

APLIKASI ALJABAR LANJAR PADA METODE NUMERIK

LAPORAN TUGAS I IF2123

Diajukan untuk memenuhi tugas mata kuliah Aljabar Geometri
pada semester I tahun akademik 2017-2018

oleh

Adam Fadhel Ramadhan 13516054

Manasye Shousen Bukti 13516122

Rizky Andyno Ramadhan 13516063



**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2017**

DAFTAR ISI

DAFTAR ISI.....	1
BAB I DESKRIPSI MASALAH.....	2
BAB II TEORI SINGKAT	3
BAB III IMPLEMENTASI PROGRAM	5
BAB IV EKSPERIMEN	21
BAB V KESIMPULAN DAN SARAN	26
REFERENSI	27

BAB I

DESKRIPSI MASALAH

1.1 Deskripsi Umum

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 meminimalkan 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.

BAB II

TEORI SINGKAT

2.1 Metode Eliminasi Gauss

Eliminasi Gauss merupakan salah satu metode untuk mendapatkan solusi dari Sistem Persamaan Linear (SPL). Eliminasi Gauss memanfaatkan matriks augmented yang didapat dari koefisien masing-masing variabel dalam Sistem Persamaan Linear. Contoh :

$$\begin{aligned}2x + y + 4z &= 8 \\3x + 2y + z &= 10 \\x + 3y + 3z &= 8\end{aligned}$$

Maka matriks augmentednya :

$$\left(\begin{array}{cccc}2 & 1 & 4 & 8 \\3 & 2 & 1 & 10 \\1 & 3 & 3 & 8\end{array}\right)$$

Setelah mendapat matriks augmentednya, terapkan Operasi Baris Elementer pada matriks augmented tersebut sehingga menjadi matriks segitiga atas :

$$\left(\begin{array}{cccc}1 & -2 & 1 & 0 \\0 & 1 & -0.25 & 1.25 \\0 & 0 & 1 & 0.538\end{array}\right)$$

Setelah didapat matriks seperti di atas, lakukan *back substitution* untuk mendapatkan solusi dari SPL yang diminta.

2.2 Metode Gauss-Jordan

Metode Gauss-Jordan mirip seperti Metode Eliminasi Gauss. Hanya saja pada metode Gauss-Jordan penerapan Operasi Baris Elementer sampai mendapat matriks identitas :

$$\left(\begin{array}{cccc}1 & 0 & 0 & 2.23 \\0 & 1 & 0 & 1.384 \\0 & 0 & 1 & 0.538\end{array}\right)$$

Setelah didapat matriks identitas tersebut, maka sudah otomatis mendapatkan solusi dari SPL yang diminta.

2.3 Pivoting

Pivoting merupakan metode yang digunakan untuk meminimalkan galat dalam perhitungan OBE tipe data *floating point*. Terdapat 2 jenis pivoting, yaitu partial pivoting dan complete pivoting. Partial pivoting merupakan metode pivoting dimana mengambil nilai maksimum dalam suatu kolom atau baris kemudian menukarnya dengan elemen-elemen baris yang sedang dioperasikan. Complete pivoting merupakan metode pivoting dimana mengambil nilai maksimum sebagai elemen pivoting kemudian mengubah baris dan kolom.

2.4 Interpolasi

Interpolasi adalah proses penghitungan nilai dari suatu fungsi yang grafiknya melewati titik-titik yang diberikan. Titik-titik tersebut dapat berasal dari hasil eksperimen ataupun dari suatu fungsi yang telah diketahui. Interpolasi juga dapat digunakan untuk menaksir nilai tengah dari titik data yang melewati suatu fungsi. Interpolasi memiliki derajat yang dalam hal umumnya digunakan polinom interpolasi. Dari data-data yang diberikan (berupa pasangan nilai x dan y), kita pertama mencari nilai koefisien fungsi yang memenuhi dengan data tersebut. Setelah itu kita dapat menentukan polinom apa yang memenuhi data-data yang diberikan.

BAB III

IMPLEMENTASI PROGRAM

3.1 Class BacaTulisFile

Class BacaTulisFile merupakan class yang berfungsi untuk segala sesuatu yang berhubungan dengan file eksternal seperti membaca file eksternal serta menulis data ke file eksternal. Terdapat beberapa method dalam class BacaTulisFile yang diimplementasikan, diantaranya :

1) tulisFile

```
import java.io.*;
import java.util.*;

public class BacaTulisFile {
    public static int CVar = 0, NPers = 0;
    public static double[][] M = new double[100][100];
    public static String[] listVar = new String[100];

    // Method tulis file
    public static void tulisFile(String text, String namaFile) {
        try {
            PrintWriter out = new PrintWriter(new BufferedWriter(new FileWriter(namaFile, true)));
            out.println(text);
            out.close();
        } catch (IOException e) {
            System.out.println("Gagal menulis isi File!");
            e.printStackTrace();
        }
    }
}
```

Method di atas digunakan untuk menulis file eksternal yang dimiliki dengan input dari program. Variabel *text* sebagai variabel untuk string yang ingin dimasukkan ke dalam file eksternal sedangkan variabel *namaFile* merupakan variabel yang diisi nama file eksternal, dalam hal ini file eksternal yang digunakan ialah persamaan.txt.

2) bacaFile

```
// Method baca file
public static String bacaFile(String namaFile) {
    BufferedReader br = null;
    String stringHasil = "";

    try {
        String sCurrentLine;
        br = new BufferedReader(new FileReader(namaFile));
        while ((sCurrentLine = br.readLine()) != null) {
            stringHasil = stringHasil + sCurrentLine + "\n";
        }
    } catch (IOException e) {
        System.out.println("Gagal membaca File " + namaFile);
        e.printStackTrace();
    } finally {
        try {
            if (br != null) {
                br.close();
            }
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }

    return stringHasil;
}
}
```

Method `bacaFile` di atas digunakan untuk membaca persamaan-persamaan yang terdapat di dalam file eksternal. Mekanismenya ialah user meng-input nama file tujuan yang ingin dibaca. Jika file tersebut tidak ada, maka file tersebut akan gagal dibuka dan akan menampilkan pesan “Gagal membaca File + `namaFile`”. Jika file ada, maka akan membaca seluruh persamaan dalam file yang dituju dan menyalinnya ke variabel *stringHasil*.

3) parsing

```

// Method parsing
public static void parsing(String hasil) {
    String[] temp = hasil.split("\n");
    Double[] tempD = new Double[100];
    String[] tempVar = new String[100];
    Double[] tempHasil = new Double[100];
    Boolean bFound = false;
    int NVar = 0, count;

    //Inisialisasi Variabel
    for(int i=0; i<=99; i++) {
        for(int j=0; j<=99; j++) {
            M[i][j] = 0.0;
        }
    }

    for(int i=0; i<=25; i++) {
        listVar[i] = "-";
    }

    NPers = 0;
    CVar = 0;

    //Pencarian NPers
    for (int i = 0; i < hasil.length(); i++) {
        if (hasil.charAt(i) == '=') {
            NPers++;
        }
    }

    //Menyalin hasil parsing ke dalam matriks
    for (int i = 0; i < NPers; i++) {
        String[] pars = temp[i].split("((?<=[a-z])|(?=[a-z]))");

        //Menghitung jumlah variabel yang berada di satu persamaan
        NVar = 0;
        while (pars[NVar].charAt(0) != '=') {
            NVar++;
        };
        NVar = NVar / 2;

        //Menyalin dari array pars ke array yang sesuai
        for (int j = 0; j < NVar; j++) {
            tempD[j] = Double.parseDouble(pars[j*2].replaceAll("\\+", ""));
            tempVar[j] = pars[j*2+1];
        }

        //Menyalin hasil persamaan di array penampungan sementara
        tempHasil[i] = Double.parseDouble(pars[NVar*2].replaceAll("\\=", ""));

        //Pencarian variabel yang berada di dalam array listVar
        //Apabila variabel ditemukan, maka nilai akan disalin ke indeks kolom matriks bersangkutan
        //Apabila tidak ditemukan, maka variabel baru akan disalin ke dalam array listVar
        for (int j = 0; j < NVar; j++) {
            int k = 0;
            bFound = false;

            while (!bFound && (k < CVar)) {
                if (tempVar[j].charAt(0) == listVar[k].charAt(0)) {bFound = true;};
                k++;
            }

            k--;
            if (bFound) {
                M[i][k] += tempD[j];
            } else {
                listVar[CVar] = tempVar[j];
                M[i][CVar] += tempD[j];
                CVar++;
            }
        }
    }

    for (int i = 0; i < NPers; i++) {
        M[i][CVar] = tempHasil[i];
    }
}

```

Method parsing digunakan untuk mendapatkan nilai-nilai yang diinginkan seperti koefisien variabel serta hasil persamaan. Nilai-nilai tersebut nantinya akan disimpan ke dalam suatu array untuk digunakan kemudian. Method parsing dapat digunakan untuk input dari console maupun dari file eksternal.

4) menuInput


```

// Method menampilkan menu input user
public static void menuInput() {
    String[] equation = new String[1000];
    String namaFile;
    String totalEq = "";
    int NPers;
    int pil;
    Scanner scan = new Scanner(System.in);

    System.out.println("Masukan pilihan input : ");
    System.out.println("1. Konsol");
    System.out.println("2. File Eksternal");
    System.out.print("Pilihan (1/2) : ");
    pil = scan.nextInt();
    scan.nextLine();
    if (pil == 1) {
        System.out.println("Masukan banyak persamaan : ");
        NPers = scan.nextInt();
        scan.nextLine();
        System.out.println("Masukan persamaan : ");
        for (int i=0; i<NPers; i++) {
            equation[i] = scan.nextLine();
            totalEq = totalEq + equation[i] + "\n";
        }
        BacaTulisFile.parsing(totalEq);
    } else if (pil == 2) {
        System.out.print("Masukan nama file : ");
        namaFile = scan.nextLine();
        String hasil = BacaTulisFile.bacaFile(namaFile);
        System.out.println("Persamaan dalam File : ");
        System.out.println(hasil);
        BacaTulisFile.parsing(hasil);
    } else {
        System.out.println("Input harus antara 1 atau 2!");
    }
}

```

Method menuInput digunakan untuk menampilkan pilihan apakah user ingin melakukan input data dari console atau dari file eksternal. Di dalam method ini juga dipanggil method bacaFile serta method parsing sehingga nantinya akan mendapatkan data-data matriks yang diinginkan.

3.2 File-file eksternal

Terdapat beberapa file eksternal yang digunakan dalam program ini, yaitu persamaan.txt untuk menyimpan data persamaan yang ingin dicari solusinya serta solusi.txt yang berisi solusi dari persamaan yang telah diselesaikan.

- 1) Persamaan.txt

```

1 -3.2x+2y+8z+5y=15
2 4.0x+7y-5z=17
3 1x+1y+2z=12
4 1x+2z=5

```

Tampilan persamaan.txt hanya berupa sistem persamaan linear yang ingin diselesaikan, perlu diperhatikan bahwa untuk variabel berkoefisien 1, harus ditulis angka 1 nya di depan variabelnya.

2) Solusi.txt

```

1 FILE FILE SOLUSI
2 a[0]:1.0
3 a[1]:-2.705659945667147
4 a[2]:4.315171775049225
5 a[3]:-4.706288495483743
6 a[4]:3.5089903784726517
7 a[5]:-1.779089674782424
8 a[6]:0.6098479128297174
9 a[7]:-0.13880777561284768
10 a[8]:0.020068221299530096
11 a[9]:-0.0016663778675825068
12 a[10]:6.04621529445226E-5
13 f(x) = 1.0 - 2.705659945667147x^1 + 4.315171775049225x^2 - 4.706288495483743x^3 + 3.5089903784726517x^4 - 1.779089674782424x^5 + 0.6098479128297174x^6 - 0.13880777561284768x^7 + 0.020068221299530096x^8 - 0.0016663778675825068x^9 + 6.04621529445226E-5x^10
14 Hasil nya adalah: 0.004243679913920939
15

```

Isi dari file solusi.txt berupa solusi dari sistem persamaan linear yang telah diselesaikan.

3.3 Fungsi-fungsi Getter

```

public static int GetBaris(){
    return NPers;
}

public static int GetKolom(){
    return CVar;
}

public static double[][] GetMatriks() {
    return M;
}

public static String GetListVar(int i) {
    return listVar[i];
}
}

```

Fungsi-fungsi di atas digunakan untuk mendapatkan variabel-variabel yang diinginkan dari class bacaFile.

3.3 Class algo

Class algo adalah class utama program yang dibuat di mana berisi program utama serta beberapa method yang digunakan. Berikut adalah penjelasan class yang dibuat :

```

import java.util.*;
import java.lang.*;

public class algeo {

    // Deklarasi Nilai EPSILON Untuk Mengatasi Galat Floating Point
    private static final double EPSILON = 3.1401849173675503e-16;

    // Prosedur Mencetak Matriks
    private static void printmatriks(double[][] A, double[] b, int m, int n){
        for (int i = 0; i < m; i++) {
            for(int j=0; j < n; j++) {
                // Mengatasi A[i][j]=-0.00
                if(Math.abs(A[i][j])<=EPSILON){
                    A[i][j]=0;
                }
                System.out.printf("%.2f",A[i][j]);
                System.out.printf(" ");
            }
            System.out.printf("%.2f",b[i]);
            System.out.println();
        }
    }
}

```

Di dalam class ini, diberikan perintah untuk mengimport *java collection* yang ingin dipakai yaitu *java.util* dan *java.lang* . Di dalam *class algeo*, dibuat constanta *EPSILON* agar dapat mengatasi galat *floating point* terhadap 0.00 dan dibuat juga prosedur tambahan *printmatriks* untuk mempermudah pencetakan matriks.

```

// Mengecek Apakah Baris Mengandung 0 Semua
private static boolean isZero(double[][] A, double[] b, int i){
    boolean cek;
    cek=true;
    int n=A[0].length;
    for(int j=0;j<n;j++){
        if(A[i][j]!=0){
            cek=false;
        }
        if(b[i]!=0){
            cek=false;
        }
    }
    return cek;
}

// Mengecek Apakah Baris Mengandung 0 Semua Dan Solusi Bukan 0
private static boolean isZeroSatu(double[][] A, double[] b, int i){
    boolean cek2 = true;
    boolean cek1 = true;
    int n=A[0].length;
    for(int j=0;j<n;j++){
        if(A[i][j]!=0){
            cek1=false;
        }
        if(b[i]==0){
            cek2=false;
        }
    }
    return (cek1&&cek2);
}

```

Di dalam *class algeo* dibuat 2 fungsi yang mengembalikan tipe *boolean*. Fungsi *isZero* mengecek apakah dalam baris *i*, elemen variabel nya 0 semua dan solusinya juga 0. Fungsi *isZeroSatu* mengecek apakah dalam baris *i*, elemen variabel nya 0 semua dan solusinya bukan 0.

```
// Eliminasi Gauss Dengan Pivoting
private static void solvegauss(double[][] A, double[] b, int m, int n) {
    for (int p = 0; p < n; p++) {
        // Mencari Maks Per Kolom Dan Menukarnya
        int max = p;

        for (int i = p + 1; i < m; i++)
            if (Math.abs(A[i][p]) > Math.abs(A[max][p])) {
                max = i;
            }

        //Menukar Variabel Max Ke Atas
        double temp = A[p];
        A[p] = A[max];
        A[max] = temp;

        //Menukar Solusi Max Ke Atas
        double t = b[p];
        b[p] = b[max];
        b[max] = t;

        // OBE Per Kolom
        if (A[p][p] != 0) {
            for (int i = p + 1; i < m; i++) {
                double factor = A[i][p] / A[p][p];
                b[i] -= factor * b[p];
                for (int j = p; j < n; j++) {
                    A[i][j] -= factor * A[p][j];
                }
            }
        }
    }
}
```

Di dalam *class* *algeo*, dibuat juga prosedur *solvegauss* yang melakukan OBE secara gauss. Awalnya dimulai dengan pivoting dan saat ingin di OBE dicek juga apakah elemen bukan 0 nya ada tepat di p. Jika tidak, maka akan dilewatkan OBE nya dan akan masuk ke parametris. Jika iya, maka akan di OBE seperti biasa.

```
for (int i = 0; i < m; i++) {
    if (!isZero(A, b, i) && (!isZeroSatu(A, b, i))) {
        int temp = 0;
        for (int j = 0; j < n; j++) {
            if (!(Math.abs(A[i][j]) <= EPSILON)) {
                temp = j;
                break;
            }
        }
        double factor = A[i][temp];
        b[i] = b[i] / factor;
        for (int j = 0; j < n; j++) {
            A[i][j] = A[i][j] / factor;
            if ((Math.abs(A[i][j]) <= EPSILON)) {
                A[i][j] = 0;
            }
        }
    }
}

System.out.println("\nHasil OBE Gauss :");
printmatriks(A, b, m, n);
```

Setelah proses sebelumnya selesai, kita mendapati matriks segitiga atas, namun koefisien 1 utama bukan 1, sehingga perlu dibagi dengan koefisien 1 utama nya/. Setelah itu dicetak hasil OBE Gauss nya.

```
// Eliminasi Gauss-Jordan Dengan Pivoting
private static void solvegaussjordan(double[][] A, double[] b,int m,int n) {

    for (int p = 0; p < n; p++) {

        // Mencari Maks Per Kolom Dan Menukarnya
        int max = p;

        for (int i = p + 1; i < m; i++)
            if (Math.abs(A[i][p]) > Math.abs(A[max][p])) {
                max = i;
            }

        //Menukar Variabel Max Ke Atas
        double[] temp = A[p];
        A[p] = A[max];
        A[max] = temp;

        //Menukar Solusi Max Ke Atas
        double t = b[p];
        b[p] = b[max];
        b[max] = t;

        // OBE Per Kolom
        if(A[p][p]!=0) {
            for (int i = p + 1; i < m; i++) {
                double factor = A[i][p] / A[p][p];
                b[i] -= factor * b[p];
                for (int j = p; j < n; j++) {
                    A[i][j] -= factor * A[p][j];
                }
            }
        }
    }
}
```

Di dalam *class* *algeo*,dibuat juga prosedur *solvegaussjordan* yang melakukan OBE secara gauss.Awalnya dimulai dengan *pivoting* dan saat ingin di OBE dicek juga apakah elemen bukan 0 nya ada tepat di p.Jika tidak ,maka akan dilewatkan OBE nya dan akan masuk ke parametris.Jika iya, maka akan di OBE seperti biasa.

```
for(int i=0;i<m;i++){
    if(!isZero(A,b,i)&&!isZeroSatu(A,b,i)) {
        int temp = 0;
        for (int j = 0; j < n; j++) {
            if (!Math.abs(A[i][j]) <= EPSILON) {
                temp = j;
                break;
            }
        }
        double factor = A[i][temp];
        b[i] = b[i] / factor;
        for (int j = 0; j < n; j++) {
            A[i][j] = A[i][j] / factor;
            if ((Math.abs(A[i][j]) <= EPSILON)) {
                A[i][j] = 0;
            }
        }
    }
}

for(int i=0;i<m;i++){
    if(!isZero(A,b,i)&&!isZeroSatu(A,b,i)) {
        int temp=cariSatu(A,i);
        for (int j = 0; j < i; j++) {
            double factor = A[j][temp] / A[i][temp];
            for (int k = 0; k < n; k++) {
                A[j][k] -= (A[i][k] * factor);
            }
            b[j] -= (b[i] * factor);
        }
    }
}
System.out.println("\nHasil OBE Gauss-Jordan:");
printmatriks(A,b,m,n);
```

Setelah proses sebelumnya selesai,kita mendapati matriks segitiga atas,namun koefisien 1 utama bukan 1,sehingga perlu dibagi dengan koefisien 1 utama nya/ .Setelah didapat matriks yang sama dengan metode Gauss,dilakukan tahap terakhir untuk menghilangkan angka yang bukan 0 selain pada diagonal utama.Setelah itu dicetak hasil OBE Gauss-Jordan nya.

```

// Mencari Solusi Simpan Dalam String
private static String[] solution(double[][] A, double[] b, int m, int n){
    double[] x = new double[m];
    String[] y = new String[m];
    for (int i = m - 1; i >= 0; i--) {
        double sum = 0.0;
        for (int j = i + 1; j < n; j++) {
            sum += A[i][j] * x[j];
        }
        x[i] = (b[i] - sum) / A[i][i];
        if (Math.abs(x[i]) <= EPSILON) {
            x[i] = 0;
        }
        y[i] = Double.toString(x[i]);
    }
    return y;
}

// Mencari Solusi Simpan Dalam Integer
private static double[] solutionint(double[][] A, double[] b, int m, int n){
    double[] x = new double[m];
    for (int i = m - 1; i >= 0; i--) {
        double sum = 0.0;
        for (int j = i + 1; j < n; j++) {
            sum += A[i][j] * x[j];
        }
        x[i] = (b[i] - sum) / A[i][i];
        if (Math.abs(x[i]) <= EPSILON) {
            x[i] = 0;
        }
    }
    return x;
}

```

Di dalam *class* *algeo*, juga dibuat 2 fungsi yang mengembalikan nilai variabel. Yang membedakan 2 fungsi tersebut hanyalah tipe data yang dipakai untuk menyimpan solusi. Fungsi pertama mengembalikan *array of string*, sementara fungsi kedua mengembalikan *array of double*.

```

// Mencari Solusi Yang Berbentuk Parametris
private static String[] solutionparametris(double[][] A, double[] b, int m, int n) {
    double[] x = new double[n];
    String[] y = new String[n];
    for (int i = 0; i < n; i++) {
        String s = String.valueOf(i+1);
        y[i] = "x"+s;
    }
    for (int i = m - 1; i >= 0; i--) {
        if (!isZero(A, b, i)) {
            if (sendiriSatu(A, i)) {
                x[cariSatu(A, i)] = b[i];
                y[cariSatu(A, i)] = Double.toString(x[cariSatu(A, i)]);
            } else {
                y[(cariSatu(A, i))] = (String.valueOf(b[i]));
                for (int j = 0; j < n; j++) {
                    if (j != cariSatu(A, i)) {
                        if (A[i][j] != 0) {
                            String s = (String.valueOf(A[i][j])).concat(" * ").concat(y[j]);
                            y[(cariSatu(A, i))] = y[(cariSatu(A, i))] + "-" + s + " + ";
                        }
                    }
                }
            }
        }
    }
    return y;
}

```

Di dalam *class* *algeo*, juga dibuat fungsi tambahan yaitu *solutionparametris* untuk mencari solusi parametris saat matriks yang dicari memiliki tak hingga solusi.

```
// Mencari 1 Utama Dalam Satu Baris
private static int cariSatu(double [][] A,int i){
    int x=-999;
    for (int j=0;j<A[0].length;j++){
        if(A[i][j]==1){
            x=j;
            break;
        }
    }
    return x;
}

// Mengecek Apakah Di Sebuah Matriks Hanya Ada 1 Sisanya 0
private static boolean sendiriSatu(double [][] A,int i){
    boolean cek=true;
    for (int j=0;j<A[0].length;j++){
        if(j!=cariSatu(A,i)) {
            if (A[i][j] != 0) {
                cek = false;
            }
        }
    }
    return cek;
}

// Menghitung Hasil Dari Suatu Fungsi
private static double fungsi(double x){
    return (Math.exp(-x)/(1+Math.sqrt(x)+Math.pow(x,2)));
}
}
```

Di dalam *class* *algeo*,dibuat fungsi *cariSatu* untuk mencari 1 utama dalam baris tertentu.Di dalam *class* *algeo*,dibuat juga fungsi *sendiriSatu* yang mengembalikan boolean yang mengecek apakah di sebuah matriks hanya ada 1 utama dan sisanya 0.Di dalam *class* *algeo*,dibuat juga fungsi *fungsi* untuk menghasilkan nilai dari fungsi yang akan dipakai di interpolasi fungsi.

```
// Program Utama
public static void main(String[] args) {
    int n, m, ans;
    double[][] A;
    double[] b;
    double[][] Matriks;
    String tampil = "";
    String nameFile = "";
    int pillTampil;

    Scanner scan = new Scanner(System.in);
    System.out.print("=====MENU UTAMA=====\\n");
    System.out.print("1. Pencarian Solusi Persamaan Dengan Metode Gauss      \\n");
    System.out.print("2. Pencarian Solusi Persamaan Dengan Metode Gauss-Jordan\\n");
    System.out.print("3. Pencarian Solusi Matriks Hilbert      \\n");
    System.out.print("4. Penaksiran Nilai Dengan Polinom Interpolasi      \\n");
    System.out.print("5. Penaksiran Nilai Fungsi Dengan Polinom Interpolasi      \\n");
    System.out.print("=====\\n");
    System.out.print("Pilih Menu Berdasarkan Angka<1-5> :");
    ans = scan.nextInt();

    if ((ans == 1) || (ans == 2)) {
        BacaTulisFile.menuInput();
        m = BacaTulisFile.GetBaris();
        n = BacaTulisFile.GetKolom();
    }
}
```

Dalam program utama,dibuat menu-menu utama program.Jika pilihan menu 1 atau 2,maka program inputan matriks ada di file *BacaTulisFile*.

```
// Agar Mudah Dibuat Jadi Matriks Persegi Dengan Paksa
if (m > n) {
    A = new double[m][m];
    b = new double[m];
} else {
    A = new double[n][n];
    b = new double[n];
}

Matriks = BacaTulisFile.GetMatriks();

// Menerima Inputan Matriks
for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++) {
        A[i][j] = Matriks[i][j];
    }
}

// Menerima Inputan Solusi Matriks
for (int i = 0; i < m; i++) {
    b[i] = Matriks[i][n];
}

System.out.println("Matriks tersebut adalah: ");
printmatriks(A, b, m, n);
```

Jika jumlah kolom lebih besar dari baris atau sebaliknya, dipilih nilai yang terbesar dan dibuat matriks perseginya untuk mempermudah OBE. Inputan user berupa persamaan seperti $3x+2y=0$. Maka $A[0][0]$ akan diisi dengan nilai 3, $A[0][1]$ akan diisi dengan nilai 2, dan $b[0]$ diisi dengan nilai 0.

```
// Matriks Yang Dapat Disolve Harus Matriks Bujursangkar
if (m == n) {
    if (ans == 1) {
        System.out.println("Tampilkan solusi persamaan ke : ");
        System.out.println("1. Layar saja");
        System.out.println("2. Layar dan File");
        System.out.print("Pilih : ");
        pilTampil = scan.nextInt();
        scan.nextLine();
        if (pilTampil == 2) {
            System.out.print("Masukan tujuan file : ");
            nameFile = scan.nextLine();
            System.out.println("Solusi sukses ditulis di file " + nameFile);
        }
        solvegauss(A, b, m, n);
        if (isMatriksZero(A, b)) {
            String [] x = solutionparametris(A, b, m, n);
            for (int i = 0; i < n; i++) {
                tampil += "\nSolusi Variabel " + BacaTulisFile.GetListVar(i) + " adalah " + x[i];
                System.out.println("\nSolusi Variabel " + BacaTulisFile.GetListVar(i) + " adalah " + x[i]);
            }
            if (pilTampil == 2) {
                BacaTulisFile.tulisFile(tampil, nameFile);
            }
        } else if (isMatriksZeroSatu(A, b)) {
            System.out.println("\nMatriks tidak memiliki solusi.");
            if (pilTampil == 2) {
                BacaTulisFile.tulisFile("Matriks tidak memiliki solusi", nameFile);
            }
        }
    }
}
```

Program diatas sampai akhir menu 1 dan 2 berisi tentang ingin ditampilkan dimana solusi persamaan tersebut. Jika pilTampil sama dengan 1 maka akan ditampilkan di layar saja, sementara jika pilTampil sama dengan 2 maka akan ditampilkan di layar sekaligus di tulis ke dalam file.


```

    }else{
        System.out.println("Tampilkan solusi persamaan ke : ");
        System.out.println("1. Layar saja");
        System.out.println("2. Layar dan File");
        System.out.print("Pilih : ");
        pillTampil = scan.nextInt();
        scan.nextLine();
        if (pillTampil == 2) {
            System.out.print("Masukan tujuan file : ");
            nameFile = scan.nextLine();
            System.out.println("Solusi sukses ditulis di file " + nameFile);
        }
        solvegaussjordan(A, b, m, n);
        if(isMatriksZero(A,b)){
            String [] x=solutionparametris(A,b,m,n);
            for (int i = 0; i < n; i++) {
                tampil += "\nSolusi Variabel " + BacaTulisFile.GetListVar(i) + " adalah " + x[i];
                System.out.println("\nSolusi Variabel " + BacaTulisFile.GetListVar(i) + " adalah " + x[i]);
            }
            if (pillTampil == 2) {
                BacaTulisFile.tulisFile(tampil,nameFile);
            }
        }else if(isMatriksZeroSatu(A,b)){
            System.out.println("\nMatriks tidak memiliki solusi.");
            if (pillTampil == 2) {
                BacaTulisFile.tulisFile("Matriks tidak memiliki solusi",nameFile);
            }
        }else {
            String [] x = solution(A, b, m, n);
            for (int i = 0; i < n; i++) {
                tampil += "\nSolusi Variabel " + BacaTulisFile.GetListVar(i) + " adalah " + x[i];
                System.out.println("\nSolusi Variabel " + BacaTulisFile.GetListVar(i) + " adalah " + x[i]);
            }
            if (pillTampil == 2) {
                BacaTulisFile.tulisFile(tampil,nameFile);
            }
        }
    }
}

// Singular Atau Tak Hingga Solusi
else {
    if(ans==1){
        System.out.println("Tampilkan solusi persamaan ke : ");
        System.out.println("1. Layar saja");
        System.out.println("2. Layar dan File");
        System.out.print("Pilih : ");
        pillTampil = scan.nextInt();
        scan.nextLine();
        if (pillTampil == 2) {
            System.out.print("Masukan tujuan file : ");
            nameFile = scan.nextLine();
            System.out.println("Solusi sukses ditulis di file " + nameFile);
        }
        if(isMatriksZero(A,b)) {
            solvegauss(A, b, m, n);
            if(isMatriksZeroSatu(A,b)){
                System.out.println("\nMatriks tidak memiliki solusi.");
                if (pillTampil == 2) {
                    BacaTulisFile.tulisFile("Matriks tidak memiliki solusi",nameFile);
                }
            }else {
                String[] x = solutionparametris(A, b, m, n);
                for (int i = 0; i < n; i++) {
                    tampil += "\nSolusi Variabel " + BacaTulisFile.GetListVar(i) + " adalah " + x[i];
                    System.out.println("\nSolusi Variabel " + BacaTulisFile.GetListVar(i) + " adalah " + x[i]);
                }
                if (pillTampil == 2) {
                    BacaTulisFile.tulisFile(tampil,nameFile);
                }
            }
        }
    }else {
        solvegauss(A, b, m, n);
        if(isMatriksZeroSatu(A,b)){
            System.out.println("\nMatriks tidak memiliki solusi.");
            if (pillTampil == 2) {
                BacaTulisFile.tulisFile("Matriks tidak memiliki solusi",nameFile);
            }
        }
    }
}

```



```

System.out.println("\nMatriks tersebut adalah: ");
printmatriks(A,b,n,n);

System.out.println("\nIngin Menyelesaikan dengan Metode: ");
System.out.println("1.Metode Gauss ");
System.out.println("2.Metode Gauss-Jordan ");
System.out.printf("Pilihan Anda Adalah: ");
int pil;
pil=scan.nextInt();
scan.nextLine();/*---*/
System.out.println("Tampilkan solusi persamaan ke : ");
System.out.println("1. Layar saja");
System.out.println("2. Layar dan File");
System.out.print("Pilih : ");
pilTampil = scan.nextInt();
scan.nextLine();
if (pilTampil == 2) {
    System.out.print("Masukan tujuan file : ");
    nameFile = scan.nextLine();
    System.out.println("Solusi sukses ditulis di file " + nameFile);
}
if(pil == 1){

```

Program diatas sampai akhir menu 3 berisi tentang ingin ditampilkan dimana solusi persamaan tersebut.Jika pilTampil sama dengan 1 maka akan ditampilkan di layar saja,sementara jika pilTampil sama dengan 2 maka akan ditampilkan di layar sekaligus di tulis ke dalam file.

```

    if(pil == 1){
        solvegauss(A, b, n, n);
        String[] x = solution(A, b, n, n);
        for (int i = 0; i < n; i++) {
            tampil += "\nSolusi Variabel ke " + "(" + (i + 1) + ")" + " adalah " + x[i];
            System.out.println("\nSolusi Variabel ke " + "(" + (i + 1) + ")" + " adalah " + x[i]);
        }
        if (pilTampil == 2) {
            BacaTulisFile.tulisFile(tampil,nameFile);
        }
    }else{
        solvegaussjordan(A, b, n, n);
        String[] x = solution(A, b, n, n);
        for (int i = 0; i < n; i++) {
            tampil += "\nSolusi Variabel ke " + "(" + (i + 1) + ")" + " adalah " + x[i];
            System.out.println("\nSolusi Variabel ke " + "(" + (i + 1) + ")" + " adalah " + x[i]);
        }
        if (pilTampil == 2) {
            BacaTulisFile.tulisFile(tampil,nameFile);
        }
    }
}
// Interpolasi Dengan Data Inputan Dari User
else if(ans==4){
    System.out.print("Masukkan jumlah data yang ingin dimasukkan: ");
    n=scan.nextInt();
    A = new double[n][n];
    b = new double[n];
    double[] c = new double[n];
    double[] a = new double[n];
    for (int i=0;i<n;i++){
        System.out.println("Masukkan nilai x["+i+"]");
        c[i]=scan.nextDouble();
        System.out.println("Masukkan nilai y["+i+"]");
        b[i]=scan.nextDouble();
    }
    for (int i=0;i<n;i++){
        for (int j=0;j<n;j++){
            if((i==0)&&(j==0)){
                A[i][j]=1;
            }else {
                A[i][j] = Math.pow(c[i], j);
            }
        }
    }
}
}

```

Jika pilihan menu sama dengan 4,maka akan masuk ke interpolasi yang mengambil inputan dari user.Dibuat matriks A nya berdasarkan inputan *user* dan solusi persamaan tersebut yaitu a_0, a_1, \dots disimpan dalam array $a[i]$.Sisa program diatas sampai akhir menu 4 berisi tentang ingin ditampilkan dimana solusi persamaan tersebut.Jika pilTampil sama dengan 1 maka akan ditampilkan di layar saja,sementara jika pilTampil sama dengan 2 maka akan ditampilkan di layar sekaligus di tulis ke dalam file.Dalam program juga akan dicetak $f(x)$ yang memuat fungsi interpolasi.Pengguna juga dapat memasukkan nilai x dan melihat hasil hampirannya di layar.

```

solvegauss(A,b,n,n);
a=solutionint(A,b,n,n);
System.out.println();
System.out.println("Tampilkan solusi persamaan ke : ");
System.out.println("1. Layar saja");
System.out.println("2. Layar dan File");
System.out.print("Pilih : ");
pillTampil = scan.nextInt();
scan.nextLine();
if (pillTampil == 2) {
    System.out.print("Masukan tujuan file : ");
    nameFile = scan.nextLine();
    System.out.println("Solusi sukses ditulis di file " + nameFile);
}
for(int i=0;i<n;i++){
    tampil += "a["+i+"]: "+a[i] + "\n";
    System.out.println("a["+i+"]: "+a[i]);
}

System.out.println();
int l=1;
for(int i=0;i<n;i++){
    if(l==1) {
        tampil += "f(x) = "+a[i];
        System.out.print("f(x) = "+a[i]);
        l++;
    }else{
        if(a[i]>0) {
            tampil += " + " + a[i] + "x^" + i + " ";
            System.out.print(" + " + a[i] + "x^" + i + " ");
        }else if(a[i]<0) {
            tampil += " - " + -a[i] + "x^" + i + " ";
            System.out.print(" - " + -a[i] + "x^" + i + " ");
        }else{
            continue;
        }
    }
}

tampil += "\n";

System.out.println();

System.out.print("\nMasukkan x yang ingin dicari hasil nya: ");
double x = scan.nextDouble();
double result = 0;
for (int i = 0; i < n; i++) {
    if ((i == 0)) result += a[i];
    else result += a[i] * Math.pow(x, i);
}
tampil += "Hasil nya adalah: " + result;
System.out.println("Hasil nya adalah: " + result);
if (pillTampil == 2) {
    BacaTulisFile.tulisFile(tampil,nameFile);
}
}

// Interpolasi Fungsi Dengan Selang Dan Jumlah Partisi Berdasarkan Inputan User
else{
    double a,z;
    double h;
    System.out.print("Masukkan awal selang tersebut: ");
    a=scan.nextDouble();
    System.out.print("Masukkan akhir selang tersebut: ");
    z=scan.nextDouble();
    System.out.print("Masukkan jumlah partisi yang diinginkan: ");
    n=scan.nextInt();
    h=(z-a)/((double)n);
    A = new double[n+1][n+1];
    double[] y = new double[n+1];
    double[] c = new double[n+1];
    double[] s = new double[n+1];

    for (int i=0;i<(n+1);i++){
        c[i]=a+((double)i * h);
        y[i]=fungsi(c[i]);
    }

    for (int i=0;i<(n+1);i++){
        for (int j=0;j<(n+1);j++){
            if((i==0)&&(j==0)){
                A[i][j]=1;
            }else {
                A[i][j] = Math.pow(c[i], j);
            }
        }
    }
}

```

Jika pilihan menu sama dengan 5, maka akan masuk ke menu interpolasi fungsi yang menerima selang dan jumlah partisi dan menghasilkan array hasil sesuai fungsi dan dicari matriks A nya dan solusi persamaan tersebut yaitu a_0, a_1, \dots dst disimpan dalam array $s[i]$. Sisa program diatas

sampai akhir menu 5 berisi tentang ingin ditampilkan dimana solusi persamaan tersebut. Jika pilTampil sama dengan 1 maka akan ditampilkan di layar saja, sementara jika pilTampil sama dengan 2 maka akan ditampilkan di layar sekaligus di tulis ke dalam file. Dalam program juga akan dicetak $f(x)$ yang memuat fungsi interpolasi. Pengguna juga dapat memasukkan nilai x dan melihat hasil hampirannya di layar.

```

solvegauss(A,y,(n+1),(n+1));
s=solutionint(A,y,(n+1),(n+1));
System.out.println();
System.out.println("Tampilkan solusi persamaan ke : ");
System.out.println("1. Layar saja");
System.out.println("2. Layar dan File");
System.out.print("Pilih : ");
pilTampil = scan.nextInt();
scan.nextLine();
if (pilTampil == 2) {
    System.out.print("Masukan tujuan file : ");
    nameFile = scan.nextLine();
    System.out.println("Solusi sukses ditulis di file " + nameFile);
}
for(int i=0;i<(n+1);i++){
    tampil += "a["+i+"]: "+s[i]+"\n";
    System.out.println("a["+i+"]: "+s[i]);
}

System.out.println();
int l=1;
for(int i=0;i<(n+1);i++){
    if(l==1) {
        tampil += "f(x) = "+s[i];
        System.out.print("f(x) = "+s[i]);
        l++;
    }else{
        if(s[i]>0) {
            tampil += " + " + s[i] + "x^" + i + " ";
            System.out.print(" + " + s[i] + "x^" + i + " ");
        }else if(s[i]<0){
            tampil += " - " + -s[i] + "x^" + i + " ";
            System.out.print(" - " + -s[i] + "x^" + i + " ");
        }else{
            continue;
        }
    }
}
tampil += "\n";
System.out.println();

System.out.print("\nMasukkan x yang ingin dicari hasil nya: ");
double x = scan.nextDouble();
double result = 0;
for (int i = 0; i < (n+1); i++) {
    if ((i == 0)) result += s[i];
    else result += s[i] * Math.pow(x, i);
}
tampil += "Hasil nya adalah: " + result;
System.out.println("Hasil nya adalah: " + result);

if (pilTampil == 2) {
    BacaTulisFile.tulisFile(tampil,nameFile);
}
}
}

```

BAB IV

EKSPERIMEN

4.1 Tampilan Menu Utama

```
=====MENU UTAMA=====
!1. Pencarian Solusi Persamaan Dengan Metode Gauss      :
!2. Pencarian Solusi Persamaan Dengan Metode Gauss-Jordan:
!3. Pencarian Solusi Matriks Hilbert                      :
!4. Penaksiran Nilai Dengan Polinom Interpolasi           :
!5. Penaksiran Nilai Fungsi Dengan Polinom Interpolasi  :
=====
Pilih Menu Berdasarkan Angka<1-5> :_
```

Ketika program dijalankan, tampilan menu utama akan muncul pertama kali yang berisi pilihan-pilihan yang dapat dilakukan oleh program. Di sini user diharuskan untuk memilih pilihan menu dengan menginput nomor yang merujuk pada hal yang ingin dijalankan. Setelah menginput nomor, maka program akan bertindak sesuai dengan nomor yang diinput, berikut adalah penjelasan masing-masing nomor :

1) Solusi SPL dengan Metode Gauss

```
=====MENU UTAMA=====
!1. Pencarian Solusi Persamaan Dengan Metode Gauss      :
!2. Pencarian Solusi Persamaan Dengan Metode Gauss-Jordan:
!3. Pencarian Solusi Matriks Hilbert                      :
!4. Penaksiran Nilai Dengan Polinom Interpolasi           :
!5. Penaksiran Nilai Fungsi Dengan Polinom Interpolasi  :
=====
Pilih Menu Berdasarkan Angka<1-5> :1
Masukan pilihan input :
1. Konsol
2. File Eksternal
Pilihan <1/2> : 1
Masukan banyak persamaan :
4
Masukan persamaan :
1x+2y+3z=7
1x+2y=3
1x+2y=3
1z+3y=5
Matriks tersebut adalah:
1.00  2.00  3.00  7.00
1.00  2.00  0.00  3.00
1.00  2.00  0.00  3.00
0.00  3.00  1.00  5.00
```

Jika user memilih menu nomor 1, maka pertama-tama user akan diminta untuk memilih jenis input SPL, apakah ingin dari console secara langsung atau melalui file eksternal. Gambar di atas merupakan contoh apabila user memilih masukan melalui console secara langsung. Setelah memilih pilihan console, user diminta untuk menginput banyaknya persamaan yang ingin diselesaikan serta SPLnya. Kemudian program akan menampilkan matriks augmented dari persamaan yang diinput oleh user seperti pada gambar.

```

tampilkan solusi persamaan ke :
1. Layar saja
2. Layar dan File
Pilih : 2
Masukan tujuan file : solusi.txt
Solusi sukses ditulis di file solusi.txt

Hasil OBE Gauss :
1.00 2.00 3.00 7.00
0.00 1.00 0.33 1.67
0.00 0.00 1.00 1.33
0.00 0.00 0.00 0.00

Solusi Variabel x adalah 0.5555555555555554
Solusi Variabel y adalah 1.2222222222222223
Solusi Variabel z adalah 1.3333333333333333

```

Setelah itu, user diminta untuk memilih jenis output solusi dari SPL, apakah ingin ditampilkan ke layar saja atau ingin ditulis ke file eksternal juga. Jika memilih pilihan 2, maka user akan diminta memasukkan tujuan file tempat untuk ditulis solusi SPLnya, dalam hal ini yang digunakan ialah solusi.txt. Setelah itu program akan menampilkan ke layar hasil OBE Gauss dari matriks augmented serta solusinya. Penampakan isi file solusi.txt nya ialah :

```

1 FILE_FILE_SOLUSI
2 a[0]:1.0
3 a[1]:-2.705659945667147
4 a[2]:4.315171775049225
5 a[3]:-4.706288495483743
6 a[4]:3.5089903784726517
7 a[5]:-1.779089674782424
8 a[6]:0.6098479128297174
9 a[7]:-0.13880777561284768
10 a[8]:0.020068221299530096
11 a[9]:-0.0016663778675825068
12 a[10]:6.04621529445226E-5
13 f(x) = 1.0 - 2.705659945667147x^1 + 4.3151717
0.13880777561284768x^7 + 0.020068221299530096
14 Hasil nya adalah: 0.004243679913920939
15
16 Solusi Variabel x adalah 0.5555555555555554
17 Solusi Variabel y adalah 1.2222222222222223
18 Solusi Variabel z adalah 1.3333333333333333
19

```

Perhatikan line 16 s.d. 18 berisi data solusi SPL yang diinginkan user.

2) Solusi SPL dengan Metode Gauss-Jordan

Mirip seperti pilihan nomor 1, hanya saja di sini dalam penyelesaiannya sampai Gauss-Jordan. Berikut adalah tampilannya :

```

=====MENU UTAMA=====
1. Pencarian Solusi Persamaan Dengan Metode Gauss
2. Pencarian Solusi Persamaan Dengan Metode Gauss-Jordan
3. Pencarian Solusi Matriks Hilbert
4. Penaksiran Nilai Dengan Polinom Interpolasi
5. Penaksiran Nilai Fungsi Dengan Polinom Interpolasi
=====
Pilih Menu Berdasarkan Angka<1-5> :2
Masukan pilihan input :
1. Konsol
2. File Eksternal
Pilihan <1/2> : 1
Masukan banyak persamaan :
2
Masukan persamaan :
3x-2y+1z=-1
5x-1y=4
Matriks tersebut adalah:
3.00 -2.00 1.00 -1.00
5.00 -1.00 0.00 4.00
Tampilkan solusi persamaan ke :
1. Layar saja
2. Layar dan File
Pilih : 2
Masukan tujuan file : solusi.txt
Solusi sukses ditulis di file solusi.txt

Hasil OBE Gauss-Jordan:
1.00 0.00 -0.14 1.29
0.00 1.00 -0.71 2.43

Solusi Variabel x adalah 1.2857142857142858-(-0.14285714285714288*x3)
Solusi Variabel y adalah 2.428571428571429-(-0.7142857142857143*x3)
Solusi Variabel z adalah x3

```

3) Solusi Matriks Hilbert

```

=====MENU UTAMA=====
1. Pencarian Solusi Persamaan Dengan Metode Gauss
2. Pencarian Solusi Persamaan Dengan Metode Gauss-Jordan
3. Pencarian Solusi Matriks Hilbert
4. Penaksiran Nilai Dengan Polinom Interpolasi
5. Penaksiran Nilai Fungsi Dengan Polinom Interpolasi
=====
Pilih Menu Berdasarkan Angka<1-5> :3
Notes:Hanya bisa sampai n=13 dikarenakan overflow.
Masukkan jumlah n yang diinginkan :5

Matriks tersebut adalah:
1.00 0.50 0.33 0.25 0.20 1.00
0.50 0.33 0.25 0.20 0.17 1.00
0.33 0.25 0.20 0.17 0.14 1.00
0.25 0.20 0.17 0.14 0.13 1.00
0.20 0.17 0.14 0.13 0.11 1.00

Ingin Menyelesaikan dengan Metode:
1.Metode Gauss
2.Metode Gauss-Jordan
Pilihan Anda Adalah: 2

```

Jika user memilih menu pilihan 3, maka user akan diminta input nilai n yang diinginkan. Kemudian program akan menampilkan Matriks Hilbert untuk input n. Setelah itu sama seperti menu 1 dan menu 2. Berikut adalah tampilan file eksternal jika dijalankan menu 3 :


```

FILE_FILE_SOLUSI

Solusi Variabel ke [1] adalah 5.0000000000000283
Solusi Variabel ke [2] adalah -120.000000000003558
Solusi Variabel ke [3] adalah 630.00000000002286
Solusi Variabel ke [4] adalah -1120.0000000004216
Solusi Variabel ke [5] adalah 630.0000000002276

```

4) Menaksir nilai dengan polinom interpolasi

```

=====MENU UTAMA=====
1. Pencarian Solusi Persamaan Dengan Metode Gauss
2. Pencarian Solusi Persamaan Dengan Metode Gauss-Jordan
3. Pencarian Solusi Matriks Hilbert
4. Penaksiran Nilai Dengan Polinom Interpolasi
5. Penaksiran Nilai Fungsi Dengan Polinom Interpolasi
=====
Pilih Menu Berdasarkan Angka<1-5> :4
Masukan pilihan input :
1. Konsol
2. File Eksternal
Pilihan <1/2> : 1
Masukkan jumlah data yang ingin dimasukkan: 2
Masukkan nilai x[0]:
2
Masukkan nilai y[0]:
3
Masukkan nilai x[1]:
4
Masukkan nilai y[1]:
5
Hasil OBE Gauss :
1.00 2.00 3.00
0.00 1.00 1.00
Tampilkan solusi persamaan ke :
1. Layar saja
Tampilkan solusi persamaan ke :
1. Layar saja
2. Layar dan File
Pilih : 2
Masukan tujuan file : solusi.txt
Solusi sukses ditulis di file solusi.txt
a[0]:1.0
a[1]:1.0

f(x) = 1.0 + 1.0x^1

Masukkan x yang ingin dicari hasil nya: 2
Hasil nya adalah: 3.0

```

Jika user memilih menu 4, maka user diminta memasukkan jumlah data yang ingin dimasukan berupa pasangan nilai x dan y yang akan dibuat persamaannya. Kemudian program akan membuat persamaan polinomnya dan menaksir nilai yang ingin dicari user. Berikut adalah tampilan file eksternal jika menerima data menu 4 :

```

FILE_FILE_SOLUSI
a[0]:1.0
a[1]:1.0
f(x) = 1.0 + 1.0x^1
Hasil nya adalah: 3.0

```

5) Penaksiran Nilai Fungsi dengan Polinom Interpolasi

```

=====MENU UTAMA=====
1. Pencarian Solusi Persamaan Dengan Metode Gauss
2. Pencarian Solusi Persamaan Dengan Metode Gauss-Jordan
3. Pencarian Solusi Matriks Hilbert
4. Penaksiran Nilai Dengan Polinom Interpolasi
5. Penaksiran Nilai Fungsi Dengan Polinom Interpolasi
=====
Pilih Menu Berdasarkan Angka<1-5> :5
Masukan pilihan input :
1. Konsol
2. File Eksternal
Pilihan <1/2> : 1
Masukkan awal selang tersebut: 1
Masukkan akhir selang tersebut: 10
Masukkan jumlah partisi yang diinginkan: 3

Hasil OBE Gauss :
1.00 1.00 1.00 1.00 0.12
0.00 1.00 11.00 111.00 -0.01
0.00 0.00 1.00 18.00 0.00
0.00 0.00 0.00 1.00 -0.00

Tampilkan solusi persamaan ke :
1. Layar saja
2. Layar dan File
Pilih : 2
Masukan tujuan file : solusi.txt
Solusi sukses ditulis di file solusi.txt
Solusi sukses ditulis di file solusi.txt
a[0]:0.21071015325761872
a[1]:-0.10292373008064712
a[2]:0.015579477366780729
a[3]:-7.394201532715299E-4

f(x) = 0.21071015325761872 - 0.10292373008064712x^1 + 0.015579477366780729x^2
- 7.394201532715299E-4x^3

Masukkan x yang ingin dicari hasil nya: 2
Hasil nya adalah: 0.06126524133727515

```

Ketika user memilih menu 5, maka user akan diminta selang untuk melakukan partisi. Setelah itu user diminta menginput jumlah partisi yang diinginkan. Kemudian program akan melakukan OBE Gauss terhadap persamaan yang terbentuk. Sisanya sama seperti sebelumnya. Berikut adalah tampilan file eksternal setelah menu 5 dijalankan :

```

1 FILE_FILE_SOLUSI
2 a[0]:0.21071015325761872
3 a[1]:-0.10292373008064712
4 a[2]:0.015579477366780729
5 a[3]:-7.394201532715299E-4
6 f(x) = 0.21071015325761872 - 0.10292373008064712x^1 + 0.015579477366780729x^2 - 7.394201532715299E-4x^3
7 Hasil nya adalah: 0.06126524133727515
8

```

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Secara garis besar, program kami dapat menerima kemungkinan-kemungkinan Sistem Persamaan Linear seperti yang ada pada spesifikasi tugas yaitu solusi unik, solusi tak hingga (solusiparametris) dan tak mempunyai solusi. Program juga dapat membaca dari console dan juga bisa membaca dari file eksternal. Solusi dari persamaan tersebut bisa ditampilkan di console dan dapat ditulis ke file eksternal juga.

Program kami juga dapat memperkirakan hampiran fungsi dengan menggunakan hampiran interpolasi. Selain itu, kami juga dapat menuliskan nilai suatu fungsi dengan nilai x yang diinput user. Fungsi yang kami gunakan adalah fungsi hampiran interpolasi yang telah didapat tersebut. Cara mendapatkan fungsi interpolasi ada dengan dua cara, yaitu dengan data yang diberikan user ataupun dengan hampiran suatu fungsi di selang tertentu.

5.2 Saran

Beberapa hal yang dapat diperbaiki dalam program kami adalah matriks Hilbert. Karena batas wajar dari matriks Hilbert buatan kami adalah jika $n=13$. Jika lebih dari 13 maka solusi akan melewati batas normal. Selain itu, program kami juga terdapat banyak duplikat terutama di program utama. Alangkah baiknya jika dibuat prosedur sehingga tinggal dipanggil saja saat dibutuhkan.

REFERENSI

- 1) <https://math.stackexchange.com/questions/1334983/gauss-elimination-difference-between-partial-and-complete-pivoting> diakses pada : 5 Oktober 2017 pukul 19.30
- 2) http://www.academia.edu/8738105/Interpolasi_Definisi_dan_Macam-macamnya diakses pada : 5 Oktober 2017 pukul 19.50
- 3) <http://arifhidayat659.blogspot.co.id/2014/04/metode-eliminasi-gauss-dan-gauss-jordan.html> diakses pada : 5 Oktober 2017 pukul 19.00