Tugas Besar 1 IF2123 Aljabar Linear dan Geometri

Aplikasi Matriks dalam Sistem Persamaan Linear, Determinan, dan Aplikasinya

13520003 – Dzaky Fattan Rizqullah

13520124 – Owen Christian Wijaya

13520157 – Thirafi Najwan Kurniatama

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2021

DAFTAR ISI

BAB 1: DESKRIPSI MASALAH

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Anda sudah mempelajari berbagai metode untuk menyelesaikan SPL, termasuk menghitung determinan matriks. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan (x = A -1b), 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, kami diminta untuk membuat satu atau lebih library aljabar linier 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 di dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi.

Spesifikasi program adalah sebagai berikut:

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

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

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

```
 3 4.5 2.8 10
-3 7 8.3 11
 0.5 -10 -9 12
```

3. Untuk persoalan interpolasi, masukannya jika dari keyboard adalah n, (x0, y0),(x1, y1), ..., (xn, yn), dan nilai x yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung. Misalnya jika titik-titik datanya adalah (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513), maka di dalam file text ditulis sebagai berikut:

```
8.0 2.0794
9.0 2.1972
9.5 2.2513
```

4. Untuk persoalan regresi, masukannya jika dari keyboard adalah n (jumlah peubah x), semua nilai-nilai $x_{1i}, x_{2i}, \ldots, x_{ni}$, nilai y_i, dan nilai-nilai x_k yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.

5. Untuk persoalan SPL, luaran (*output*) program adalah solusi SPL. Jika solusinya tunggal, tuliskan nilainya. Jika solusinya tidak ada, tuliskan solusi tidak ada. Jika solusinyabanyak, maka tuliskan solusinya dalam bentuk parametrik (misalnya $x_4 = -2, x_3 = 2s - t, x_2 = s,$ dan $x_1 = t$).

6. Untuk persoalan determinan dan matriks balikan, maka luarannya sesuai dengan persoalan masing-masing.

7. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan.

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

9. Bahasa program yang digunakan adalah Java.

10. Program dapat diubah dengan pilihan menu bebas. Misalnya, menu

```
MENU
1. Sistem Persamaaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
```

Bersama dengan submodul sesuai metode yang ada (ekspansi kofaktor dan reduksi baris untuk determinan, metode entri kofaktor dan reduksi baris untuk matriks balikan).

BAB 2: TEORI SINGKAT

2.1 Metode Eliminasi Gauss

Metode eliminasi Gauss adalah metode yang digunakan untuk mengeliminasi sistem persamaan linear menggunakan matriks *augmented.* Metode ini memanfaatkan operasi baris elementer pada awalnya dengan:

- Mengalikan sebuah baris dengan konstanta tidak nol
- Menukarkan dua buah baris
- Menambahkan sebuah baris dengan kelipatan baris lainnya

Metode eliminasi Gauss dilakukan hingga matriks *augmented* mencapai matriks eselon baris (matriks yang memiliki 1 utama pada setiap baris). Dengan mengambil persamaan di tiap baris, maka akan didapatkan persamaan atau hasil akhir dari setiap variabel yang ada dengan melakukan substitusi mundur (*backwards substitution)*. Bentuk matriks eselon baris adalah sebagai berikut.

$$\begin{bmatrix} 1 & ... & ... & ... \\ 0 & 1 & ... & ... \\ 0 & 0 & 1 & ... \end{bmatrix}$$

Ada tiga kemungkinan solusi metode eliminasi Gauss:

- Solusi unik, ketika baris terakhir tidak 0 semua dan:
$$x_n = c$$
- Solusi banyak/tidak terhingga, ketika baris terakhir 0 semua sehingga hasil yang didapatkan berbentuk parametrik
- Tidak ada solusi, ketika ada 1 utama di posisi n x n matriks
$$0x_0 + 0x_1 + 0x_2 + \cdots + 0x_n = 1$$

2.2 Metode Eliminasi Gauss-Jordan

Metode Eliminasi Gauss-Jordan mempunyai kemiripan dengan metode eliminasi Gauss, namun di metode ini, matriks direduksi ke bentuk matriks eselon baris tereduksi, di mana setiap baris mempunyai 1 kolom utama dan setiap kolom hanya mempunyai 1 nilai berupa 1 utama tersebut. Dengan demikian, tidak perlu dilakukan substitusi mundur, namun diperoleh nilai untuk setiap variabelnya tanpa harus mengeliminasi persamaan yang didapatkan. Bentuk matriks eselon baris tereduksi adalah sebagai berikut.

$$\begin{bmatrix} 1 & 0 & 0 & ... \\ 0 & 1 & 0 & ... \\ 0 & 0 & 1 & ... \end{bmatrix}$$

2.3 Determinan

Determinan adalah nilai yang dapat dihitung dari unsur-unsur dalam sebuah matriks persegi, biasanya dituliskan dengan notasi det(A) atau | a | (determinan dari matriks A). Untuk matriks 2×2,

$$|A| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

Untuk matriks berukuran lebih dari 2×2, ada beberapa metode untuk menentukan determinan, namun yang akan dibahas adalah pencarian determinan dengan metode reduksi baris dan metode ekspansi kofaktor.

2.3.1 Metode Reduksi Baris

Metode reduksi baris dilakukan dengan metode eliminasi Gauss untuk mendapatkan matriks segitiga, di mana semua nilai di atas atau di bawah diagonal utama adalah 0, namun tidak seperti matriks eselon baris yang mempunyai 1 utama. Setelah itu, determinan dapat didapatkan melalui persamaan berikut.

$$\det(A) = (-1)^p \times a_{11} \times a_{22} \times \ldots \times a_{nn}$$

dengan p = jumlah pertukaran baris yang dilakukan dan n = ukuran matriks.

2.3.2 Metode Ekspansi Kofaktor

Metode ekspansi kofaktor dilakukan dengan menghitung minor entri dengan menggunakan acuan berupa elemen pada baris atau kolom yang dipilih untuk mencari jumlah kofaktor entri. entri.

Minor entri adalah determinan submatriks yang elemen-elemennya tidak berada pada baris I dan kolom j. Misalkan minor entri dari matriks A:

$$A = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}, m_{00} = \begin{vmatrix} e & f \\ h & i \end{vmatrix}, m_{11} = \begin{vmatrix} a & c \\ g & i \end{vmatrix}, \ m_{01} = \begin{vmatrix} d & f \\ g & i \end{vmatrix}$$

Kofaktor entri dihitung menggunakan minor entri dan nilai positif/negatif berdasarkan letak matriks tersebut. Kofaktor entri dihitung dengan persamaan berikut.

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

Determinan sebuah matriks dapat dihitung dengan menjumlahkan kofaktor entri dari baris atau kolom yang dipakai sebagai acuan.

$$\det(A) = a_{1n}C_{1n} + a_{2n}C_{2n} + \cdots + a_{nn}C_{nn}$$

$$\det(A) = a_{n1}C_{n1} + a_{n2}C_{n2} + \cdots + a_{nn}C_{nn}$$

Menggunakan acuan baris 0,

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

$$= a(ei - hf) - b(di - gf) + c(dh - eg)$$

Menggunakan acuan kolom 1,

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

$$= \ -b\ (di - gf) + e\ (ai - cg) - h(af - cd)$$

2.4 Matriks Balikan

Matriks balikan dari suatu matriks adalah matriks yang jika dikalikan dengan matriks aslinya akan menghasilkan matriks identitas. Adanya matriks balikan ditentukan oleh nilai determinan. Apabila nilai determinan suatu matriks 0, maka matriks tersebut tidak mempunyai balikan. Sebaliknya, matriks akan mempunyai balikan jika matriks mempunyai nilai determinan. Ada dua metode mencari matriks balikan yang akan dibahas di tugas ini, yaitu menggunakan operasi baris elementer dan kofaktor entri. Metode operasi baris elementer akan meng-"ekspansi"-kan matriks dengan matriks identitas yang berukuran sama, sehingga akan berbentuk seperti demikian:

$$misalkan\ A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}, ekspansi\ menjadi \left|\begin{matrix} a & b & c \\ d & e & f \\ g & h & i \end{matrix}\right|\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

Setelah itu, reduksi ekspansi matriks menggunakan operasi baris elementer sehingga matriks di bagian kiri menjadi matriks identitas.

$$\left|\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}\right|\begin{matrix} \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \end{matrix}$$

Matriks invers yang didapatkan adalah matriks di sisi kanan. Untuk metode menggunakan kofaktor entri akan dibahas lebih lanjut di subbab 2.5.

2.5 Matriks Kofaktor dan Matriks Adjoin

Matriks kofaktor adalah matriks yang setiap elemennya digantikan kofaktor entri dari elemen tersebut.

Misalkan matriks A = $\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}$, maka matriks kofaktor dari A adalah $\begin{vmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{vmatrix}$

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

Matriks adjoin adalah transpose dari matriks kofaktor.

$$adj(A) = \begin{vmatrix} C_{11} & C_{21} & C_{31} \\ C_{12} & C_{22} & C_{32} \\ C_{13} & C_{23} & C_{33} \end{vmatrix}$$

Invers dari matriks A dapat dihitung dengan membagi setiap elemen di matriks adjoin dengan determinan A.

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

2.6 Kaidah Cramer

Kaidah Cramer adalah metode yang menggunakan determinan untuk menyelesaikan sistem persamaan linear yang mempunyai jumlah persamaan yang sama dengan jumlah variabel yang akan dicari. Kolom variabel yang hendak dicari akan diubah dengan kolom konstanta, lalu nilai variabel tersebut akan dicari dengan hasil pembagian determinan matriks dengan pergantian kolom dan determinan matriks konstanta. Kaidah Cramer dapat dilakukan dengan syarat det(A) $\neq 0$ dan ukuran matriks $n \times (n+1)$.

Misalkan persamaan linear:

$$\begin{gathered} ax + by + cz = c_1 \\ dx + ey + fz = c_2 \\ gx + hy + iz = c_3 \end{gathered}$$

Dalam bentuk persamaan Ax = B,

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}, B = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Dengan mengganti kolom di matriks A menggunakan matriks B, diperoleh

$$A_x = \begin{bmatrix} c_1 & b & c \\ c_2 & e & f \\ c_3 & h & i \end{bmatrix}, A_y = \begin{bmatrix} a & c_1 & c \\ d & c_2 & f \\ g & c_3 & i \end{bmatrix}, A_z = \begin{bmatrix} a & b & c_1 \\ d & e & c_2 \\ g & h & c_3 \end{bmatrix}$$

Akan diperoleh solusi unik berupa

$$x = \frac{\det(A_x)}{\det(A)}, y = \frac{\det(A_y)}{\det(A)}, z = \frac{\det(A_z)}{\det(A)}$$

2.7 Interpolasi Polinom

Interpolasi polinom dapat digunakan untuk menerka nilai yang akan dihasilkan oleh sebuah input x apabila dimasukkan ke suatu persamaan polinomial yang tidak diketahui. Kita mampu mengetahui persamaan polinomial tersebut apabila kita mempunyai informasi mengenai titik-titik apa saja yang melewati polinomial tersebut sedemikian rupa sehingga $y_n = p(x_n)$. Derajat tertinggi polinomial yang dapat dihasilkan adalah jumlah pasangan titik yang dimasukkan – 1. Apabila ada tiga pasangan titik, maka akan dihasilkan persamaan polinomial dengan pangkat tertinggi 2 yang akan berbentuk persamaan kuadrat.

Misalkan program menerima n pasang titik $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$, maka sistem persamaan linear yang terbentuk adalah:

$$a_0 + a_1x_0 + a_2x_0^2 + \cdots + a_nx_0^n = y_0$$

$$a_0 + a_1x_1 + a_2x_1^2 + \cdots + a_nx_1^n = y_1$$

$$\dots \dots$$

$$a_0 + a_1x_n + a_2x_n^2 + \cdots + a_nx_n^n = y_n$$

Melalui substitusi nilai x ke persamaan polinomial yang ada, kita mampu membuat matriks *augmented* berukuran $n \times n + 1$, lalu menggunakan Kaidah Cramer, kita dapat menemukan koefisien dari tiap operator yang ada. Setelah itu, kita dapat mensubstitusikan nilai x yang hendak dicari hasil bayangannya dan mendapatkan hasil aproksimasi p(x) dengan x merupakan titik sembarang.

2.8 Regresi Linear Berganda

Regresi linear berganda adalah metode untuk memperoleh nilai aproksimasi, sama dengan metode interpolasi polinom. Hanya saja, jika metode interpolasi polinomial menghasilkan persamaan polinomial, metode regresi linear berganda menghasilkan persamaan linear dengan n peubah.

$$f(x) = \beta_0 + \beta_1x_{1i} + \beta\ _2x_{2i} + \cdots + \beta_kx_{ki} + \epsilon_i$$

Dapat digunakan *Normal Estimation Equation for Multiple Linear Regression* sebagai berikut:

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

Sistem persamaan ini dapat diselesaikan dengan metode eliminasi Gauss.

BAB 3: IMPLEMENTASI PUSTAKA DAN PROGRAM DALAM JAVA

3.1 ADT Matriks (Matrix.java)

File ADT Matriks berisi assignment matriks apabila dipanggil. ADT Matriks ini yang digunakan untuk mengolah matriks di fungsi-fungsi yang akan dibahas selanjutnya. File ADT Matriks ini juga mengandung fungsi-fungsi inti yang mampu membantu pengolahan, seperti memberikan output terformat untuk sistem persamaaan linear, mengakses atau mengubah nilai elemen matriks, membaca elemen dan memberikan luaran matriks, mengubah baris yang ada, mengalikan / menjumlahkan elemen / baris matriks.

3.2 Program Utama (mainpr.java)

Mekanisme untuk memunculkan menu utama serta navigasinya diatur dalam modul mainpr.java. Di dalam modul ini diinisialisasi Scanner dan Matrix sementara yang nantinya akan diisi oleh input pengguna dari keyboard ataupun dari file .txt. Modul-modul pemrosesan matriks dapat diakses melalui submenu-submenu pada menu utama ini. Untuk keluar dari program, cukup dengan memilih opsi “Keluar”.

3.3 Sistem Persamaan Linear menggunakan Eliminasi Gauss (SPLGauss.java)

Fungsi akan menerima representasi SPL berupa matriks augmented. Program lalu akan mengubah matriks augmented tersebut menjadi bentuk Echelon Form dengan mencari Leading One di setiap baris dan memindahkan baris dengan isi nol semua ke bawah. Bila dalam pencarian ini terdapat baris dengan nol semua di kiri, namun tidak nol di kanan, Program akan mereturn 0 menandakan bahwa SPL tidak konsisten. Bila tidak ditemukan masalah, akan dilanjutkan dengan teknik backwards substitution dan akan mereturn 1 bila telah selesai menyederhanakan Echelon Form yang ada.

3.4 Sistem Persamaan Linear menggunakan Eliminasi Gauss-Jordan (SPLGaussJordan.java)

Fungsi akan menerima representasi SPL berupa matriks augmented. Program lalu akan mengubah matriks augmented tersebut menjadi bentuk Row Echelon Form dengan mencari Leading One di setiap baris dan memindahkan baris dengan isi nol semua ke bawah. Bila dalam pencarian ini terdapat baris dengan nol semua di kiri, namun tidak nol di kanan, Program akan mereturn 0 menandakan bahwa SPL tidak konsisten. Bila tidak ditemukan masalah, akan mereturn 1.

3.5 Sistem Persamaan Linear menggunakan Matriks Invers (SPLInverse.java)

Fungsi akan menerima representasi SPL berupa matriks augmented. Lalu program akan memvalidasi apakah bagian kiri matriks augmented adalah matriks singular atau bukan. Jika ya, maka SPL tidak dapat diselesaikan dengan cara ini. Jika tidak, maka dilanjutkan dengan mengalikan inverse kiri matriks augmented dengan kanan matriks augmented.

3.6 Sistem Persamaan Linear Kaidah Cramer (SPLCramer.java)

Program akan menerima parameter berupa matriks berukuran $n \times n$. Program kemudian akan menginisialisasi matriks baru untuk meletakkan matriks dengan pergantian kolom konstanta dan matriks baru untuk mengambil hanya elemen koefisien dari matriks m. Program kemudian akan menginisialisasi perulangan dengan *for loop* untuk mengisikan matriks kolom konstanta sesuai posisinya dan memetakan nilai hasil pembagian determinan matriks kolom konstanta tersebut ke dalam array float. Setelah itu, array akan di-return ke fungsi yang membutuhkan, seperti fungsi interpolasi polinom.

3.7 Matriks Invers menggunakan Operasi Baris Elementer (InverseOBE.java)

Program akan menerima matriks sembarang, lalu akan dilakukan validasi apakah matriks tersebut adalah matriks persegi, memiliki determinan nol, atau tidak. Bila bukan matriks persegi atau memiliki determinan yang nol, maka akan mereturn 0 yang menandakan matriks singular. Bila tidak, akan dilanjutkan dengan membuat matriks baru berukuran BARIS x 2*KOLOM dan membuat bagian kiri matriks tersebut menjadi Row Echelon Form. Lalu akan diduplikat matriks yang menjadi masukan dengan bagian kanan matriks baru tersebut dan akan mereturn 1 menandakan operasi berhasil dilakukan.

3.8 Matriks Invers menggunakan Matriks Kofaktor (InverseCofactor.java)

Program akan menerima parameter berupa matriks m berukuran $n \times n$. Apabila determinan matriks m sama dengan 0, program akan mengembalikan nilai 0 sebagai pertanda matriks tidak mempunyai balikan. Jika tidak, program kemudian akan membuat matriks baru berukuran $n \times n$ untuk kofaktor dan matriks penyimpanan sementara untuk mencari nilai kofaktor. Untuk setiap loop, program akan menghitung nilai determinan matriks penyimpanan sementara untuk dijadikan elemen matriks kofaktor. Setelah itu, matriks kofaktor akan di-transpose untuk memperoleh matriks adjoin. Tiap elemen matriks adjoin kemudian akan dikalikan dengan determinan untuk menghasilkan matriks invers.

3.9 Determinan menggunakan Ekspansi Kofaktor (Det.java)

Program akan menerima parameter berupa matriks m berukuran $n \times n$. Untuk kasus basis, program akan memberikan luaran berupa $m[0][0] \times m[1][1] - m[0][1] \times m[1][0]$ apabila ukuran matriks 2×2, dan luaran berupa $m[0][0]$ apabila matriks berukuran 1×1. Di luar kasus tersebut, program akan membuat matriks baru berukuran $(n-1) \times (n-1)$. Karena operasi perhitungan dapat dilakukan untuk satu baris saja, program akan memanfaatkan seleksi di baris 0. Program kemudian akan mengisikan matriks baru dengan elemen di matriks m yang tidak berada di baris 0 dan di kolom 0. Setelah itu, program akan menjumlahkan nilai determinan sementara dengan nilai yang ada sebelumnya menggunakan kofaktor entri. Program kemudian akan berjalan secara rekursif hingga basis (ukuran matriks 2×2) tercapai.

3.10 Determinan menggunakan Eliminasi Gauss (Det.java)

Program menerima matriks m berukuan $n \times n$ lalu menganalisis baris per baris. Program kemudian melakukan pengubahan baris apabila diperlukan dan menghitung jumlah terjadinya perubahan baris. Setelah itu, program akan mengubah baris menjadi bentuk triangular dan mengalikan nilai semua elemen diagonal yang ada. Program kemudian akan mengalikan nilai tersebut dengan $(-1)^p$ dengan p merupakan jumlah perubahan baris. \

3.11 Interpolasi Polinom (InterpolasiPolinom.java)

Program menerima banyaknya pasangan titik yang hendak dimasukkan ke dalam program, lalu menginisialisasi perulangan untuk menerima titik. Program kemudian meng-assign nilai x yang diperlukan ke matriks koefisien dan nilai y ke kolom terakhir sehingga terbentuk matriks *augmented* yang mewakili persamaan polinomial. Setelah itu, matriks tersebut akan diproses menggunakan penyelesaian sistem persamaan linear menggunakan Kaidah Cramer. Program kemudian akan memberikan luaran berupa persamaan polinomial Bersama koefisien persamaan yang telah ditemukan menggunakan Kaidah Cramer. Program kemudian menerima input berupa angka yang hendak ditaksir lalu memberikan luaran berupa hasil aproksimasi angka tersebut.

3.12 Regresi Linear Berganda

Program akan membaca matriks berupa serangkaian data persamaan. Lalu program akan menghitung berdasarkan rumus umum yang diberikan. Setelah selesai, program akan menampilkan prediksi dengan persamaan untuk parameter yang diberikan.

3.13 Mekanisme Baca Tulis file .txt

Mekanisme untuk membaca dan menulis file .txt diatur dalam modul ReadWriteText.java. Pembacaan dilakukan oleh fungsi readtxt() yang memiliki parameter Matrix dan Scanner. Pembacaan diawali dengan meminta input nama file dari user (dipastikan letak file .txt berada di dalam folder "/test" dan sudah terisi matriks sesuai format yang ditentukan), kemudian program akan membaca matriks di dalam .txt dan menyalinnya ke dalam parameter Matrix. Bila file tidak ditemukan, akan dimunculkan pesan error.

Sementara itu, Penulisan dilakukan oleh fungsi writetxt() yang memiliki parameter String dan Scanner. Penulisan diawali dengan meminta input nama file output dari user. Selanjutnya, bila file tersebut belum ada, maka program akan otomatis membuat file terlebih dahulu. Bila sudah ada, maka program akan langsung menuliskan string dari parameter String (pada baris baru jika sebelumnya sudah ada isinya. Setelah selesai menulis ke dalam .txt, akan dimunculkan pesan bahwa penulisan pada file selesai.

BAB 4: EKSPERIMEN

4.1 Percobaan 1: Solusi SPL dengan Format Ax = b

4.1.1 Percobaan 1a

a.

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

```
SISTEM PERSAMAAN LINIER
[1] Metode eliminasi Gauss
[2] Metode eliminasi Gauss-Jordan
[3] Metode matriks balikan
[4] Kaidah Cramer
[5] Kembali
Masukkan pilihan menu (1/2/3/4/5) :
1
1 Selected.
Masukkan pilihan input [1] keyboard | [2] file .txt (1/2) :
1
[1] Masukan Ax = B
[2] Matriks Augmented
1
Masukkan ukuran baris matriks A: 4
Masukkan ukuran kolom matriks A: 4
Masukkan matriks A:
1 1 -1 -1
2 5 -7 -5
2 -1 1 3
5 2 -4 2
Masukkan matriks B (ukuran baris x 1) dalam bentuk baris: 1 -2 4 6
1.00 1.00 -1.00 -1.00 1.00
2.00 5.00 -7.00 -5.00 -2.00
2.00 -1.00 1.00 3.00 4.00
5.00 2.00 -4.00 2.00 6.00
Hasil perhitungan menggunakan metode eliminasi Gauss:
Sistem inkonsisten.
1.00 1.00 -1.00 -1.00 1.00
0.00 1.00 -1.67 -1.00 -1.33
0.00 0.00 1.00 -1.00 1.00
0.00 0.00 0.00 0.00 1.00
```

Percobaan dilakukan dengan memasukkan matriks A dan B secara terpisah. Matriks A dan B kemudian akan digabungkan untuk menghasilkan matriks *augmented* untuk penyelesaian menggunakan eliminasi Gauss. Hasil yang diberikan adalah sistem inkonsisten, dengan eliminasi Gauss yang tidak memenuhi persyaratan.

4.1.2 Percobaan 1b

b.

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

```
SISTEM PERSAMAAN LINIER
[1] Metode eliminasi Gauss
[2] Metode eliminasi Gauss-Jordan
[3] Metode matriks balikan
[4] Kaidah Cramer
[5] Kembali
Masukkan pilihan menu (1/2/3/4/5) :
2
2 Selected.
Masukkan pilihan input [1] keyboard | [2] file .txt (1/2) :
1

[1] Masukan Ax = B
[2] Matriks Augmented
1
1 Selected.
Masukkan ukuran baris matriks A: 4
Masukkan ukuran kolom matriks A: 5
Masukkan matriks A:
1 -1 0 0 1
1 1 0 -3 0
2 -1 0 1 -1
-1 2 0 -2 -1
Masukkan matriks B (ukuran baris x 1) dalam bentuk baris: 3 6 5 -1
Hasil perhitungan menggunakan metode eliminasi Gauss-Jordan:
x1 = 3.00 + t1
x2 = 2.00t1
x4 = -1.00 + t1
```

Percobaan dilakukan dengan memasukkan matriks A dan B secara terpisah. Matriks A dan B kemudian akan digabungkan untuk menghasilkan matriks *augmented* untuk penyelesaian menggunakan eliminasi Gauss-Jordan. Hasil yang diberikan adalah solusi parametrik terkait variabel yang membutuhkan hasil parametris, sementara variabel yang bersifat bebas tidak dimasukkan.

4.1.3 Percobaan 1c

c.

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

```
SISTEM PERSAMAAN LINIER
[1] Metode eliminasi Gauss
[2] Metode eliminasi Gauss-Jordan
[3] Metode matriks balikan
[4] Kaidah Cramer
[5] Kembali
Masukkan pilihan menu (1/2/3/4/5) :
2
2 Selected.
Masukkan pilihan input [1] keyboard | [2] file .txt (1/2) :
1

[1] Masukan Ax = B
[2] Matriks Augmented
1
1 Selected.
Masukkan ukuran baris matriks A: 3
Masukkan ukuran kolom matriks A: 6
Masukkan matriks A:
0 1 0 0 1 0
0 0 0 1 1 0
0 1 0 0 0 1
Masukkan matriks B (ukuran baris x 1) dalam bentuk baris: 2 -1 1
Hasil perhitungan menggunakan metode eliminasi Gauss-Jordan:
x2 = 1.00 - t1
x4 = -2.00 - t1
x5 = 1.00 + t1
```

Percobaan dilakukan dengan memasukkan matriks A dan B secara terpisah. Matriks A dan B kemudian akan digabungkan untuk menghasilkan matriks *augmented* untuk penyelesaian menggunakan eliminasi Gauss-Jordan. Hasil yang diberikan adalah solusi parametrik terkait variabel yang membutuhkan hasil parametris, sementara variabel yang bersifat bebas tidak dimasukkan.

4.1.4 Percobaan 1.4

```
Masukkan pilihan menu (1/2/3/4/5/6/7/0) :
6
6 Selected.
Masukkan dimensi matriks: 6

Matriks Hilbert:
1.00 0.50 0.33 0.25 0.20 0.17 1.00
0.50 0.33 0.25 0.20 0.17 0.14 0.00
0.33 0.25 0.20 0.17 0.14 0.13 0.00
0.25 0.20 0.17 0.14 0.13 0.11 0.00
0.20 0.17 0.14 0.13 0.11 0.10 0.00
0.17 0.14 0.13 0.11 0.10 0.09 0.00

Melalui sistem penyelesaian Gauss-Jordan, diperoleh solusi:
x0: 36.00000000090669
x1: -630.0000000272028
x2: 3360.000000189634
x3: -7560.000000503366
x4: 7560.000000563855
x5: -2772.0000002246784
```

Pengujian matriks Hilbert dengan n = 6

```
6 Selected.
Masukkan dimensi matriks: 10

Matriks Hilbert:
1.00 0.50 0.33 0.25 0.20 0.17 0.14 0.13 0.11 0.10 1.00
0.50 0.33 0.25 0.20 0.17 0.14 0.13 0.11 0.10 0.09 0.00
0.33 0.25 0.20 0.17 0.14 0.13 0.11 0.10 0.09 0.08 0.00
0.25 0.20 0.17 0.14 0.13 0.11 0.10 0.09 0.08 0.08 0.00
0.20 0.17 0.14 0.13 0.11 0.10 0.09 0.08 0.08 0.07 0.00
0.17 0.14 0.13 0.11 0.10 0.09 0.08 0.08 0.07 0.07 0.00
0.14 0.13 0.11 0.10 0.09 0.08 0.08 0.07 0.07 0.06 0.00
0.13 0.11 0.10 0.09 0.08 0.08 0.07 0.07 0.06 0.06 0.00
0.11 0.10 0.09 0.08 0.08 0.07 0.07 0.06 0.06 0.06 0.00
0.10 0.09 0.08 0.08 0.07 0.07 0.06 0.06 0.06 0.05 0.00

Melalui sistem penyelesaian Gauss-Jordan, diperoleh solusi:
x0: 99.99766989068321
x1: -4949.79760423883
x2: 79195.6718050914
x3: -600560.5288201892
x4: 2522331.2542866915
x5: -6305780.0627520755
x6: 9608745.465436822
x7: -8750772.98231808
x8: 4375365.278450846
x9: -923684.2944614044
```

Pengujian matriks Hilbert dengan n = 10

4.2 Percobaan 2: Solusi SPL dengan Matrix *Augmented*

4.2.1 Percobaan 2a

a.

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

```
SISTEM PERSAMAAN LINIER
[1] Metode eliminasi Gauss
[2] Metode eliminasi Gauss-Jordan
[3] Metode matriks balikan
[4] Kaidah Cramer
[5] Kembali
Masukkan pilihan menu (1/2/3/4/5) :
2
2 Selected.
Masukkan pilihan input [1] keyboard | [2] file .txt (1/2) :
1

[1] Masukan Ax = B
[2] Matriks Augmented
[3] Matriks Hilbert
2
2 Selected.
Masukkan ukuran baris matriks: 4
Masukkan ukuran kolom matriks: 5
Masukkan matriks:
1 -1 2 -1 -1
2 1 -2 -2 -2
-1 2 -4 1 1
3 0 0 -3 -3
Hasil perhitungan menggunakan metode eliminasi Gauss-Jordan:
x1 = -1.00 + t1
x2 = 2.00t2
```

Percobaan dilakukan langsung dengan memasukkan matriks *augmented* ke dalam sistem, lalu sistem akan menghasilkan jawaban sesuai hasil reduksi akhir. Dalam kasus ini, hasil yang diperoleh adalah hasil banyak, sehingga dihasilkan persamaan parametrik. Variabel yang bersifat parametrik akan ditunjukkan, sementara variabel yang bersifat bebas tidak ditunjukkan.

4.2.2 Percobaan 2b

b.

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

```
SISTEM PERSAMAAN LINIER
[1] Metode eliminasi Gauss
[2] Metode eliminasi Gauss-Jordan
[3] Metode matriks balikan
[4] Kaidah Cramer
[5] Kembali
Masukkan pilihan menu (1/2/3/4/5) :
2
2 Selected.
Masukkan pilihan input [1] keyboard | [2] file .txt (1/2) :
1

[1] Masukan Ax = B
[2] Matriks Augmented
[3] Matriks Hilbert
2
2 Selected.
Masukkan ukuran baris matriks: 6
Masukkan ukuran kolom matriks: 5
Masukkan matriks:
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
Hasil perhitungan menggunakan metode eliminasi Gauss-Jordan:
x1 = 0
x2 = 2.00
x3 = 1.00
x4 = 1.00
```

Percobaan dilakukan langsung dengan memasukkan matriks *augmented* ke dalam sistem, lalu sistem akan menghasilkan jawaban sesuai hasil reduksi akhir. Dalam kasus ini, hasil yang diperoleh adalah solusi unik, sehingga sistem akan memberikan luaran berupa nilai tiap variabel x. Karena selisih ukuran baris matriks dan kolom matriks melebihi 1, maka kaidah Cramer tidak dapat digunakan.

4.3 Percobaan 3: Solusi SPL dengan Input SPL yang diubah menjadi Matriks *Augmented*

4.3.3 Percobaan 3a

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

```
SISTEM PERSAMAAN LINIER
[1] Metode eliminasi Gauss
[2] Metode eliminasi Gauss-Jordan
[3] Metode matriks balikan
[4] Kaidah Cramer
[5] Kembali
Masukkan pilihan menu (1/2/3/4/5) :
4
4 Selected.
Masukkan pilihan input [1] keyboard | [2] file .txt (1/2) :
1

[1] Masukan Ax = B
[2] Matriks Augmented
[3] Matriks Hilbert
2
2 Selected.
Masukkan ukuran baris matriks: 4
Masukkan ukuran kolom matriks: 5
Masukkan matriks:
8 1 3 2 0
2 9 -1 -2 1
1 3 2 -1 2
1 0 6 4 3
Hasil perhitungan menggunakan kaidah Crammer:
x1 = -0.22
x2 = 0.18
x3 = 0.71
x4 = -0.26
```

Percobaan dilakukan dengan mengubah sistem persamaan linear menjadi matriks *augmented*:

$$
\begin{bmatrix}
8 & 1 & 3 & 2 & 0\\
2 & 9 & -1 & -2 & 1\\
1 & 3 & 2 & -1 & 2\\
1 & 0 & 6 & 4 & 3
\end{bmatrix}
$$

Diperoleh matriks *augmented* dengan ukuran 4×5, dan karena selisih ukuran baris matriks dna kolom matriks 1, maka kaidah Cramer dapat digunakan. Menggunakan kaidah Cramer, didapatkan

$$x_1 = -0.22, x_2 = 0.18, x_3 = 0.71, x_4 = -0.26$$

dengan x_1, x_2, x_3, x_4 dibulatkan ke dua bilangan di belakang koma.

Percobaan 3b

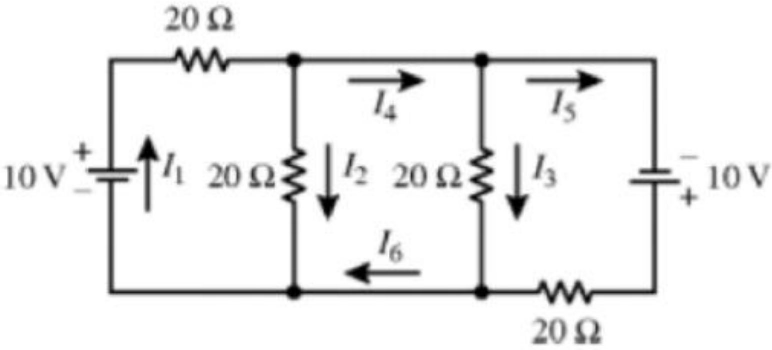
b.

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

```
[1] Masukan Ax = B
[2] Matriks Augmented
[3] Matriks Hilbert
2
2 Selected.
Masukkan ukuran baris matriks: 12
Masukkan ukuran kolom matriks: 10
Masukkan matriks:
0 0 0 0 0 0 1 1 1 13
0 0 0 1 1 1 0 0 0 15
1 1 1 0 0 0 0 0 0 8
0 0 0.04289 0 0.04289 0.75 0.04289 0.75 0.61396 14.79
0 0.25 0.91421 0.25 0.91421 0.25 0.91421 0.25 0 14.31
0.61396 0.75 0.04289 0.75 0.04289 0 0.04289 0 0 3.81
0 0 1 0 0 1 0 0 1 18
0 1 0 0 1 0 0 1 0 12
1 0 0 1 0 0 1 0 0 6
0.04289 0.75 0.61396 0 0.04289 0.75 0 0 0.04289 10.51
0.91421 0.25 0 0.25 0.91421 0.25 0 0.25 0.91421 16.13
0.04289 0 0 0.75 0.04289 0 0.61396 0.75 0.04289 7.04
Hasil perhitungan menggunakan metode eliminasi Gauss:
1.00 1.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 8.00
0.00 1.00 0.00 0.00 1.00 0.00 0.00 1.00 0.00 12.00
0.00 0.00 1.00 0.00 0.00 1.00 0.00 0.00 1.00 18.00
0.00 0.00 0.00 1.00 1.00 1.00 0.00 0.00 0.00 15.00
0.00 0.00 0.00 0.00 1.00 -2.21 2.21 0.00 -2.21 -21.48
0.00 0.00 0.00 0.00 0.00 1.00 -0.93 0.07 0.63 10.69
0.00 0.00 0.00 0.00 0.00 0.00 1.00 1.00 1.00 13.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 -9,022,440.00 -45,119,920.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 5.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 -4.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 -0.01
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
Sistem inkonsisten.
```

Persamaan linear diubah menjadi matriks *augmented*, kemudian dimasukkan ke dalam program. Program memberikan output sistem inkonsisten bersama dengan hasil reduksi yang tidak memungkinkan adanya hasil.

4.4 Percobaan 4: Mencari Arus Listrik dalam Rangkaian



Menggunakan analisis *mesh*, didapatkan persamaan berikut:

$$-10 + 40I_1 - 20I_4 = 0$$

$$40I_4 - 20I_1 - 20I_5 = 0$$

$$-10 + 40I_5 - 20I_4 = 0$$

$$I_2 = I_1 - I_4$$

$$I_3 = I_4 - I_5$$

$$I_6 = I_4$$

Persamaan tersebut dapat diubah menjadi sistem persamaan linear yang kemudian dapat diubah menjadi matriks *augmented:*

$$\begin{bmatrix} 40 & 0 & 0 & -20 & 0 & 0 & 10 \\ -20 & 0 & 0 & 40 & -20 & 0 & 0 \\ 0 & 0 & 0 & -20 & 40 & 0 & 10 \\ -1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 \end{bmatrix}$$

```
Masukkan ukuran baris matriks: 6
Masukkan ukuran kolom matriks: 7
Masukkan matriks:
40 0 0 -20 0 0 10
-20 0 0 40 -20 0 0
0 0 0 -20 40 0 10
-1 1 0 1 0 0 0
0 0 1 -1 1 0 0
0 0 0 -1 0 1 0
Hasil perhitungan menggunakan kaidah Crammer:
x1 = 0.50
x2 = 0.00
x3 = 0.00
x4 = 0.50
x5 = 0.50
x6 = 0.50
```

Didapatkan hasil $I_1 = 0.5\ A, I_2 = 0\ A, I_3 = 0\ A, I_4 = 0.5\ A, I_5 = 0.5\ A, I_6 = 0.5\ A$ menggunakan kaidah Cramer.

4.5 Percobaan 5: Sistem Reaktor

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

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

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.

Dengan mensubstitusikan konstanta-konstanta yang ada ke sistem persamaan linear, didapatkan persamaan sebagai berikut:

$$1300 + 60x_b - 40x_a - 80x_a = 0$$

$$40x_a - 60x_b - 20x_b = 0$$

$$200 + 80x_a + 20\ x_b - 150x_c = 0$$

didapatkan matriks A dan B sebagai berikut:

$$A = \begin{bmatrix} -120 & 60 & 0 \\ 40 & -80 & 0 \\ 80 & 20 & -150 \end{bmatrix}, B = \begin{bmatrix} -1300 \\ 0 \\ -200 \end{bmatrix}$$

```
SISTEM PERSAMAAN LINIER
[1] Metode eliminasi Gauss
[2] Metode eliminasi Gauss-Jordan
[3] Metode matriks balikan
[4] Kaidah Cramer
[5] Kembali
Masukkan pilihan menu (1/2/3/4/5) :
4
4 Selected.
Masukkan pilihan input [1] keyboard | [2] file .txt (1/2) :
1

[1] Masukan Ax = B
[2] Matriks Augmented
[3] Matriks Hilbert
1
1 Selected.
Masukkan ukuran baris matriks A: 3
Masukkan ukuran kolom matriks A: 3
Masukkan matriks A:
-120 60 0
40 -80 0
80 20 -150
Masukkan matriks B (ukuran baris x 1) dalam bentuk baris: -1300 0 -200
Hasil perhitungan menggunakan kaidah Crammer:
x1 = 14.44
x2 = 7.22
x3 = 10.00
```

Menggunakan kaidah Cramer, didapatkan bahwa $x_a = 14.44\frac{mg}{m^3}, x_b = 7.22\frac{mg}{m^3}, x_c = \frac{10mg}{m^3}$

4.6 Percobaan 6: Interpolasi Polinom

4.6.1 Percobaan 6a

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

```
Masukkan banyaknya pasangan titik: 7

Masukkan titik (x, y):
0.1 0.003
0.3 0.067
0.5 0.148
0.7 0.248
0.9 0.370
1.1 0.518
1.3 0.697

Matriks augmented persamaan:
1.00 0.10 0.01 0.00 0.00 0.00 0.00 0.00
1.00 0.30 0.09 0.03 0.01 0.00 0.00 0.07
1.00 0.50 0.25 0.13 0.06 0.03 0.02 0.15
1.00 0.70 0.49 0.34 0.24 0.17 0.12 0.25
1.00 0.90 0.81 0.73 0.66 0.59 0.53 0.37
1.00 1.10 1.21 1.33 1.46 1.61 1.77 0.52
1.00 1.30 1.69 2.20 2.86 3.71 4.83 0.70
x1 = -0.02
x2 = 0.24
x3 = 0.20
x4 = 0.00
x5 = 0.03
x6 = 0.00
x7 = 0.00

Persamaan polinom: -0.022977 + 0.240000x^1 + 0.197396x^2 + 0.000000x^3 + 0.026042x^4 + 0.00000
0x^5 -0.000000x^6
Masukkan nilai x yang akan diaproksimasi (input -999 untuk keluar): 0.2
Aproksimasi nilai 0.200000 terhadap interpolasi polinom: 0.032960937500000065

Masukkan nilai x yang akan diaproksimasi (input -999 untuk keluar): 0.55
Aproksimasi nilai 0.550000 terhadap interpolasi polinom: 0.17111865234374998

Masukkan nilai x yang akan diaproksimasi (input -999 untuk keluar): 0.85
Aproksimasi nilai 0.850000 terhadap interpolasi polinom: 0.33723583984374994

Masukkan nilai x yang akan diaproksimasi (input -999 untuk keluar): 1.28
Aproksimasi nilai 1.280000 terhadap interpolasi polinom: 0.6775418374999997
```

Didapatkan aproksimasi:

- f(0.2) = 0.03296
- f(0.55) = 0.1711
- f(0.85) = 0.3372
- f(1.28) = 0.677

Nilai aproksimasi yang didapatkan masih berada di rentang nilai antara y_1 dan y_7 karena nilai x yang dimasukkan masih berada di antara x_1 dan x_7 , sehingga aproksimasi masih valid (tidak terdapat *outlier* yang berlebih)

4.6.2 Percobaan 6b: Estimasi Kasus COVID-19

Estimasi di tanggal

- 7/07/2021 (7.225)
- 16/07/2021 (7.52)
- 10/08/2021 (8.322)
- 05/09/2021 (9.167)

Tanggal	Tanggal (desimal)	Jumlah Kasus Baru
17/06/2021	6,567	12.624
30/06/2021	7	21.807
08/07/2021	7,258	38.391
14/07/2021	7,451	54.517
17/07/2021	7,548	51.952
26/07/2021	7,839	28.228
05/08/2021	8,161	35.764
15/08/2021	8,484	20.813
22/08/2021	8,709	12.408
31/08/2021	9	10.534

```
Persamaan polinom: 7200093878806.706000 -9362131140303.973000x^1 + 5342010946202.457000x^2 -17
59156377519.361000x^3 + 369003453733.159550x^4 -51190022492.628580x^5 + 4700779581.075389x^6 -
275747650.190244x^7 + 9381587.281704x^8 -141117.811992x^9
Masukkan nilai x yang akan diaproksimasi (input -999 untuk keluar): 7.225
Aproksimasi nilai 7.225000 terhadap interpolasi polinom: 33804.587890625

Masukkan nilai x yang akan diaproksimasi (input -999 untuk keluar): 8.322
Aproksimasi nilai 8.322000 terhadap interpolasi polinom: 36355.34375

Masukkan nilai x yang akan diaproksimasi (input -999 untuk keluar): 9.167
Aproksimasi nilai 9.167000 terhadap interpolasi polinom: -667846.8203125
```

```
Persamaan polinom: 7200093878806.706000 -9362131140303.973000x^1 + 534201094
6202.457000x^2 -1759156377519.361000x^3 + 369003453733.159550x^4 -5119002249
2.628580x^5 + 4700779581.075389x^6 -275747650.190244x^7 + 9381587.281704x^8
-141117.811992x^9
Masukkan nilai x yang akan diaproksimasi (input -999 untuk keluar): 7.52
Aproksimasi nilai 7.520000 terhadap interpolasi polinom: 53383.51953125
```

Diperoleh aproksimasi:

- f(7.225) = 33804.58789
- f(7.52) = 53383.51
- f(8.322) = 36355.34375
- f(9.167) = -667846.8203125

Dapat dilihat bahwa nilai percobaan di titik 7.225 dan 8.322 menghasilkan nilai yang tidak jauh berbeda dibandingkan dengan nilai-nilai yang ada di tabel. Hal ini dikarenakan kedua titik tersebut yang masih berada di rentang $x_1 - x_{10}$ dan interpolasi polinom melakukan pendekatan terhadap nilai-nilai yang berada di rentang tersebut. Oleh karena itu, aproksimasi f(7.225) dan f(8.322) mempunyai nilai yang tidak termasuk *outlier*. Sementara itu, pengujian di titik 9.167 menghasilkan *outlier* bawah senilai -667846.82. Hal ini dikarenakan titik 9.167 yang berada di luar rentang $x_1 - x_{10}$ $(6.567 - 9.000)$ sehingga tidak masuk ke rentang interpolasi yang valid. Oleh karena itu, didapatkan hasil yang tidak mungkin.

4.6.3 Percobaan 6c: Penyederhanaan Fungsi

Misalkan derajat yang diambil = 4, maka pasangan titik yang perlu dimasukkan adalah sebagai berikut:

- (0.5, 0.4453)
- (1, 0.537)
- (1.5, 0.581)
- (2, 0.576)

```
Masukkan banyaknya pasangan titik: 4

Masukkan titik (x, y):
0.5 0.4453
1 0.537
1.5 0.581
2 0.576

Matriks augmented persamaan:
1.00 0.50 0.25 0.13 0.45
1.00 1.00 1.00 1.00 0.54
1.00 1.50 2.25 3.38 0.58
1.00 2.00 4.00 8.00 0.58
x1 = 0.31
x2 = 0.32
x3 = -0.09
x4 = 0.00

Persamaan polinom: 0.307200 + 0.321733x^1 -0.090200x^2 -0.001733x^3
Masukkan nilai x yang akan diaproksimasi (input -999 untuk keluar): 0.5
Aproksimasi nilai 0.500000 terhadap interpolasi polinom: 0.4452999999999999
```

Misalkan derajat yang diambil = 5, maka pasangan titik yang perlu dimasukkan adalah sebagai berikut:

- (0.4, 0.418)
- (0.8, 0.507)
- (1.2, 0.561)
- (1.6, 0.583)
- (2, 0.576)

```
Masukkan banyaknya pasangan titik: 5

Masukkan titik (x, y):
0.4 0.418
0.8 0.507
1.2 0.561
1.6 0.583
2 0.576

Matriks augmented persamaan:
1.00 0.40 0.16 0.06 0.03 0.42
1.00 0.80 0.64 0.51 0.41 0.51
1.00 1.20 1.44 1.73 2.07 0.56
1.00 1.60 2.56 4.10 6.55 0.58
1.00 2.00 4.00 8.00 16.00 0.58
x1 = 0.29
x2 = 0.37
x3 = -0.13
x4 = 0.01
x5 = 0.00

Persamaan polinom: 0.291000 + 0.367500x^1 -0.128125x^2 + 0.007812x^3 + 0.000000x^4
Masukkan nilai x yang akan diaproksimasi (input -999 untuk keluar): 0.5
Aproksimasi nilai 0.500000 terhadap interpolasi polinom: 0.4436953125
```

Misalkan derajat yang diambil = 8, maka pasangan titik yang perlu dimasukkan adalah sebagai berikut:

- (0.25, 0.368)
- (0.5, 0.4453)
- (0.75, 0.49826)
- (1, 0.537)
- (1.25, 0.565)
- (1.5, 0.581)
- (1.75, 0.58435)
- (2, 0.576)

```
Masukkan banyaknya pasangan titik: 8

Masukkan titik (x, y):
0.25 0.368
0.5 0.4453
0.75 0.49826
1 0.537
1.25 0.565
1.5 0.581
1.75 0.58435
2 0.576

Matriks augmented persamaan:
1.00 0.25 0.06 0.02 0.00 0.00 0.00 0.00 0.37
1.00 0.50 0.25 0.13 0.06 0.03 0.02 0.01 0.45
1.00 0.75 0.56 0.42 0.32 0.24 0.18 0.13 0.50
1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.54
1.00 1.25 1.56 1.95 2.44 3.05 3.81 4.77 0.56
1.00 1.50 2.25 3.38 5.06 7.59 11.39 17.09 0.58
1.00 1.75 3.06 5.36 9.38 16.41 28.72 50.27 0.58
1.00 2.00 4.00 8.00 16.00 32.00 64.00 128.00 0.58
x1 = 0.26
x2 = 0.49
x3 = -0.12
x4 = -0.53
x5 = 0.86
x6 = -0.60
x7 = 0.20
x8 = -0.03

Persamaan polinom: 0.258960 + 0.487329x^1 -0.118476x^2 -0.525566x^3 + 0.860569x^4
-0.597774x^5 + 0.197348x^6 -0.025389x^7
Masukkan nilai x yang akan diaproksimasi (input -999 untuk keluar): 0.5
Aproksimasi nilai 0.500000 terhadap interpolasi polinom: 0.44530000000000003
```

Nilai aproksimasi x = 0.5 pada derajat n = 4, 5, dan 8 mempunyai kemiripan, yaitu jika dibulatkan menjadi dua angka decimal, nilai hampiran aproksimasi x = 0.5 adalah y = 0.44.

4.7 Percobaan 7: Regresi Linear Berganda

Menggunakan data berikut dengan urutan yang sama:

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.

Diperoleh hasil demikian untuk humiditas 50% (x1), temperature $76°F$ (x2), dan tekanan 29.3 (x3), prakiraan kadar NO sekitar 0.94.

```
Masukkan x1: 50
Masukkan x2: 76
Masukkan x3: 29.3


Prediksi: 0.94

SPL yang terbentuk:
20.0x0 + 863.10004x1 + 1530.4001x2 + 587.84x3 = 19.42
863.10004x0 + 54876.89x1 + 67000.086x2 + 25283.398x3 = 779.47705
1530.4001x0 + 67000.086x1 + 117912.32x2 + 44976.87x3 = 1483.437
587.84x0 + 25283.398x1 + 44976.87x2 + 17278.51x3 = 571.1219
```

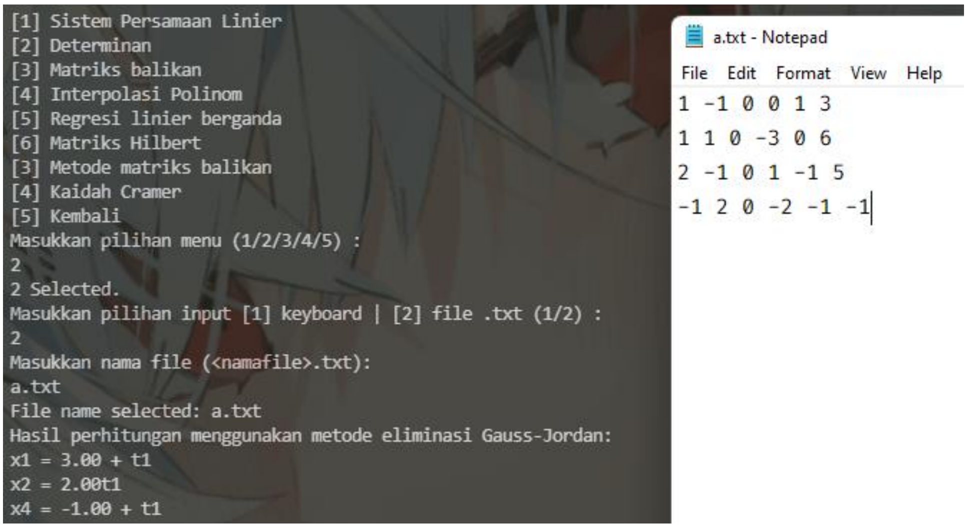
Program juga menerima luaran berupa sistem-sistem persamaan linear yang terbentuk. Model regresi yang terbentuk kemudian ditampilkan dalam bentuk persamaan.

```
Model regresi yang terbentuk:
-3.51171875 -0.00262451171875x1 + 7.992833852767944E-4x2 + 0.154296875x3
```

4.8 Percobaan 8: Baca Tulis File

4.8.1 Percobaan 8a: Baca File

Menggunakan data dalam percobaan 1b sebagai contoh (lihat 4.6.2):



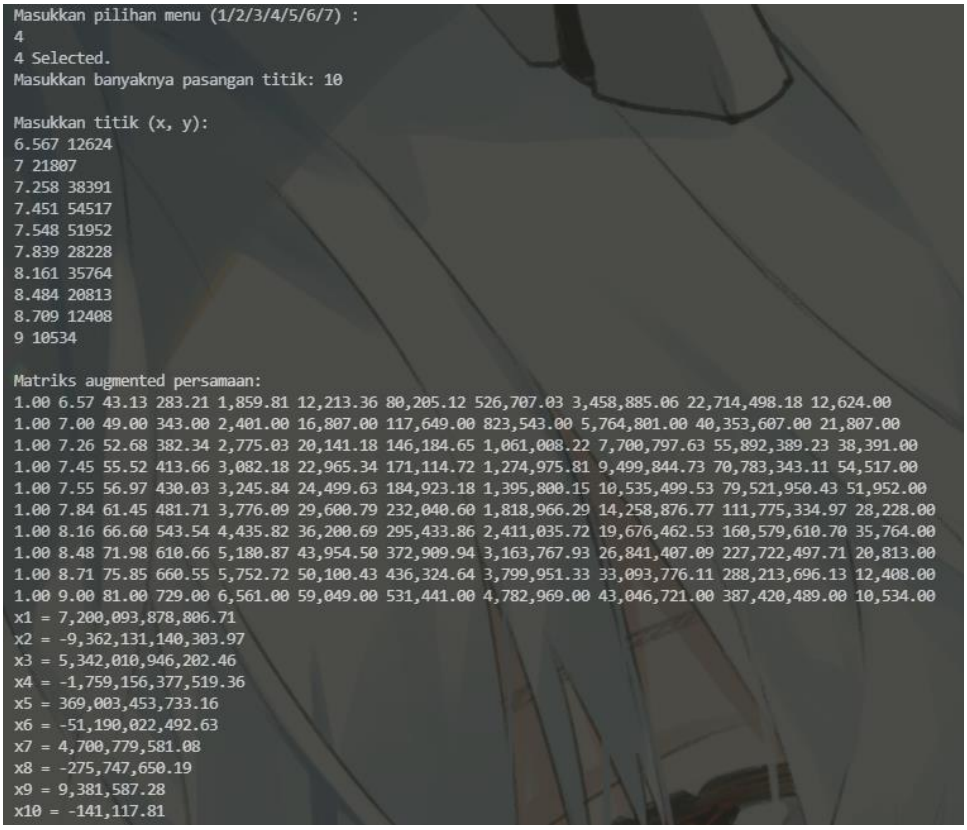
```
[1] Sistem Persamaan Linier
[2] Determinan
[3] Matriks balikan
[4] Interpolasi Polinom
[5] Regresi linier berganda
[6] Matriks Hilbert
[3] Metode matriks balikan
[4] Kaidah Cramer
[5] Kembali
Masukkan pilihan menu (1/2/3/4/5) :
2
2 Selected.
Masukkan pilihan input [1] keyboard | [2] file .txt (1/2) :
2
Masukkan nama file (<namafile>.txt):
a.txt
File name selected: a.txt
Hasil perhitungan menggunakan metode eliminasi Gauss-Jordan:
x1 = 3.00 + t1
x2 = 2.00t1
x4 = -1.00 + t1
```

```
a.txt - Notepad
File  Edit  Format  View  Help
1 -1 0 0 1 3
1 1 0 -3 0 6
2 -1 0 1 -1 5
-1 2 0 -2 -1 -1
```

Dengan memilih input dari file .txt, akan diminta input nama file dan hasil perhitungan yang sama akan dimunculkan.

4.8.2 Percobaan 8b: Tulis File

Menggunakan data dalam percobaan Interpolasi Polinom sebagai contoh:

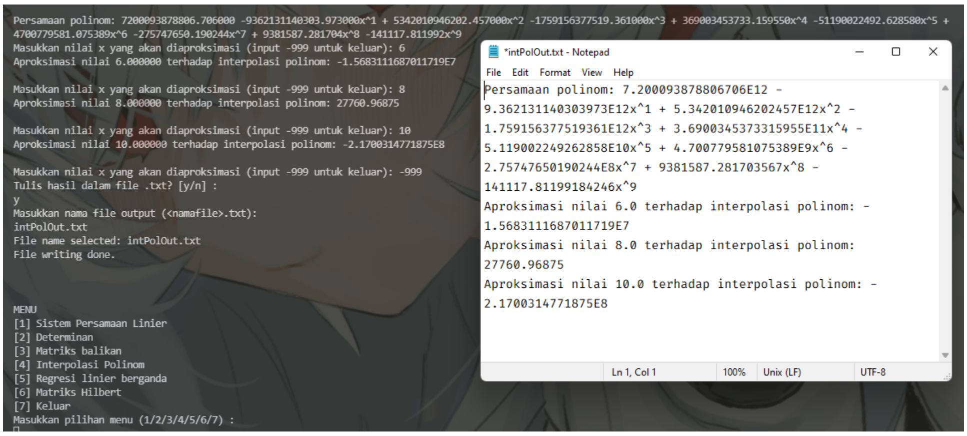


```
Masukkan pilihan menu (1/2/3/4/5/6/7) :
4
4 Selected.
Masukkan banyaknya pasangan titik: 10

Masukkan titik (x, y):
6.567 12624
7 21807
7.258 38391
7.451 54517
7.548 51952
7.839 28228
8.161 35764
8.484 20813
8.709 12408
9 10534

Matriks augmented persamaan:
1.00 6.57 43.13 283.21 1,859.81 12,213.36 80,205.12 526,707.03 3,458,885.06 22,714,498.18 12,624.00
1.00 7.00 49.00 343.00 2,401.00 16,807.00 117,649.00 823,543.00 5,764,801.00 40,353,607.00 21,807.00
1.00 7.26 52.68 382.34 2,775.03 20,141.18 146,184.65 1,061,008.22 7,700,797.63 55,892,389.23 38,391.00
1.00 7.45 55.52 413.66 3,082.18 22,965.34 171,114.72 1,274,975.81 9,499,844.73 70,783,343.11 54,517.00
1.00 7.55 56.97 430.03 3,245.84 24,499.63 184,923.18 1,395,800.15 10,535,499.53 79,521,950.43 51,952.00
1.00 7.84 61.45 481.71 3,776.09 29,600.79 232,040.60 1,818,966.29 14,258,876.77 111,775,334.97 28,228.00
1.00 8.16 66.60 543.54 4,435.82 36,200.69 295,433.86 2,411,035.72 19,676,462.53 160,579,610.70 35,764.00
1.00 8.48 71.98 610.66 5,180.87 43,954.50 372,909.94 3,163,767.93 26,841,407.09 227,722,497.71 20,813.00
1.00 8.71 75.85 660.55 5,752.72 50,100.43 436,324.64 3,799,951.33 33,093,776.11 288,213,696.13 12,408.00
1.00 9.00 81.00 729.00 6,561.00 59,049.00 531,441.00 4,782,969.00 43,046,721.00 387,420,489.00 10,534.00
x1 = 7,200,093,878,806.71
x2 = -9,362,131,140,303.97
x3 = 5,342,010,946,202.46
x4 = -1,759,156,377,519.36
x5 = 369,003,453,733.16
x6 = -51,190,022,492.63
x7 = 4,700,779,581.08
x8 = -275,747,650.19
x9 = 9,381,587.28
x10 = -141,117.81
```

Dicoba mengambil x sembarang untuk diaproksimasi, kemudian dicoba meng-output-kan hasil ke dalam file “intPolOut.txt” menjadi:



BAB V: KESIMPULAN

5.1 Kesimpulan

Pada akhir tugas besar ini, kami telah mengimplementasikan:

- Matriks secara matematis dalam bentuk *abstract data type*
- Operasi-operasi secara elemen (perkalian, pengubahan baris)
- Operasi pencarian solusi sistem persamaan linear
- Metode perhitungan determinan matriks
- Metode pencarian matriks balikan dari sebuah matriks
- Aplikasi matriks dan operasi-operasi lainnya dalam kasus-kasus dunia nyata, seperti interpolasi polinom dan regresi linear berganda

Implementasi ini berhasil dilakukan menggunakan sifat modularitas dan *object-orientation* dari bahasa pemrograman Java sehingga satu fungsi dapat diterapkan untuk fungsi-fungsi lainnya.

5.2 Saran

Disarankan untuk eksperimen selanjutnya untuk memanfaatkan sistem modularitas dan *object-oriented programming* pada bahasa Java dengan lebih baik, terutama pada implementasi sifat-sifat matriks, karena fungsi yang sama dapat digunakan berulang kali untuk tujuan yang berbeda-beda. Setiap algoritma yang dipakai untuk menyelesaikan sebuah masalah mempunyai keunggulan dan kekurangannya masing-masing, sehingga penulis menyarankan untuk bijak memilih algoritma yang diperlukan untuk menyelesaikan permasalahan tertentu. Selain itu, penulis juga menyarankan untuk mengutamakan presisi *floating point* ketimbang efisiensi dalam konteks tugas besar ini dikarenakan *test case* yang mungkin mengambil *floating point* yang cukup banyak.

5.3 Refleksi

Melalui tugas besar ini, penulis menyadari banyak hal yang dapat dimanfaatkan dari sifat matriks dan kegunaannya di bidang aljabar linear, terutama di bidang keinformatikaan dan implementasinya di struktur data. Matriks dapat digunakan mulai dari menyimpan informasi sederhana hingga mengaproksimasi sebuah nilai polinom dari titik-titik yang abstrak. Oleh karena itu, penulis merefleksikan kembali kegunaan matriks dan implementasi-implementasi yang memungkinkan ke depannya.

Penulis juga merefleksikan kegunaan dan perbedaan tiap-tiap algoritma penyelesaian yang telah diciptakan. Bahwa sesungguhnya tidak ada satu algoritma yang lebih baik dari algoritma lainnya, dan perbedaan algoritma seharusnya dimanfaatkan untuk membuat sebuah program dan penyelesaian yang lebih baik tergantung dari kasus yang diberikan. Penulis juga merefleksikan kembali perbedaan tipe data dan presisi yang ada, sehingga melalui tugas ini penulis juga menambah pengalaman pemrograman dan kerja tim.

BAB IV: DAFTAR PUSTAKA

- Bahan pembelajaran mata kuliah IF2123 Aljabar Linear dan Geometri 2020/2021, oleh Rinaldi Munir
- Modul Tugas Besar I IF2123 Aljabar Linear dan Geometri
- Munir, Rinaldi. Lidya, Leoni. 2016. *Algoritma dan Pemrograman dalam Bahasa Pascal, C, dan C++: Edisi Keenam.* Bandung: Penerbit INFORMATIKA.