolation* sebagaimana dijelaskan lebih lanjut pada bab IV.

2. Saran

a. Program dapat ditingkatkan di bagian *error handling*.
b. Persamaan parametrik terutama ketika eliminasi Gauss dapat ditingkatkan lagi dengan melakukan operasi aritmatika pada koefisien dengan variabel yang sama.
c. Dapat mengaplikasikan GUI.

3. Refleksi

Kami belajar banyak dalam tugas ini. Pengalaman kami selama mengerjakan tugas beasr ini cukup memuaskan, tetapi karena ditambah banyak kuis dari mata kuliah lain serta baru mengenal bahasa pemrograman Java, terasa lebih berat. Dari segi kekurangan, kerja sama kami dapat ditingkatkan lagi sebagai suatu tim. Selain itu, komunikasi antaranggota juga masih kurang optimal sehingga menyebabkan ketidakmaksimalan pada penyelesaian dan pengembangan program.

DAFTAR PUSTAKA

https://celiksoftware.blogspot.com/2017/02/cara-mengecek-keberadaan-file-di-java.html (Diakses pada 30 September 2023).

https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-02-Matriks-Eselon.pdf (Diakses 26 September 2023).

https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-03-Sistem-Persamaan-Linier.pdf (Diakses 26 September 2023).

https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-05-Sistem-Persamaan-Linier-2.pdf (Diakses 26 September 2023).

https://stackoverflow.com/questions/3415507/how-can-i-compile-and-run-a-java-class-in-a-different-directory (Diakses 24 September 2023).

http://www.kitainformatika.com/2015/07/menulis-dan-membaca-file-berbentuk-txt.html (Diakses pada 28 September 2023).

https://www.mssc.mu.edu/~daniel/pubs/RoweTalkMSCS_BiCubic.pdf (Diakses pada 29 September 2023).

https://www.petanikode.com/java-file/ (Diakses pada 27 September 2023).