# LAPORAN TUGAS BESAR I
# IF2123-ALJABAR GEOMETRI

## "Implementasi Program Java Untuk Menyelesaikan Persoalan Sistem Persamaan Lanjar"


disusun oleh:

**K03-Teknik Informatika 2016**

| | |
|---|---|
| Ricky Kennedy | 13516105 |
| Daniel Ryan Levyson | 13516132 |
| Christian Wibisono | 13516147 |


**INSTITUT TEKNOLOGI BANDUNG**
**BANDUNG**
**2017**

# BAB I

# PENDAHULUAN

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.

Secara umum untuk menyelesaikan sistem persamaan lanjar (SPL) dengan *n* peubah (*variable*) dan *m* persamaan:

$$
\begin{array}{l}
a_{11} x_1 + a_{12} x_2 + .... + a_{1n} x_n = b_1 \\
a_{21} x_1 + a_{22} x_2 + .... + a_{2n} x_n = b_2 \\
\quad : \qquad\qquad\qquad\qquad : \\
\quad : \qquad\qquad\qquad\qquad : \\
a_{m1} x_1 + a_{m2} x_2 + .... + a_{mn} x_n = b_m
\end{array}
$$

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) $0.31x_1 + 0.14x_2 + 0.30x_3 + 0.27x_4 = 1.02$

$0.26x_1 + 0.32x_2 + 0.18x_3 + 0.24x_4 = 1.00$

$0.61x_1 + 0.22x_2 + 0.20x_3 + 0.31x_4 = 1.34$

$0.40x_1 + 0.34x_2 + 0.36x_3 + 0.17x_4 = 1.27$

b) $x_1 + 7x_2 – 2x_3 + 8x_5 = -3$

$x_1 + 7x_2 – x_3 + x_4 = 2$

$2x_1 + 14x_2 – 4x_3 + x_4 – 13x_5 = 3$

$2x_1 + 14x_2 – 4x_3 + 16x_5 = -6$

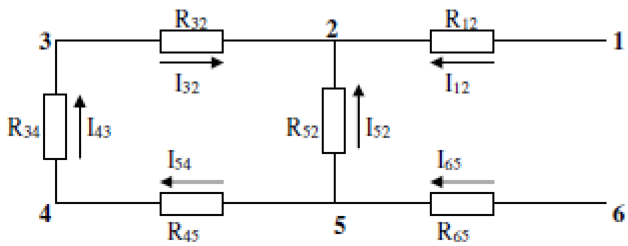
c) $HX = B$, yang dalam hal ini $H$ adalah matriks Hilbert yang memiliki bentuk sebagai berikut:

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

dan $B = (1, 1, 1, \ldots, 1)^T$. 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 :

$I_{12} + I_{52} + I_{32} = 0$

$I_{65} - I_{52} - I_{54} = 0$

$I_{43} - I_{32} = 0$

$I_{54} - I_{43} = 0$

Dari hukum Ohm didapat :

$I_{32}R_{32} - V_3 + V_2 = 0$

$I_{43}R_{43} - V_4 + V_3 = 0$

$I_{65}R_{65} + V_5 = 0$

$I_{12}R_{12} + V_2 = 0$

$I_{54}R_{54} - V_5 + V_4 = 0$

$I_{52}R_{52} - V_5 + V_2 = 0$

Tentukan $I_{12}$ , $I_{52}$ , $I_{32}$ , $I_{65}$ , $I_{54}$ , $I_{13}$ , $V_2$ , $V_3$ , $V_4$ , $V_5$ bila :

$R_{12}$ = 5 ohm, $R_{52}$ = 10 ohm, $R_{32}$ = 10 ohm,

$R_{65}$ = 20 ohm, $R_{54}$ = 15 ohm, $R_{14}$ = 5 ohm,

$V_1$ = 200 volt, $V_6$ = 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 | 75160 |
| 1967 | 76,160 |
| 1968 | 84690 |
| 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 II

# DASAR TEORI

## 2.1. Operasi Baris Elementer

Operasi baris elementer atau OBE adalah operasi-operasi dasar pada baris matriks, yang isinya adalah sebagai berikut:

1. Suatu baris dapat dikurangi dengan k kali baris lainnya
2. Suatu baris dapat dikali/dibagi dengan suatu k
3. Suatu baris dapat ditukarkan dengan baris lainnya

## 2.2.Metode Eliminasi Gauss

Eliminasi Gauss adalah suatu metode untuk mengoperasikan nilai-nilai di dalam matriks sehingga menjadi matriks yang lebih sederhana lagi. Dengan melakukan operasi baris sehingga matriks tersebut menjadi matriks yang baris. Ini dapat digunakan sebagai salah satu metode penyelesaian persamaan linear dengan menggunakan matriks. Caranya dengan mengubah persamaan linear tersebut ke dalam matriks teraugmentasi dan mengoperasikannya. Setelah menjadi matriks baris, lakukan substitusi balik untuk mendapatkan nil dari variabel-variabel tersebut.

Ciri ciri Metode Gauss adalah

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

1. Jika suatu baris tidak semua nol, maka bilangan pertama yang tidak nol adalah 1 (1utama)
2. Baris nol terletak paling bawah.
3. 1 utama baris berikutnya berada dikanan 1 utama baris diatasnya.
4. Dibawah 1 utama harus nol.

Kelebihan dan Kekurangan

Metode ini digunakan dalam analisis numerik untuk meminimalkan mengisi selama eliminasi, dengan beberapa tahap

Keuntungan :

a. Menentukan apakah sistem konsisten.

b. Menghilangkan kebutuhan untuk menulis ulang variabel setiap langka.

c. Lebih mudah untuk memecahkan

Kelemahan :

a. Memiliki masalah akurasi saat pembulatan desimal

Dengan menggunakan metode Eliminasi Gaus dalam menyelesaikan SPL ada tiga kemungkinan bentuk matriks augmented yaitu sebagai berikut:

A. Mempunyai solusi yang unik (tunggal)

Matriks eselon masih terdapat angka non-satu utama pada kolom terakhir di masing-masing baris. Bentuk ini menunjukkan bahwa terdapat solusi unik dari sistem persamaan lanjar. Contoh dari bentuk ini adalah sebagai berikut.

**Solusi unik/tunggal**

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 0 \\ 2 & 3 & 1 & 1 \\ 3 & 1 & 2 & 1 \end{array}\right] \xrightarrow{\text{Eliminasi Gauss}} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 1 & 1 \end{array}\right]$$

Solusi: $x_1 = 1, x_2 = 0, x_3 = -1$

B. Mempunyai tak berhingga banyak solusi

Matriks eselon yang terdapat baris yang isinya semuanya nol. Bentuknya menunjukkan bahwa terdapat banyak solusi dalam sistem persamaan lanjar ini.

**Solusi banyak/tidak terhingga**

$$\left[\begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 2 & -1 & 1 & 2 \\ 1 & 2 & 3 & 6 \end{array}\right] \xrightarrow{\text{Eliminasi Gauss}} \left[\begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

Contoh matriks ini seperti yang terdapat di bawah ini.

Perhatikan hasil eliminasi Gauss pada baris terakhir. Persamaan yang bersesuaian dengan baris terakhir tersebut adalah

$$0x_1 + 0x_2 + 0x_3 = 0$$

yang dipenuhi oleh banyak nilai x. Solusinya diberikan dalam benruk parameter.

Misalkan $x_3 = k$

maka $x_2 = 2 - k$ dan $x_1 = 4 - x_2 - 2x_3 = 4 - (2 - k) - 2k = 2 - k$,

dengan $k$ € R. Terdapat tidak berhingga nilai $k$

C. Tidak ada solusi sama sekali.

Matriks eselon yang pada baris terakhir, satu utama terletak pada kolom terakhir. Bentuk ini menunjukkan bahwa tidak ada solusi dari sistem persamaan lanjar ini. Contoh dari matriks ini seperti yang terdapat di bawah ini.

**Tidak ada solusi**

$$\left[\begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 2 & -1 & 1 & 2 \\ 1 & 2 & 3 & 7 \end{array}\right] \xrightarrow{\text{Eliminasi Gauss}} \left[\begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

Perhatikan hasil eliminasi Gauss pada baris terakhir. Persamaan yang bersesuaian Dengan baris terakhir tersebut adalah

$$0x_1 + 0x_2 + 0x_3 = 0$$

yang dalam hal ini, tidak nilai $x_1$ yang memenuhi, $i$ = 1,2,3

## 2.3. Metode Eliminasi Gauss Jordan

Salah satu metode yang dapat digunakan untuk menyelesaikan sistem persamaan linier adalah metode eliminasi Gauss-Jordan. Metode ini diberi nama Gauss-Jordan untuk menghormati CarlFriedrich Gauss dan Wilhelm Jordan. Metode ini sebenarnya adalah modifikasi dari metode eliminasi Gauss, yang dijelaskan oleh Jordan di tahun 1887.

Metode Gauss-Jordan ini menghasilkan matriks dengan bentuk baris eselon yang tereduksi(reduced row echelon form), sementara eliminasi Gauss hanya menghasilkan matriks sampai padabentuk baris eselon (row echelon form). Selain untuk menyelesaikan sistem persamaan linier, metode eliminasi Gauss-Jordan ini dapat. Metode Eliminasi Gauss : metode yang dikembangkan dari metode eliminasi, yaitu menghilangkanatau mengurangi jumlah variable sehingga dapat diperoleh nilai dari suatu variable yang bebas.

Eliminasi Gauss-Jordan adalah pengembangan dari eliminasi Gauss yang hasilnya lebih sederhana lagi. Caranya adalah dengan meneruskan operasi baris dari eliminasi Gauss sehingga menghasilkan matriks yang Eselon-baris. Ini juga dapat digunakan sebagai salah satu metode penyelesaian persamaan linear dengan menggunakan matriks.

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

Metode ini digunakan untuk mencari invers dari sebuah matriks. Prosedur umum untuk metode eliminasi Gauss-Jordan ini adalah Ubah sistem persamaan linier yang ingin dihitung menjadi matriks augmentasi. Lakukan operasi baris elementer pada matriks augmentasi (A|b) untuk mengubah matriks A menjadi dalam bentuk baris eselon yang tereduksi.

Kelebihan dan Keuntungan :

Mengubah sistem persamaan linier yang ingin dihitung menjadi matriks augmentasi. merupakan variasi dari eliminasi gauss dengan kebutuhan dapat mgenyelesaikan matriks invers.

## 2.4. Tatancang Pemorosan

Untuk menghindari adanya galat (error) dalam perhitungan eliminasi Gauss dan eliminasi Gauss-Jordan, maka dilakukan tatancang pemorosan (pivoting strategy). Tatancang pemorosan sangatlah sederhana, yaitu memindahkan baris yang nilai di belakang satu utamanya terbesar menjadi baris di atas dari baris lainnya sesuai aturan matriks eselon.

## 2.5. Interpolasi

Interpolasi adalah proses pencarian dan perhitungan nilai suatu fungsi yang grafiknya melewati sekumpulan titik yang diberikan. Titik-titik tersebut mungkin merupakan hasil eksperimen dalam sebuah percobaan atau diperoleh dari suatu fungsi yang diketahui.

Interpolasi memiliki beberapa macam seperti interpolasi lanjar, interpolasi kuadratik, interpolasi kubik, dan interpolasi derajat n.

### 2.5.1. Interpolasi lanjar

Interpolasi lanjar adalah interpolasi dua buah titik dengan sebuah garis lurus. Misal diberikan dua buah titik, $(x0,y0)$ dan $(x1,y1)$. Polinom yang menginterpolasi kedua titik itu adalah $p1(x) = a0+a1x$ .

### 2.5.2. Interpolasi kuadratik

Misal diberikan tiga buah titik data, $(x0, y0)$, $(x1, y1)$, dan $(x2, y2)$. Polinom yang menginterpolasi ketiga buah titik itu adalah polinom kuadrat yang berbentuk : $p2(x) = a0 + a1x + a2x2$.

### 2.5.3. Interpolasi Kubik

Misal diberikan empat buah titik data, (*x*0, *y*0), (*x*1, *y*1), (*x*2, *y*2), dan (*x*3, *y*3). Polinom yang menginterpolasi keempat buah titik itu adalah polinom kuadrat yang berbentuk : *p*3(*x*) = *a*0 + *a*1*x* + *a*2*x*2+ *a*3*x*3.

2.5.4. Interpolasi Derajat N

Diberikan N+1 buah titik data, (*x*0, *y*0), (*x*1, *y*1), (*x*2, *y*2), ... , (*x*n, *y*n). Polinom yang akan menginterpolasi ke-(N+1) buah titik tersebut berbentuk polinom derajat-n yaitu : *p*n(*x*) = *a*0 + *a*1*x* + *a*2*x*2+ ... + *a*n*x*n.

# BAB III

# Implementasi Program

Program menghitung solusi Sistem Persamaan Linier secara numerik ini diimplementasikan dalam bahasa Java dengan metode eliminasi Gauss-Jordan. Adapun dalam pembuatan program ini kami yang dibuat adalah kelas bernama GaussJordanMatrix. Kelas GaussJordanMatrix ini memiliki beberapa pustaka, atribut, dan metode yang mendukung berjalannya program ini.

Pustaka

- *import java.util.*;*
- *import java.lang.*;*
- *import java.io.File;*
- *import java.io.PrintWriter;*
- *import java.io.BufferedReader;*
- *import java.io.IOException;*
- *import java.io.LineNumberReader;*
- *import java.io.FileReader;*
- *import java.io.UnsupportedEncodingException;*

Kelas

- *public class GaussJordanMatrix*

Atribut

- *private final int N* adalah atribut global bertipe *integer* yang menyimpan nilai dimensi baris dari matriks.
- *public static int col* adalah atribut public static global bertipe *integer* yang menyimpan nilai dimensi kolom dari matriks
- *private double[][] a* adalah atribut global yang digunakan sebagai Augmented Matriks yang mengalami pemrosesan menjadi Matriks Eselon Tereduksi.

- *private static double[] Result* adalah atribut global yang digunakan untuk mengekstraksi solusi dari sistem persamaan dalam bentuk array. Atribut ini berisi nilai masing-masing parameter hasil metode eliminasi Gauss Jordan.
- *private static Scanner in* adalah atribut global yang digunakan untuk melakukan *scanning* atau pembacaan terhadap input pengguna
- *private static int optMenu* adalah atribut global bertipe *integer[1..7]* yang berguna untuk menampung pilihan menu yang dipilih pengguna

    1 = untuk menyelesaikan sistem persamaan lanjar biasa

    2 = aplikasi SPL pada matriks Hilbert

    3 = aplikasi SPL untuk menghitung pajak perusahaan

    4 = aplikasi SPL untuk menyelesaikan rangkaian listrik

    5 = aplikasi SPL untuk menyelesaikan permasalahan interpolasi

    6 = aplikasi SPL untuk menyelesaikan hampiran fungsi
- *private static int optInput* adalah atribut global yang berguna untuk menyimpan pilihan input dari pengguna;

    1 = masukan menggunakan papan ketik

    2= masukan dari file

Method

- *public GaussJordanMatrix(double[][] A, double[] b):* definisi dan konstruktor kelas untuk mengkonstruksi sebuah matriks GaussJordan. Keluarannya menghasilkan matriks yang sudah merupakan matriks eselon tereduksi.
- *public static void printAugmentedMatrix(double[][] A, double[] b):* mencetak matriks ke layar dalam bentuk matriks augmented.
- *public static void main(String[] args):* program utama yang memanggil semua metode lain yang ada lewat *mainMenu().*
- *public static void mainMenu():* metode ini digunakan untuk menampilkan menu utama yang berisi pilihan menu yang tersedia pada program
- *public static void chosenMenu(int opt):* metode ini memerlukan argumen opt menu untuk menampilkan keluaran pesan berkaitan dengan menu yang dipilih oleh pengguna

- *public static void chooseInput(int optMenu):* metode ini digunakan untuk menampilkan pilihan masukan pengguna dan mengarahkan pengguna ke metode sesuai pilihan jenis masukan yang dikehendaki.
- *public void solve():* metode *solve* digunakan untuk menyelesaikan matriks yang sudah dimasukkan oleh pengguna menjadi matriks eselon tereduksi. Adapun di dalam metode ini dilakukan pemorosan dan operasi baris elementer pada matriks teraugmentasi.
- *public int check():* digunakan untuk melakukan pengecekan jenis solusi dari matriks.
  Metode ini akan mengembalikan nilai 1 apabila solusi persamaannya unik.
  Nilai 2 apabila persamaan tidak memiliki solusi.
  Nilai 3 apabila persamaan memiliki solusi banyak.
- *public void solveManySolutions():* metode yang digunakan untuk melakukan parameterisasi untuk persamaan dengan solusi banyak dan menampilkan hasilnya.
- *public static void test(double[][] A, double[] b):* metode test digunakan untuk menyelesaikan sistem persamaan lanjar dengan membentuk matriks eselon tereduksi dan menampilkannya di layar kemudian menuliskan solusi sesuai dengan jenis solusi dari persamaan tersebut.
- *public double[] primal():* metode untuk mengakses solusi unik dari suatu persamaan lanjar
- *public boolean isFeasible():* metode untuk melakukan pengecekan apakah sistem persamaan memiliki solusi unik
- *private void show()*: metode untuk menampilkan matriks eselon tereduksi
- *public static void matrixKeyboardInput():* metode untuk menyelesaikan persamaan SPL biasa dengan masukan dari papan ketik.
- *public static void matrixFileInput():* metode untuk menyelesaikan persamaan SPL biasa dengan masukan dari file eksternal.
- *public static void interpolationKeyboardInput():* metode untuk menyelesaikan persoalan interpolasi yang masukannya berasal dari papan ketik
- *public static void interpolationFileInput()*: metode untuk menyelesaikan persoalan interpolasi yang masukannya berasal dari file eksternal
- *public static void solveInterpolation(double[][] A, double [] b):* metode yang berguna untuk membangun matriks interpolasi kemudian menampilkan hasilnya pada layar

- *public static void hilbertMatrixFileInput():* metode untuk mengekstrak solusi dari matriks Hilbert dengan masukan dari file eksternal
- *public static void constructHilbertMatrix(int n):* metode yang dipanggil untuk mengkonstruksi matriks Hilber berdasarkan input n;
- *public static void hilbertMatrixKeyboardInput():* metode untuk mengekstrak solusi dari matriks Hilbert dengan masukan dari papan ketik
- *public static void companiesTaxes():* metode untuk menyelesaikan aplikasi SPL untuk menghitung pajak sebuah perusahaan
- *public static void electricalCircuit()*: metode untuk menyelesaikan persamaan rangkaian listrik sesuai contoh kasus uji.
- *public static void interpolationFileFunction():* metode untuk menyelesaikan hampiran persamaan pada contoh kasus uji dengan masukan dari file eksternal.
- *public static void interpolationFunction()*: metode untuk menyelesaikan hampiran persamaan pada contoh kasus uji dengan masukan dari papan ketik.
- *private void exportMatrixToTxt()*: metode untuk menulis dan menyimpan data matriks eselon tereduksi dalam bentuk file txt/csv.

# BAB IV

# EKSPERIMEN

## 1. Kasus 1 Sistem Persamaan Linear (Solusi Unik)

```
> Insert element of matrix :
> 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
> Augmented Matrix :
  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
> Reduced Row Echelon Matrix :
  1.00 0.00 0.00 | 0.08
  0.00 1.00 0.00 | 0.03
  0.00 0.00 1.00 | 0.28
  0.00 0.00 0.00 | 0.16
> Solution to Ax = b
  x1 = 0.08
  x2 = 0.03
  x3 = 0.28
  x4 = 0.16
```

## 2. Kasus 2 Sistem Persamaan Linear (Solusi banyak)

```
> Insert element of matrix :
>  1  7  -2   0   8  -3
>  1  7  -1   1   0   2
>  2 14  -4   1 -13   3
>  2 14  -4   0  16  -6
> Augmented Matrix :
   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

> Reduced Row Echelon Matrix :

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

> System have many solutions

> Let :

> x2 = t

> x5 = u

> x1 = -11.00-7.00t-50.00u

> x2 = t

> x3 = -4.00-21.00u

> x4 = 9.00+29.00u

> x5 =

# 3. Kasus 3 Sistem Persamaan Linier (Tidak Ada Solusi)

> Insert element of matrix :

> 1 1 2 4

> 2 -1 1 2

> 1 2 3 7

> Augmented Matrix :

1.00 1.00 2.00 | 4.00

2.00 -1.00 1.00 | 2.00

1.00 2.00 3.00 | 7.00

> Reduced Row Echelon Matrix :

1.00 0.00 1.00 | 2.20

0.00 1.00 1.00 | 2.40

0.00 0.00 0.00 | -0.60

> System doesn't have any solution

# 4. Kasus 4 Hilbert Matriks

```
> Insert element of matrix :
 > Hx = B, H is a Hilbert matrix
 > Input n : 20
 > Augmented Matrix :
   1.00 0.50 0.33 0.25 0.20 0.17 0.14 0.13 0.11 0.10 0.09 0.08 0.08 0.07 0.07 0.06 0.06 0.06 0.05 0.05 | 1.00
   0.50 0.33 0.25 0.20 0.17 0.14 0.13 0.11 0.10 0.09 0.08 0.08 0.07 0.07 0.06 0.06 0.06 0.05 0.05 0.05 | 1.00
   0.33 0.25 0.20 0.17 0.14 0.13 0.11 0.10 0.09 0.08 0.08 0.07 0.07 0.06 0.06 0.06 0.05 0.05 0.05 0.05 | 1.00
   0.25 0.20 0.17 0.14 0.13 0.11 0.10 0.09 0.08 0.08 0.07 0.07 0.06 0.06 0.06 0.05 0.05 0.05 0.05 0.04 | 1.00
   0.20 0.17 0.14 0.13 0.11 0.10 0.09 0.08 0.08 0.07 0.07 0.06 0.06 0.06 0.05 0.05 0.05 0.05 0.04 0.04 | 1.00
   0.17 0.14 0.13 0.11 0.10 0.09 0.08 0.08 0.07 0.07 0.06 0.06 0.06 0.05 0.05 0.05 0.05 0.04 0.04 0.04 | 1.00
   0.14 0.13 0.11 0.10 0.09 0.08 0.08 0.07 0.07 0.06 0.06 0.06 0.05 0.05 0.05 0.05 0.04 0.04 0.04 0.04 | 1.00
   0.13 0.11 0.10 0.09 0.08 0.08 0.07 0.07 0.06 0.06 0.06 0.05 0.05 0.05 0.05 0.04 0.04 0.04 0.04 0.04 | 1.00
   0.11 0.10 0.09 0.08 0.08 0.07 0.07 0.06 0.06 0.06 0.05 0.05 0.05 0.05 0.04 0.04 0.04 0.04 0.04 0.04 | 1.00
   0.10 0.09 0.08 0.08 0.07 0.07 0.06 0.06 0.06 0.05 0.05 0.05 0.05 0.04 0.04 0.04 0.04 0.04 0.04 0.03 | 1.00
   0.09 0.08 0.08 0.07 0.07 0.06 0.06 0.06 0.05 0.05 0.05 0.05 0.04 0.04 0.04 0.04 0.04 0.04 0.03 0.03 | 1.00
   0.08 0.08 0.07 0.07 0.06 0.06 0.06 0.05 0.05 0.05 0.05 0.04 0.04 0.04 0.04 0.04 0.04 0.03 0.03 0.03 | 1.00
   0.08 0.07 0.07 0.06 0.06 0.06 0.05 0.05 0.05 0.05 0.04 0.04 0.04 0.04 0.04 0.04 0.03 0.03 0.03 0.03 | 1.00
   0.07 0.07 0.06 0.06 0.06 0.05 0.05 0.05 0.05 0.04 0.04 0.04 0.04 0.04 0.04 0.03 0.03 0.03 0.03 0.03 | 1.00
   0.07 0.06 0.06 0.06 0.05 0.05 0.05 0.05 0.04 0.04 0.04 0.04 0.04 0.04 0.03 0.03 0.03 0.03 0.03 0.03 | 1.00
   0.06 0.06 0.06 0.05 0.05 0.05 0.05 0.04 0.04 0.04 0.04 0.04 0.04 0.03 0.03 0.03 0.03 0.03 0.03 0.03 | 1.00
   0.06 0.06 0.05 0.05 0.05 0.05 0.04 0.04 0.04 0.04 0.04 0.04 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 | 1.00
   0.06 0.05 0.05 0.05 0.05 0.04 0.04 0.04 0.04 0.04 0.04 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 | 1.00
   0.05 0.05 0.05 0.05 0.04 0.04 0.04 0.04 0.04 0.04 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 | 1.00
   0.05 0.05 0.05 0.04 0.04 0.04 0.04 0.04 0.04 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 | 1.00
 > Reduced Row Echelon Matrix :
   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 | -2.38
   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 | 118.60
   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 | 5836.31
   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 | -283209.87
   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 | 4294852.46
   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 | -32604188.87
   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 | 140920011.57
   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 | -358603944.92
   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 | 516060225.43
   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 | -354202035.32
   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 | 77417477.71
   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 | -114497532.48
   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 | 36426524.50
   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 | 632494250.63
   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 | -966004654.14
   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 | 491211445.14
   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 | -318442750.69
   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 | 616542374.35
   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 | -510223966.00
   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 1.00 | 139489423.41
```

> Solution to Ax = b

x1 = -2.38

x2 = 118.60

x3 = 5836.31

x4 = -283209.87

x5 = 4294852.46

x6 = -32604188.87

x7 = 140920011.57

x8 = -358603944.92

x9 = 516060225.43

x10 = -354202035.32

x11 = 77417477.71

x12 = -114497532.48

x13 = 36426524.50

x14 = 632494250.63

x15 = -966004654.14

x16 = 491211445.14

x17 = -318442750.69

x18 = 616542374.35

x19 = -510223966.00

x20 = 139489423.41

## 5. Kasus 5 Keuangan Perusahaan

> Input company's revenue : $ 100000

> Augmented Matrix :

2.00 1.00 1.00 | 100000.00

0.00 20.00 1.00 | 100000.00

0.00 0.00 10.00 | 100000.00

> Reduced Row Echelon Matrix :

1.00 0.00 0.00 | 42750.00

0.00 1.00 0.00 | 4500.00

0.00 0.00 1.00 | 10000.00

> Solution to Ax = b

x1 = 42750.00

x2 = 4500.00

x3 = 10000.00

> Federal Taxes = $42750.00

> Regional Taxes = $4500.00

> Corporate Social Responsibility = $10000.00

## 5. Kasus 5 Persamaan Hukum Kirchoff dan Hukum Ohm

> Input the value of the resistor :

> 1. r12 : 5

> 2. r52 : 10

> 3. r32 : 10

> 4. r65 : 20

> 5. r54 : 15

> 6. r43 : 5

> Masukan nilai tegangan :

> 1. v1 : 200

> 2. v6 : 0

> Augmented Matrix :

-0.40 0.10 0.00 0.10 | -40.00

0.10 0.00 0.07 -0.22 | 0.00

0.10 -0.30 0.20 0.00 | 0.00

0.00 0.20 -0.27 0.07 | 0.00

> Reduced Row Echelon Matrix :

1.00 0.00 0.00 0.00 | 169.23

-0.00 1.00 0.00 0.00 | 153.85

-0.00 -0.00 1.00 0.00 | 146.15

-0.00 -0.00 -0.00 1.00 | 123.08

> Solution to Ax = b

x1 = 169.23

x2 = 153.85

x3 = 146.15

x4 = 123.08

> List of Voltages

v1 = 200.00

v2 = 169.23

v3 = 153.85

v4 = 146.15

v5 = 123.08

v6 = 0.00

> List of Current

i12 = 6.15

i32 = -1.54

i52 = -4.62

i43 = -1.54

i54 = -1.54

i65 = -6.15

## 6. Kasus 6 Hampiran Fungsi

> Insert the degree of the curve : 5

> Data #1 :

> X = 0

> f(x) = 1

> Data #2 :

> X = 1

> f(x) = 0.1226264

> Data #3 :

> X = 2

> f(x) = 0.0210992

> Data #4 :

> X = 3

> f(x) = 0.0042436

> Data #5 :

> X = 4

> f(x) = 0.0009639

> Data #6 :

> X = 5

> f(x) = 0.0002386

> Augmented Matrix :

0.00 0.00 0.00 0.00 0.00 1.00 | 1.00

1.00 1.00 1.00 1.00 1.00 1.00 | 0.12

32.00 16.00 8.00 4.00 2.00 1.00 | 0.02

243.00 81.00 27.00 9.00 3.00 1.00 | 0.00

1024.00 256.00 64.00 16.00 4.00 1.00 | 0.00

3125.00 625.00 125.00 25.00 5.00 1.00 | 0.00

> Reduced Row Echelon Matrix :

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 | 0.07

0.00 0.00 1.00 0.00 0.00 0.00 | -0.43

0.00 0.00 0.00 1.00 0.00 0.00 | 1.25

-0.00 -0.00 -0.00 -0.00 1.00 0.00 | -1.76

0.00 0.00 0.00 0.00 0.00 1.00 | 1.00

> Do you want to save it to file? (y/n) :n

> So the polinomial found is:

> f(x) = -0.005x^5+0.073x^4-0.434x^3+1.251x^2-1.763x+1.000

> Insert the degree of the curve : 10

> Data #1 :

> X = 0

> f(x) = 1

> Data #2 :

> X = 0.5

> f(x) = 0.3099118

> Data #3 :

> X = 1.0

> f(x) = 0.1226264

> Data #4 :

> X = 1.5

> f(x) = 0.0498643

> Data #5 :

> X = 2.0

> f(x) = 0.0210992

> Data #6 :

> X = 2.5

> f(x) = 0.0092945

> Data #7 :

> X = 3.0

> f(x) = 0.0042436

> Data #8 :

> X = 3.5

> f(x) = 0.1997071

> Data #9 :

> X = 4.0

> f(x) = 0.0009639

> Data #10 :

> X = 4.5

> f(x) = 0.0004753

> Data #11 :

> X = 5.0

> f(x) = 0.0002386

> Augmented Matrix :

```
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 | 1.00
0.00 0.00 0.00 0.01 0.02 0.03 0.06 0.13 0.25 0.50 1.00 | 0.31
1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 | 0.12
57.67 38.44 25.63 17.09 11.39 7.59 5.06 3.38 2.25 1.50 1.00 | 0.05
1024.00 512.00 256.00 128.00 64.00 32.00 16.00 8.00 4.00 2.00 1.00 | 0.02
9536.74 3814.70 1525.88 610.35 244.14 97.66 39.06 15.63 6.25 2.50 1.00 | 0.01
59049.00 19683.00 6561.00 2187.00 729.00 243.00 81.00 27.00 9.00 3.00 1.00 | 0.00
275854.74 78815.64 22518.75 6433.93 1838.27 525.22 150.06 42.88 12.25 3.50 1.00 | 0.20
1048576.00 262144.00 65536.00 16384.00 4096.00 1024.00 256.00 64.00 16.00 4.00 1.00 | 0.00
3405062.89 756680.64 168151.25 37366.95 8303.77 1845.28 410.06 91.13 20.25 4.50 1.00 | 0.00
9765625.00 1953125.00 390625.00 78125.00 15625.00 3125.00 625.00 125.00 25.00 5.00 1.00 | 0.00
```

> Reduced Row Echelon Matrix :

```
1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | -0.01
0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | 0.16
0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | -1.63
```

0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | 9.29

0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 | -32.42

0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 | 71.34

0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 | -98.02

0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 | 79.80

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 | -33.46

-0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 1.00 0.00 | 4.07

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 | 1.00

> Do you want to save it to file? (y/n) :n

> So the polinomial found is:

> f(x) = -0.007x^10+0.159x^9-1.627x^8+9.286x^7-32.421x^6+71.338x^5-98.019x^4+79.796x^3-33.457x^2+4.073x+1.000

> Insert the degree of the curve : 12

> Data #1 :

> X = 0

> f(x) = 1

> Data #2 :

> X = 0.4167

> f(x) = 0.3623749

> Data #3 :

> X = 0.8333

> f(x) = 0.1666944

> Data #4 :

> X = 1.25

> f(x) = 0.0778432

> Data #5 :

> X = 2.0833

> f(x) = 0.0183560

> Data #6 :

> X = 2.5

> f(x) = 0.0092949

> Data #7 :

> X = 2.9167

> f(x) = 0.0048249

> Data #8 :

> X = 3.3333

> f(x) = 0.0025598

> Data #9 :

> X = 3.75

> f(x) = 0.0013834

> Data #10 :

> X = 4.1667

> f(x) = 0.0007598

> Data #11 :

> X = 4.5833

> f(x) = 0.0002327

> Data #12 :

> X = 5

> f(x) = 0.0002386

> Data #13 :

> X = 1.6667

> f(x) = 0.0372604

> Augmented Matrix :

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 | 1.00

0.00 0.00 0.00 0.00 0.00 0.00 0.01 0.01 0.03 0.07 0.17 0.42 1.00 | 0.36

0.11 0.13 0.16 0.19 0.23 0.28 0.33 0.40 0.48 0.58 0.69 0.83 1.00 | 0.17

14.55 11.64 9.31 7.45 5.96 4.77 3.81 3.05 2.44 1.95 1.56 1.25 1.00 | 0.08

6683.77 3208.26 1539.99 739.21 354.83 170.32 81.75 39.24 18.84 9.04 4.34 2.08 1.00 | 0.02

59604.64 23841.86 9536.74 3814.70 1525.88 610.35 244.14 97.66 39.06 15.63 6.25 2.50 1.00 | 0.01

379054.05 129959.90 44557.17 15276.57 5237.62 1795.74 615.67 211.09 72.37 24.81 8.51 2.92 1.00 | 0.00

1881450.63 564440.83 169333.94 50800.69 15240.36 4572.15 1371.66 411.50 123.45 37.04 11.11 3.33 1.00 | 0.00

7733484.38 2062262.50 549936.67 146649.78 39106.61 10428.43 2780.91 741.58 197.75 52.73 14.06 3.75 1.00 | 0.00

27384624.58 6572257.32 1577329.14 378555.96 90852.70 21804.47 5233.03 1255.92 301.42 72.34 17.36 4.17 1.00 | 0.00

85928933.01 18748267.19 4090560.77 892492.48 194727.05 42486.21 9269.79 2022.51 441.28 96.28 21.01 4.58 1.00 | 0.00

244140625.00 48828125.00 9765625.00 1953125.00 390625.00 78125.00 15625.00 3125.00 625.00 125.00 25.00 5.00 1.00 | 0.00

459.50 275.70 165.41 99.25 59.55 35.73 21.44 12.86 7.72 4.63 2.78 1.67 1.00 | 0.04

> Reduced Row Echelon Matrix :

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 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 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | 0.01

0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | -0.06

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 | 0.36

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 | -1.40

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 | 3.80

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 | -7.30

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 | 9.90

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 | -9.37

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 | 6.23

-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 | -3.04

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 | 1.00

> Do you want to save it to file? (y/n) :n

> So the polinomial found is:

> f(x) = 0.000x^12-0.001x^11+0.008x^10-0.065x^9+0.364x^8-1.401x^7+3.800x^6-7.304x^5+9.896x^4-9.367x^3+6.229x^2-3.036x+1.000

## 7. Kasus 7 Interpolasi Pada Fungsi Polinom

> Insert the degree of the curve : 6

> Data #1 :

> X = 0.1

> f(x) = 0.003

> Data #2 :

> X = 0.3

> f(x) = 0.67

> Data #3 :

> X = 0.5

> f(x) = 0.148

> Data #4 :

> X = 0.7

> f(x) = 0.248

> Data #5 :

> X = 0.9

> f(x) = 0.370

> Data #6 :

> X = 1.1

> f(x) = 0.518

> Data #7 :

> X = 1.3

> f(x) = 0.697

> Augmented Matrix :

```
0.00 0.00 0.00 0.00 0.01 0.10 1.00 | 0.00
0.00 0.00 0.01 0.03 0.09 0.30 1.00 | 0.67
0.02 0.03 0.06 0.13 0.25 0.50 1.00 | 0.15
0.12 0.17 0.24 0.34 0.49 0.70 1.00 | 0.25
0.53 0.59 0.66 0.73 0.81 0.90 1.00 | 0.37
1.77 1.61 1.46 1.33 1.21 1.10 1.00 | 0.52
4.83 3.71 2.86 2.20 1.69 1.30 1.00 | 0.70
```

> Reduced Row Echelon Matrix :

```
1.00 0.00 0.00 0.00 0.00 0.00 0.00 | -78.52
0.00 1.00 0.00 0.00 0.00 0.00 0.00 | 361.17
0.00 0.00 1.00 0.00 0.00 0.00 0.00 | -655.58
0.00 0.00 0.00 1.00 0.00 0.00 0.00 | 592.01
0.00 0.00 0.00 0.00 1.00 0.00 0.00 | -272.72
0.00 0.00 0.00 0.00 0.00 1.00 0.00 | 57.60
0.00 0.00 0.00 0.00 0.00 0.00 1.00 | -3.56
```

> Do you want to save it to file? (y/n) :n

> So the polinomial found is:

> f(x) = -78.516x^6+361.172x^5-655.579x^4+592.008x^3-272.715x^2+57.599x-3.560

> Input X to find the f(x) (enter -999 to stop): 0.2

> f(0.20) = 0.849

> Input X to find the f(x) (enter -999 to stop): 0.55

> f(0.55) = 0.133

> Input X to find the f(x) (enter -999 to stop): 0.85

> f(0.85) = 0.355

> Input X to find the f(x) (enter -999 to stop): 1.28

> f(1.28) = 0.735

## 8. Kasus 8 Interpolasi Harga Rumah

> Insert the degree of the curve : 7

> Data #1 :

> X = 1950

> f(x) = 33.525

> Data #2 :

> X = 1955

> f(x) = 46.519

> Data #3 :

> X = 1960

> f(x) = 53.941

> Data #4 :

> X = 1965

> f(x) = 72.319

> Data #5 :

> X = 1966

> f(x) = 75.160

> Data #6 :

> X = 1967

> f(x) = 76.160

> Data #7 :

> X = 1968

> f(x) = 84.690

> Data #8 :

> X = 1969

> f(x) = 90.866

> Augmented Matrix :

107211723967968740000000.00 54980371265625000000.00 28195062187500000.00 14459006250000.00 7414875000.00 3802500.00 1950.00 1.00 | 33.53

10915090279051655000000000.00 55831663831466260000.00 28558395821721876.00 14607875100625.00 7472058875.00 3822025.00 1955.00 1.00 | 46.52

11112006825558017000000000.00 56693912375296000000.00 28925465497600000.00 14757890560000.00 7529536000.00 3841600.00 1960.00 1.00 | 53.94

11311960578883387000000000.00 57567229409075765000.00 29296299953728124.00 14909058500625.00 7587307125.00 3861225.00 1965.00 1.00 | 72.32

11352319213907911000000000.00 57743230996479710000.00 29370921157924576.00 14939430904336.00 7598896696.00 3865156.00 1966.00 1.00 | 75.16

11392801206854492000000000.00 57919680766926750000.00 29445694340074608.00 14969849689921.00 7610498063.00 3869089.00 1967.00 1.00 | 76.16

11433406871770574000000000.00 58096579632980560000.00 29520619732205568.00 15000314904576.00 7622111232.00 3873024.00 1968.00 1.00 | 84.69

11474136523342886000000000.00 58273928508597690000.00 29595697566580848.00 15030826595521.00 7633736209.00 3876961.00 1969.00 1.00 | 90.87

> Reduced Row Echelon Matrix :

1.00 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 1.00 0.00 0.00 0.00 0.00 0.00 | -2.13

0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 | 5824.91

0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 | -9170424.47

0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 | 8142261153.02

0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 | -3593145494280.70

-0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 1.00 | 522600807113123.40

> Do you want to save it to file? (y/n) :n

> So the polinomial found is:

> f(x) = -0.000x^7+0.000x^6-2.134x^5+5824.907x^4-9170424.470x^3+8142261153.024x^2-3593145494280.703x+522600807113123.400

> Input X to find the f(x) (enter -999 to stop): 1957

> f(1957.00) = -120.625

> Input X to find the f(x) (enter -999 to stop): 1964

> f(1964.00) = -109.625

> Input X to find the f(x) (enter -999 to stop): 1970

> f(1970.00) = -105.625

> Input X to find the f(x) (enter -999 to stop): 1975

> f(1975.00) = -509.625

# 9. Kasus 9 Interpolasi

> Insert the degree of the curve : 5
> Data #1 :
> X = 40
> f(x) = 1.66
> Data #2 :
> X = 50
> f(x) = 1.41
> Data #3 :
> X = 60
> f(x) = 1.22
> Data #4 :
> X = 70
> f(x) = 1.06
> Data #5 :
> X = 80
> f(x) = 0.93
> Data #6 :
> X = 90
> f(x) = 0.84
> Augmented Matrix :
102400000.00 2560000.00 64000.00 1600.00 40.00 1.00 | 1.66
312500000.00 6250000.00 125000.00 2500.00 50.00 1.00 | 1.41
777600000.00 12960000.00 216000.00 3600.00 60.00 1.00 | 1.22
1680700000.00 24010000.00 343000.00 4900.00 70.00 1.00 | 1.06
3276800000.00 40960000.00 512000.00 6400.00 80.00 1.00 | 0.93
5904900000.00 65610000.00 729000.00 8100.00 90.00 1.00 | 0.84
> Reduced Row Echelon Matrix :
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 0.00 0.00 | -0.00
0.00 0.00 0.00 1.00 0.00 0.00 | 0.01
-0.00 -0.00 -0.00 -0.00 1.00 0.00 | -0.27
0.00 0.00 0.00 0.00 0.00 1.00 | 6.03
> Do you want to save it to file? (y/n) :n

> So the polinomial found is:

> f(x) = -0.000x^5+0.000x^4-0.000x^3+0.007x^2-0.268x+6.030

> Input X to find the f(x) (enter -999 to stop): 62

> f(62.00) = 1.186

> Input X to find the f(x) (enter -999 to stop): 75

> f(75.00) = 0.991

# BAB V
# KESIMPULAN DAN SARAN

## 5.1. Kesimpulan

Metode Eliminasi Gauss dan pengembangannya Eliminasi Gauss-Jordan yang dapat menghasilkan matrik Echelon yang dapat memnujukan dari persoalan-persoalan secara sistem persamaan liniear. Dalam kehidupan sehari-hari juga banyak persoalan yang dapat di ubah kedalam persoalan secara matematik seperti penyelesaian hukum kirchoff, permasalahan interpolasi, permasalahan perhitungan pajak perusahaan dan lain lain sehingga Metode Eliminasi Gauss dan Eliminasi Gauss-Jordan yang dibuat dalam program berikut ini sangat bermanfaat untuk membantu perhitungan terhadap masalah-masalah disekitar kita yang dapat diselesaikan.

## 5.2 Saran

Penyelesaian masalah sistem persamaan linier dengan menggunakan metode Eliminasi Gauss-Jordan dapat diajarkan semenjak Sekolah Menengah Atas Karena banyak sekali manfaat dan kegunaan untuk menyelesaikan permasalahan sekitar kita. Selain itu  untuk orang yang ingin menyelesaikan persoalan yang sama disarankan untuk menggunakan tipe data BigDecimal sehingga memiliki tingkat presisi yang lebih tinggi.