**TUGAS BESAR 1 IF2123**

**Aljabar Linear dan Geometri**

**2022/2023**



**KELOMPOK 41 - Go Yoon Jung**

**Brian Kheng 13521049**

**Farizki Kurniawan 13521082**

**Frankie Huang 13521092**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**BAB 1**

**DESKRIPSI MASALAH**

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-jordan, metode matriks balikan (), dan kaidah *Cramer* (khusus untuk SPL dengan *n* peubah dan *n* persamaan). Solusi sebuah SPL mungkin tidak ada, banyak (tidak berhingga), atau hanya satu (unik/tunggal).

Di dalam Tugas Besar 1 ini, kami membuat *library* aljabar linier dalam Bahasa Java. Library tersebut berisi fungsi-fungsi seperti eliminasi Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung determinan, kaidah *Cramer*. Selanjutnya *library* tersebut digunakan di dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi.

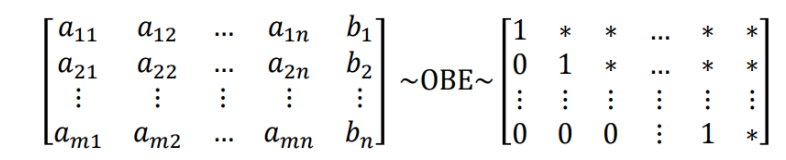
**BAB 2**

**TEORI SINGKAT**

**2.1. Sistem Persamaan Linier**

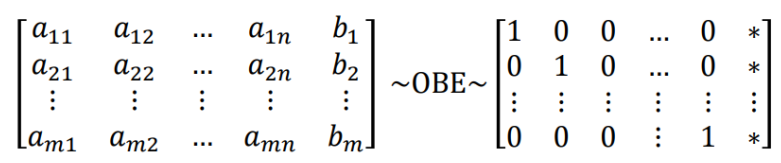
**2.1.1 Metode Eliminasi Gauss**

Metode yang dinamai oleh matematikawan Carl Friedrich Gauss (1777-1855) ini merupakan salah satu algoritma dalam penyelesaian SPL. Metode eliminasi Gauss dalam penyelesaian SPL membentuk matriks augmented menjadi matriks eselon baris dengan menggunakan operasi baris elementer. Yang nantinya akan dilakukan teknik penyulihan mundur untuk mendapatkan nilai x.



**2.1.2. Metode Eliminasi Gauss Jordan**

Metode ini merupakan variasi dari eliminasi Gauss yang dijelaskan oleh Wilhelm Jordan pada tahun 1888. Metode eliminasi Gauss Jordan membentuk matriks augmented menjadi matriks eselon baris tereduksi (elemen di atas dan di bawah 1 utama bernilai 0). Sehingga, nantinya tidak perlu dilakukan teknik penyulihan mundur untuk mendapatkan nilai x.

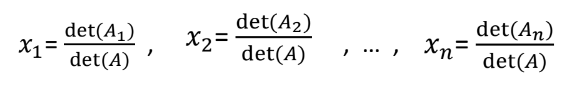
****

**2.1.3. Metode Matriks Balikan**

Metode penentuan SPL dengan menggunakan matriks balikan hanya dapat digunakan pada matriks persegi dan juga ketika determinan ≠ 0. Pada metode ini, solusi SPL didapatkan dengan x = A-1 b.

**2.1.4. Kaidah Cramer**

Menurut Kaidah Cramer, jika Ax = b adalah SPL yang terdiri dari n persamaan linier dengan n peubah (variable) sedemikian sehingga det(A) ≠ 0, maka SPL tersebut memiliki solusi yang unik, yaitu



dimana Ai adalah matriks yang diperoleh dengan mengganti entri pada kolom ke-i dari A dengan entri dari matriks b.

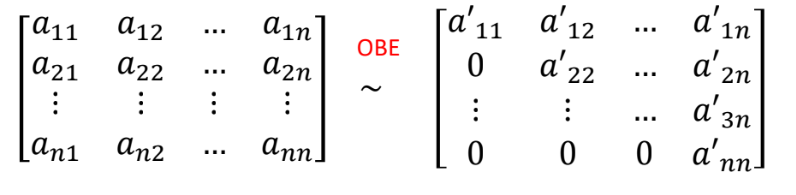
**2.2. Determinan**

Determinan adalah sebuah abstraksi yang melambangkan suatu nilai yang bisa didapatkan dari sebuah matriks persegi. Determinan dari suatu matriks persegi A umumnya dilambangkan dengan det(A). Terdapat beberapa cara untuk menentukan determinan dari suatu matriks persegi, dua diantaranya adalah dengan metode reduksi baris dan dengan metode ekspansi kofaktor.

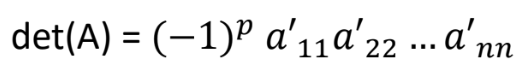
**2.2.1. Metode Reduksi Baris**

Determinan matriks A dapat diperoleh dengan melakukan OBE pada matriks A sampai diperoleh matriks segitiga (segitiga bawah atau atas).





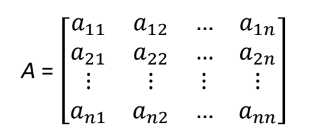
Sehingga didapatkan



dimana p menyatakan banyaknya operasi pertukaran baris di dalam OBE.

**2.2.2. Metode Ekspansi Kofaktor**

Didefinisikan sebuah matriks persegi A sebagai berikut

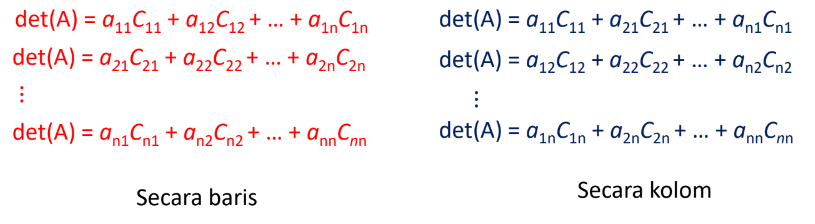


Definisikan notasi Mij sebagai minor dari entri aij, yaitu determinan dari sub-matriks yang elemen-elemennya adalah elemen matriks A yang tidak berada pada baris i dan kolom j.

Lalu didefinisikan juga Cij sebagai kofaktor dari entri aij, yaitu



Maka, determinan dari matriks A dapat ditentukan dengan salah satu dari persamaan berikut.

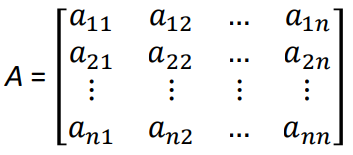


**2.3. Balikan Matriks**

Invers dari suatu matriks adalah suatu matriks yang jika dikalikan dengan matriks semula akan menghasilkan matriks identitas. Agar suatu matriks dapat memiliki determinan, harus ada 2 syarat yang harus dipenuhi, yaitu determinan matriks tersebut tidak boleh nol dan harus berbentuk persegi. Terdapat beberapa cara untuk mencari balikan matriks, dua diantaranya adalah dengan menggunakan matriks adjoin dan eliminasi gauss jordan.

**2.3.1. Metode Matriks Adjoin**

Didefinisikan suatu matriks sebagai berikut



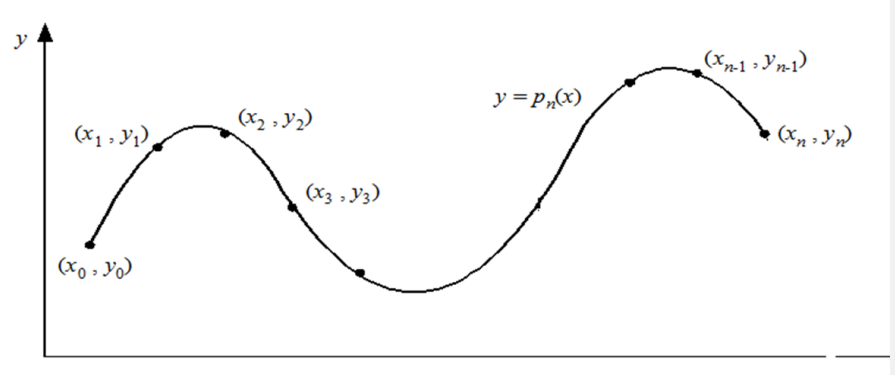
Maka entri kofaktor pada baris *i* dan kolom *j* adalah -1 pangkat i+j dikali dengan determinan dari matriks minor pada baris *i* dan kolom *j*, yaitu matriks semula yang tidak memiliki elemen pada baris *i* dan kolom *j*. Setelah kofaktor dari suatu matriks didapatkan, balikan matriks dapat dicari dengan mentranspose matriks adjoin dan mengalikan tiap elemen matriks tersebut dengan determinan matriks semula.

**2.3.2. Metode Eliminasi Gauss Jordan**

Didefinisikan suatu matriks *A*, balikan dari matriks tersebut dapat dicari dengan melakukan eliminasi Gauss Jordan pada matriks *A* yang telah diaugmented dengan matriks identitas.

**2.4. Interpolasi Polinom**

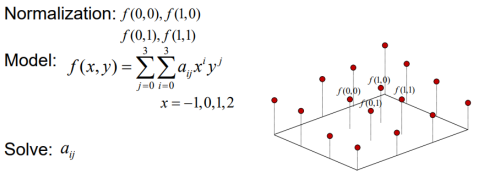
Interpolasi polinom merupakan teknik interpolasi dengan mengasumsikan pola data yang kita miliki mengikuti pola polinomial baik berderajat satu (linier) maupun berderajat tinggi. Interpolasi dengan metode ini dilakukan dengan terlebih dahulu membentuk persamaan polinomial Pn(x) dari n+1 buah titik berbeda, (x0, y0), (x1, y1), …, (xn, yn) sehingga yi = Pn(xi) untuk i = 0, 1, 2, …, n. Sehingga, kita dapat menggunakan persamaan polinomial tersebut untuk menghitung perkiraan nilai y di x sembarang.



**2.5. Bicubic Interpolation**

Bicubic interpolation merupakan teknik interpolasi pada data 2 dimensi yang merupakan pengembangan dari interpolasi cubic. Interpolasi ini dilakukan dengan mengambil data-data yang sudah diketahui untuk memprediksi nilai baru yang tidak diketahui sebelumnya.

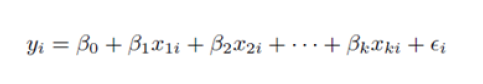
Semisal untuk mencari nilai (x,y) yang berada pada rentang (0,0), (0,1), (1,0), hingga (1,1); kita akan memerlukan 16 titik yang sama atau bersebelahan dengan range titik yang ingin dicari. Titik-titik tersebut kemudian diplot dalam suatu fungsi berikut



**2.6. Regresi Linear Berganda**

Regresi linear adalah metode untuk memprediksi nilai dari suatu variabel dependen jika diberikan satu atau lebih variabel independen. Hal ini dilakukan dengan memprediksi persamaan yang berlaku menggunakan data-data yang ada hingga membentuk sebuah persamaan linear.

Rumus yang umum digunakan untuk regresi linear berganda adalah sebagai berikut



dimana

y = variabel dependen yang akan ditentukan nilainya

β = koefisien regresi

x = variabel independen

**BAB 3**

**IMPLEMENTASI PUSTAKA DAN PROGRAM DALAM JAVA**

1. **Folder lib (berisi pustaka yang digunakan dalam program utama)**

**1.1. Matrix.java**

Class ini digunakan untuk mengimplementasikan matriks yang akan digunakan di program utama.

* **Atribut**

| **Atribut** | **Deskripsi** |
| --- | --- |
| private int row | Jumlah baris yang digunakan pada matriks berupa integer |
| private int col | Jumlah kolom yang digunakan pada matriks berupa integer |
| private double[][] Matrix | Elemen-elemen dari matriks berupa double |

* **Konstruktor**

| **Konstruktor** | **Deskripsi** |
| --- | --- |
| public Matrix(int row, int col) | Dibuat matriks dengan mengisi atribut row = row dan mengisi atribut col = col, dan menginisiasi array static berukuran row x col |

* **Method**

| **Method** | **Deskripsi** |
| --- | --- |
| public double getElmt(int i, int j) | Mengembalikan nilai elemen pada indeks posisi (i, j) |
| public int getRowEff() | Mengembalikan jumlah baris matriks |
| public int getColEff() | Mengembalikan jumlah kolom matriks |
| public void setElmt(int i, int j, double x) | Mengubah nilai elemen pada indeks posisi (i, j) dengan nilai x |
| public static Matrix inputMatrix() | Melakukan prosedur input matriks |
| public void printMatrix() | Melakukan prosedur output matriks |

**1.2. SPL.java**

Class ini digunakan untuk menyelesaikan suatu persoalan sistem persamaan linear.

* **Atribut**

| **Atribut** | **Deskripsi** |
| --- | --- |
| private double[] x | Nilai x hasil melakukan penyelesaian SPL dengan metode yang ada (solusi tunggal) |
| private String[] ans | Output dari hasil penyelesaian SPL (solusi tunggal, banyak, atau tidak ada) |
| private Integer nEff | Jumlah elemen pada array ans |

* **Konstruktor**

| **Konstruktor** | **Deskripsi** |
| --- | --- |
| public SPL() | Melakukan inisiasi nilai pada atribut yang ada |

* **Method**

| **Method** | **Deskripsi** |
| --- | --- |
| public static void DriverSPL () | Prosedur untuk menyelesaikan persoalan SPL, berisi input matriks dan metode yang ingin digunakan |
| public void Gauss(Matrix M) | Penyelesaian SPL dengan metode Gauss |
| public void GaussJordan (Matrix M) | Penyelesaian SPL dengan metode Gauss Jordan |
| public void InversMatrix (Matrix M) | Penyelesaian SPL dengan metode invers matriks |
| public void Cramer(Matrix M) | Penyelesaian SPL dengan kaidah cramer |
| public Matrix EselonBaris (Matrix M) | Mengubah bentuk matriks semula dengan OBE menghasilkan matriks eselon baris |
| public Matrix EselonBarisTereduksi (Matrix M) | Mengubah bentuk matriks semula dengan OBE menghasilkan matriks eselon baris tereduksi |
| public void SolveManySolution (Matrix M) | Prosedur untuk menyelesaikan SPL dengan solusi banyak |

**1.3. Determinant.java**

Class ini digunakan untuk menyelesaikan suatu persoalan pencarian determinan.

* **Atribut**

-

* **Konstruktor**

-

* **Method**

| **Method** | **Deskripsi** |
| --- | --- |
| public static void DriverDeterminan() | Prosedur untuk menyelesaikan persoalan SPL, input berupa matriks persegi dan metode yang ingin digunakan. Output berupa determinan matriks. |
| public static double DetOBE(Matrix M) | Pencarian determinan dengan metode reduksi baris. |
| public static double DetCofactor(Matrix M) | Pencarian determinan dengan metode ekspansi kofaktor. |

**1.4. Balikan.java**

Class ini digunakan untuk mencari invers dari suatu matriks.

* **Atribut**

-

* **Konstruktor**

| **Method** | **Deskripsi** |
| --- | --- |
| public static boolean isInversExist | Nilai kebenaran apakah invers suatu matriks input ada |

* **Method**

| **Method** | **Deskripsi** |
| --- | --- |
| public static void DriverBalikan() | Prosedur untuk mencari invers dari suatu matrix.  Input berupa matriks dan metode yang ingin digunakan.  Output berupa invers dari matrix. |
| public static Matrix swapRow(Matrix matrix, int n, int m) | Fungsi untuk menukar row n dan m pada matrix. |
| public static Matrix Adjoin(Matrix matrix) | Fungsi untuk mencari adjoin dari matrix. |
| public static Matrix BalikanAdjoin(Matrix matrix) | Fungsi untuk mencari invers dari matrix menggunakan matrix adjoin. |
| public static Matrix BalikanGaussJordan(Matrix matrix) | Fungsi untuk mencari invers dari matrix menggunakan Gauss Jordan. |

**1.5. Kofaktor.java**

* **Atribut**

-

* **Konstruktor**

-

* **Method**

| **Method** | **Deskripsi** |
| --- | --- |
| public static double findDetKofaktor(Matrix matrixKofaktor, int row, int col) | Fungsi untuk mencari nilai determinan kofaktor pada indeks (row, col) |
| public static Matrix Kofaktor(Matrix matrix) | Fungsi untuk mencari kofaktor dari matrix. |

1. **Interpolate.java**

Class ini digunakan untuk menyelesaikan persoalan interpolasi polinom.

* **Atribut**

-

* **Konstruktor**

-

* **Method**

| **Method** | **Deskripsi** |
| --- | --- |
| public static void SolveInterpolate() | Prosedur untuk menyelesaikan persoalan interpolasi polinom, berisi input titik-titik dan nilai x yang ingin diperkirakan nilainya dan output berupa fungsi f(x) beserta nilainya |

1. **Regresi.java**

Class ini digunakan untuk menyelesaikan persoalan regresi linear berganda.

* **Atribut**

-

* **Konstruktor**

-

* **Method**

| **Method** | **Deskripsi** |
| --- | --- |
| public static void SolveRegression() | Prosedur untuk menyelesaikan persoalan regresi linear berganda. Input berupa titik-titik x1,x2..,xk,y dan nilai x1,x2,..,xk yang ingin diperkirakan hasilnya. Output berupa fungsi y beserta nilai hampirannya. |

1. **Bicubic.java**

Class ini digunakan untuk menyelesaikan persoalan bicubic interpolation.

* **Atribut**

-

* **Konstruktor**

-

* **Method**

| **Method** | **Deskripsi** |
| --- | --- |
| public static void SolveBicubic() | Prosedur untuk menyelesaikan persoalan bicubic interpolation.  Input berupa masukan dari file yang berisi titik-titik f(-1,-1), f(-1,0), f(-1,1), dst. dan nilai a dan b, yaitu titik yang nilainya ingin dicari.  Output berupa fungsi f(a,b) beserta nilai hampirannya. |
| public static double getValue(Matrix matrix, double a, double b) | Fungsi untuk mencari nilai hasil bicubic interpolation. |

1. **ImageProcessing.java**

Class ini digunakan untuk memperbesar citra dengan bicubic interpolation.

* **Atribut**

| **Method** | **Deskripsi** |
| --- | --- |
| public static int[][] columnExtendMatrix | Array 2d hasil pembesaran lebar. |
| public static int[][] columnInterpolateMatrix | Array 2d hasil interpolasi columnExtendMatrix. |
| public static int[][] rowExtendMatrix | Array 2d hasil pembesaran tinggi. |
| public static int[][] rowInterpolateMatrix | Array 2d hasil interpolasi rowExtendMatrix. |
| public static int[][] interpolateMatrix | Array 2d hasil interpolasi salah satu dimensi. |
| public static Matrix xValueMatrix | Matrix berisi nilai konstanta yang digunakan pada double cubic interpolation. |
| public static double[][] valueXandY | Array 2d berisi nilai konstanta yang digunakan pada bicubic interpolation. |

* **Konstruktor**

-

* **Method**

| **Method** | **Deskripsi** |
| --- | --- |
| public static void ImageProcessingDriver() | Prosedur untuk memperbesar citra dengan menggunakan bicubic interpolation atau double cubic interpolation.  Input berupa masukan nama file pada folder test/image/, nama file hasil output image, faktor pembesaran lebar, dan faktor pembesaran tinggi.  Output berupa file dengan nama yang telah diinput. |
| public static void bicubic(String readDir, String writeDir, int heightFactor, int widthFactor) | Prosedur untuk memperbesar citra dengan menggunakan bicubic interpolation. |
| public static double bicubicInterpolation(int[] zValue, double a, double b) | Fungsi untuk menghitung nilai hasil bicubic interpolation. |
| public static int checkValue(double val) | Fungsi untuk mengecek apakah nilai val melebihi 0xff atau kurang dari 0x0. |
| public static void doubleCubic(String readDir, String writeDir, int heightFactor, int widthFactor) | Prosedur untuk memperbesar citra dengan menggunakan cubic interpolation dua kali. |
| public static int[][] interpolatePoints(int width, int height, int factor, boolean interpolateHeight) | Fungsi untuk menghitung nilai elemen matriks hasil perbesaran interpolasi cubic. |
| public static int getValue(int index, double x, boolean interpolateHeight) | Fungsi untuk menghitung nilai ARGB hasil interpolasi. |
| public static int RGBValue(int[] ordinate, double x) | Fungsi untuk menghitung nilai pada titik f(x). |

1. **Main.java**

Class ini digunakan sebagai penyatu semua program.

* **Atribut**

-

* **Konstruktor**

-

* **Method**

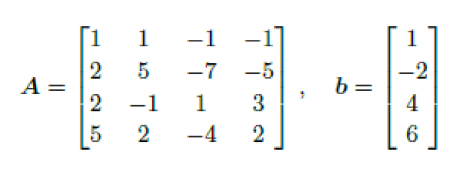
| **Method** | **Deskripsi** |
| --- | --- |
| public static void main(String[] args) | Prosedur utama untuk menjalankan program. |

**BAB 4**

**EKSPERIMEN**

4.1. Temukan solusi SPL *Ax = b*, berikut:

1. Test case 1



* Metode Gauss:



* Metode Gauss Jordan:



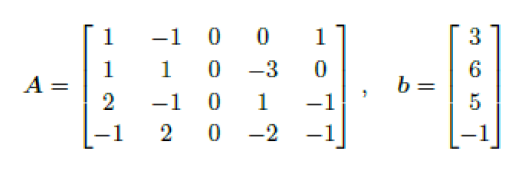
* Metode Matriks Balikan:



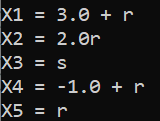
* Kaidah Cramer:



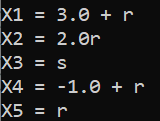
1. Test case 2



* Metode Gauss:



* Metode Gauss Jordan:



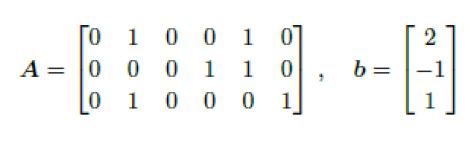
* Metode Matriks Balikan:



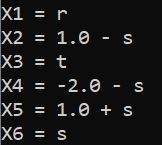
* Kaidah Cramer:



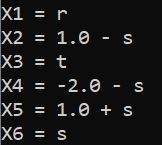
1. Test case 3



* Metode Gauss:



* Metode Gauss Jordan:



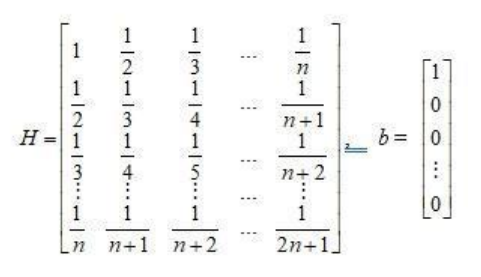
* Metode Matriks Balikan:



* Kaidah Cramer:

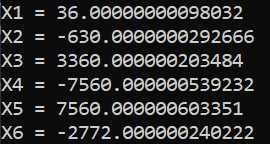


1. Test case 4

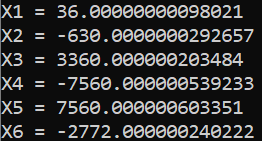


Kasus n = 6:

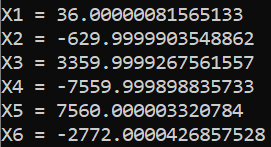
* Metode Gauss:



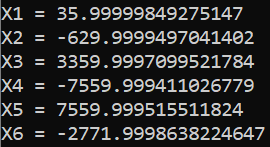
* Metode Gauss Jordan:



* Metode Matriks Balikan:



* Kaidah Cramer:

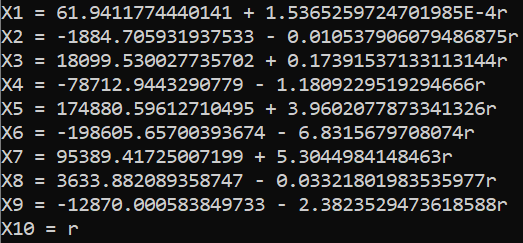


Kasus n = 10:

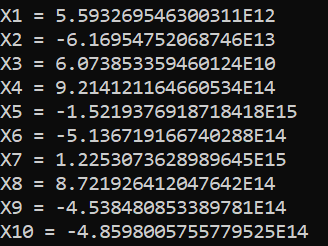
* Metode Gauss:



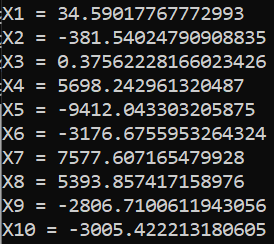
* Metode Gauss Jordan:



* Metode Matriks Balikan:

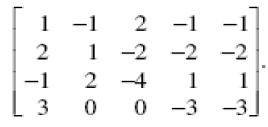


* Kaidah Cramer:

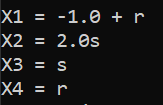


4.2. SPL berbentuk matriks augmented

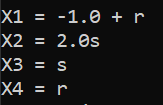
1. Test case 1



* Metode Gauss:



* Metode Gauss Jordan:



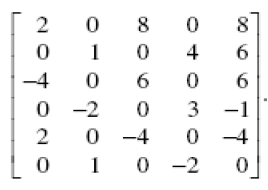
* Metode Matriks Balikan:



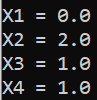
* Kaidah Cramer:



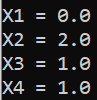
1. Test case 2



* Metode Gauss:



* Metode Gauss Jordan:



* Metode Matriks Balikan:

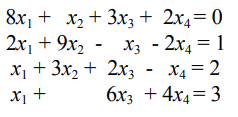


* Kaidah Cramer:

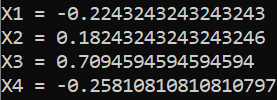


4.3. SPL berbentuk

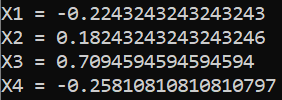
1. Test case 1



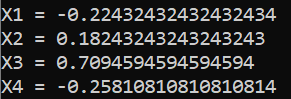
* Metode Gauss:



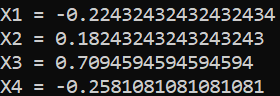
* Metode Gauss Jordan:



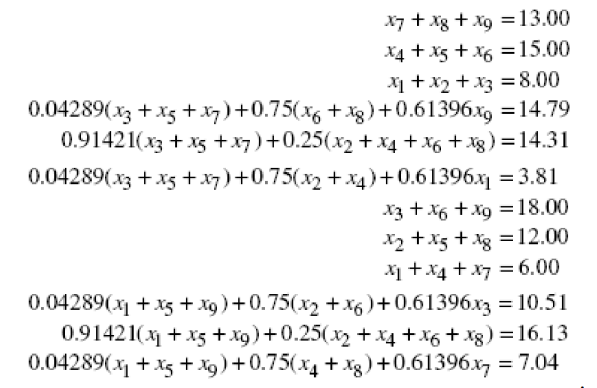
* Metode Matriks Balikan:



* Kaidah Cramer:



1. Test case 2



* Metode Gauss:



* Metode Gauss Jordan:



* Metode Matriks Balikan:

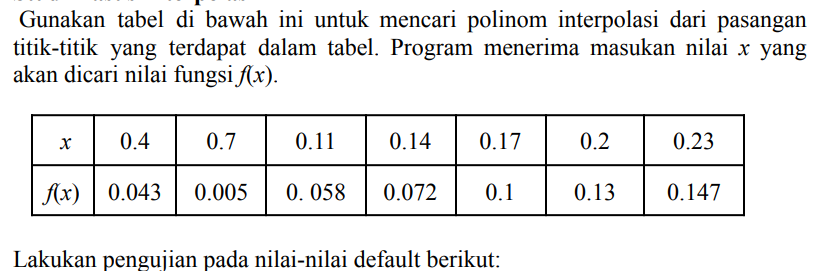


* Kaidah Cramer:



4.4. Studi Kasus Interpolasi

1. Test case 1



* x = 0.2, f(x) = ? 
* x = 0.55, f(x) = ?



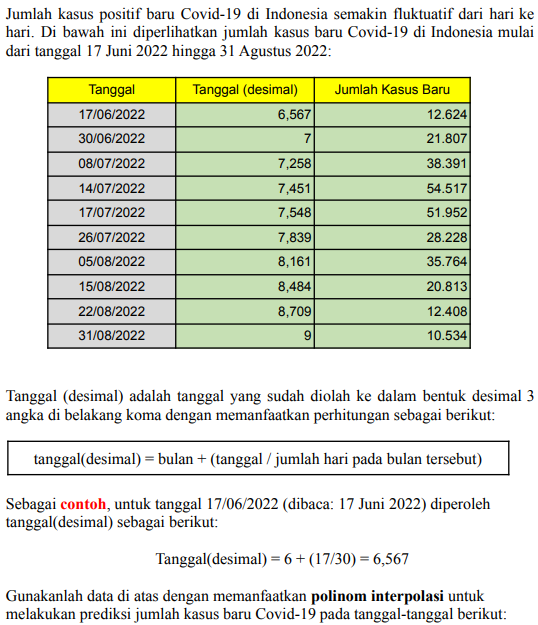
* x = 0.85, f(x) = ?



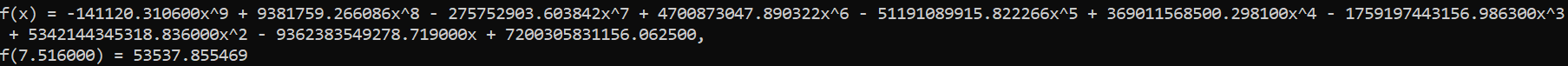
* x = 1.28, f(x) = ?



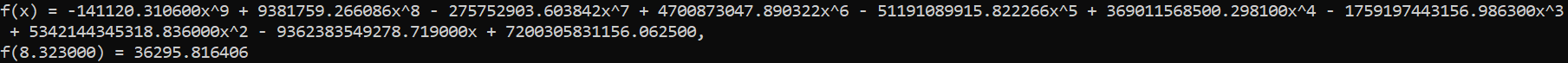
1. Test case 2



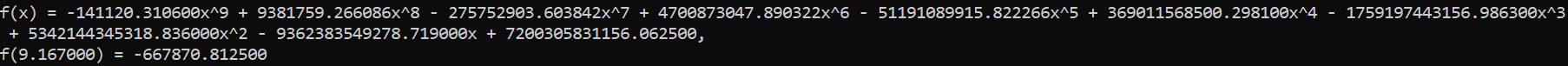
* 16/07/2022 (7.516)



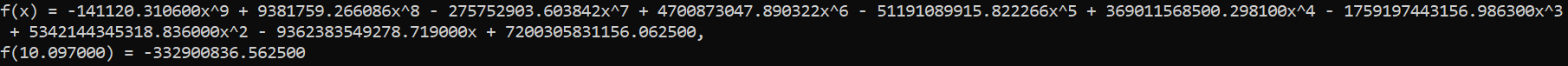
* 10/08/2022 (8.323)



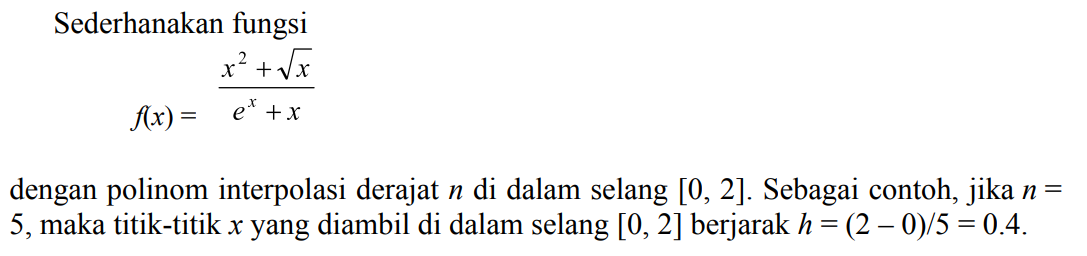
* 05/09/2022 (9.167)



* 03/10/2022 (10.097)



1. Test case 3



* Untuk n = 5, titik-titik x yang diambil = {0, 0.4, 0.8, 1.2, 1.6, 2.0}



4.5 Studi Kasus Interpolasi Bicubic

1. TC-Bicubic-1.txt



1. TC-Bicubic-2.txt



1. TC-Bicubic-3.txt



1. TC-Bicubic-4.txt



4.6 Studi Kasus Pembesaran Citra dengan Interpolasi Bicubic

1. TC-ImageProcessing-1.jpg

Faktor pembesaran : 2

Input :



Output :



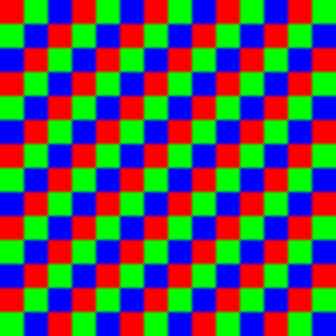
1. TC-ImageProcessing-2.png

Faktor pembesaran : 4

Input :



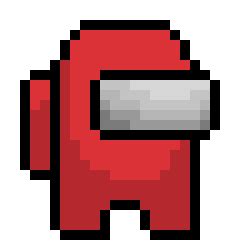
Output :



1. TC-ImageProcessing-3.jpg

Faktor pembesaran : 3

Input :



Output :



1. TC-ImageProcessing-4.png

Faktor pembesaran : 64

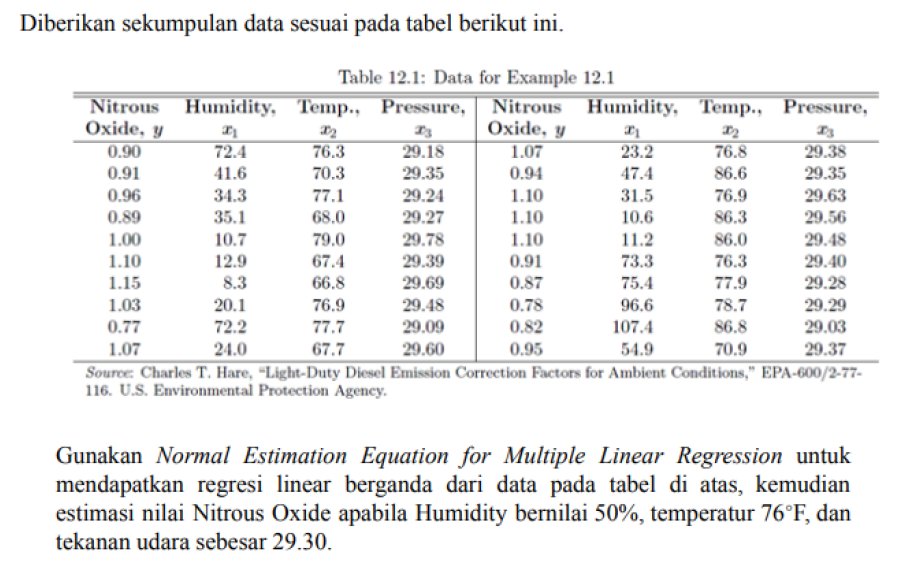
Input :



Output :



4.7 Studi Kasus Linear Berganda



Persamaan yang didapatkan:

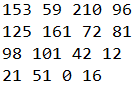


Hampiran nilai y dari data yang diinginkan:



4.8. Studi kasus pencarian determinan

1. Test case 1



-Metode reduksi baris



-Metode ekspansi kofaktor



1. Test case 2



-Metode reduksi baris



-Metode ekspansi kofaktor



**BAB 5**

**KESIMPULAN**

**5.1. Kesimpulan**

Terdapat berbagai metode dalam penyelesaian SPL, pencarian determinan, dan penentuan matriks balikan. Cara penyelesaian SPL di antaranya adalah menggunakan metode eliminasi Gauss, metode Gauss-Jordan, metode matriks balikan, dan kaidah Cramer. Cara pencarian determinan di antaranya adalah menggunakan metode reduksi baris dan metode ekspansi kofaktor.

Basis-basis di atas ternyata dapat menyelesaikan berbagai masalah dengan pengaplikasian yang sesuai. Beberapa di antara pengaplikasiannya adalah ekspansi bikubik, interpolasi polinom, dan juga regresi linear berganda.

Pada tugas ini, kami mengimplementasi sebuah program yang dapat mengimplementasi metode-metode dan pengaplikasiannya yang telah disebutkan di atas. Program ini diimplementasi dalam bahasa Java dan dapat menyelesaikan berbagai permasalahan yang melibatkan matriks, seperti penggunaan algoritma ekspansi bikubik untuk memperbesar ukuran gambar.

**5.2. Saran**

Beberapa saran pengembangan dari program ini adalah sebagai berikut:

* Penggunaan *rounding* untuk menghasilkan hasil kalkulasi yang lebih akurat.
* Penggunaan metode pencarian determinan yang lebih beragam.
* Penjabaran *step-by-step* dari metode yang digunakan pada penyelesaian SPL atau pencarian determinan sebagai output apabila diminta.

**5.3. Refleksi**

Pengerjaan tubes ini tergolong cukup lancar karena progresinya yang cukup konstan per harinya dan pembagian tugas yang sudah cukup jelas. Komunikasi yang konstan juga menjamin pengerjaan tubes tidak stagnan. Selain itu, dibutuhkan ketelitian lebih dalam membaca instruksi yang telah diberikan agar tidak perlu banyak *adjustment* pada program yang telah dibuat.

**DAFTAR PUSTAKA**

Informatika.stei.itb.ac.id. (2022). Determinan (Bagian 1). Diakses pada 28 September 2022, dari [https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Alg](https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-08-Determinan-bagian1.pdf)

[eo-08-Determinan-bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-08-Determinan-bagian1.pdf)

Informatika.stei.itb.ac.id. (2022). Determinan (Bagian 2). Diakses pada 28 September 2022, dari [https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Alg](https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-09-Determinan-bagian2.pdf)

[eo-09-Determinan-bagian2.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-09-Determinan-bagian2.pdf)

Informatika.stei.itb.ac.id. (2022). Sistem persamaan linier (Bagian 1: Metode eliminasi Gauss). Diakses pada 20 September 2022, dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-03-Sistem-Persamaan-Linier.pdf>

Informatika.stei.itb.ac.id. (2022). Sistem persamaan linier (Bagian 2: Metode eliminasi Gauss-Jordan). Diakses pada 22 September 2022, dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-05-Sistem-Persamaan-Linier-2.pdf>

Link repository: <https://github.com/briankheng/Algeo01-21049>