BAB I

DESKRIPSI MASALAH

1.1 Deskripsi Umum Tugas Besar

Tugas besar ini adalah membuat program menghitung solusi Sistem Persamaan Linier (SPL) secara numerik dalam bahasa pemrograman Java dengan menggunakan metode eliminasi Gauss dan/atau Gauss-Jordan. SPL dapat memiliki solusi unik, banyak solusi, atau solusi tidak ada. SPL juga digunakan dalam menentukan persamaan polinom interpolasi. Karena perhitungan menggunakan representasi bilangan titik-kambang (floating point) di dalam komputer, maka untuk meminumkan galat perhitungan, digunakan strategi pivoting dalam memilih baris yang dijadikan basis dalam operasi baris elementer. Bahasa Java digunakan sebagai bahan belajar penggunaan bahasa pemrograman selain C dan Pascal yang sudah digunakan selama ini.

1.2 Prosedur Pengerjaan

1. Tugas dikerjakan secara berkelompok yang terdiri dari 3 orang.
2. Tugas ini dikumpulkan hari Jumat 6 Oktober 2017 paling lambat pukul 7.30 pagi di atas loker Lab IRK. Silakan isi daftar pengumpulan dan tanggal demo program.

1.3 Bahasa Pemrograman

1. Bahasa program yang digunakan adalah Java dengan kakas pengembangan program adalah JDK.
2. Program tidak harus berbasis GUI, cukup text-based saja.

1.4 Program

Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan dirancang oleh masing-masing kelompok.

1.5 Laporan

Cover: Cover laporan ada foto anggota kelompok (foto bertiga, bebas gaya). Foto ini menggantikan logo “gajah” ganesha.

Bab 1: Deskripsi masalah (dapat meng-copy paste file tugas ini) Bab 2: Teori singkat mengenai metode eliminasi Gauss, metode eliminasi Gauss- Jordan, tatancang pemorosan, polinom interpolasi, dll.

Bab 3: Implementasi program dalam Java, meliputi kelas Java yang dibuat, lengkap dengan method dan atributnya.

Bab 4: Eksperimen. Bab ini berisi hasil eksekusi program terhadap contoh-contoh kasus yang diberikan berikut analisis hasil eksekusi tersebut.

Bab 5: Kesimpulan dan saran (hasil yang dicapai, saran pengembangan).

Tuliskan juga referensi (buku, web), yang dipakai/diacu di dalam Daftar Referensi.

Keterangan laporan dan program:

a) Laporan ditulis dalam bahasa Indonesia yang baik dan benar, tidak perlu panjang tetapi tepat sasaran dan jelas.

b) Laporan tidak perlu memakai cover mika dan dijilid. Cukup dibuat agar laporan tidak akan tercecer bila dibaca.

c) Laporan boleh menggunakan kertas rius, boleh bolak-balik, boleh dalam satu halaman kertas terdapat dua halaman tulisan asalkan masih terbaca.

d) Identitas per halaman harus jelas (misalnya : halaman, kode kuliah).

e) Listing program ataupun algoritma tidak perlu disertakan pada laporan.

f) Program disimpan di dalam folder Algeo-xxxxx. Lima digit terakhir adalah NIM anggota terkecil. Didalam folder tersebut terdapat tiga folder bin, src dan doc yang masing-masing berisi :

1) Folder bin berisi java byte code (.class)

2) Folder src berisi source code dari program java

3) Folder test berisi data uji

4) Folder doc berisi dokumentasi program dan readme

1.6 Pengumpulan Tugas

1. Yang diserahkan saat pengumpulan tugas adalah:

a) CD/DVD yang berisi program sumber (source code) dan arsip java yang sudah dikompilasi tanpa ada kesalahan.

b) Laporan

2. Java bytecode di dalam CD/DVD dapat dijalankan. Asisten pemeriksa tidak akan melakukan setting atau kompilasi lagi agar program dapat berjalan. Program yang tidak dapat dijalankan tidak akan diberi nilai.

3. CD dan laporan akan dikembalikan setelah dinilai.

1.7 Penilaian

Komposisi penilaian umum adalah sebagai berikut :

1. Program: 80 %

2. Laporan : 20 %

1.8 Spesifikasi Umum

1. Program harus dapat menerima input data dari

• Papan ketik

• File

2. Keluaran program harus dapat ditampilkan ke:

• Layar monitor

• Simpan ke dalam arsip

Format luaran (misalnya dalam bentuk tabel) didefinisikan sendiri. Luaran harus mudah dibaca dan informatif.

1.9 Spesifikasi Materi

Tulislah program java secara umum untuk menyelesaikan sistem persamaan lanjar (SPL) dengan n peubah (variable) dan m persamaan:

$$a11\ x1 + a12\ x2 + ... + a1n\ xn = b1$$

$$a21\ x1 + a22\ x2 + ... + a2n\ xn = b2$$

$$\vdots \qquad \vdots$$

$$\vdots \quad \vdots$$

am1 x1 + am2 x2 + ... + amn xn = bm

SPL dapat diselesaikan secara numerik dengan metode eliminasi Gauss dan metode eliminasi Gauss-Jordan. Di dalam kedua metode tersebut diterapkan tatancang pemorosan (pivoting) untuk mengurangi galat pembulatan. Program harus dapat menangani kasus-kasus sebagai berikut:

a) SPL memiliki solusi unik, tampilkan solusinya

b) SPL memiliki solusi tak terbatas, tampilkan solusinya dalam bentuk parameter

c) SPL tidak memiliki solusi, tuliskan tidak ada solusinya.

Contoh-contoh SPL yang dijadikan data eksperimen:

a) 0.31x1 + 0.14x2 + 0.30x3 + 0.27x4 = 1.02

0.26x1 + 0.32x2 + 0.18x3 + 0.24x4 = 1.00

0.61x1 + 0.22x2 + 0.20x3 + 0.31x4 = 1.34

0.40x1 + 0.34x2 + 0.36x3 + 0.17x4 = 1.27

b) x1 + 7x2 – 2x3 + 8x5 = -3

x1 + 7x2 – x3 + x4 = 2

2x1 + 14x2 – 4x3 + x4 – 13x5 = 3

2x1 + 14x2 – 4x3 + 16x5 = -6

c) HX = B, yang dalam hal ini H adalah matriks Hilbert yang memiliki bentuk

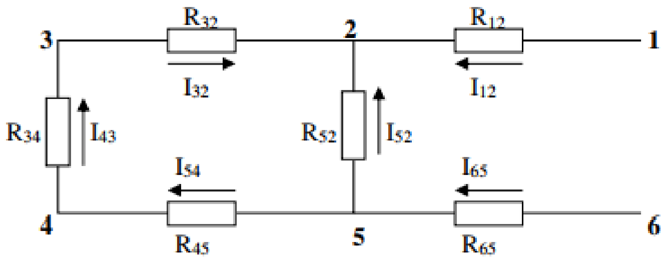
sebagai berikut:

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

dan B = (1, 1, 1, …,1)^T. Uji untuk n = 10 dan n = 20.

d) Sebuah perusahaan di AS memperoleh keuntungan (sebelum dipotong pajak) sebesar \$100,000. Perusahaan setuju untuk mengkontribusikan 10% dari keuntungannya (setelah dipotong pajak) untuk Corporate Social Responsibility (CSR). Perusahaan membayar pajak daerah sebesar 5% dari keuntungannya (setelah dipotong CSR) dan pajak federal sebesar 40% dari keuntungangannya (setelah dipotong CSR dan pajak daerah). Berapa banyak uang yang dibayarkan perusahaan untuk pajak daerah, pajak federal, dan CSR? Modelkan ke dalam SPL dan selesaikan dengan Gauss/Gauss-Jordan.

e) Diberikan sebuah rangkaian listrik sbb :



Anda diminta menghitung arus pada masing-masing rangkaian. Arah arus dimisalkan seperti diatas. Dengan hukum Kirchoff diperoleh persamaanpersamaan berikut :

I12 + I52 + I32 = 0

I65 - I52 - I54 = 0

I43 - I32 = 0

I54 - I43 = 0

Dari hukum Ohm didapat :

I32R32 - V3 + V2 = 0

I43R43 - V4 + V3 = 0

I65R65 + V5 = 0

I12R12 + V2 = 0

I54R54 - V5 + V4 = 0

I52R52 - V5 + V2 = 0

Tentukan I12 , I52 , I32 , I65 , I54 , I13 , V2 , V3 , V4 , V5 bila :

R12 = 5 ohm, R52 = 10 ohm, R32 = 10 ohm,

R65 = 20 ohm, R54 = 15 ohm, R14 = 5 ohm,

V1 = 200 volt, V6 = 0 volt

(f) (Interpolasi) Hampiri fungsi berikut

$$f(x) = \frac{e^{-x}}{1 + \sqrt{x} + x^2}$$

\

dengan polinom interpolasi derajat n:

$$p(x) = a_0 + a_1x + a_2x_2 + \ldots + a_nx_n$$

di dalam selang [a, b]. Gunakan titik-titik selebar h, yang dalam hal ini $h = (b - a)/n$. Sebagai tes, gunakan selang [0, 5] dan selang [-2, 2], n = 5, 10, dan 12. Tentukan persamaan polinom interpolasi yang dihasilkan.

(g) **(Interpolasi)** Gunakan tabel di bawah ini untuk mencari polinom interpolasi dari pasangan titik-titik yang terdapat dalam tabel.

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

Selanjutnya, estimasi nilai fungsi f(x) pada nilai-nilai x berikut:

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

(h) **(Interpolasi)** Harga rumah baru dari tahun 1950 hingga 1969 mengalami perubahan yang tercatat sebagai berikut:

Tahun	Harga ($ juta)
1950	33,525
1955	46,519
1960	53,941
1965	72,319
1966	75,160
1967	76,160

1968	84,690
1969	90,866

Carilah polinom yang menginterpolasi data tersebut, lalu prediksilah harga rumah baru pada tahun 1957, 1964, 1970, 1975 (atau nilai lain sesuai masukan user) dengan menggunakan polinom interpolasi derajat 7.

(i) **(Interpolasi)** Viskositas kinematika air, v, diukur pada suhu-suhu tertentu dan diperoleh hasil sebagai berikut:

T (°F)	40	50	60	70	80	90
v (10^{-5} ft^2/detik)	1.66	1.41	1.22	1.06	0.93	0.84

Carilah polinom yang menginterpolasi data tersebut, dan taksirlah viskositas pada suhu T tertentu (misalnya T = 62°F, T = 75°F, dll)

BAB 2

TEORI SINGKAT

2.1 Sistem Persamaan Lanjar

Sistem persamaan lanjar (SPL) dengan dengan n peubah dinyatakan sebagai

a11 x1 + a12 x2 + ... + a1n xn = b1

a21 x1 + a22 x2 + ... + a2n xn = b2

: :

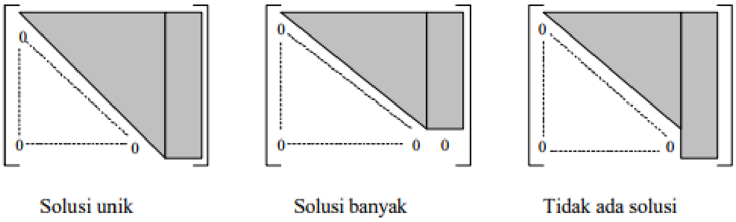
: :

an1 x1 + an2 x2 + ... + ann xn = bn

Untuk mendapatkan solusi dari SPL, bisa menggunakan metode eliminasi Gauss dan metode eliminasi Gauss-Jordan. Kemungkinan yang terjadi dari solusi SPL ada tiga:

a) mempunyai satu solusi unik
b) mempunyai banyak solusi
c) tidak ada solusi

Untuk membedakannya, dapat dilihat dari bentuk matriks setelah proses eliminasi seperti gambar di bawah ini.



Gambar 1 Matriks Solusi SPL

1) Apabila pada matriks hasil eliminasi tidak ada barisan yang berisi nol seluruhnya, maka solusi dari SPL tersebut unik.
2) Apabila pada matriks hasil eliminasi terdapat minimal 1 baris yang seluruhnya berisi nol, maka solusi dari SPL tersebut adalah tak hingga.
3) Jika pada hasil eliminasi Gauss terdapat baris yang semuanya bernilai 0 tetapi elemen pada baris yang bersesuaian pada vektor kolom b tidak 0, maka SPL tidak mempunyai solusi.

2.2 Eliminasi Gauss

Eliminasi Gauss adalah proses eliminasi sistem persamaan variabel melalui bentuk matriks menggunakan operasi baris elementer (OBE) sehingga terbentuk matriks eselon. Suatu matriks dikatakan eselon jika memenuhi syarat berikut.

a. Bila ada baris yang tidak seluruhnya terdiri dari nol, maka elemen pertama yang bukan nol harus bilangan 1. Bilangan ini disebut *leading 1*.
b. Bila ada baris yang berisi nol semua, maka baris itu diletakkan di paling bawah matriks.
c. Dalam dua baris berturut-turut yang tidak seluruhnya terdiri dari nol, *leading* 1 di baris keduanya harus lebih jauh ke kanan daripada *leading 1* di atasnya.
d. Tiap kolom yang memiliki *leading* 1, baris dibawahnya harus berisi nilai nol seluruhnya.

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

Gambar 2 Matriks Eselon Baris

2.3 Eliminasi Gauss-Jordan

Eliminasi Gauss adalah proses eliminasi sistem persamaan variabel melalui bentuk matriks menggunakan operasi baris elementer (OBE) sehingga terbentuk matriks eselon baris tereduksi. Metode eliminasi Gauss-Jordan merupakan variasi dari metode eliminasi Gauss. Definisi dari operasi baris elementer dapat dibagi menjadi tiga hal, yaitu:

a. Perkalian suatu baris dengan konstanta bukan nol
b. Pertukaran antara dua baris
c. Menambah konstanta kali suatu baris ke baris lain

Syarat suatu matriks dikatakan matriks eselon tereduksi adalah sama untuk aturan a-c pada matriks eselon. Namun perbedaannya adalah pada aturan keempat. Untuk tiap kolom yang memiliki *leading 1*, baris dibawah maupun diatasnya harus berisi nilai nol seluruhnya. Jadi, kolom tersebut hanya memiliki dua nilai yaitu nol dan satu.

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

Gambar 3 Matriks Eselon Baris Tereduksi

Terdapat tiga fakta mengenai bentuk matriks eselon baris dan juga matriks eselon baris tereduksi, yaitu:

a. Tiap matriks memiliki bentuk eselon baris tereduksi yang unik. bentuk eliminasi apapun (Gauss-Jordan maupun bukan) akan tetap menghasilkan matriks eselon baris tereduksi yang sama untuk matriks yang sama.
b. Bentuk matriks eselon baris tidak unik, satu matriks memiliki banyak bentuk matriks eselon baris.
c. Walaupun bentuk matriks eselon baris tidak unik, seluruh matriks eselon baris dari suatu matriks A memiliki jumlah baris nol yang sama dan letak *leading 1* yang sama. Hal ini disebut posisi *pivot* pada A. Sebuah kolom yang memiliki posisi pivot dinamakan kolom *pivot*.

2.4 Tatancang Pemorosan

Nilai ar, r pada posisi (r, r) yang digunakan untuk mengeliminasi xr pada baris r + 1, r + 2, ..., N dinamakan elemen pivot dan persamaan pada baris ke-r disebut persamaan pivot. Ada kemungkinan pivot bernilai nol sehingga pembagian dengan nol tidak dapat dielakkan. Tata-ancang eliminasi yang tidak mempedulikan nilai pivot adalah tatancang yang naif (naive) atau sederhana. Metode eliminasi Gauss seperti ini dinamakan metode eliminasi Gauss naif (naive Gaussian elimination), karena metodenya tidak melakukan pemeriksaan kemungkinan pembagian dengan nol. Pada metode eliminasi Gauss naif tidak ada operasi pertukaran baris dalam rangka menghindari pivot yang bernilai nol itu.

Prinsip tata-ancang pivoting adalah sebagai berikut: jika ap,p$^{(p-1)}$ = 0, cari baris k dengan ak,p ¹ 0 dan k > p, lalu pertukarkan baris p dan baris k. Metode eliminasi Gauss dengan tata-ancang pivoting disebut metode eliminasi Gauss yang diperbaiki (modified Gaussian elimination). Terdapat dua macam tatancang *pivoting:*

a. Pivoting sebagian (partial pivoting)

Pada tatancang pivoting sebagian, pivot dipilih dari semua elemen pada kolom p yang mempunyai nilai mutlak terbesar,

| ak , p | = max{|ap,p|, |ap+1,p|,…, |an-1,p|,|an,p|}

lalu pertukarkan baris ke-k dengan baris ke-p.

$$\left[\begin{array}{ccccc|c} x & x & x & x & & x \\ 0 & x & x & x & & x \\ 0 & x & x & x & & x \\ 0 & x & x & x & & x \end{array}\right]$$

Cari $|x|$ terbesar, lalu pertukarkan barisnya dengan baris ke-2

Teknik pivoting sebagian juga sekaligus menghindari pemilihan pivot = 0 (sebagaimana pada simple pivoting) karena 0 tidak akan pernah menjadi elemen dengan nilai mutlak terbesar, kecuali jika seluruh elemen di kolom yang diacu adalah 0. Apabila setelah melakukan pivoting sebagian ternyata elemen pivot = 0, itu berarti sistem persamaan lanjar tidak dapat diselesaikan (singular system).

b. Pivoting lengkap (complete pivoting)

Jika disamping baris, kolom juga diikutkan dalam pencarian elemen terbesar dan kemudian dipertukarkan, maka tatancang ini disebut pivoting lengkap. Pivoting lengkap jarang dipakai dalam program sederhana karena pertukaran kolom mengubah urutan suku x dan akibatnya menambah kerumitan program secara berarti.

2.5 Polinom Interpolasi

Interpolasi merupakan suatu pendekatan numerik yang perlu dilakukan, bila kita memerlukan nilai suatu fungsi y = f(x) yang tidak diketahui perumusannya secara tepat. Pada nilai argumen x tertentu, bila nilainya pada argument lain di sekitar argumen yang diinginkan diketahui.

Interpolasi polinomial digunakan untuk mencari titik-titik antara dari n buah titik P1(x_1,y_1), P2(x_2,y_2), P3(x_3,y_3) …, Pn(xn,yn) dengan menggunakan pendekatan fungsi polinomial pangkat n-1:

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1}$$

Penyelesaian persamaan simultan sebelumnya adalah nilai-nilai $a_0, a_1, a_2, \ldots, a_n$ yang merupakan nilai-nilai koefisien dari fungsi pendekatan polinomial yang akan digunakan. Dengan memasukkan nilai x dari titik yang dicari pada fungsi polinomialnya, akan diperoleh nilai y dari titik tersebut.

2.6 Matriks Hilbert

Dalam aljabar linier, matriks Hilbert, yang diperkenalkan oleh Hilbert (1894), adalah matriks bujursangkar dengan entri menjadi pecahan unit. Matriks Hilbert adalah suatu matriks berordo m x n, yang nilai setiap elemennya mempunyai aturan

$$A(i,j) = \frac{1}{(i+j-1)}$$

Contoh Matriks Hilbert ordo 5 x 5 =

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

BAB 3

IMPLEMENTASI PROGRAM

Untuk pembuatan Eliminasi Gauss-Jordan dengan pemrograman Java, kami membuat kelas yang bernama MatriksAlgeo dengan spesifikasi variabel sebagai berikut.

```
public class MatriksAlgeo{

	//Variabel global
	private double[][] mat = new
double[50][50];
	private int neffbrs;
	private int neffkol;
	double divider;
	double factor;
	double[] X;
	double[] Y;

	final int ADAsol = 1;
	final int  INFsol = 0;
	final int  NOsol = -1;

	private String[] hasil = new String[50];
	int neffhsl;

	private int isAdaSolusi = ADAsol;
	Scanner input = new Scanner(System.in);
```

Di dalam kelas, kami membuat beberapa *method* untuk memperlengkap program kami. Spesifikasi dan kegunaan tiap *method* adalah sebagai berikut.

```
public MatriksAlgeo()
Konstruktor untuk kelas MatriksAlgeo

public void isiMatriks()
Pembacaan matriks dengan input berasal dari
keyboard pengguna

public void tulisMatriks()
Penulisan hasil ke layar

public int countLines(String filename) throws
IOException {
Menghitung banyaknya barisan pada suatu file
eksternal
```

public void bacaFile()
Pembacaan matriks dengan input dari file eksternal

public void tulisFile()
Penulisan hasil kedalam suatu file eksternal

void swap(int i, int k, int j)
Pertukaran posisi suatu baris ke baris lainnya

void divide(int i, int j)
prosedur pembagian

void eliminate(double[][] mat, int i, int j)
prosedur eliminasi antara dua baris

public void GaussJordanBismillah()
Pembentukan matriks eselon baris tereduksi

static double pangkat(double d, double t)
Proses perpangkatan d dengan t

public void Interpolasi()
Pencarian titik-titik dari n buah titik dengan pendekatan polinomial

public void hasilInterpolasi()
Menampilkan hasil dari interpolasi

void fungsiE ()
Mencari nilai f(x) dari suatu fungsi yang mengandung eksponen dengan polinom interpolasi derajat n pad selang [a,b]

void MatriksHilbert ()
Menghitung polinomial dengan matriks Hilbert

void solutionValue()
Penentuan apakah solusi dari eliminasi unik, banyak solusi, maupun tidak ada solusi

boolean sebarisNol(int i)
Pengecakan suatu baris yang berisi seluruhnya nol

public void cetakSolusiINF()
Pencetakan hasil eliminasi apabila banyak solusi

public void hasilSolusi()
Pencetakan hasil eliminasi dalam bentuk SPL

Int firstNonZero(int brs)
Mencari elemen pertama yang bukan nol dalam suatu baris (leading 1)

public static void main(String args[])
Prosedur untuk penjalanan program

Pertama, terdapat konstruktor untuk kelas MatriksAlgeo, yaitu sebagai berikut.

```
public MatriksAlgeo(){
            for (int i = 0; i < 50; i++){
                  for (int j = 0; j < 50; j++){
                        mat[i][j] = 0;
                  }
            }
      }
```

Untuk pembacaan input dari pengguna dan pengembalian hasil ke layar pengguna digunakan *method* sebagai berikut.

```
public void isiMatriks(){
System.out.println("------PENGISIAN AUGMENTED MATRIKS--
----");
System.out.print("Baris: ");
neffbrs = input.nextInt();
System.out.print("Kolom: ");
neffkol = input.nextInt();

for(int i=0; i<neffbrs; i++){
System.out.println("Masukkan persamaan "+(i+1)+" :");
for(int j=0; j<neffkol; j++){
if (j!=neffkol-1){
System.out.print("Masukkan koefisien x"+(j+1)+" : ");
mat[i][j]=input.nextDouble();
}
else{
System.out.print("Masukkan konstanta persamaan
"+(i+1)+" : ");
mat[i][j]=input.nextDouble();
}
}
}
System.out.println();
}

public void tulisMatriks(){
System.out.println("Matriks Augmentednya adalah");

for( int i=0; i<neffbrs; i++){
System.out.print("| ");
for(int j=0; j<neffkol; j++){
if (j!=neffkol-1)
```

```
System.out.print(String.format("%.2f ", mat[i][j]));
else
System.out.print(String.format("| %.2f |",
mat[i][j]));
}
System.out.println();
}
System.out.println();
}
```

Untuk pembacaan matriks dari file eksternal dan penulisan matriks ke dalam file eksternal, digunakan *method* sebagai berikut. Penggunaan *method* CountLines digunakan untuk penentuan jumlah baris pada file eksternal.

```
public void  bacaFile() {
           int k=1;
           String array[] = new String[1000];
           String line;
           try{
                 System.out.print("Masukkan nama file
yang akan dibaca: ");
                 String fileName = input.next();
                 FileReader fr = new
FileReader(fileName);
                 BufferedReader br = new
BufferedReader(fr);
                 neffbrs = countLines(fileName);

                 System.out.println("Jumlah baris adalah
" + neffbrs);
                 while ((line = br.readLine()) != null)
{
                       String[] kosong =
line.split("\\s+");
                       for (String part : kosong) {
                             array[k]= part;
                             k++;
                       }
                 }
                 neffkol=k/neffbrs;
                 k=1;
                 System.out.println("Jumlah kolom adalah
" + neffkol);
                 for (int i=0; i<neffbrs;i++) {
                       for(int j=0; j<neffkol;j++) {
                             mat[i][j]=
Double.parseDouble(array[k]);
```

```
                            }
                  }
                  br.close();
        }
            catch(Exception e){
                  System.out.println("Pembacaan file
error.");
            }
      }

public int countLines(String filename) throws
IOException {
    InputStream is = new BufferedInputStream(new
FileInputStream(filename));
    try {
        byte[] c = new byte[1024];
        int count = 0;
        int readChars = 0;
        boolean empty = true;
        while ((readChars = is.read(c)) != -1) {
            empty = false;
            for (int i = 0; i < readChars; ++i) {
                if (c[i] == '\n') {
                    ++count;
                }
            }
        }
        return (count == 0 && !empty) ? 1 : count;
    } finally {
        is.close();
    }
}

public void tulisFile() {
      System.out.print("Masukan nama file yang akan
ditulis: ");
      String namaFile = input.next();

      BufferedWriter bw = null;
      FileWriter fw = null;

      try {
            fw = new FileWriter(namaFile);
            bw = new BufferedWriter(fw);
            for (int i=0; i<neffbrs; i++) {
                  for (int j=0;j<neffkol; j++) {
```

```
bw.write(String.format("%.2f", mat[i][j]) + " ");
                  }
                  bw.write("\r\n");
            }
            bw.write("\r\n");
            for (int i =1; i<=neffhsl;i++) {
                  bw.write(hasil[i] + " \r\n");
            }
            bw.close();
            fw.close();
      } catch (Exception e) {
            System.out.println("penulisan file
error.");
      }
}
```

Setelah itu, terdapat *method* swap, divide, eliminate, dan GaussJordanBismillah. Bentuk realisasinya adalah sebagai berikut.

```
void swap(double[][] mat, int i, int k, int j){
      double temp;
      for(int q=j; q<=neffkol; q++){
            temp = mat[i][q];
            mat[i][q] = mat[k][q];
            mat[k][q] = temp;
      }
}

void divide( int i, int j){
      for(int q=j+1; q<neffkol; q++)
            mat[i][q] /= mat[i][j];
      mat[i][j] = 1;
   }

void eliminate(double[][] mat, int i, int j){
      for(int p=0; p<neffbrs; p++){
            if( p!=i && mat[p][j]!=0 ){
                  for(int q=j+1; q<neffkol; q++){
                        mat[p][q] -=  mat[p][j]*mat[i][q];
                  }
                  mat[p][j] = 0;
            }
      }
}

void gaussJordanBismillah(){
      int i = 0;
      int j = 0;

      while( i<neffbrs && j<neffkol ){
```

```
int k = i;
      while( k<neffbrs && mat[k][j]==0 )
      k++;
            if( k<neffbrs )
                  if( k!=i ) {
                        swap(mat, i, k, j);
                  }
                  if( mat[i][j]!=1 ){
                        divide(mat, i, j);
                  }

                  eliminate(mat, i, j);

                  i++;
            }
            j++;
      }
}
```

Setelah itu, terdapat *method* pangkat, Interpolasi, hasilInterpolasi, fungsiE, dan MatriksHilbert. Bentuk realisasi untuk keempat *method* tersebut adalah sebagai berikut.

```
static double pangkat(double d, int t) {
      double hasil = 1;
      if (t==0) {
            return 1;
      }
      for (int i = 0; i<t; i++) {
            hasil *= d;
      }
      return hasil;
}

public void Interpolasi() {
      System.out.print("Masukkan jumlah titik: \n");
      Scanner scan = new Scanner(System.in);
      int titik = scan.nextInt  ();
      double[] X = new double [titik];      double[] Y =
new double [titik];

      neffbrs = titik;
      neffkol = titik+1;
      //memasukkan input x dan y(hasil)
      for (int i=0; i<titik; i++) {
            System.out.print("Masukkan nilai x" + i + " :
");
```

```
X[i] = scan.nextDouble();
          System.out.println();
          for (int j=0; j< titik; j++) {
               mat[i][j] = pangkat(X[i],   j);
          }

          System.out.print("Masukkan nilai y" + i + "
: ");
          Y[i] = scan.nextDouble();
          System.out.println();
          mat[i][titik] = Y[i];
     }
     System.out.print("Dengan metode Interpolasi
Polinom: \n");
     tulisMatriks();
     System.out.println("Hasil matriks setelah
menggunakan metode Gauss Jordan: \n");
     gaussJordanBismillah();
     tulisMatriks();
     hasilInterpolasi();

     }
public void hasilInterpolasi(){
     gaussJordanBismillah();
     solutionValue();

     Scanner scan = new Scanner(System.in);
     if (isAdaSolusi==NOsol)
          {System.out.println("Tidak ada solusi untuk
sistem persamaan diatas.");
     }else if (isAdaSolusi==ADAsol){
          int j = neffkol-1;
          for (int i=0; i<neffbrs; i++){
               System.out.println(String.format("a%d =
%.2f",i,mat[i][j]));
          }
                    System.out.print("Persamaannya
adalah: \n");
     System.out.print("f(x) = ");
     for (int i=0; i<neffbrs; i++){
          if (i==0) {

     System.out.print(String.format(" %.2f ",
mat[i][j]));
          } else {

     System.out.print(String.format(" %.2f x^%d ",
mat[i][j], i));
          }
               if ((i != (neffbrs-1)) && (mat[i+1][j])
>= 0) {
                    System.out.print("+");
                         }
```

```
                if (i==neffbrs-1) {
                            System.out.println();
                    }
            }

      System.out.print("Apakah ingin memasukkan
input x? (Y/N)\n");
      char inputx = input.next(".").charAt(0);
      if (inputx == 'Y') {
            double hasil = 0;
            double inp;

      System.out.print("Masukkan nilai x: \n");
      inp = scan.nextDouble();
      for (int i=0; i<neffbrs; i++) {
            hasil += (pangkat(inp, i) *
(mat[i][j]));
            }

      System.out.print(String.format("f(x) =
%.2f\n", hasil));

      } else if (inputx == 'N') {
System.out.print("---Interpolasi Polinomial Telah
Selesai---\n");
                    }
            }
                else if (isAdaSolusi==INFsol)
                    {cetakSolusiINF();}
            }
```

void fungsiE () {

```
      Scanner scan = new Scanner(System.in);
      double a, b; //selang
      double x;
      double y;
      int n;

      System.out.println("Menghampiri fungsi dengan
polinom interpolasi pada batas [a,b]");
      System.out.println("Masukkan a: ");
      a = scan.nextDouble();
      System.out.println("Masukkan b: ");
      b = scan.nextDouble();
      System.out.println("Masukkan n: ");
      n = scan.nextInt();

      double h;
      h = (b-a)/((double)n);
      neffbrs = n;
      neffkol = n + 1;
```

```
      for (int i = 0; i < neffbrs; i++) {
            x = a+(h * (double)i);
            y = (Math.exp(-x)) / (1 + Math.pow(x,2)
+ Math.sqrt(x));
            for (int j = 0; j < neffkol-1; j++) {
                mat[i][neffkol-2-j] = pangkat(x, j);
             mat[i][neffkol-1] = y;
             }

      }


      System.out.println("================HASIL====
============");
      gaussJordanBismillah();
      tulisMatriks();
}

void MatriksHilbert () {
      System.out.println("===========MATRIKS
HILBERT============");
      Scanner scan = new Scanner(System.in);
      System.out.print("Masukkan nilai n: ");
      int n =  scan.nextInt();

neffbrs = n;
      neffkol = n+1;

      for (int i = 0; i< neffbrs; i++) {
            for (int j= 0; j< neffkol-1; j++) {
                  mat[i][j] =
((float)1/((float)(i+1)+(j+1)-1));
                        }
            mat[i][neffkol-1] = 1.00;
      }

      System.out.print("Matriks Hilbert: \n");
      tulisMatriks();

      System.out.print("Akan dilakukan perkalian
dengan transpose Matriks B = (1, 1, 1, ..., 1)\n");
      System.out.println("HX=B, akan dihasilkan:
");
      gaussJordanBismillah();
      tulisMatriks();
}
```

Untuk penentuan berapa buah solusi yang dihasilkan serta pencetakan hasil solusi digunakan *method* berikut.

```
void solutionValue(){
      int j = 0;
      int countzerow = 0;

      for (int i=neffbrs-1; i>=0; i--) {
            j=0;
            while ((j<neffkol-1)&&(mat[i][j]==0))
                  j++;
            if (j==neffkol-1){
                  if (mat[i][j] == 0)
                        countzerow++;
                  else{
                        isAdaSolusi = NOsol;
                        break;
                  }
            }
      }
      if (isAdaSolusi != NOsol){
            if ((neffbrs-countzerow) < (neffkol-1))
                  isAdaSolusi = INFsol;
            else
                  neffbrs -= countzerow;
      }
}

boolean sebarisNol(int i){
      int j=0;
      boolean sNol = false;
      while (j<neffkol-1 && mat[i][j]==0){
            j++;
      }
      if (mat[i][j]==0){
            sNol = true;
      }
      return sNol;
}

public void cetakSolusiINF(){
            String currentkata = "";
            int[] unwrittenX = new int[50];
            int neffX = 0;
            int neffhsl = 0;
            int[] ParX= new int[50];
            int neffpar = 0;
            int maxPar = 0;
            int countpar = 0;
            char[] par = 
{'a','b','c','d','e','f','g','h','i','j','k','l','m
','n','o','p','q','r','s','t','u','v'};
            for (int i=0; i<neffbrs; i++) {
```

```
      int kol = firstNonZero(i);
      System.out.print("x"+(kol+1)+" = ");
      if (!((mat[i][neffkol-1]< 0.01) &&
(mat[i][neffkol-1]>9.0E-25))){
      System.out.print(String.format("%.2f",mat[i][
neffkol-1]));
      currentkata +=
String.format("%.2f",mat[i][neffkol-1]);
      }
      for (int j=kol+1; j<neffkol-1; j++){
            if (mat[i][j]!=0){
                  if (mat[i][j]<0){
                        mat[i][j]*=-1;
                        if
(String.format("%.2f",mat[i][j])!="0.00"){

      System.out.print(String.format(" + %.2f",
mat[i][j])+par[j]);
      currentkata += String.format(" + %.2f",
mat[i][j])+par[j];
      neffpar++;
      ParX[neffpar] = j;
            }
      }
      else if (mat[i][j]>0){
                  if
(String.format("%.2f",mat[i][j])!="0.00"){

      System.out.print(String.format(" - %.2f",
mat[i][j])+par[j]);
                              currentkata +=
String.format(" - %.2f", mat[i][j])+par[j];
                              neffpar++;
                              ParX[neffpar] = j;
                        }
                  }
            }
      }
      neffhsl++;
      hasil[neffhsl] = currentkata;
      currentkata = "";
      System.out.println();
      }
      for (int p=1; p<neffkol-1; p++){
            if (firstNonZero(p)!=p){
                  neffX++;
                  unwrittenX[neffX] = p;
      System.out.println("x"+(unwrittenX[neffX]+1
+" = "+par[ParX[neffX]]);
            }
      }
}
```

```
public void hasilSolusi(){
            gaussJordanBismillah();
            solutionValue();

            if (isAdaSolusi==NOsol)
                  System.out.println("Tidak ada 
solusi untuk sistem persamaan diatas.");
            else if (isAdaSolusi==ADAsol){
                  int j = neffkol-1;
                  for (int i=0; i<neffbrs; i++){

      System.out.println(String.format("x%d = 
%.2f",i+1,mat[i][j]));
                        neffhsl++;
                        hasil[neffhsl] = 
String.format("x%d = %.2f",i+1,mat[i][j]);
                  }
            }
            else if (isAdaSolusi==INFsol)
                  cetakSolusiINF();

     }

int firstNonZero(int brs){
            int kol = -1;
            for (int j=0; j<neffkol-1; j++){
                  if (mat[brs][j]!=0){
                        kol = j;
                        break;
                  }
                  else
                        kol = -1;
            }
            return kol;
      }
```

Terakhir adalah program utama untuk eliminasi Gauss-Jordan. Realisasi dari *method* untuk program utamanya adalah sebagai berikut.

```
public static void main(String args[]){
          MatriksAlgeo coba=new MatriksAlgeo();
          System.out.println("-_-_-_-_-_-_-_-_-_-
_-_-_-_-_-_WELCOME TO TUGAS BESAR 1 ALJABAR
GEOMETRI_-_-_-_-_-_-_-_-_-_-_-_-_-");
          System.out.println();
          System.out.println("*****************A
program by**************");
          System.out.println("Haifa Fadhila Ilma|
13516076 |      K01");
```

```
else if (menu == 3){
                      System.out.print(">a/b/c: ");
                      char s =
input.next(".").charAt(0);
                      if (s=='a') {
                      coba.Interpolasi();
                      }else if (s=='b') {
                           coba.MatriksHilbert();
                      }else if (s=='c') {
                           coba.fungsiE();
                           coba.hasilSolusi();
                      }
                 }

                 System.out.println(">Simpan ke
file eksternal? (Y/N)");
                 char simpanfile =
input.next(".").charAt(0);
                 if (simpanfile == 'Y' ||
simpanfile== 'y'){
                      coba.tulisFile();
                 }

                 System.out.println(">Ulangi
program? (Y/N)");
                 ulang = input.next(".").charAt(0);
            }while (ulang == 'Y' || ulang == 'y');
      }
}
```

BAB 4

EKSPERIMEN

a)

```
Matriks Augmentednya adalah
| 0,31 0,14 0,30 0,27 | 1,02 |
| 0,26 0,32 0,18 0,24 | 1,00 |
| 0,61 0,22 0,20 0,31 | 1,34 |
| 0,40 0,34 0,36 0,17 | 1,27 |

x1 = 1,00
x2 = 1,00
x3 = 1,00
x4 = 1,00
x0 = 1,00
x1 = 1,00
x2 = 1,00
x3 = 1,00
Matriks Augmentednya adalah
| 1,00 0,00 0,00 0,00 | 1,00 |
| 0,00 1,00 0,00 0,00 | 1,00 |
| 0,00 0,00 1,00 0,00 | 1,00 |
| 0,00 0,00 0,00 1,00 | 1,00 |
```

b)

```
Matriks Augmentednya adalah
| 1,00 7,00 -2,00 0,00 8,00 | -3,00 |
| 1,00 7,00 -1,00 1,00 0,00 | 2,00 |
| 2,00 14,00 -4,00 1,00 -13,00 | 3,00 |
| 2,00 14,00 -4,00 0,00 16,00 | -6,00 |

x1 = -11,00 - 7,00b - 50,00e
x3 = -4,00 - 21,00e
x4 = 9,00 + 29,00e
x0 = 0,00
x2 = b
x3 = e
x4 = e
x5 = e
Matriks Augmentednya adalah
| 1,00 7,00 0,00 0,00 50,00 | -11,00 |
| 0,00 0,00 1,00 0,00 21,00 | -4,00 |
| 0,00 0,00 0,00 1,00 29,00 | 9,00 |
| 0,00 0,00 0,00 0,00 0,00 | 0,00 |
```

c) Matriks Hilbert

Untuk n= 20:

```
Akan dilakukan perkalian dengan transpose Matriks B = (1, 1, 1, ..., 1)
HX=B, akan dihasilkan:
Matriks Augmentednya adalah
| 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | 9,08 |
| 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | -694,57 |
| 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | 10845,54 |
| 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | -63088,51 |
| 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | 163291,28 |
| 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | -192434,71
| 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | 120897,08 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | -135196,73
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | 122689,56 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | 20309,59 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | -31806,31 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | 109276,28 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | -224279,54
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 | 18753,87 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 | 111190,97 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 | -56496,99 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 | 76784,97 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 | -97148,43 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 | 80681,52 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 | -33481,29 |

x1 = 9,08
x2 = -694,57
x3 = 10845,54
x4 = -63088,51
x5 = 163291,28
x6 = -192434,71
x7 = 120897,08
x8 = -135196,73
x9 = 122689,56
x10 = 20309,59
x11 = -31806,31
x12 = 109276,28
x13 = -224279,54
x14 = 18753,87
x15 = 111190,97
x16 = -56496,99
x17 = 76784,97
x18 = -97148,43
x19 = 80681,52
x20 = -33481,29
```

Untuk n=10 :

```
Matriks Augmentednya adalah
| 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | 12,98 |
| 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | -465,53 |
| 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | 3279,88 |
| 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 | -2394,97 |
| 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 | -37612,70 |
| 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 | 120258,12 |
| 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 | -127468,07 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 | 14098,08 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 | 55729,67 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 | -25389,60 |

x0 = 12,98
x1 = -465,53
x2 = 3279,88
x3 = -2394,97
x4 = -37612,70
x5 = 120258,12
x6 = -127468,07
x7 = 14098,08
x8 = 55729,67
x9 = -25389,60
```

d) Keuntungan Perusahaan

```
Masukkan koefisien x1 : 0,1
Masukkan koefisien x2 : 0,1
Masukkan koefisien x3 : 1
Masukkan konstanta persamaan 1 : 10000
Masukkan persamaan 2 :
Masukkan koefisien x1 : 1
Masukkan koefisien x2 : 0
Masukkan koefisien x3 : 0,05
Masukkan konstanta persamaan 2 : 5000
Masukkan persamaan 3 :
Masukkan koefisien x1 : 0,4
Masukkan koefisien x2 : 1
Masukkan koefisien x3 : 0,4
Masukkan konstanta persamaan 3 : 40000

Matriks Augmentednya adalah
| 0,10 0,10 1,00 | 10000,00 |
| 1,00 0,00 0,05 | 5000,00 |
| 0,40 1,00 0,40 | 40000,00 |

x1 = 4702,19
x2 = 35736,68
x3 = 5956,11
Matriks Augmentednya adalah
| 1,00 0,00 0,00 | 4702,19 |
| 0,00 1,00 0,00 | 35736,68 |
| 0,00 0,00 1,00 | 5956,11 |
```

e) Rangkaian Listrik

File eksternal **matriiix.t**:

matriiix.txt | MatriksAlgeo.java | tesmatrix.txt

```
1 1 1 0 0 0 0 0 0 0 0
0 -1 0 1 -1 0 0 0 0 0 0
0 0 -1 0 0 1 0 0 0 0 0
0 0 0 0 1 -1 0 0 0 0 0
0 0 10 0 0 0 -1 1 0 0 0
0 0 0 0 0 5 0 1 -1 0 0
0 0 0 20 0 0 0 0 0 1 0
5 0 0 0 0 0 1 0 0 0 200
0 0 0 0 15 0 0 0 1 -1 0
0 10 0 0 0 0 1 0 0 -1 0
```

Hasil eksekusi:

MinGW Command Prompt

```
>b. Matriks Hilbert
>c. Fungsi E
>Masukkan pilihan menu: 2
Masukkan nama file yang akan dibaca: matriiix.txt
Jumlah baris adalah 10
Jumlah kolom adalah 11
x1 = 6.67
x2 = -3.33
x3 = -3.33
x4 = -6.67
x5 = -3.33
x6 = -3.33
x7 = 166.67
x8 = 200.00
x9 = 183.33
x10 = 133.33
x1 = 6.67
x2 = -3.33
x3 = -3.33
x4 = -6.67
x5 = -3.33
x6 = -3.33
x7 = 166.67
x8 = 200.00
x9 = 183.33
x10 = 133.33
Matriks Augmentednya adalah
| 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | 6.67 |
| 0.00 1.00 -0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | -3.33 |
| 0.00 0.00 1.00 -0.00 -0.00 0.00 0.00 0.00 0.00 0.00 | -3.33 |
| 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 | -6.67 |
| 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 | -3.33 |
| 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 | -3.33 |
| 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 | 166.67 |
| 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 | 200.00 |
| 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 | 183.33 |
| 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 | 133.33 |

>Simpan ke file eksternal? (Y/N)
y
Masukan nama file yang akan ditulis: tesmatrix.txt
>Ulangi program? (Y/N)
n
```

Hasil pada file eksternal **tesmatrix.txt**:

matriiix.txt MatriksAlgeo.java tesmatrix.txt

```
1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 6.67
0.00 1.00 -0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 -3.33
0.00 0.00 1.00 -0.00 -0.00 0.00 0.00 0.00 0.00 0.00 -3.33
0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 -6.67
0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 -3.33
0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 -3.33
0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 166.67
0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 200.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 183.33
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 133.33

x1 = 6.67
x2 = -3.33
x3 = -3.33
x4 = -6.67
x5 = -3.33
x6 = -3.33
x7 = 166.67
x8 = 200.00
x9 = 183.33
x10 = 133.33
```

f) Interpolasi dengan polinom interpolasi derajat *n* di dalam selang [*a*, *b*]
[0,5] n=5

```
Masukkan koefisien x1 : 0,1
Masukkan koefisien x2 : 0,1
Masukkan koefisien x3 : 1
Masukkan konstanta persamaan 1 : 10000
Masukkan persamaan 2 :
Masukkan koefisien x1 : 1
Masukkan koefisien x2 : 0
Masukkan koefisien x3 : 0,05
Masukkan konstanta persamaan 2 : 5000
Masukkan persamaan 3 :
Masukkan koefisien x1 : 0,4
Masukkan koefisien x2 : 1
Masukkan koefisien x3 : 0,4
Masukkan konstanta persamaan 3 : 40000

Matriks Augmentednya adalah
| 0,10 0,10 1,00 | 10000,00 |
| 1,00 0,00 0,05 | 5000,00 |
| 0,40 1,00 0,40 | 40000,00 |

x1 = 4702,19
x2 = 35736,68
x3 = 5956,11
Matriks Augmentednya adalah
| 1,00 0,00 0,00 | 4702,19 |
| 0,00 1,00 0,00 | 35736,68 |
| 0,00 0,00 1,00 | 5956,11 |
```

[0,5] n=10

```
================HASIL================
Matriks Augmentednya adalah
| 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | -0,00 |
| 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | 0,01 |
| 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | -0,07 |
| 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 | 0,37 |
| 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 | -1,27 |
| 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 | 2,83 |
| 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 | -4,15 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 | 4,07 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 | -2,66 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 | 1,00 |

x1 = -0,00
x2 = 0,01
x3 = -0,07
x4 = 0,37
x5 = -1,27
x6 = 2,83
x7 = -4,15
x8 = 4,07
x9 = -2,66
x10 = 1,00
```

[0,5] n=12

```
================HASIL================
Matriks Augmentednya adalah
| 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | -0,00 |
| 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | 0,00 |
| 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | -0,03 |
| 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | 0,19 |
| 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | -0,87 |
| 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 | 2,67 |
| 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 | -5,69 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 | 8,35 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 | -8,45 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 | 5,92 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 | -2,99 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 | 1,00 |

x1 = -0,00
x2 = 0,00
x3 = -0,03
x4 = 0,19
x5 = -0,87
x6 = 2,67
x7 = -5,69
x8 = 8,35
x9 = -8,45
x10 = 5,92
x11 = -2,99
x12 = 1,00
```

[-2,2] n=5

```
Masukkan a:
2
Masukkan b:
2
Masukkan n:
5
================HASIL================
Matriks Augmentednya adalah
| 1,00 0,50 0,25 0,13 0,06 | 0,00 |
| 0,00 0,00 0,00 0,00 0,00 | 0,00 |
| 0,00 0,00 0,00 0,00 0,00 | 0,00 |
| 0,00 0,00 0,00 0,00 0,00 | 0,00 |
| 0,00 0,00 0,00 0,00 0,00 | 0,00 |

x1 =  - 0,50b - 0,25c - 0,13d - 0,06e
x0 = 0,00
x0 = 0,00
x0 = 0,00
x0 = 0,00
x2 = b
x3 = c
x4 = d
x5 = e
```

[-2,2] n=10

```
================HASIL================
Matriks Augmentednya adalah
| 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | NaN |
| 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | NaN |
| 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | NaN |
| 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 | NaN |
| 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 | NaN |
| 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 | NaN |
| 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 | NaN |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 | NaN |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 | NaN |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 | 1,00 |

x1 = NaN
x2 = NaN
x3 = NaN
x4 = NaN
x5 = NaN
x6 = NaN
x7 = NaN
x8 = NaN
x9 = NaN
x10 = 1,00
```

[-2,2] n= 12

```
================HASIL================
Matriks Augmentednya adalah
| 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | NaN |
| 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | NaN |
| 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | NaN |
| 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | NaN |
| 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | NaN |
| 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 | NaN |
| 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 | NaN |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 | NaN |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 | NaN |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 | NaN |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 | NaN |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 | 1,00 |

x1 = NaN
x2 = NaN
x3 = NaN
x4 = NaN
x5 = NaN
x6 = NaN
x7 = NaN
x8 = NaN
x9 = NaN
x10 = NaN
x11 = NaN
x12 = 1,00
```

g) Polinomial Interpolasi dari pasangan titik-titik

```
Masukkan nilai x0 : 0,1

Masukkan nilai y0 : 0,003

Masukkan nilai x1 : 0,3

Masukkan nilai y1 : 0,067

Masukkan nilai x2 : 0,5

Masukkan nilai y2 : 0,148

Masukkan nilai x3 : 0,7

Masukkan nilai y3 : 0,248

Masukkan nilai x4 : 0,9

Masukkan nilai y4 : 0,370

Masukkan nilai x5 : 1,1

Masukkan nilai y5 : 0,518

Masukkan nilai x6 : 1,3

Masukkan nilai y6 : 0,697

Dengan metode Interpolasi Polinom:
Matriks Augmentednya adalah
| 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 |

Hasil matriks setelah menggunakan metode Gauss Jordan:

Matriks Augmentednya adalah
| 1,00 0,00 0,00 0,00 0,00 0,00 0,00 | -0,02 |
| 0,00 1,00 0,00 0,00 0,00 0,00 0,00 | 0,24 |
| 0,00 0,00 1,00 0,00 0,00 0,00 0,00 | 0,20 |
| 0,00 0,00 0,00 1,00 0,00 0,00 0,00 | 0,00 |
| 0,00 0,00 0,00 0,00 1,00 0,00 0,00 | 0,03 |
| 0,00 0,00 0,00 0,00 0,00 1,00 0,00 | 0,00 |
| 0,00 0,00 0,00 0,00 0,00 0,00 1,00 | -0,00 |

a0 = -0,02
a1 = 0,24
a2 = 0,20
a3 = 0,00
a4 = 0,03
a5 = 0,00
a6 = -0,00
Persamaannya adalah:
f(x) =  -0,02 + 0,24 x^1 + 0,20 x^2 + 0,00 x^3 + 0,03 x^4 + 0,00 x^5  -0,00 x^6
```

x=0,2

```
Persamaannya adalah:
f(x) =  -0,02 + 0,24 x^1 + 0,20 x^2 + 0,00 x^3 + 0,03 x^4 + 0,00 x^5  -0,00 x^6
Apakah ingin memasukkan input x? (Y/N)
Y
Masukkan nilai x:
0,2
f(x) = 0,03
```

x=0,55

```
Persamaannya adalah:
f(x) =  -0,02 + 0,24 x^1 + 0,20 x^2 + 0,00 x^3 + 0,03 x^4 + 0,00 x^5  -0,00 x^6
Apakah ingin memasukkan input x? (Y/N)
Y
Masukkan nilai x:
0,55
f(x) = 0,17
```

x= 0,85

```
Persamaannya adalah:
f(x) =  -0,02 + 0,24 x^1 + 0,20 x^2 + 0,00 x^3 + 0,03 x^4 + 0,00 x^5  -0,00 x^6
Apakah ingin memasukkan input x? (Y/N)
Y
Masukkan nilai x:
0,85
f(x) = 0,34
```

x= 1,28

```
a0 = -0,02
a1 = 0,24
a2 = 0,20
a3 = 0,00
a4 = 0,03
a5 = 0,00
a6 = -0,00
Persamaannya adalah:
f(x) =  -0,02 + 0,24 x^1 + 0,20 x^2 + 0,00 x^3 + 0,03 x^4 + 0,00 x^5  -0,00 x^6
Apakah ingin memasukkan input x? (Y/N)
Y
Masukkan nilai x:
1,28
f(x) = 0,68
```

h) Soal Harga Rumah Baru Per Tahun

```
Matriks Augmentednya adalah
| 1,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 | 4646448533068549000000,00 |
| 0,00 1,00 0,00 0,00 0,00 0,00 0,00 0,00 | -1403920876039834620,00 |
| 0,00 0,00 1,00 0,00 0,00 0,00 0,00 0,00 | 1758286370228289,20 |
| 0,00 0,00 0,00 1,00 0,00 0,00 0,00 0,00 | -1163481659836,77 |
| 0,00 0,00 0,00 0,00 1,00 0,00 0,00 0,00 | 424634357,69 |
| 0,00 0,00 0,00 0,00 0,00 1,00 0,00 0,00 | -78334,34 |
| 0,00 0,00 0,00 0,00 0,00 0,00 1,00 0,00 | 4,55 |
| 0,00 0,00 0,00 0,00 0,00 0,00 0,00 1,00 | 0,00 |

a0 = 4646448533068549000000,00
a1 = -1403920876039834620,00
a2 = 1758286370228289,20
a3 = -1163481659836,77
a4 = 424634357,69
a5 = -78334,34
a6 = 4,55
a7 = 0,00
Persamaannya adalah:
f(x) =  4646448533068549000000,00  -1403920876039834620,00 x^1 + 1758286370228289,20 x^2
-1163481659836,77 x^3 + 424634357,69 x^4  -78334,34 x^5 + 4,55 x^6 + 0,00 x^7
```

Tahun= 1957

```
Persamaannya adalah:
f(x) =  4646448533068549000000,00  -1403920876039834620,00 x^1 + 1758286370228289,20 x^2
-1163481659836,77 x^3 + 424634357,69 x^4  -78334,34 x^5 + 4,55 x^6 + 0,00 x^7
Apakah ingin memasukkan input x? (Y/N)
Y
Masukkan nilai x:
1957
f(x) = 14110720,00
```

i) Viskositas pada suhu T tertentu. Misalkan pada suhu 67 derajat Fahrenheit:

```
Dengan metode Interpolasi Polinom:
Matriks Augmentednya adalah
| 1.00 40.00 1600.00 64000.00 2560000.00 102400000.00 | 1.66 |
| 1.00 50.00 2500.00 125000.00 6250000.00 312500000.00 | 1.41 |
| 1.00 60.00 3600.00 216000.00 12960000.00 777600000.00 | 1.22 |
| 1.00 70.00 4900.00 343000.00 24010000.00 1680700000.00 | 1.06 |
| 1.00 80.00 6400.00 512000.00 40960000.00 3276800000.00 | 0.93 |
| 1.00 90.00 8100.00 729000.00 65610000.00 5904900000.00 | 0.84 |

Hasil matriks setelah menggunakan metode Gauss Jordan:

Matriks Augmentednya adalah
| 1.00 0.00 0.00 0.00 0.00 0.00 | 6.03 |
| 0.00 1.00 0.00 0.00 0.00 0.00 | -0.27 |
| 0.00 0.00 1.00 0.00 0.00 0.00 | 0.01 |
| 0.00 0.00 0.00 1.00 0.00 0.00 | -0.00 |
| 0.00 0.00 0.00 0.00 1.00 0.00 | 0.00 |
| 0.00 0.00 0.00 0.00 0.00 1.00 | -0.00 |

a0 = 6.03
a1 = -0.27
a2 = 0.01
a3 = -0.00
a4 = 0.00
a5 = -0.00
Persamaannya adalah:
f(x) =  6.03  -0.27 x^1 + 0.01 x^2  -0.00 x^3 + 0.00 x^4  -0.00 x^5
Apakah ingin memasukkan input x? (Y/N)
Y
Masukkan nilai x:
62
f(x) = 1.19
```

BAB 5

KESIMPULAN DAN SARAN

Dari seluruh pengerjaan tugas besar mengenai penyelesaian sistem persamaan linier (SPL) secara numerik dalam bahasa pemrograman Java dengan menggunakan metode eliminasi Gauss dan/atau Gauss-Jordan, dapat disimpulkan bahwa sistem persamaan linier dalam bentuk apapun dapat diselesaikan dengan metode eliminasi Gauss/ Gauss-Jordan. Selain itu, dapat disimpulkan pula bahwa penyelesaian dengan metode-metode tersebut juga dapat dibuat dengan sebuah program dengan prosedur dan fungsi tertentu. Perhitungan dengan menggunakan metode Gauss/Gauss-Jordan ini dilakukan dengan berbagai macam operasi dalam program, termasuk pembagian, maka dari itu, kami menggunakan tipe data floating point dan untuk meminumkan galat perhitungan, kami menggunakan pivoting dalam memilih baris yang dijadikan basis dalam operasi baris elementer.

Metode ini juga dapat digunakan untuk menyelesaikan soal-soal perhitungan sehari-hari. Yang perlu dilakukan adalah memodelkan permasalahan tersebut dalam suatu sistem persamaan kemudian membuat augmented matrix-nya. Selain itu, program ini juga dapat menyelesaikan masalah polinom Interpolasi. Interpolasi adalah cara menentukan nilai yang berada di antara dua nilai diketahui berdasarkan suatu fungsi persamaan. Interpolasi linear adalah cara menentukan nilai yang berada di antara dua nilai diketahui berdasarkan persamaan linear (persamaan garis lurus). Karena berupa persamaan, pastinya dapat dijadikan augmented matrix lalu diselesaikan dengan metode Gauss/Gauss-Jordan tersebut.

Jadi, secara umum, penyelesaian masalah yang berhubungan dengan matematika dapat dimudahkan dengan pengaplikasian metode sistem persamaan linear dan juga matriks didalamnya. Metode pemecahan solusi dalam matriks dapat dilakukan dengan cara Gauss/Gauss-Jordan dan metode tersebut dapat dibuat lebih praktis lagi dengan dijadikan prosedur dan fungsi pada kode program.

DAFTAR PUSTAKA

Anton, Howard. 2010. *Elementary Linear Algebra.* Edisi kesepuluh.

https://en.wikipedia.org/wiki/Hilbert_matrix

http://informatika.stei.itb.ac.id/~rinaldi.munir/Buku/Metode%20Numerik/BAb-%2004%20Solusi%20Sistem%20Persamaan%20Lanjar.pdf

http://tarbiyahmatematika.blogspot.com/2016/03/eselon-baris-dan-eseolon-baris-tereduksi.html

https://www.slideshare.net/muhammadkennedy/presentasi-interpolasi-polinomial