Laporan Tugas Besar 1
Aljabar Linear dan Geometri

KELOMPOK 21 : BONEK

Anggota Kelompok :

1. Ahmad Alfani Handoyo - 13520023
2. Saul Sayers - 13520094
3. Rizky Ramadhana P. K. - 13520151

BAB I
Deskripsi Masalah

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Pada mata kuliah Aljabar Linear dan Geometri IF2123, telah dipelajari berbagai metode untuk menyelesaikan sistem persamaan linear (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).

Namun, yang menjadi permasalahan ialah ketika mengerjakan sebuah soal yang membutuhkan penerapan metode-metode di atas, langkah-langkah yang dilakukan terkesan repetitif. Sehingga, akan menjadi baik bila metode-metode tersebut digeneralisir menjadi sebuah algoritma yang dapat direalisasikan dalam bentuk *code* atau program dalam bahasa Java. Bahasa Java dipilih karena sifatnya yang multiplatform, bisa berjalan di banyak platform. Dengan menggeneralisir prosedur-prosedur yang harus dilakukan pada metode-metode di atas, diharapkan proses perhitungan dapat dipercepat dan tentunya kesalahan perhitungan dapat diminimalisir. Penggeneralisiran metode-metode tersebut juga dapat melatih kemampuan dalam melihat pola yang berulang. Pada tugas kali ini, akan terdapat lima bagian besar yaitu menyelesaikan SPL, menghitung determinan matriks, mencari matriks balikan, dan penerapan metode-metode tersebut dalam interpolasi polinom serta regresi linear berganda.

BAB II
Teori Singkat

A. Sistem Persamaan Linear

Sistem persamaan linear merupakan salah satu model dan masalah matematika yang banyak diterapkan dalam berbagai ilmu. Suatu sistem persamaan linear terdiri atas sejumlah berhingga persamaan linear dalam sejumlah berhingga variabel. Sistem Persamaan Linear dalam bentuk persamaan perkalian matriks dapat ditulis

$$Ax = B$$

dengan A adalah matriks koefisien yang berukuran n × n, serta x dan b merupakan matriks kolom yang berukuran n × 1.

Sistem persamaan linear berbentuk matriks tersebut dapat diselesaikan secara langsung ataupun secara augmented. Secara augmented, terdapat 2 metode yang dapat digunakan yakni metode Eliminasi Gauss dan metode Eliminasi Gauss-Jordan. Apabila secara langsung, maka persamaan linear tersebut dapat diselesaikan dengan metode Cramer ataupun dengan metode matriks balikan.

1) Metode Eliminasi Gauss

Eliminasi Gauss diperkenalkan Karl Friendrich Gauss (1777 – 1855) dengan meng-augmentasi matriks A dengan matriks B, lalu matriks gabungan tersebut diubah menjadi suatu matriks eselon baris dengan memanfaatkan Operasi Baris Elementer (OBE). Operasi yang diliput oleh OBE adalah penjumlahan atau pengurangan suatu baris dengan k kali baris yang lain (dengan k bebas) dan operasi tukar baris.

Suatu matriks dikatakan memiliki bentuk eselon baris jika :

a. Entri bukan nol pertama dalam setiap baris adalah 1.
b. Jika baris tidak seluruhnya mengandung nol, maka banyaknya entri nol di bagian muka pada baris lebih besar dari banyaknya entri nol di bagian muka pada baris .
c. Jika terdapat baris-baris yang entrinya semuanya nol, maka baris - baris ini berada dibawah baris-baris yang memiliki entri-entri nol.

$$A = \begin{bmatrix} 0 & 1 & 3 & 7 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Contoh matriks eselon baris

Setelah matriks berhasil diubah menjadi matriks Eselon Baris, maka dapat digunakan substitusi untuk mendapatkan solusi tiap variabel.

2) Metode Eliminasi Gauss-Jordan

Metode Eliminasi Gauss-Jordan dicetuskan oleh Wilhelm Jordan, seorang insinyur Jerman, pada tahun 1887 dengan cara mengembangkan metode Eliminasi Gauss. Langkah - langkah metode ini sama seperti metode Eliminasi Gauss, namun setelah matriks berhasil diubah menjadi matriks eselon baris, perlu dilanjutkan untuk mengubah matriks tersebut menjadi matriks eselon baris tereduksi.

Suatu matriks dikatakan memiliki bentuk eselon baris tereduksi jika :

a. Memenuhi syarat yang sama dengan eselon baris
b. Setiap kolom yang memiliki 1 utama memiliki nol di tempat lain.

$$\begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 7 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

Contoh matriks eselon baris tereduksi

Manfaat daripada menggunakan metode ini dibanding metode eliminasi Gauss biasa adalah bisa didapatkan nilai dari variabel tanpa harus melakukan proses substitusi.

3) Metode matriks Balikan

Karena bentuk matriks dari sistem persamaan linear adalah

$$Ax = B$$

Maka langsung didapatkan x dengan mengalikan kedua ruas invers matriks A sehingga didapat

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

Hasil perkalian invers matriks A dengan B langsung berupa solusi dari variabel x, urut dari atas ke bawah.

4) Metode Cramer

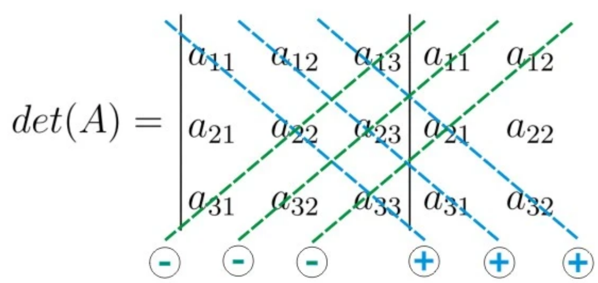
Metode ini dapat digunakan dengan prekondisi determinan matriks A bukan 0. Jika Ax = B adalah SPL yang terdiri dari n persamaan linier dengan n peubah, SPL tersebut memiliki solusi yang unik yaitu

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

matriks A_i adalah sebuah matriks A dengan seluruh elemen pada kolom i ditukar dengan elemen pada matriks B.

B. Determinan matriks

Determinan suatu matriks didefinisikan sebagai selisih antara perkalian elemen-elemen pada diagonal utama dengan perkalian elemen-elemen pada diagonal sekunder. Determinan matriks hanya dapat ditentukan pada matriks persegi. Determinan dari matriks A dapat dituliskan det(A) atau |A|. Apabila sebuah matriks berordo 1x1, maka determinannya adalah elemen itu sendiri. Untuk menghitung determinan matriks berordo 2x2 dan 3x3, kurangi perkalian elemen secara diagonal ke kanan dengan perkalian secara diagonal ke kiri.



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

Ilustrasi menghitung determinan matriks 2x2 dan 3x3

Apabila matriks memiliki ordo n x n dengan n lebih besar daripada 3, maka determinan dapat dihitung menggunakan dua metode yakni dengan reduksi baris dan dengan kofaktor.

1) Metode reduksi baris (OBE)

Dengan menggunakan operasi baris elementer (OBE), matriks awal diubah menjadi matriks baru yang menjadi matriks berjenis segitiga atas atau segitiga bawah. Dengan demikian, determinan matriks cukup dihitung dengan mengalikan diagonal utama pada matriks.

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

Contoh matriks segitiga atas

$$det(A) = \frac{(-1)^p a_{11} a_{22} ... a_{mm}}{k_1 k_2 ... k_{mm}}$$

Dengan p adalah banyaknya operasi tukar baris yang terjadi, k adalah seluruh bilangan yang pernah digunakan untuk membagi suatu baris, dan a adalah elemen matriks akhir setelah dioperasikan menjadi segitiga atas atau segitiga bawah

2) Metode Kofaktor

Determinan suatu matriks juga dapat ditemukan melalui metode kofaktor. Misalkan matriks A sebagai berikut

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

Maka untuk setiap elemen a_{ij}, terdapat entri kofaktornya C_{ij} sehingga matriks kofaktornya dapat ditulis sebagai berikut:

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

Untuk mendapatkan nilai dari C_{ij}, perlu dicari minor sebuah matriks terlebih dahulu. Minor suatu matriks A dilambangkan dengan M_{ij} adalah determinan dari matriks bagian matriks A yang diperoleh dengan cara menghilangkan semua elemen lain pada baris ke-i dan kolom ke-j. Dengan demikian, dapat dicari nilai C_{ij} sebagai berikut

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

Setelah didapatkan nilai entri kofaktor, dapat dihitung determinan sebuah matriks dengan salah satu persamaan sebagai berikut:

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

Secara baris

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

Secara kolom

C. Inverse matriks (Balikan)

Misalkan A adalah matriks persegi berukuran n x n. Balikan (*inverse*) matriks A adalah A^{-1} sedemikian sehingga $AA^{-1} = A^{-1}A = I$, I adalah matriks identitas. Terdapat 2 metode mencari invers dari sebuah matriks, yaitu dengan eliminasi Gauss-Jordan dan dengan adjoin.

1) Metode eliminasi Gauss-Jordan

Langkah pertama dari metode ini adalah meng-augmentasi matriks A dengan matriks Identitas di sebelah kanannya. Lalu, lakukan operasi eliminasi gauss-jordan pada matriks A sehingga matriks A menjadi matriks identitas. matriks hasil inversnya dapat dilihat dari matriks identitas awal yang telah berubah karena operasi eliminasi gauss jordan tersebut.

$$[A|I] \ -> \ [I|A^{-1}]$$

Ilustrasi metode eliminasi gauss jordan

2) Metode Adjoin

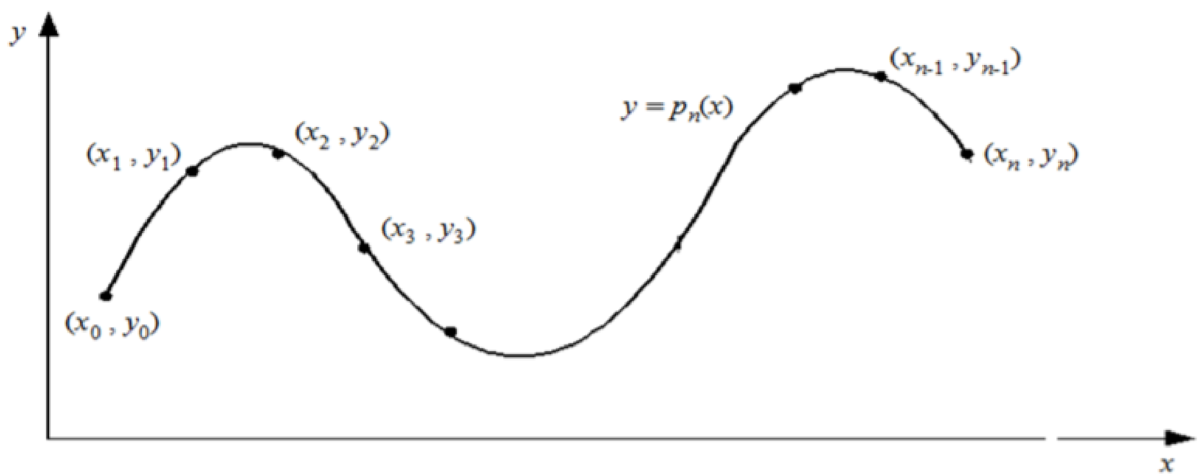
Invers dari sebuah matriks bisa didapat dengan membagi adjoin dari matriksnya dengan determinan matriks itu sendiri. Metode ini dapat dituliskan dengan persamaan sebagai berikut:

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

Adjoin dari sebuah matriks adalah transpose dari matriks kofaktor matriks A itu sendiri. Transpose adalah operasi menukar tiap elemen a_{ij} pada sebuah matriks menjadi a_{ji} sehingga ordonya juga ikut berubah dari n x m menjadi m x n.

D. Interpolasi Polinom

Persoalan interpolasi polinom adalah diberikan n+1 buah titik berbeda, (x0 , y0), (x1 , y1), ..., (xn , yn). Tentukan polinom $p_n(x)$ yang melewati semua titik-titik tersebut sedemikian rupa sehingga $y_i = p_n(x_i)$ untuk i = 0, 1, 2, …, n.



Setelah polinom interpolasi $p_n(x)$ ditemukan, $p_n(x)$ dapat digunakan untuk menghitung perkiraan nilai y di sembarang titik di dalam selang [x0 , xn].

Polinom interpolasi derajat n yang menginterpolasi titik-titik (x0 , y0), (x1 , y1), ..., (xn , yn). adalah berbentuk $p_n(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n$. Jika hanya ada dua titik, (x0 , y0) dan (x1 , y1), maka polinom yang melewati kedua titik tersebut adalah $p_1(x) = a_0 + a_1x$ yaitu berupa persamaan garis lurus. Jika tersedia tiga titik, (x0 , y0), (x1 , y1), dan (x2 , y2), maka polinom yang melewati ketiga titik tersebut adalah $p_2(x) = a_0 + a_1x + a_2x^2$ atau persaman kuadrat dan kurvanya berupa parabola. Jika tersedia empat titik, (x0 , y0), (x1 , y1), (x2 , y2), dan (x3 , y3), polinom yang melewati keempat titik tersebut adalah $p_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3$, demikian seterusnya. Dengan cara yang sama dapat dibuat polinom interpolasi berderajat n untuk n yang lebih tinggi asalkan tersedia (n+1) buah titik data. Dengan menyulihkan (xi , yi) ke dalam persamaan polinom $p_n(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n$ untuk i = 0, 1, 2, …, n, akan diperoleh n buah sistem persamaan lanjar dalam a_0, a_1, a_2, ..., a_n.

$$a_0 + a_1x_0 + a_2x_0^2 + ... + a_nx_o^n = y_0$$

$$a_0 + a_1x_1 + a_2x_1^2 + ... + a_nx_1^n = y_1$$

$$\dots$$

$$a_n + a_1x_n + a_2x_n^2 + ... + a_nx_n^n = y_n$$

Solusi sistem persamaan lanjar ini, yaitu nilai a0 , a1 , …, an , diperoleh dengan menggunakan metode eliminasi Gauss yang sudah dipelajari. Sebagai contoh, misalkan diberikan tiga buah titik yaitu (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513). Tentukan polinom interpolasi kuadratik lalu estimasi nilai fungsi pada x = 9.2. Polinom kuadratik berbentuk $p_2(x) = a_0 + a_1x + a_2x^2$. Dengan menyulihkan ketiga buah titik data ke dalam polinom tersebut, diperoleh sisten persamaan lanjar yang terbentuk adalah

$$a_0 + 8a_1 + 64a_2 = 2.0794$$

$$a_0 + 9a_1 + 81a_2 = 2.1972$$

$$a_0 + 9.5a_1 + 90.25a_2 = 2.2513$$

Penyelesaian sistem persamaan dengan metode eliminasi Gauss menghasilkan

$$a_0 = 0.6762,\ a_1 = 0.2266,\ a_2 =- 0.0064$$

Polinom interpolasi yang melalui ketiga buah titik tersebut adalah

$$p_2(x) = 0.6762 + 0.2266x - 0.0064x^2$$

Dengan menggunakan polinom ini, maka nilai fungsi pada x = 9.2 dapat ditaksir sebagai berikut

$$p_2(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(9.2)^2 = 2.2192$$

E. Regresi Linear Berganda

Regresi Linear (akan dipelajari lebih lanjut di Probabilitas dan Statistika) merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Meskipun sudah ada rumus jadi untuk menghitung regresi linear sederhana, terdapat rumus umum dari regresi linear yang bisa digunakan untuk regresi linear berganda, yaitu.

$$y_1 = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + ... + \beta_{k_i} + \epsilon_i$$

Untuk mendapatkan nilai dari setiap βi dapat digunakan *Normal Estimation Equation for Multiple Linear Regression* sebagai berikut:

$$nb_0 + b_1\Sigma x_{1i} + b_2\Sigma x_{2i} + ... + b_k\Sigma x_{ki} = \Sigma y_i$$

$$b_0\Sigma x_{1i} + b_1\Sigma x_{1i}^{\ 2} + b_2\Sigma x_{1i}x_{2i} + ... + b_k\Sigma x_{1i}x_{ki} = \Sigma x_{1i}y_i$$

$$\cdots$$

$$b_0\Sigma x_{ki} + b_1\Sigma x_{ki}x_{1i} + b_2\Sigma x_{ki}x_{2i} + ... + b_k\Sigma x_{ki}^{\ 2} = \Sigma x_{ki}y_i$$

Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

BAB III

Implementasi dalam Java

Pada tugas ini, kelompok dibuat sebuah public class yaitu class matriks. Atribut dari class matriks ini adalah sebagai berikut

```
public class Matrix {
    //ATRIBUT
    int rows;
    int cols;
    double[][] matrix;
```

Class matriks memiliki atribut integer rows (baris), cols (kolom), dan double matriks yang merupakan array dua dimensi. Konstruktor class Matrix adalah sebagai berikut

```
    public Matrix(int rows, int cols) { //KONSTRUKTOR
        this.rows = rows;
        this.cols = cols;
        this.matrix = new double[rows][cols];
    }
```

Konstruktor class Matrix menggunakan parameter integer rows dan cols, yang akan menyatakan banyaknya baris dan kolom efektif yang dimiliki oleh objek tersebut. Atribut matriks juga akan dideklarasikan secara statis dengan elemen yang berisi double. Selanjutnya ada method - method primitif dalam class kami sebagai berikut :

```
    public void readMatrix(int rows, int cols) {
        int i, j;
        double elemen;
        Scanner sc = new Scanner(System.in);
        this.rows = rows;
        this.cols = cols;
        for (i = 1; i <= rows; i++) {
            for (j = 1; j <= cols; j++) {
                System.out.print("Masukkan nilai untuk baris ke-"+i+" kolom ke-"+j+": ");
                elemen = sc.nextDouble();
                this.matrix[i - 1][j - 1] = elemen;
            }
```

```
        }
    }
```

Prosedur readmatriks memiliki parameter int rows dan int cols yang bermakna banyaknya baris dan kolom dari matriks tersebut. Kemudian, prosedur akan meminta input pengguna untuk tiap elemen matriks[i][j] sebanyak baris x kolom.

```
public void displayMatrix(){
    int i, j;
    for (i = 0; i < this.rows; i++) {
        for (j = 0; j < this.cols; j++) {
            System.out.printf("%f ", this.matrix[i][j]);
        }
        System.out.println();
    }
}
```

Prosedur displaymatriks tidak menerima parameter. Prosedur tersebut mencetak seluruh elemen dengan bentuk matriks, dan diakhiri dengan newline.

```
public void createMatrix(int rows, int cols) {
    int i, j;
    Scanner sc = new Scanner(System.in);
    for (i = 0; i < this.rows; i++) {
        for (j = 0; j < this.cols; j++) {
            this.matrix[i][j] = 0;
        }
    }
}
```

Prosedur creatematriks menerima parameter rows dan cols, dimana semua elemen dalam scope baris dan kolom tersebut di set menjadi 0.

```
public Matrix copyMatrix(){
    Matrix mOutput = new Matrix(this.rows, this.cols);
    int i, j;
    mOutput.rows = this.rows;
    mOutput.cols = this.cols;
    for (i = 0; i < this.rows; i++) {
```

```
        for (j = 0; j < this.cols; j++) {
            mOutput.matrix[i][j] = this.matrix[i][j];
        }
    }
    return mOutput;
}
```

Fungsi copymatriks tidak menerima parameter. Fungsi ini berguna untuk meng-copy sebuah matriks input dan me-return sebuah matriks duplikat yang seluruh atribut dan elemennya sama persis dengan matriks originalnya.

```
public Matrix extendMatrix(int row, int col){
    Matrix mOutput = new Matrix(this.rows + row, this.cols + col);
    int i, j;
    for (i = 0; i < this.rows; i++) {
        for (j = 0; j < this.cols; j++) {
            mOutput.matrix[i][j] = this.matrix[i][j];
        }
    }
    //inisialisasi elemen baru dengan 0
    for (i = this.rows; i < mOutput.rows; i++) {
        for (j = this.cols; j < mOutput.cols; j++) {
            mOutput.matrix[i][j] = 0;
        }
    }
    return mOutput;
}
```

Fungsi extendmatriks menerima parameter int row dan int col. Fungsi ini mengembalikan sebuah matriks yang memiliki elemen yang sama dengan matriks originalnya, namun baris matriks baru tersebut diperpanjang sebanyak rows dan kolomnya diperpanjang sebanyak cols. Elemen yang berada pada scope baris dan kolom baru tersebut diisi dengan 0. Selanjutnya ada beberapa fungsi boolean :

1) **isRowsZero()**

Fungsi yang mengembalikan kondisi boolean yang bernilai true apabila terdapat sebuah baris yang elemennya 0 semua (sehingga determinannya 0)

2) **isColsZero()**

Fungsi yang mengembalikan kondisi boolean yang bernilai true apabila terdapat sebuah kolom yang elemennya 0 semua (sehingga determinannya 0)

3) **isSegitigaAtas()**

Fungsi yang mengembalikan kondisi boolean yang bernilai true apabila seluruh elemen dibawah matriks diagonal bernilai 0

4) **isIdentity()**

Fungsi yang mengembalikan kondisi boolean yang bernilai true apabila matriks tersebut termasuk matriks identitas. Prekondisi fungsi : matriks harus persegi.

5) **isUndef()**

Fungsi yang mengembalikan kondisi boolean yang bernilai true apabila matriks tersebut berordo 1x1 dengan satu satunya elemen adalah -999

Adapun fungsi dan prosedur yang berkaitan dengan penyelesaian sistem persamaan linear (SPL) sebagai berikut :

1) **cekEselon()**

Fungsi yang mengembalikan nilai boolean, true apabila matriks tersebut merupakan matriks eselon baris.

2) **convertToEselon()**

Fungsi yang mengembalikan hasil operasi baris elementer dari sebuah matriks. Fungsi ini akan mengembalikan hasil OBE berupa matriks eselon baris bila terdapat solusi tunggal, namun mengembalikan matriks undef bila tidak memiliki solusi atau memiliki solusi tak hingga banyaknya.

3) **substitusiBalik()**

Fungsi yang mengembalikan array berisi solusi dari sebuah matriks eselon baris yang dijamin memiliki solusi tunggal.

4) **solusiTakHingga()**

Prosedur yang menampilkan ke layar solusi parametrik dari sebuah matriks eselon baris tereduksi. Apabila metode yang dipilih pengguna adalah metode eliminasi Gauss, matriks *augmented* tetap akan dikonversi ke matriks eselon baris tereduksi bila ingin dicari solusi parametriknya.

Adapun fungsi dan prosedur yang berkaitan dengan perhitungan determinan dari sebuah matrix adalah sebagai berikut :

1) **convertToSegitigaAtas()**

   Fungsi yang mengembalikan hasil operasi baris elementer dari sebuah matriks hingga menjadi segitiga atas. Fungsi ini akan mengembalikan hasil OBE berupa matriks segitiga atas apabila bisa ditransformasikan menjadi segitiga atas. Fungsi tidak akan mengubah matriks apabila sudah berupa segitiga atas atau tidak bisa ditransformasikan.

2) **determinanOBE()**

   Fungsi yang memanfaatkan fungsi convertToSegitigaAtas() dalam penerapannya untuk menghitung determinan. Fungsi ini akan mencetak tidak dapat dihitung determinannya apabila matriks bukan berupa persegi. Apabila persegi, fungsi akan mengembalikan nilai determinan berupa double yang dihitung dengan mengalikan seluruh elemen pada diagonal utama matriks yang berupa segitiga atas. Fungsi ini mengembalikan hasil determinannya yang berupa double.

3) **getMinor(<u>input:</u> <u>int</u> i, <u>input:</u> <u>int</u> j)**

   Fungsi ini menerima parameter i dan j yang berupa integer. Kegunaan fungsi ini adalah untuk mencari minor dari suatu matriks di baris ke- i kolom ke-j dengan cara mengambil matriks bagian yang tidak terletak pada baris ataupun kolom tersebut, kemudian menghitungkan determinannya menggunakan determinanKofaktor secara rekursif. Fungsi ini mengembalikan hasil determinan minornya yang berupa double.

4) **getKofaktor(<u>input:</u> <u>int</u> i, <u>input:</u> <u>int</u> j)**

   Fungsi ini menerima parameter i dan j yang berupa integer. Kegunaan fungsi ini adalah untuk mencari minor dari suatu matriks di baris ke- i kolom ke-j dengan cara memanfaatkan fungsi getMinor sebelumnya, lalu mengalikan tandanya (1 apabila i+j berjumlah genap, dan -1 apabila i+j berjumlah ganjil). Fungsi ini mengembalikan entri kofaktor dari sebuah matriks yang berupa double.

5) **determinanKofaktor()**

Fungsi ini memanfaatkan getKofaktor(i,j) dalam penerapannya untuk mencari entri kofaktor yang dibutuhkan. Dengan fungsi ini, determinan dihitung dengan cara mengambil tiap elemen pada baris pertama, kemudian dikalikan dengan entri kofaktor yang bersesuaian lalu ditotal jumlahnya. Fungsi ini mengembalikan determinan dari sebuah matriks yang berupa double.

Adapun fungsi dan prosedur yang berkaitan dengan pencarian invers (balikan) dari sebuah matrix adalah sebagai berikut :

1) **InverseOBE()**

   mempersiapkan matriks dalam bentuk [A|I] untuk dioperasikan OBE secara Gauss-Jordan

2) **InverseKofaktor()**

   mengembalikan matriks balikan dengan metode kofaktor dengan pre-kondisi matriks persegi nxn dan sudah dicek sebelumnya apakah det(A) = 0.

   Ditemukan matriks kofaktor terlebih dahulu dengan MatriksKofaktor() yang lalu ditranspose dengan Transpose() sehingga menghasilkan adjoint. Terakhir semua anggota elemen matriks dibagi determinan untuk mendapatkan invers.

3) **MatriksKofaktor()**

   membentuk matriks kofaktor suatu matriks dengan memanggil getKofaktor() pada setiap elemen.

4) **Transpose()**

   mengembalikan transpose dari matriks dengan membuat matriks yang jumlah baris dan kolomnya ditukar dengan originalnya yang lalu semua elemennya diisi dengan baris dan kolom yang tertukar itu.

Adapun fungsi dan prosedur yang berkaitan dengan subprogram interpolasi dan regresi linear berganda adalah sebagai berikut :

1) **getRegresiElement(<u>input:</u> <u>int</u> i, <u>input:</u> <u>int</u> j)**

   Fungsi ini diterapkan pada sebuah objek berkelas matriks yang tersusun secara k baris dan n+2 kolom, dengan k adalah banyaknya titik yang diinput dan n adalah banyaknya variabel peubah (2 kolom extra untuk ordinat, dan untuk konstanta yang koefisiennya selalu 1). Kemudian, fungsi ini menerima parameter i dan j yang berupa integer dan mengembalikan sebuah nilai double yang berupa elemen matriks sesuai *Normal Estimation Equation for Multiple Linear Regression* pada baris ke-i dan elemen ke-j dengan cara mengalikan variabel peubah yang bersesuaian pada indeks tersebut untuk tiap titik kemudian dijumlahkan semua hasil kalinya.

BAB IV

Eksperimen (Studi Kasus)

A. Hasil Eksekusi Program

Studi kasus Sistem Persamaan Linear :

```
Silahkan menginput matriks A:
Ketik 1 untuk menambah matriks by input user
Ketik 2 untuk input matriks dari file txt
Masukkan sub-pilihan anda: 3
Pilihan salah, ulangi
Masukkan sub-pilihan anda: 1
Masukkan baris matriksnya: 4
Masukkan kolom matriksnya: 4
Masukkan nilai untuk baris ke-1 kolom ke-1: 1
Masukkan nilai untuk baris ke-1 kolom ke-2: 1
Masukkan nilai untuk baris ke-1 kolom ke-3: -1
Masukkan nilai untuk baris ke-1 kolom ke-4: -1
Masukkan nilai untuk baris ke-2 kolom ke-1: 2
Masukkan nilai untuk baris ke-2 kolom ke-2: 5
Masukkan nilai untuk baris ke-2 kolom ke-3: -7
Masukkan nilai untuk baris ke-2 kolom ke-4: -5
Masukkan nilai untuk baris ke-3 kolom ke-1: 2
Masukkan nilai untuk baris ke-3 kolom ke-2: -1
Masukkan nilai untuk baris ke-3 kolom ke-3: 1
Masukkan nilai untuk baris ke-3 kolom ke-4: 3
Masukkan nilai untuk baris ke-4 kolom ke-1: 5
Masukkan nilai untuk baris ke-4 kolom ke-2: 2
Masukkan nilai untuk baris ke-4 kolom ke-3: -4
Masukkan nilai untuk baris ke-4 kolom ke-4: 2
Determinan matriks A adalah 0 sehingga tidak bisa menggunakan metode ini karena tidak ada inversnya.
Akan dikembalikan ke menu utama
```

Soal 1a

```
Masukkan baris matriksnya: 4
Masukkan kolom matriksnya: 5
Masukkan nilai untuk baris ke-1 kolom ke-1: 1
Masukkan nilai untuk baris ke-1 kolom ke-2: -1
Masukkan nilai untuk baris ke-1 kolom ke-3: 0
Masukkan nilai untuk baris ke-1 kolom ke-4: 0
Masukkan nilai untuk baris ke-1 kolom ke-5: 1
Masukkan nilai untuk baris ke-2 kolom ke-1: 1
Masukkan nilai untuk baris ke-2 kolom ke-2: 1
Masukkan nilai untuk baris ke-2 kolom ke-3: 0
Masukkan nilai untuk baris ke-2 kolom ke-4: -3
Masukkan nilai untuk baris ke-2 kolom ke-5: 0
Masukkan nilai untuk baris ke-3 kolom ke-1: 2
Masukkan nilai untuk baris ke-3 kolom ke-2: -1
Masukkan nilai untuk baris ke-3 kolom ke-3: 0
Masukkan nilai untuk baris ke-3 kolom ke-4: 1
Masukkan nilai untuk baris ke-3 kolom ke-5: -1
Masukkan nilai untuk baris ke-4 kolom ke-1: -1
Masukkan nilai untuk baris ke-4 kolom ke-2: 2
Masukkan nilai untuk baris ke-4 kolom ke-3: 0
Masukkan nilai untuk baris ke-4 kolom ke-4: -2
Masukkan nilai untuk baris ke-4 kolom ke-5: -1
Matriks A tidak persegi, tidak bisa menggunakan metode ini.
Akan dikembalikan ke menu utama
-------------------------------------------------------
```

Soal 1b

```
Matrix awal
0.000000 1.000000 0.000000 0.000000 1.000000 0.000000 2.000000
0.000000 0.000000 0.000000 1.000000 1.000000 0.000000 -1.000000
0.000000 1.000000 0.000000 0.000000 0.000000 1.000000 1.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
```

```
Baris ke-3 dibagi dengan 1.000000
0.000000 1.000000 0.000000 0.000000 0.000000 1.000000 1.000000
0.000000 0.000000 0.000000 1.000000 0.000000 1.000000 -2.000000
-0.000000 -0.000000 -0.000000 -0.000000 1.000000 -1.000000 1.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
x2 = 1.0- 1.0*F
x4 = -2.0- 1.0*F
x5 = 1.0 + 1.0*F
x1 = A
x3 = C
x6 = F
```

Matriks awal, matriks eselon baris tereduksi, dan solusi 1c

```
Menggunakan metode matriks balikan, bentuk matriks Ax = B
Silahkan menginput matriks A:
Matrix A awal adalah
1.000000 0.500000 0.333333 0.250000 0.200000 0.166667
0.500000 0.333333 0.250000 0.200000 0.166667 0.142857
0.333333 0.250000 0.200000 0.166667 0.142857 0.125000
0.250000 0.200000 0.166667 0.142857 0.125000 0.111111
0.200000 0.166667 0.142857 0.125000 0.111111 0.100000
0.166667 0.142857 0.125000 0.111111 0.100000 0.090909
Silahkan menginput matriks B:
Masukkan matriks B baris ke-1: 1
Masukkan matriks B baris ke-2: 0
Masukkan matriks B baris ke-3: 0
Masukkan matriks B baris ke-4: 0
Masukkan matriks B baris ke-5: 0
Masukkan matriks B baris ke-6: 0
Invers matrix A adalah
35.999998 -629.999959 3359.999775 -7559.999501 7559.999577 -2771.999864
-629.999950 14699.998972 -88199.994161 211679.986184 -220499.986154 83159.994832
3359.999710 -88199.993925 564479.963160 -1411199.908660 1511999.902708 -582119.962564
-7559.999411 211679.985938 -1411199.908773 3628799.766015 -3968999.743996 1552319.899930
7559.999516 -220499.985900 1511999.902789 -3968999.744481 4409999.716145 -1746359.887281
-2771.999864 83159.994860 -582119.962672 1552319.900129 -1746359.887796 698543.954658
Matrix B adalah
1.000000
0.000000
0.000000
0.000000
0.000000
0.000000
Hasil kali invers matrix A dengan matrix B adalah
35.999998
-629.999950
3359.999710
-7559.999411
7559.999516
-2771.999864
Sehingga solusi dari paling atas adalah
X1 adalah 35.999998
X2 adalah -629.999950
X3 adalah 3359.999710
X4 adalah -7559.999411
X5 adalah 7559.999516
X6 adalah -2771.999864
```

Soal 1d dengan n = 6

```
Matrix awal
1.000000 -1.000000 2.000000 -1.000000 -1.000000
2.000000 1.000000 -2.000000 -2.000000 -2.000000
-1.000000 2.000000 -4.000000 1.000000 1.000000
3.000000 0.000000 0.000000 -3.000000 -3.000000
Tukar baris ke-1 dan baris ke-4
3.000000 0.000000 0.000000 -3.000000 -3.000000
2.000000 1.000000 -2.000000 -2.000000 -2.000000
-1.000000 2.000000 -4.000000 1.000000 1.000000
1.000000 -1.000000 2.000000 -1.000000 -1.000000
Baris ke-2 dikurangi baris ke-1 dikali 0.666667
3.000000 0.000000 0.000000 -3.000000 -3.000000
0.000000 1.000000 -2.000000 0.000000 0.000000
-1.000000 2.000000 -4.000000 1.000000 1.000000
1.000000 -1.000000 2.000000 -1.000000 -1.000000
Baris ke-3 dikurangi baris ke-1 dikali -0.333333
3.000000 0.000000 0.000000 -3.000000 -3.000000
0.000000 1.000000 -2.000000 0.000000 0.000000
0.000000 2.000000 -4.000000 0.000000 0.000000
1.000000 -1.000000 2.000000 -1.000000 -1.000000
Baris ke-4 dikurangi baris ke-1 dikali 0.333333
3.000000 0.000000 0.000000 -3.000000 -3.000000
0.000000 1.000000 -2.000000 0.000000 0.000000
0.000000 2.000000 -4.000000 0.000000 0.000000
0.000000 -1.000000 2.000000 0.000000 0.000000
Tukar baris ke-2 dan baris ke-3
3.000000 0.000000 0.000000 -3.000000 -3.000000
0.000000 2.000000 -4.000000 0.000000 0.000000
0.000000 1.000000 -2.000000 0.000000 0.000000
0.000000 -1.000000 2.000000 0.000000 0.000000
Baris ke-3 dikurangi baris ke-2 dikali 0.500000
3.000000 0.000000 0.000000 -3.000000 -3.000000
0.000000 2.000000 -4.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 -1.000000 2.000000 0.000000 0.000000
Baris ke-4 dikurangi baris ke-2 dikali -0.500000
3.000000 0.000000 0.000000 -3.000000 -3.000000
0.000000 2.000000 -4.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
```

```
Baris ke-1 dibagi dengan 3.000000
1.000000 0.000000 0.000000 -1.000000 -1.000000
0.000000 2.000000 -4.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
Baris ke-2 dibagi dengan 2.000000
1.000000 0.000000 0.000000 -1.000000 -1.000000
0.000000 1.000000 -2.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
Memiliki tak hingga solusi
Matrix awal
1.000000 0.000000 0.000000 -1.000000 -1.000000
0.000000 1.000000 -2.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
Baris ke-1 dikurangi baris ke-2 dikali 0.000000
1.000000 0.000000 0.000000 -1.000000 -1.000000
0.000000 1.000000 -2.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
Baris ke-1 dibagi dengan 1.000000
1.000000 0.000000 0.000000 -1.000000 -1.000000
0.000000 1.000000 -2.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
Baris ke-2 dibagi dengan 1.000000
1.000000 0.000000 0.000000 -1.000000 -1.000000
0.000000 1.000000 -2.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
x1 = -1.0 + 1.0*D
x2 = 0.0 + 2.0*C
x3 = C
x4 = D
```

Proses pengerjaan soal 2a menggunakan metode gauss

```
Matrix awal
2.000000 0.000000 8.000000 0.000000 8.000000
0.000000 1.000000 0.000000 4.000000 6.000000
-4.000000 0.000000 6.000000 0.000000 6.000000
0.000000 -2.000000 0.000000 3.000000 -1.000000
2.000000 0.000000 -4.000000 0.000000 -4.000000
0.000000 1.000000 0.000000 -2.000000 0.000000
```

```
1.000000 0.000000 0.000000 0.000000 0.000000
0.000000 1.000000 0.000000 0.000000 2.000000
0.000000 0.000000 1.000000 0.000000 1.000000
0.000000 0.000000 0.000000 1.000000 1.000000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
x1 = 0.0
x2 = 2.0
x3 = 1.0
x4 = 1.0
```

Matriks awal dan akhir soal 2b

```
Matrix awal
8.000000 1.000000 3.000000 2.000000 0.000000
2.000000 9.000000 -1.000000 -2.000000 1.000000
1.000000 3.000000 2.000000 -1.000000 2.000000
1.000000 0.000000 6.000000 4.000000 3.000000

1.000000 0.000000 0.000000 0.000000 -0.224324
0.000000 1.000000 0.000000 0.000000 0.182432
0.000000 0.000000 1.000000 0.000000 0.709459
0.000000 0.000000 0.000000 1.000000 -0.258108
Penyelesaian persamaan tersebut adalah
[-0.22432432432432442, 0.18243243243243248, 0.7094594594594594, -0.25810810810810797]
```

Soal 3a dengan metode gauss jordan

```
Matrix awal
-120.000000 60.000000 0.000000 -1300.000000
40.000000 -80.000000 0.000000 0.000000
80.000000 20.000000 -150.000000 -200.000000

1.000000 0.000000 0.000000 14.444444
-0.000000 1.000000 -0.000000 7.222222
-0.000000 -0.000000 1.000000 10.000000
Penyelesaian persamaan tersebut adalah
[14.444444444444443, 7.222222222222222, 10.0]
```

Soal no 5 dengan metode gauss jordan

Studi Kasus Interpolasi:

```
Matrix awal
1.000000 0.100000 0.010000 0.001000 0.000100 0.000010 0.000001 0.003000
1.000000 0.300000 0.090000 0.027000 0.008100 0.002430 0.000729 0.067000
1.000000 0.500000 0.250000 0.125000 0.062500 0.031250 0.015625 0.148000
1.000000 0.700000 0.490000 0.343000 0.240100 0.168070 0.117649 0.248000
1.000000 0.900000 0.810000 0.729000 0.656100 0.590490 0.531441 0.370000
1.000000 1.100000 1.210000 1.331000 1.464100 1.610510 1.771561 0.518000
1.000000 1.300000 1.690000 2.197000 2.856100 3.712930 4.826809 0.697000
```

```
Koefisien-koefisien dari polinomial yang bersesuaian, dimulai dari derajat 0 di paling kiri
[-0.022976562500000387, 0.2400000000000693, 0.19739583333329394, 9.985982746837269E-14, 0.026041666666540748, 7.710258574544819E-14, -1.82723
8296772476E-14]
Masukkan nilai ordinat yang ingin diinterpolasi
0.2
Hasil interpolasi adalah : 0.032961
```

```
Masukkan nilai ordinat yang ingin diinterpolasi
0.55
Hasil interpolasi adalah : 0.171119
```

```
Masukkan nilai ordinat yang ingin diinterpolasi
0.85
Hasil interpolasi adalah : 0.337236
```

```
Masukkan nilai ordinat yang ingin diinterpolasi
1.28
Hasil interpolasi adalah : 0.677542
```

Soal 6a

```
Matrix awal
1.000000 6.657000 44.315649 295.009275 1963.876746 13073.527500 87030.472568 579361.855885 3856811.874623 25674796.649367 12624.000000
1.000000 7.000000 49.000000 343.000000 2401.000000 16807.000000 117649.000000 823543.000000 5764801.000000 40353607.000000 21807.000000
1.000000 7.258000 52.678564 382.341018 2775.031105 20141.175761 146184.653672 1061008.216352 7700797.634284 55892389.229634 38391.00000
1.000000 7.451000 55.517401 413.660155 3082.181814 22965.336695 171114.723711 1274975.806373 9499844.733287 70783343.107724 54517.00000
1.000000 7.548000 56.972304 430.026951 3245.843423 24499.626157 184923.178235 1395800.149321 10535499.527076 79521950.430373 51952.0000
1.000000 7.839000 61.449921 481.705931 3776.092791 29600.791388 232040.603690 1818966.292325 14258876.765534 111775334.965022 28228.000
```

```
Koefisien-koefisien dari polinomial yang bersesuaian, dimulai dari derajat 0 di paling kiri
[2.343777280369853E12, -3.819734918748623E12, 2.5330377883669336E12, -9.2939122858921E11, 2.1156278391676788E11, -3.129126307661515E10,
255227701799703E9, -1.851560608416585E8, 6526319.208728365, -101154.79349020561]
Masukkan nilai ordinat yang ingin diinterpolasi
7.516
Hasil interpolasi adalah : 53541.610352
```

Soal 6b menggunakan tanggal 16/07/2021

```
Matrix awal
1.000000 0.400000 0.160000 0.064000 0.025600 0.418884
1.000000 0.800000 0.640000 0.512000 0.409600 0.507158
1.000000 1.200000 1.440000 1.728000 2.073600 0.560925
1.000000 1.600000 2.560000 4.096000 6.553600 0.583686
1.000000 2.000000 4.000000 8.000000 16.000000 0.576652
```

```
Koefisien-koefisien dari polinomial yang bersesuaian, dimulai dari derajat 0 di paling kiri
[0.2903119999999999, 0.3780595833333338, -0.15058802083333397, 0.02402604166666706, -0.003727213541666749]
Masukkan nilai ordinat yang ingin diinterpolasi
1
Hasil interpolasi adalah : 0.538082
```

Soal 6c

Studi Kasus Regresi Linear Berganda :

```
Ketik 1 untuk memasukkan titik melalui CLI atau 0 untuk memasukkan titik melalui file txt
1
Jumlah variabel peubah yang ingin dimasukkan :
3
Jumlah titik yang ingin dimasukkan :
20
```

```
Matrix awal
20.00 863.10 1530.40 587.84 19.42
863.10 54876.89 67000.09 25283.40 779.48
1530.40 67000.09 117912.32 44976.87 1483.44
587.84 25283.40 44976.87 17278.51 571.12
```

```
1.00 43.78 77.05 29.39 0.97
0.00 1.00 0.03 -0.00 -0.00
-0.00 -0.00 1.00 -0.00 0.00
0.00 0.00 0.00 1.00 0.15
nilai dari B yang bersesuaian, dimulai dari derajat 0 di paling kiri
[-3.507778140873786, -0.0026249907458798733, 7.989410472207764E-4, 0.1541550301979756]
Masukkan nilai nilai Xk yang ingin ditaksir fungsinya
Masukkan nilai dari X1: 50
Masukkan nilai dari X2: 76
Masukkan nilai dari X3: 29.3
Hasil taksiran adalah : 0.94
```

Soal no 7

B. Analisis

Pada studi kasus SPL, program dapat mengidentifikasi SPL yang memiliki solusi unik, solusi banyak, ataupun tidak memiliki solusi. Pada soal 1a dan 1d, SPL memiliki solusi unik menggunakan metode eliminasi Gauss. Pada soal 1b, SPL tidak memiliki solusi. Pada soal 2a, SPL memiliki banyak solusi sehingga program menampilkan solusinya secara parametrik. Untuk soal no 3a dan 5 juga memiliki solusi unik menggunakan metode eliminasi Gauss-Jordan. Untuk tiap persoalan SPL, program kami juga menampilkan pembahasan / cara dari penyelesaian dari permasalahan tersebut.

Pada studi kasus interpolasi, program mampu membuat persamaan polinomial dari data titik yang diberikan. Pada soal 6a, program dapat mengkalkulasi koefisien dari polinomial berdasarkan sebaran titik dari tabel yang diberikan. Pada soal 6c, program mampu menyederhanakan fungsi yang sulit dengan mengambil beberapa titik dari fungsi tersebut dan menaksir nilai dari sebuah ordinat untuk titik yang lain. Untuk studi kasus regresi linear berganda, terlihat program mampu menyusun matrix berdasarkan *Normal Estimation Equation for Multiple Linear Regression* dari n jumlah titik untuk k jumlah variabel peubah. Terlihat juga bahwa matrix yang tersusun sama persis dengan hasil yang disediakan pada soal, maka kalkulasi program terbukti benar. Adapun dokumentasi dari eksperimen kami ada yang terpotong langkahnya karena terlalu panjang, tersedia dokumentasi lengkapnya pada folder docs repo kami.

BAB V

Kesimpulan

A. Kesimpulan

Pada tugas kali ini diketahui bahwa seluruh metode perhitungan matriks yang dipelajari di kelas dapat digeneralisir menjadi sebuah algoritma. Algoritma tersebut juga memungkinkan untuk dituliskan dalam bentuk kode di bahasa Java. Hal ini membuat perhitungan menjadi cepat dan minim kesalahan, mengingat kesalahan menghitung ketika seseorang melakukan operasi-operasi tersebut cukup tinggi.

Sebetulnya, sudah banyak algoritma untuk metode eliminasi Gauss, eliminasi Gauss Jordan, mencari matriks balikan, dan mencari determinan yang ditulis dalam bahasa Java di internet. Namun, kode-kode tersebut kebanyakan tidak bisa menyelesaikan sistem persamaan linear yang memiliki solusi parametrik. Pada tugas ini, ditemukan cara bagaimana mengatasi situasi tersebut. Program tidak hanya menampilkan solusi untuk persamaan yang memiliki solusi tunggal, tapi juga bisa menampilkan solusi berbentuk parametrik.

B. Saran

Pada program yang dibuat pada tugas ini, belum tercapai modularisasi program yang sempurna. Terdapat beberapa fungsi yang digunakan beberapa kali, namun belum dibuatkan fungsi secara khusus. Hal ini menyebabkan baris kode yang lebih banyak. Pada algoritma perhitungan solusi parametrik, matriks augmented juga harus dikonversi ke matriks eselon baris tereduksi terlebih dahulu, meskipun pengguna memilih metode Gauss. Pada tugas ini, belum bisa ditemukan bagaimana cara mencari solusi parametrik dari sebuah matriks *augmented* yang berbentuk matrik eselon baris.

C. Refleksi

Pada tugas ini, hal yang penting untuk dipelajari dan diingat adalah bagaimana berkomunikasi dalam suatu tim. Sesungguhnya, menulis kode untuk program yang kecil itu mudah. Menulis kode untuk program yang lima kali lebih besar, bukan berarti kesulitannya bertambah lima kali lipat, tapi lebih dari itu. Hal ini menjadi penting karena suatu hari nanti tidak mungkin satu program hanya dikerjakan oleh satu orang. Suatu hari nanti pasti dibutuhkan kemampuan untuk berkoordinasi dan berdiskusi dengan *programmer* lain. Semoga pengalaman dalam mengerjakan tugas ini bisa berguna di masa depan khususnya dalam konteks berkomunikasi dengan orang lain

DAFTAR REFERENSI

Varyani, Y. (2021, April 15). *Gaussian elimination to solve linear equations*. GeeksforGeeks. Retrieved September 23, 2021, from https://www.geeksforgeeks.org/gaussian-elimination/.

Bisht, A. (2021, July 16). *Program for Gauss-Jordan Elimination Method*. GeeksforGeeks. Retrieved September 23, 2021, from https://www.geeksforgeeks.org/program-for-gauss-jordan-elimination-method/.