**LAPORAN TUGAS BESAR I**

**ALJABAR GEOMETRI**

**LAPORAN**

Diajukan untuk memenuhi Tugas Besar Aljabar Geometri

oleh

**YUSUF RAHMAT PRATAMA**

**16516062**

****

**TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**BANDUNG**

**2017**

**BAB I**

**DESKRIPSI MASALAH**

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.

**BAB II**

**TEORI SINGKAT**

**2.1 Metode Eliminasi Gauss**

Metode eliminasi Gauss adalah metode yang digunakan untuk mereduksi sebuah *matrix augmented* menjadi *echelon form* menggunakan OBE (Operasi Baris Elementer) dan memanipulasi persamaan yang didapatkan dari *echelon form* tersebut untuk mengeliminasi variabel dan mendapatkan solusi daru sebuah SPL (Sistem Persamaan Linier) dalam bentuk matrix tersebut.

**2.2 Metode Eliminasi Gauss-Jordan**

Metode eliminasi Gauss-Jordan adalah metode yang mirip dengan metode eliminasi Gauss, tetapi pada metode ini *matrix augmented* diubah menjadi *reduced echelon form* menggunakan OBE (Operasi Baris Elementer). Penambahan yang digunakan dalam Gauss-Jordan ini adalah *backward phase* dimana bilangan selain *leading one* (jika ada) pada suatu kolom dijadikan 0.

**2.3 Interpolasi**

Interpolasi digunakan untuk mencari fungsi yang menghampiri beberapa titik yang diketahui. Dari fungsi yang didapatkan, dapat dicari hampiran dari suatu titik yang kemungkinan satu fungsi dengan titik yang diketahui. Semakin banyak titik yang dimasukkan sebagai patokan, semakin kecil galat dari fungsi tersebut.

**BAB III**

**IMPLEMENTASI PROGRAM**

|  |
| --- |
| **FILE MATRIKS.java**  import java.io.\*;  import java.io.BufferedReader;  import java.io.FileReader;  import java.util.Arrays;  import java.util.Scanner;  public class Matrix {      private int rowSize;      private int colSize;      private double[][] data;      public double[][] dataInterpolasi;      private double[] dataSolved;      //initializer      //create 100-by-100 matrix of 0      public Matrix(int m, int n) {          this.rowSize = m;          this.colSize = n + 1;          this.data = new double[100][100];      }      //getter      //get number of row used      public int getRowSize() {          return rowSize;      }      //get number of column used      public int getColSize() {          return colSize;      }      public double getData(int i, int j) {          return this.data[i][j];      }      //setter      public void setData() {          Scanner scanner = new Scanner(System.in);          for (int i = 0; i < this.rowSize; i++) {              System.out.println("Persamaan " + (i + 1) + " : ");              for (int j = 0; j < this.colSize; j++) {                  if (j == colSize - 1) {                      System.out.print("Konstanta : ");                  } else {                      System.out.print("X" + (j + 1) + " : ");                  }                  this.data[i][j] = scanner.nextDouble();              }              System.out.println();          }      }      public void setDataInterpolasi(int n) {          Scanner scanner = new Scanner(System.in);          dataInterpolasi = new double[2][n];          for (int i = 0; i < n; i++) {              System.out.println("Titik " + (i + 1) + ": ");              System.out.print("  X  = ");    dataInterpolasi[0][i] = scanner.nextDouble();              System.out.print("f(X) = ");    dataInterpolasi[1][i] = scanner.nextDouble();              System.out.println();          }      }      public void readFileSPL(String namaFile, Matrix mat) {          File file = new File(namaFile);  try {              Scanner input = new Scanner(file);              int col = 0;              int row = 0;              if(input.hasNextLine()) {                  row++;                  Scanner test = new Scanner(input.nextLine());                  while (test.hasNextDouble()) {                      ++col;                      test.nextDouble();                  }                  while (input.hasNextLine()) {                      row++;                      while (test.hasNextDouble()) {                          ++col;                          test.nextDouble();                      }                      input.nextLine();                  }                  mat.rowSize = row;                  mat.colSize = col;                  input.close();                  mat = new Matrix(row, col);                  input = new Scanner(file);                  for (int i = 0; i < rowSize; i++) {                      for (int j = 0; j < colSize; j++) {                          if (input.hasNextDouble()) {                              data[i][j] = input.nextDouble();                          }                      }                  }              }  } catch(IOException e){  System.out.println("File tidak ditemukan");  }  }      public void readFileInterpolasi(String namaFile, Matrix mat) {          File file = new File(namaFile);  BufferedReader reader = null;  try {              Scanner input = new Scanner(file);              int col = 0;              int row = 0;              if(input.hasNextLine()) {                  row++;                  Scanner test = new Scanner(input.nextLine());                  while (test.hasNextDouble()) {                      ++col;                      test.nextDouble();                  }                  while (input.hasNextLine()) {                      row++;                      while (test.hasNextDouble()) {                          col++;                          test.nextDouble();                      }                      input.nextLine();                  }                  rowSize = col;                  colSize = col + 1;                  input.close();                  input = new Scanner(file);                  dataInterpolasi = new double[2][col];                  for (int i = 0; i < col; i++) {                      dataInterpolasi[0][i] = input.nextDouble();                  }                  for (int i = 0; i < col; i++) {                      dataInterpolasi[1][i] = input.nextDouble();                  }              }  } catch(IOException e){  System.out.println("File tidak ditemukan");              //panggil fungsinya lagi  }  }      public void changeData(int i, int j, double d) {          this.data[i][j] = d;      }      //output      public void printMatrix() {          for (int i = 0; i < this.rowSize; i++) {              for (int j = 0; j < this.colSize; j++) {                  System.out.printf("%.2f ", this.data[i][j]);              }              System.out.println();          }          System.out.println();      }      public void printToFile(boolean spl) {          Scanner scanner = new Scanner(System.in);          System.out.print("Simpan solusi? (y/n) : ");          char c = scanner.next().charAt(0);          if (c == 'y') {              System.out.printf("Masukkan nama file penyimpanan : ");              String namaFile = scanner.next();              System.out.println();              if (spl) {                  try {                      PrintWriter writer = new PrintWriter(namaFile, "UTF-8");                      int indeks;                      int count;                      int rowCheck = rowSize;                      int[] free = new int[colSize - 1]; Arrays.fill(free, (-1));                      boolean check = false;                      boolean first;                      if (weirdRow(rowSize-1)) {                          writer.println("SPL TIDAK memiliki solusi");                      }                      else {                          while (zeroRow(rowCheck - 1)) {                              rowCheck--;                          }                          if (rowCheck < (colSize - 1)) {                              writer.println("SPL memiliki BANYAK SOLUSI");                              check = true;                          } else {                              writer.println("SPL memiliki SOLUSI UNIK");                          }                          for (int i = 0; i < rowCheck; i++) {                              indeks = -1;                              count = 0;                              first = false;                              for (int j = 0; j < colSize; j++) {                                  if ((data[i][j] > 0.99) && (data[i][j] < 1.01) && (!first)) {                                      indeks = j;                                      first = true;                                  }                                  if (indeks == j) {                                      writer.print("X" + (indeks + 1) + " =");                                  } else if ((data[i][j] < -0.01) || (data[i][j] > 0.01) || (j == (colSize - 1))) {                                      double temp = data[i][j];                                      if (j != colSize - 1) {                                          temp \*= (-1);                                      }                                      writer.print(" ");                                      if (count != 0) {                                          if (temp > 0) {                                              writer.print("+ ");                                          } else {                                              temp \*= (-1);                                              writer.print("- ");                                          }                                      }                                      writer.printf("%.2f",temp);                                      if (check) {                                          if (j != (colSize - 1)) {                                              writer.print("X" + (j + 1));                                              free[j] = j;                                          }                                      }                                      count++;                                  }                              }                              writer.println();                          }                          if (check) {                              for (int l = 0; l < (colSize - 1); l++) {                                  if (free[l] != (-1)) {                                      writer.println("X" + (free[l] + 1) + " = free");                                  }                              }                          }                      }                      writer.close();                  } catch ( IOException e ) {                      System.out.println("Error dalam penyimpanan file");                  }              } else {                  try {                      PrintWriter writer = new PrintWriter(namaFile, "UTF-8");                      //Print                      writer.println("Persamaan Interpolasi : ");                      writer.print("P(X) = ");                      int count=0;                      if(dataSolved[colSize - 2] != 0) {                          if((dataSolved[colSize-2] > 0.01) || (dataSolved[colSize-2] < -0.01)) {                              writer.printf("%.2fX^%d", dataSolved[colSize - 2], (colSize - 2));                              count++;                          }                      }                      for(int i = colSize - 3; i >= 0; i--) {                          if((dataSolved[i] > 0.01) || (dataSolved[i] < -0.01)) {                              if(dataSolved[i] < 0) {                                  writer.printf(" - %.2f",(dataSolved[i]\*(-1)));                                  count++;                              }                              else {                                  if(count == 0) {                                      writer.printf(" %.2f",dataSolved[i]);                                  }                                  else {                                      writer.printf(" + %.2f",dataSolved[i]);                                  }                                  count++;                              }                              if(i > 0) {                                  writer.printf("X^%d",i);                              }                          }                      }                      writer.close();                  } catch (IOException e) {                      System.out.println("Error dalam penyimpanan file");                  }              }          }      }      public void copyMatrix(Matrix M) {          for (int i = 0; i < rowSize; i++) {              for (int j = 0; j < colSize; j++) {                  this.data[i][j] = M.data[i][j];              }          }      }      //methods for gauss jordan      //swap i'th row with the k'th row      private void swapRow(int i, int k) {          if (i != k) {              for (int j = 0; j < colSize; j++) {                  double temp = data[i][j];                  data[i][j] = data[k][j];                  data[k][j] = temp;              }          }      }      private int maxAbsCol(int i, int j) {          int idxMax = i;          for (int k = i; k < rowSize; k++) {              if ((data[idxMax][j] == 0) || ((data[k][j] != 0)&&(data[k][j] > data[idxMax][j]))&&(!zeroRow(k))&&(!weirdRow(k))) {                  idxMax = k;              }          }          return idxMax;      }      private void pivotingPoint(int i, int j) {          swapRow(i, maxAbsCol(i, j));      }      private void moveZeroRow() {          for(int i = 0; i < rowSize; i++) {              boolean check = false;              for (int j = 0; j < (colSize - 1); j++) {                  if(data[i][j] != 0) {                      check = true;                      break;                  }              }              if(!check) {                  swapRow(i,(rowSize - 1));              }          }      }      private void divideRow(int i, double d) {          for (int j = 0; j < colSize; j++) {              data[i][j] /= d;          }      }      private void substractRow(int i, double d, int k) {          for (int j = 0; j < colSize; j++) {              data[k][j] -= d \* data[i][j];          }      }      private boolean zeroRow(int i) {          for (int j = 0; j < colSize; j++) {              if (data[i][j] != 0) {                  return false;              }          }          return true;      }      private boolean weirdRow(int i) {          for (int j = 0; j < colSize - 1; j++) {              if (data[i][j] != 0) return false;          }          return !(data[i][colSize - 1] == 0);      }      public void gauss() {          //diperhatikan Kalo leading one nya ga di i, i dan kalo doi ga pembaginya 0          int i = 0;          moveZeroRow();          for (int l = 0; l < colSize; l++) {              pivotingPoint(i, l);              if (data[i][l] != 0) {                  for (int k = i + 1; k <= rowSize; k++) {                      if(k != rowSize) { //Kalo k=rowsize kan gaada data nya jadi ini mungkin error. Cmn perlu divideRow aja                          double t = data[k][l] / data[i][l];                          substractRow(i, t, k);                      }                      divideRow(i, data[i][l]);                  }                  i++;              }          }      }      public void gaussJordan() {          int j;          boolean check;          gauss();          for (int i = 0; i < rowSize; i++) {              check = false;              j = 0;              while ((!check) && (j < colSize)) {                  if(data[i][j] > 0.99 && data[i][j] < 1.01) {                      check = true;                  }                  else {                      j++;                  }              }              if (check) {                  for (int k = 0; k < rowSize; k++) {                      if(k != i) {                          double t = data[k][j]/data[i][j];                          substractRow(i, t, k);                      }                  }              }          }      }      public void solutionMatrix() {          int indeks;          int count;          int rowCheck = rowSize;          int[] free = new int[colSize - 1]; Arrays.fill(free, (-1));          boolean check = false;          boolean first;          if (weirdRow(rowSize-1)) {              System.out.println("SPL TIDAK memiliki solusi");          }          else {              while (zeroRow(rowCheck - 1)) {                  rowCheck--;              }              if (rowCheck < (colSize - 1)) {                  System.out.println("SPL memiliki BANYAK SOLUSI");                  check = true;              } else {                  System.out.println("SPL memiliki SOLUSI UNIK");              }              for (int i = 0; i < rowCheck; i++) {                  indeks = -1;                  count = 0;                  first = false;                  for (int j = 0; j < colSize; j++) {                      if ((data[i][j] > 0.99) && (data[i][j] < 1.01) && (!first)) {                          indeks = j;                          first = true;                      }                      if (indeks == j) {                          System.out.print("X" + (indeks + 1) + " =");                      } else if ((data[i][j] < -0.01) || (data[i][j] > 0.01) || (j == (colSize - 1))) {                          double temp = data[i][j];                          if (j != colSize - 1) {                              temp \*= (-1);                          }                          System.out.print(" ");                          if (count != 0) {                              if (temp > 0) {                                  System.out.print("+ ");                              } else {                                  temp \*= (-1);                                  System.out.print("- ");                              }                          }                          System.out.printf("%.2f",temp);                          if (check) {                              if (j != (colSize - 1)) {                                  System.out.print("X" + (j + 1));                                  free[j] = j;                              }                          }                          count++;                      }                  }                  System.out.println();              }              if (check) {                  for (int l = 0; l < (colSize - 1); l++) {                      if (free[l] != (-1)) {                          System.out.println("X" + (free[l] + 1) + " = free");                      }                  }              }          }      }      private double powerOf(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 (int t) {          Scanner scanner = new Scanner(System.in);          dataSolved = new double[colSize - 1];          //Membuat Matriks dari input titik          for(int i = 0; i < rowSize; i++) {              for(int j = 0; j < (colSize - 1); j++) {                  this.data[i][j] = powerOf(dataInterpolasi[0][i], (colSize - j - 2));              }              this.data[i][colSize - 1] = dataInterpolasi[1][i];          }          printMatrix();          gaussJordan();          for(int i = 0; i < rowSize; i++) {              dataSolved[i] = data[rowSize - i - 1][colSize - 1];          }          //Print          System.out.println("Persamaan Interpolasi : ");          System.out.print("P(X) = ");          int count=0;          if(dataSolved[colSize - 2] != 0) {              if((dataSolved[colSize-2] > 0.01) || (dataSolved[colSize-2] < -0.01)) {                  System.out.printf("%.2fX^%d", dataSolved[colSize - 2], (colSize - 2));                  count++;              }          }          for(int i = colSize - 3; i >= 0; i--) {              if((dataSolved[i] > 0.01) || (dataSolved[i] < -0.01)) {                  if(dataSolved[i] < 0) {                      System.out.printf(" - %.2f",(dataSolved[i]\*(-1)));                      count++;                  }                  else {                      if(count == 0) {                          System.out.printf(" %.2f",dataSolved[i]);                      }                      else {                          System.out.printf(" + %.2f",dataSolved[i]);                      }                      count++;                  }                  if(i > 0) {                      System.out.printf("X^%d",i);                  }              }          }          System.out.println();      }      public double hampiranInterpolasi(double x) {      //Asumsi dataSolved sudah terdefinisi          double d = 0;          for(int i = 0; i < colSize-1; i++) {              d += dataSolved[i] \* powerOf(x, i);          }          return d;      }  } |

|  |
| --- |
| **FILE MAIN.java**  import java.util.Scanner;  import java.io.\*;  import java.io.BufferedReader;  import java.io.FileReader;  import java.util.Arrays;  import java.lang.Math;  public class Main {      public static void printMenu() {          System.out.printf("MENU : \n");          System.out.printf("1. Solusi persamaan lanjar\n");          System.out.printf("2. Hampiran Interpolasi\n");          System.out.printf("3. Matriks Hilbert\n");          System.out.printf("4. Fungsi e\n");          System.out.printf("0. Keluar\n");      }      public static void linearSolution() {          Scanner scanner = new Scanner(System.in);          String fileName;          System.out.printf("Pilihan metode input :\n");          System.out.printf("1. Masukan dari Keyboard\n");          System.out.printf("2. Masukan dari File Eksternal\n\n");          System.out.printf("Pilih metode input :  ");          int m = scanner.nextInt();          while (m != 1 && m != 2) {              System.out.printf("Input Salah\n");              System.out.printf("Pilih metode Input :  ");              m = scanner.nextInt();          }          Matrix mat = new Matrix(0, 0);          switch (m) {              case 1 :                  System.out.print("Masukkan jumlah persamaan : ");                  int l = scanner.nextInt();                  System.out.print("Masukkan jumlah variabel : ");                  int n = scanner.nextInt();                  mat = new Matrix(l, n);                  mat.setData();                  System.out.println();                  break;              case 2 :                  System.out.printf("Masukkan nama file eksternal input : ");                  fileName = scanner.next();                  //Proses Pembacaan File External                  mat.readFileSPL(fileName, mat);                  break;          }          System.out.println("Matriks Input : ");          mat.printMatrix();          System.out.println("Hasil Eliminasi Gauss : ");          mat.gauss();          mat.printMatrix();          System.out.println("Hasil Eliminasi Gauss - Jordan : ");          mat.gaussJordan();          mat.printMatrix();          System.out.println("Solusi : ");          mat.solutionMatrix();          mat.printToFile(true);          System.out.println();      }      public static void interpolation() {          Scanner scanner = new Scanner(System.in);          String fileName;          System.out.printf("Pilihan metode input :\n");          System.out.printf("1. Masukan dari Keyboard\n");          System.out.printf("2. Masukan dari File Eksternal\n\n");          System.out.printf("Pilih metode input :  ");          int m = scanner.nextInt();          int l = 0;          while (m != 1 && m != 2) {              System.out.printf("\nMasukan Salah\n");              System.out.printf("Pilih metode Input :  ");              m = scanner.nextInt();          }          Matrix mat = new Matrix(0, 0);          switch (m) {              case 1 :                  System.out.print("\nMasukkan jumlah titik : ");                  l = scanner.nextInt();                  mat = new Matrix(l, l);                  mat.setDataInterpolasi(l);                  System.out.println();                  break;              case 2 :                  System.out.printf("\nMasukkan nama file eksternal input : ");                  fileName = scanner.next();                  //Proses Pembacaan File External                  mat.readFileInterpolasi(fileName, mat);                  break;          }          mat.interpolasi(l);          System.out.printf("\nApakah anda ingin mencari nilai hampiran? (y/n) : ");          char c = scanner.next().charAt(0);          while(c == 'y') {              System.out.print("X = ");              double d = scanner.nextDouble();              System.out.printf("f(%.2f) = %.2f\n\n", d, mat.hampiranInterpolasi(d));              System.out.printf("Apakah anda ingin mencari nilai hampiran? (y/n) : ");              c = scanner.next().charAt(0);          }          mat.printToFile(false);          System.out.println();      }       public static void hilbert() {          Scanner scanner = new Scanner(System.in);          int n;          System.out.printf("Masukkan dimensi Matriks Hilbert n : ");          n = scanner.nextInt();          Matrix mat = new Matrix(n,n);          for(int i=0; i<n; i++) {              for(int j=0; j<n; j++) {                  double d = (1.0 /(i+j+1));                  mat.changeData(i,j,d);              }              mat.changeData(i,n,1);          }          System.out.println("Matriks Input : ");          mat.printMatrix();          System.out.println("Hasil Eliminasi Gauss : ");          mat.gauss();          mat.printMatrix();          System.out.println("Hasil Eliminasi Gauss - Jordan : ");          mat.gaussJordan();          mat.printMatrix();          System.out.println("Solusi : ");          mat.solutionMatrix();          mat.printToFile(true);          System.out.println();      }      public static double f(double d) {          return (1.0 / (Math.exp(d)\*(1 + Math.sqrt(d) + (d \* d))));      }      public static void fungsiE() {          Scanner scanner = new Scanner(System.in);          double a, b;          int n;          double h;          System.out.println("Fungsi yang akan dihampiri :                                        ");          System.out.println();          System.out.println("    f(x) =  e^(-x) / (1 + √x + x^2)                                 ");          System.out.println();          System.out.println("    di selang           [a , b]     | a, b  bilangan bulat positif  ");          System.out.println("    dan jumlah titik       n        | n     bilangan bulat positif  ");          System.out.println();          System.out.print("Masukkan nilai a : ");          a = scanner.nextDouble();          System.out.print("Masukkan nilai b : ");          b = scanner.nextDouble();          System.out.print("Masukkan nilai n : ");          n = scanner.nextInt();          h = (b - a) / n;          Matrix mat = new Matrix(n, n);          mat.dataInterpolasi = new double[2][n+2];          int i = 0;          double x = a;          while(x<=b) {              mat.dataInterpolasi[0][i] = x;              mat.dataInterpolasi[1][i] = f(x);              x+=h;              i++;          }          mat.interpolasi(0);          System.out.printf("\nApakah anda ingin mencari nilai hampiran? (y/n) : ");          char c = scanner.next().charAt(0);          while(c == 'y') {              System.out.print("X = ");              double d = scanner.nextDouble();              System.out.printf("f(%.2f) = %.2f\n\n", d, mat.hampiranInterpolasi(d));              System.out.printf("Apakah anda ingin mencari nilai hampiran? (y/n) : ");              c = scanner.next().charAt(0);          }          mat.printToFile(false);          System.out.println();      }      public static void main(String[] args) {          Scanner scanner = new Scanner(System.in);          boolean jalan = true;          char c;          int m;          while (jalan) {              System.out.print("\033[H\033[2J");              System.out.flush();              System.out.println("Selamat Datang di       ");              System.out.println();              System.out.println("  ,-.----.                                                                                                                                   ");              System.out.println("  \\    /  \\                                 ,--,                      \_\_\_\_    .--.--.               ,--,                                   ");              System.out.println("  |   :    \\                      ,---,   ,--.'|                    ,'  , `. /  /    '.           ,--.'|                                    ");              System.out.println("  |   |  .\\ :  \_\_  ,-.   ,---.  ,---.'|   |  | :                 ,-+-,.' \_ ||  :  /`. /    ,---.  |  | :                       \_\_  ,-.      ");              System.out.println("  .   :  |: |,' ,'/ /|  '   ,'\\ |   | :   :  : '              ,-+-. ;   , ||;  |  |--`    '   ,'\\ :  : '       .---.         ,' ,'/ /|     ");              System.out.println("  |   |   \\ :'  | |' | /   /   |:   : :   |  ' |      ,---.  ,--.'|'   |  |||  :  ;\_     /   /   ||  ' |     /.  ./|  ,---.  '  | |' |      ");              System.out.println("  |   : .   /|  |   ,'.   ; ,. ::     |,-.'  | |     /     \\|   |  ,', |  |, \\  \\    `. .   ; ,. :'  | |   .-' . ' | /     \\ |  |   ,'   ");              System.out.println("  ;   | |`-' '  :  /  '   | |: :|   : '  ||  | :    /    /  |   | /  | |--'   `----.   \\'   | |: :|  | :  /\_\_\_/ \\: |/    /  |'  :  /       ");              System.out.println("  |   | ;    |  | '   '   | .; :|   |  / :'  : |\_\_ .    ' / |   : |  | ,      \_\_ \\  \\  |'   | .; :'  : |\_\_.   \\  ' .    ' / ||  | '       ");              System.out.println("  :   ' |    ;  : |   |   :    |'   : |: ||  | '.'|'   ;   /|   : |  |/      /  /`--'  /|   :    ||  | '.'|\\   \\   '   ;   /|;  : |        ");              System.out.println("  :   : :    |  , ;    \\   \\  / |   | '/ :;  :    ;'   |  / |   | |`-'      '--'.     /  \\   \\  / ;  :    ; \\   \\  '   |  / ||  , ;    ");              System.out.println("  |   | :     ---'      `----'  |   :    ||  ,   / |   :    |   ;/            `--'---'    `----'  |  ,   /   \\   \\ |   :    | ---'         ");              System.out.println("  `---'.|                       /    \\  /  ---`-'   \\   \\  /'---'                                  ---`-'     '---\" \\   \\  /            ");              System.out.println("    `---`                       `-'----'             `----'                                                          `----'                  ");              System.out.println("                                                                         By :                                                      ");              System.out.printf(  "                                                           13516083 / Abram Perdanaputra          \n");              System.out.printf(  "                                                           13516086 / Dandy Arif Rahman           \n");              System.out.printf(  "                                                           13516062 / Yusuf Rahmat Pratama        \n");              printMenu();              System.out.println();              System.out.printf("Masukkan pilihan anda : ");              m = scanner.nextInt();              System.out.println();              switch (m) {                  case 0 :                      jalan = false;                      break;                  case 1 :                      linearSolution();                      System.out.printf("Apakah anda ingin kembali ke menu utama? (y/n) : ");                      c = scanner.next().charAt(0);                      System.out.println();                      if(c == 'n') {                          jalan = false;                      }                      break;                  case 2 :                      interpolation();                      System.out.printf("Apakah anda ingin kembali ke menu utama? (y/n) : ");                      c = scanner.next().charAt(0);                      System.out.println();                      if(c == 'n') {                          jalan = false;                      }                      break;                  case 3 :                      hilbert();                      System.out.printf("Apakah anda ingin kembali ke menu utama? (y/n) : ");                      c = scanner.next().charAt(0);                      System.out.println();                      if(c == 'n') {                          jalan = false;                      }                      break;                  case 4 :                      fungsiE();                      System.out.printf("Apakah anda ingin kembali ke menu utama? (y/n) : ");                      c = scanner.next().charAt(0);                      System.out.println();                      if(c == 'n') {                          jalan = false;                      }                      break;                  default :                      System.out.printf("Input Salah\n");                      break;              }              System.out.print("\033[H\033[2J");              System.out.flush();              if (m == 0 || jalan == false) {                  System.out.println("Terimakasih sudah menggunakan program kami...");                  System.out.println("13516062");                  System.out.println("13516083");                  System.out.println("13516086");              }          }      }  } |

**BAB IV**

**EKSPERIMEN/CONTOH**

**4.1 Catatan**

Tulisan yang digarisbawahi merupakan input dari user.

**4.2 Contoh SPL 1**

Soal:

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

Implementasi program:

Input jumlah persamaan : 4

Input jumlah variabel : 4

Persamaan :

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

Matriks Input :

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

Hasil Eliminasi Gauss :

1.00 0.36 0.33 0.51 2.20

0.00 1.00 0.42 0.48 1.90

0.00 0.00 1.00 0.53 1.53

-0.00 -0.00 -0.00 1.00 1.00

Hasil Eliminasi Gauss - Jordan :

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

Solusi :

SPL memiliki SOLUSI UNIK

X1 = 1.00

X2 = 1.00

X3 = 1.00

X4 = 1.00

**4.3 Contoh SPL 2**

Soal:

x1 + 7x2 - 2x3 + 8x5 = -3

x1 + 7x2 - x3 + x4 = 2

2x1 + 14x2 - 4x3 + x4 - 13x5 = 3

2x1 + 14x2 - 4x3 + 16x5 = -6

Implementasi Program:

Input jumlah persamaan : 4

Input jumlah variabel : 5

Persamaan:

1 7 -2 0 8 -3

1 7 -1 1 0 2

2 14 -4 1 -13 3

2 14 -4 0 16 -6

Matriks Input :

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

Hasil Eliminasi Gauss :

1.00 7.00 -2.00 0.50 -6.50 1.50

0.00 0.00 1.00 0.50 6.50 0.50

-0.00 -0.00 -0.00 1.00 -29.00 9.00

0.00 0.00 0.00 0.00 0.00 0.00

Hasil Eliminasi Gauss - Jordan :

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

Solusi :

SPL memiliki BANYAK SOLUSI

X1 = -7.00X2 - 50.00X5 - 11.00

X3 = -21.00X5 - 4.00

X4 = 29.00X5 + 9.00

X2 = free

X5 = free

**4.4 Contoh SPL 3**

Soal:

Matriks Hilbert dengan n = 10 dan n = 20

Implementasi Program:

n =10

Masukkan jumlah n: 10

Matriks Input :

1.00 0.50 0.33 0.25 0.20 0.17 0.14 0.13 0.11 0.10 1.00

0.50 0.33 0.25 0.20 0.17 0.14 0.13 0.11 0.10 0.09 1.00

0.33 0.25 0.20 0.17 0.14 0.13 0.11 0.10 0.09 0.08 1.00

0.25 0.20 0.17 0.14 0.13 0.11 0.10 0.09 0.08 0.08 1.00

0.20 0.17 0.14 0.13 0.11 0.10 0.09 0.08 0.08 0.07 1.00

0.17 0.14 0.13 0.11 0.10 0.09 0.08 0.08 0.07 0.07 1.00

0.14 0.13 0.11 0.10 0.09 0.08 0.08 0.07 0.07 0.06 1.00

0.13 0.11 0.10 0.09 0.08 0.08 0.07 0.07 0.06 0.06 1.00

0.11 0.10 0.09 0.08 0.08 0.07 0.07 0.06 0.06 0.06 1.00

0.10 0.09 0.08 0.08 0.07 0.07 0.06 0.06 0.06 0.05 1.00

Hasil Eliminasi Gauss :

1.00 0.50 0.33 0.25 0.20 0.17 0.14 0.13 0.11 0.10 1.00

0.00 1.00 1.07 1.00 0.91 0.83 0.76 0.70 0.65 0.60 8.00

0.00 0.00 1.00 1.70 2.14 2.40 2.55 2.62 2.65 2.65 75.00

0.00 0.00 0.00 1.00 2.10 3.02 3.74 4.28 4.67 4.94 220.00

0.00 0.00 0.00 0.00 1.00 2.69 4.69 6.73 8.66 10.43 1470.00

0.00 0.00 0.00 0.00 0.00 1.00 3.17 6.25 9.90 13.83 5460.00

0.00 0.00 0.00 0.00 0.00 0.00 1.00 3.64 8.04 13.98 20697.60

0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 4.05 9.74 62511.43

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 4.51 231659.60

-0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 1.00 923697.12

Hasil Eliminasi Gauss - Jordan :

1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 -10.00

0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 989.82

0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 -23756.24

0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 240205.70

0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 -1261096.12

0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 3783328.85

0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 -6725978.96

0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 7000563.19

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 -3937843.38

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 923697.12

Solusi :

SPL memiliki SOLUSI UNIK

X1 = -10.00

X2 = 989.82

X3 = -23756.24

X4 = 240205.70

X5 = -1261096.12

X6 = 3783328.85

X7 = -6725978.96

X8 = 7000563.19

X9 = -3937843.38

X10 = 923697.12

n = 20

Masukkan jumlah n: 20

Matriks Input :

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

Hasil Eliminasi Gauss :

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

0.00 1.00 1.07 1.00 0.91 0.83 0.76 0.70 0.65 0.60 0.56 0.52 0.49 0.46 0.44 0.42 0.40 0.38 0.36 0.35 8.00

0.00 0.00 1.00 1.70 2.14 2.40 2.55 2.62 2.65 2.65 2.62 2.58 2.54 2.49 2.43 2.38 2.32 2.27 2.21 2.16 75.00

0.00 0.00 0.00 1.00 2.10 3.02 3.74 4.28 4.67 4.94 5.13 5.25 5.32 5.35 5.35 5.33 5.30 5.25 5.19 5.13 220.00

0.00 0.00 0.00 0.00 1.00 2.77 4.95 7.26 9.55 11.71 13.71 15.52 17.13 18.57 19.84 20.95 21.91 22.75 23.47 24.09 2520.00

0.00 0.00 0.00 0.00 0.00 1.00 3.25 6.55 10.61 15.10 19.80 24.51 29.12 33.53 37.70 41.60 45.22 48.55 51.60 54.38 9100.00

0.00 0.00 0.00 0.00 0.00 0.00 1.00 3.73 8.40 14.88 22.87 31.98 41.85 52.18 62.71 73.22 83.56 93.61 103.29 112.55 33633.60

0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 4.13 10.14 19.29 31.46 46.33 63.43 82.27 102.38 123.33 144.75 166.32 187.78 99282.86

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 4.60 12.35 25.38 44.25 69.06 99.53 135.13 175.16 218.91 265.62 314.57 360359.79

-0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 1.00 5.09 14.89 32.98 61.41 101.59 154.20 219.31 296.48 384.91 483.53 1409936.41

-0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 1.00 5.57 17.67 41.97 83.16 145.41 231.98 345.13 486.06 655.03 5537829.84

-0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 1.00 6.04 20.54 52.01 109.15 201.00 336.12 521.98 764.56 21408707.78

-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 6.38 22.57 59.35 129.22 246.42 425.95 682.45 65060716.21

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 -6.10 -43.12 -148.65 -379.81 -808.33 -1516.77 -637862075.66

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 5.27 16.72 40.99 85.12 157.13 60953760.19

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 1.22 -3.61 -22.58 -70.17 -402343598.56

-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 5.02 15.69 38.41 127986886.93

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 -1732.74 -8934.12 -329548984942.56

-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 5.15 189721222.51

-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 132997786.07

Hasil Eliminasi Gauss - Jordan :

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 -3.11

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 0.00 217.80

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 2687.66

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 -245214.88

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 4129245.77

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 -33011138.67

0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 148458585.60

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 -394062458.82

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 601744647.09

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 -461108641.79

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 117564343.24

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 -69887405.77

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 22038840.59

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 577617374.26

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 -985843812.46

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 642450762.38

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 -454284453.37

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 646621273.75

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 -495182380.44

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 132997786.07

Solusi :

SPL memiliki SOLUSI UNIK

X1 = -3.11

X2 = 217.80

X3 = 2687.66

X4 = -245214.88

X5 = 4129245.77

X6 = -33011138.67

X7 = 148458585.60

X8 = -394062458.82

X9 = 601744647.09

X10 = -461108641.79

X11 = 117564343.24

X12 = -69887405.77

X13 = 22038840.59

X14 = 577617374.26

X15 = -985843812.46

X16 = 642450762.38

X17 = -454284453.37

X18 = 646621273.75

X19 = -495182380.44

X20 = 132997786.07

**4.5 Contoh SPL 4**

Soal:

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.

Model Matriks:

x1 + x2 + x3 + x4 = 100.000

10x2 = 100.000

20x3 = 90.000

2.5x4 = 85.500

Dengan x1 = Keuntungan, x2 = CSR, x3 = Pajak daerah; x4 = Pajak federal

Implementasi Program:

Input jumlah persamaan : 4

Input jumlah variabel : 4

Persamaan:

1 1 1 1 100000

0 10 0 0 100000

0 0 20 0 90000

0 0 0 2.5 85500

Matriks Input :

1.00 1.00 1.00 1.00 100000.00

0.00 10.00 0.00 0.00 100000.00

0.00 0.00 20.00 0.00 90000.00

0.00 0.00 0.00 2.50 85500.00

Hasil Eliminasi Gauss :

1.00 1.00 1.00 1.00 100000.00

0.00 1.00 0.00 0.00 10000.00

0.00 0.00 1.00 0.00 4500.00

0.00 0.00 0.00 1.00 34200.00

Hasil Eliminasi Gauss - Jordan :

1.00 0.00 0.00 0.00 51300.00

0.00 1.00 0.00 0.00 10000.00

0.00 0.00 1.00 0.00 4500.00

0.00 0.00 0.00 1.00 34200.00

Solusi :

SPL memiliki SOLUSI UNIK

X1 = 51300.00

X2 = 10000.00

X3 = 4500.00

X4 = 34200.00

**4.6 Contoh SPL 5**

Soal:

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 = 5ohm, R52 =10ohm, R32 =10ohm, R65 =20ohm, R54 =15ohm, R14 = 5ohm, V1=200volt, V6 =0volt

Didapatkan matriks:

10I32 - V3 + V2 = 0

20I65 + V5 = 0

5I12 + V2 = 0

15I54 - V5 + V4 = 0

10I52 - V5 + V2 = 0

**4.7 Contoh Interpolasi 1**

Soal:

Hampiri fungsi

dengan polinom interpolasi derajat 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.

Implementasi Program:

[0,5], n=5

Fungsi yang akan dihampiri :

    f(x) =  e^(-x) / (1 + √x + x^2)

    di selang           [a , b]     | a, b  bilangan bulat positif

    dan jumlah titik       n        | n     bilangan bulat positif

Masukkan nilai a : 0

Masukkan nilai b : 5

Masukkan nilai n : 5

0.00 0.00 0.00 0.00 1.00 1.00

1.00 1.00 1.00 1.00 1.00 0.12

16.00 8.00 4.00 2.00 1.00 0.02

81.00 27.00 9.00 3.00 1.00 0.00

256.00 64.00 16.00 4.00 1.00 0.00

Persamaan Interpolasi :

P(X) = 0.03X^4 - 0.27X^3 + 1.02X^2 - 1.65X^1 + 1.00

[0,5], n=10

Fungsi yang akan dihampiri :

    f(x) =  e^(-x) / (1 + √x + x^2)

    di selang           [a , b]     | a, b  bilangan bulat positif

    dan jumlah titik       n        | n     bilangan bulat positif

Masukkan nilai a : 0

Masukkan nilai b : 5

Masukkan nilai n : 10

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 0.01 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 0.12

38.44 25.63 17.09 11.39 7.59 5.06 3.38 2.25 1.50 1.00 0.05

512.00 256.00 128.00 64.00 32.00 16.00 8.00 4.00 2.00 1.00 0.02

3814.70 1525.88 610.35 244.14 97.66 39.06 15.63 6.25 2.50 1.00 0.01

19683.00 6561.00 2187.00 729.00 243.00 81.00 27.00 9.00 3.00 1.00 0.00

78815.64 22518.75 6433.93 1838.27 525.22 150.06 42.88 12.25 3.50 1.00 0.00

262144.00 65536.00 16384.00 4096.00 1024.00 256.00 64.00 16.00 4.00 1.00 0.00

756680.64 168151.25 37366.95 8303.77 1845.28 410.06 91.13 20.25 4.50 1.00 0.00

Persamaan Interpolasi :

P(X) =  - 0.07X^7 + 0.37X^6 - 1.27X^5 + 2.83X^4 - 4.15X^3 + 4.07X^2 - 2.66X^1 + 1.00

[0,5], n=12

Fungsi yang akan dihampiri :

    f(x) =  e^(-x) / (1 + √x + x^2)

    di selang           [a , b]     | a, b  bilangan bulat positif

    dan jumlah titik       n        | n     bilangan bulat positif

Masukkan nilai a : 0

Masukkan nilai b : 5

Masukkan nilai n : 12

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 0.01 0.01 0.03 0.07 0.17 0.42 1.00 0.36

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

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

275.64 165.38 99.23 59.54 35.72 21.43 12.86 7.72 4.63 2.78 1.67 1.00 0.04

3208.83 1540.24 739.31 354.87 170.34 81.76 39.25 18.84 9.04 4.34 2.08 1.00 0.02

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

129943.57 44552.08 15275.00 5237.14 1795.59 615.63 211.07 72.37 24.81 8.51 2.92 1.00 0.00

564502.93 169350.88 50805.26 15241.58 4572.47 1371.74 411.52 123.46 37.04 11.11 3.33 1.00 0.00

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

6571678.99 1577202.96 378528.71 90846.89 21803.25 5232.78 1255.87 301.41 72.34 17.36 4.17 1.00 0.00

18749767.12 4090858.28 892550.90 194738.38 42488.37 9270.19 2022.59 441.29 96.28 21.01 4.58 1.00 0.00

Persamaan Interpolasi :

P(X) =  - 0.03X^9 + 0.19X^8 - 0.87X^7 + 2.67X^6 - 5.68X^5 + 8.35X^4 - 8.44X^3 + 5.92X^2 - 2.99X^1 + 1.00

Untuk selang [-2,2], hasil interpolasi NaN (*Not a Number*) karena terdapat bilangan negatif yang tidak dapat dioperasikan dengan akar (*sqrt*).

**4.8 Contoh Interpolasi 2**

Soal:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *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 |

Hitung hampiran dari x=0.2, x=0.55, x=0.85, x=1.28

Implementasi Program:

Masukkan jumlah titik : 7

Titik 1:

  X  = 0.1

f(X) = 0.003

Titik 2:

  X  = 0.3

f(X) = 0.067

Titik 3:

  X  = 0.5

f(X) = 0.148

Titik 4:

  X  = 0.7

f(X) = 0.248

Titik 5:

  X  = 0.9

f(X) = 0.370

Titik 6:

  X  = 1.1

f(X) = 0.518

Titik 7:

  X  = 1.3

f(X) = 0.697

Persamaan Interpolasi :

P(X) =  0.03X^4 + 0.20X^2 + 0.24X^1 - 0.02

Apakah anda ingin mencari nilai hampiran? (y/n) : y

X = 0.2

f(0.20) = 0.03

Apakah anda ingin mencari nilai hampiran? (y/n) : y

X = 0.55

f(0.55) = 0.17

Apakah anda ingin mencari nilai hampiran? (y/n) : y

X = 0.85

f(0.85) = 0.34

Apakah anda ingin mencari nilai hampiran? (y/n) : y

X = 1.28

f(1.28) = 0.68

Apakah anda ingin mencari nilai hampiran? (y/n) : n

**4.9 Contoh Interpolasi 3**

Soal:

|  |  |
| --- | --- |
| 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 |

Cari polinom dan prediksi harga pada tahun 1957, 1964, 1970, 1975

Implementasi Program:

**4.10 Contoh Interpolasi 4**

Soal:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T (°F) | 40 | 50 | 60 | 70 | 80 | 90 |
| *v* (10-5 ft2/detik) | 1.66 | 1.41 | 1.22 | 1.06 | 0.93 | 0.84 |

Implementasi Program:

Masukkan jumlah titik : 6

Titik 1:

  X  = 40

f(X) = 1.66

Titik 2:

  X  = 50

f(X) = 1.41

Titik 3:

  X  = 60

f(X) = 1.22

Titik 4:

  X  = 70

f(X) = 1.06

Titik 5:

  X  = 80

f(X) = 0.93

Titik 6:

  X  = 90

f(X) = 0.84

Persamaan Interpolasi :

P(X) =  - 0.27X^1 + 6.03

Apakah anda ingin mencari nilai hampiran? (y/n) : y

X = 62

f(62.00) = 1.19

Apakah anda ingin mencari nilai hampiran? (y/n) : y

X = 75

f(75.00) = 0.99

Apakah anda ingin mencari nilai hampiran? (y/n) : n

**BAB V**

**KESIMPULAN**

Dapat dibuat program dengan algoritma untuk menyelesaikan SPL dalam bentuk matriks, dan juga dengan interpolasi mengeluarkan hampiran nilai berdasarkan titik-titik yang diterima. Untuk penyelesaian SPL dapat digunakan metode eliminasi Gauss dan eliminasi Gauss-Jordan, dengan kaidah Operasi Baris Elementer (OBE). Sedangkan untuk interpolasi digunakan pendekatan antara titik dan hasil titik untuk mencari konstanta dalam persamaan hampiran, dan diselesaikan menggunakan eliminasi Gauss dan eliminasi Gauss-Jordan.

Dari tugas besar ini dapat disimpulkan bahwa permasalahan yang bersifat matematis dapat diselesaikan dengan bantuan komputer jika digunakan algoritma yang benar sesuai dengan permasalahan yang ingin diselesaikan dalam program tersebut.

Hasil yang didapatkan dari permasalahan ini yaitu dapat menyelesaikan berbagai permasalahan lebih lanjut, seperti penyelesaian rangkaian listrik dengan metode SPL, dan juga *modelling* prediksi dengan menggunakan data-data yang dicacah ke dalam polinom interpolasi.

Pengembangan lebih lanjut yang dapat diimplementasikan dalam program adalah dengan mencari dan mengaplikasikan algoritma yang lebih efisien untuk mengurangi memori dan waktu penyelesaian yang dibutuhkan komputer untuk menyelesaikan permasalahan khususnya dalam tugas ini.

**Daftar Pustaka**

Munir, Rinaldi. “Metode Eliminasi Gauss”, diambil 4 Oktober 2017 <http://informatika.stei.itb.ac.id/~rinaldi.munir/>