IF2123 ALJABAR LINIER DAN GEOMETRI
Sistem Persamaan Linier, Determinan, dan Aplikasinya
Semester I Tahun 2024/2025

Dipersiapkan oleh:

Kelompok Mboten Ngertos

Bertha Soliany Frandi / 13523026

Ahmad Ibrahim / 13523089

Michael Alexander Angkawijaya / 13523102

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
JL. GANESA 10, BANDUNG 40132

2024

Daftar Isi

BAB 1
DESKRIPSI MASALAH

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Andstrea sudah mempelajari berbagai metode untuk menyelesaikan SPL, termasuk menghitung determinan matriks. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan ($x = A^{-1} b$), 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 Linear

Sistem Persamaan Linear (SPL) mempunyai *m* buah persamaan dan *n* variabel yang dapat dinyatakan dalam bentuk persamaan, matriks perkalian $Ax=b$, atau matriks *augmented*.

$$\begin{aligned} x_1 + 3x_2 - 6x_3 &= 9 \\ 2x_1 - 6x_2 + 4x_3 &= 7 \\ 5x_1 + 2x_2 - 5x_3 &= -2 \end{aligned} \quad \Longrightarrow \quad \begin{bmatrix} 1 & 3 & -6 & 9 \\ 2 & -6 & 4 & 7 \\ 5 & 2 & -5 & -2 \end{bmatrix}$$

Gambar 1 Matriks *Augmented*

Terdapat tiga Operasi Baris Elementer (OBE) yang bisa diterapkan pada matriks *augmented*. Ketiga operasi tersebut dilakukan untuk mengubah bentuk matriks menjadi matriks eselon baris atau matriks eselon baris tereduksi.

Matriks eselon baris adalah matriks yang memiliki 1 utama pada setiap baris, kecuali baris yang seluruhnya bernilai 0. Perbedaannya dengan matriks eselon baris tereduksi adalah elemen di atas dan di bawah 1 utama bernilai 0 yang dimana pada matriks eselon baris hanya elemen di bawah 1 utama saja yang bernilai 0. Berikut adalah perbandingan dari matriks eselon baris (gambar sebelah kiri) dengan matriks eselon baris tereduksi (gambar sebelah kanan).

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

Gambar 2 Perbandingan Matriks Eselon

2.1.1 Metode Eliminasi Gauss

Metode eliminasi Gauss adalah salah satu metode untuk mendapatkan solusi SPL. Metode dilakukan dengan mengubah SPL dalam bentuk matriks *augmented* menjadi matriks eselon baris dengan menggunakan OBE. Setelah matriks eselon baris terbentuk, dilakukan penyelesaian persamaan yang berkoresponden pada matriks eselon baris dengan penyulihan mundur untuk mendapatkan nilai x.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_n \end{bmatrix} \sim\text{OBE}\sim \begin{bmatrix} 1 & * & * & \dots & * & * \\ 0 & 1 & * & \dots & * & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \vdots & 1 & * \end{bmatrix}$$

Gambar 3 Metode Eliminasi Gauss

2.1.2 Metode Eliminasi Gauss-Jordan

Metode eliminasi Gauss-Jordan adalah pengembangan dari metode eliminasi Gauss. Dengan menggunakan OBE, dihasilkan matriks eselon baris tereduksi. Pada metode eliminasi Gauss-Jordan, tidak perlu melakukan penyulihan mundur untuk memperoleh solusi SPL dikarenakan solusi langsung didapatkan dari matriks *augmented* akhir.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_m \end{bmatrix} \sim\text{OBE}\sim \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & * \\ 0 & 1 & 0 & \dots & 0 & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \vdots & 1 & * \end{bmatrix}$$

Gambar 4 Metode Eliminasi Gauss-Jordan

2.1.3 Metode Matriks Balikan

Metode matriks balikan dilakukan dengan melakukan metode eliminasi Gauss-Jordan. Metode ini hanya bisa diterapkan pada matriks persegi n x n dan matriks dengan determinan $\neq 0$. Penyelesaian SPL didapatkan dengan melakukan perkalian $x=A^{-1}b$.

$$[A|I] \overset{\text{G-J}}{\sim} [I|A^{-1}]$$

Gambar 5 Metode Matriks Balikan

2.1.4 Kaidah *Cramer*

Kaidah *Cramer* mengatakan jika $Ax=b$ adalah SPL yang terdiri dari n buah persamaan dan n variabel dan determinan matriks A $\neq 0$, maka SPL tersebut memiliki solusi yang unik sebagai berikut.

$$x_1 = \frac{\det(A_1)}{\det(A)}, \quad x_2 = \frac{\det(A_2)}{\det(A)}, \quad \dots, \quad x_n = \frac{\det(A_n)}{\det(A)}$$

Gambar 6 Kaidah *Cramer*

A_j adalah matriks yang diperoleh dengan mengganti kolom ke-j dari A menjadi matriks b pada $Ax=b$.

2.2 Determinan

Determinan adalah sebuah nilai yang didapatkan dari sebuah matriks persegi dan dilambangkan sebagai det(A). Jika suatu matriks diperlakukan OBE, maka determinan matriks tersebut terpengaruh dengan operasi yang dilakukan (aturan determinan). Determinan bisa didapat dari menghasilkan matriks segitiga atau dengan kofaktor.

2.2.1 Metode Reduksi Baris

Metode reduksi baris adalah metode untuk mendapatkan determinan suatu matriks. Metode ini dilakukan dengan melakukan OBE pada suatu matriks persegi hingga diperoleh matriks segitiga. Berikut adalah contoh untuk matriks segitiga bawah.

$$[A] \overset{\text{OBE}}{\sim} [\text{matriks segitiga bawah}]$$

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \overset{\text{OBE}}{\sim} \begin{bmatrix} a'_{11} & a'_{12} & \dots & a'_{1n} \\ 0 & a'_{22} & \dots & a'_{2n} \\ \vdots & \vdots & \dots & a'_{3n} \\ 0 & 0 & 0 & a'_{nn} \end{bmatrix}$$

Gambar 7 Metode Reduksi Baris

Sehingga det(A) adalah

$$\det(A) = \frac{(-1)^p \, a'_{11} \, a'_{22} \dots a'_{nn}}{k_1 k_2 \dots k_m}$$

Gambar 8 Determinan Matriks Segitiga

dengan p menyatakan banyaknya operasi pertukaran baris di dalam OBE dan k adalah pengali dalam operasi perkalian baris dalam OBE.

2.2.2 Metode Ekspansi Kofaktor

Metode ekspansi kofaktor adalah metode untuk mencari determinan matriks persegi dengan menggunakan minor entri dan kofaktor. Pada matriks segitiga, didefinisikan M_{ij} adalah minor entri a_{ij} atau determinan submatriks yang elemennya tidak berada pada baris i dan kolom j. Didefinisikan pula kofaktor entri a_{ij}, yaitu $C_{ij} = (-1)^{i+j}M_{ij}$. Dengan kedua hal tersebut, determinan matriks dapat dihitung dengan

$$\begin{aligned} \det(A) &= a_{11}C_{11} + a_{12}C_{12} + \dots + a_{1n}C_{1n} \\ \det(A) &= a_{21}C_{21} + a_{22}C_{22} + \dots + a_{2n}C_{2n} \\ &\vdots \\ \det(A) &= a_{n1}C_{n1} + a_{n2}C_{n2} + \dots + a_{nn}C_{nn} \end{aligned}$$

Secara baris

$$\begin{aligned} \det(A) &= a_{11}C_{11} + a_{21}C_{21} + \dots + a_{n1}C_{n1} \\ \det(A) &= a_{12}C_{12} + a_{22}C_{22} + \dots + a_{n2}C_{n2} \\ &\vdots \\ \det(A) &= a_{1n}C_{1n} + a_{2n}C_{2n} + \dots + a_{nn}C_{nn} \end{aligned}$$

Secara kolom

Gambar 9 Determinan Metode Ekspansi Kofaktor

2.3 Matriks Balikan

Matriks balikan adalah matriks persegi sedemikian rupa sehingga sebuah $AA^{-1} = A^{-1}A = I$, dimana I adalah matriks identitas. Syarat sebuah matriks mempunyai matriks balikan adalah determinan tidak bernilai 0. Matriks identitas bisa diperoleh dari dua metode, yaitu dengan menggunakan matriks adjoin dan OBE.

2.3.1 Metode Matriks Adjoin

Matriks adjoin adalah matriks yang diperoleh dari hasil transpose sebuah matriks kofaktor. Definisi matriks kofaktor sendiri adalah matriks yang berisi kofaktor entri a_{ij} (C_{ij}). Sedangkan definisi transpose adalah matriks yang diperoleh dengan menukar elemen baris menjadi kolom dan berlaku sebaliknya.

$$\begin{bmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ C_{n1} & C_{n2} & \dots & C_{nn} \end{bmatrix}$$

Gambar 10 Matriks Kofaktor

Dengan adjoin, matriks balikan dihitung dengan rumus sebagai berikut.

$$A^{-1} = \frac{1}{\det(A)} \operatorname{adj}(A)$$

Gambar 11 Matriks Balikan dengan Adjoin

2.3.2 Metode OBE

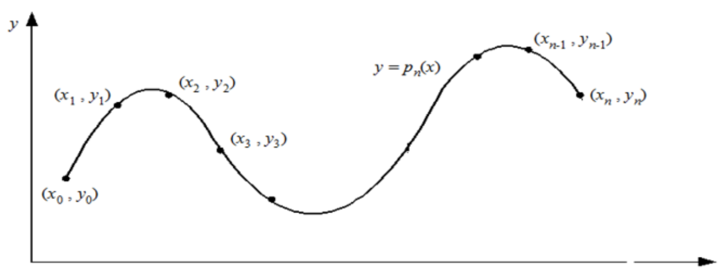
Untuk sebuah matriks A, dapat diperoleh matriks balikannya dengan menggunakan OBE. OBE diterapkan pada suatu matriks *augmented* antara matriks A dengan matriks identitas (I) hingga terbentuk matriks eselon baris tereduksi.

$$[A|I] \overset{\text{G-J}}{\sim} [I|A^{-1}]$$

Gambar 12 Ilustrasi Matriks *Augmented* Gabungan

2.4 Interpolasi Polinom

Interpolasi polinom adalah teknik interpolasi dengan mengasumsikan pola data yang dimiliki mengikuti pola polinomial baik berderajat satu (linear) maupun berderajat tinggi. Dibentuk terlebih dahulu persamaan polinomial sehingga persamaan tersebut dapat digunakan untuk menghitung nilai y di x sembarang.

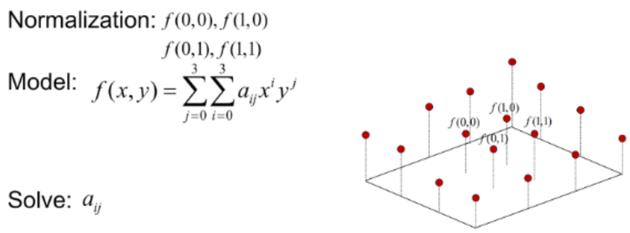


Gambar 13 Ilustrasi Interpolasi Polinom

2.5 Interpolasi *Bicubic Spline*

Bicubic spline interpolation adalah metode interpolasi yang digunakan untuk mengaproksimasi fungsi di antara titik-titik data yang diketahui. Bicubic spline interpolation melibatkan konsep spline dan konstruksi serangkaian polinomial kubik di dalam setiap sel segi empat dari data yang diberikan. Pendekatan ini menciptakan permukaan yang halus dan kontinu, memungkinkan untuk perluasan data secara visual yang lebih akurat daripada metode interpolasi linear.

Dalam pemrosesan menggunakan interpolasi bicubic spline digunakan 16 buah titik, 4 titik referensi utama di bagian pusat, dan 12 titik di sekitarnya sebagai aproksimasi turunan dari keempat titik referensi untuk membagun permukaan bikubik. Bentuk pemodelannya adalah sebagai berikut.



Gambar 14 Pemodelan Interpolasi *Bicubic Spline*

2.6 Regresi Berganda

Regresi adalah metode prediksi nilai hubungan antara satu variabel dependen dengan serangkaian variabel lain atau independen. Terdapat 2 jenis regresi, yaitu regresi linier berganda dan regresi kuadratik berganda.

2.6.1 Regresi Linier Berganda

Rumus untuk regresi linier berganda adalah sebagai berikut dimana y adalah variabel dependen, β adalah koefisien regresi, dan x adalah variabel independen.

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki} + \epsilon_i$$

Gambar 15 Rumus Regresi Linier Berganda

Dengan menggunakan *Normal Estimation Equation for Multiple Linear Regression*, didapatkan nilai dari setiap β.

2.6.2 Regresi Kuadratik Berganda

Terdapat 3 bentuk persamaan dari regresi kuadratik, yaitu variabel linier, variabel kuadrat, dan variabel interaksi. Setiap n-peubah, jumlah variabelnya akan berbeda-beda. Semakin banyak peubah *n*, semakin besar ukuran matriks dibandingkan dengan regresi linier berganda.

$$\begin{pmatrix} N & \sum u_i & \sum v_i & \sum u_i^2 & \sum u_i v_i & \sum v_i^2 \\ \sum u_i & \sum u_i^2 & \sum u_i v_i & \sum u_i^3 & \sum u_i^2 v_i & \sum u_i v_i^2 \\ \sum v_i & \sum u_i v_i & \sum v_i^2 & \sum u_i^2 v_i & \sum u_i v_i^2 & \sum v_i^3 \\ \sum u_i^2 & \sum u_i^3 & \sum u_i^2 v_i & \sum u_i^4 & \sum u_i^3 v_i & \sum u_i^2 v_i^2 \\ \sum u_i v_i & \sum u_i^2 v_i & \sum u_i v_i^2 & \sum u_i^3 v_i & \sum u_i^2 v_i^2 & \sum u_i v_i^3 \\ \sum v_i^2 & \sum u_i v_i^2 & \sum v_i^3 & \sum u_i^2 v_i^2 & \sum u_i v_i^3 & \sum v_i^4 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum y_i u_i \\ \sum y_i v_i \\ \sum y_i u_i^2 \\ \sum y_i u_i v_i \\ \sum y_i v_i^2 \end{pmatrix}$$

Gambar 16 Regresi Kuadratik n=2

BAB 3
IMPLEMENTASI PUSTAKA DAN PROGRAM DALAM JAVA

3.1 Folder lib

Berisi pustaka yang digunakan dalam program utama. Pada folder ini, hanya terdapat Matrix.java dan *class* Matrix dengan sejumlah konstruktor, selektor, atribut, dan *method* yang berguna untuk mengimplementasikan operasi yang dapat digunakan di program utama.

- Konstruktor

Konstruktor	Deskripsi
`public Matrix(int x, int y)`	Melakukan pembuatan matriks dengan x adalah jumlah baris dan y adalah jumlah kolom
`public Matrix()`	Melakukan pembuatan matriks 1×1

- Selektor

Selektor	Deskripsi
`public double getMat(int r, int c)`	Fungsi mengembalikan elemen sebuah matriks pada baris r dan kolom c
`public int getRow()`	Fungsi mengembalikan jumlah baris pada matriks
`public int getCol()`	Fungsi mengembalikan jumlah kolom pa
`public void setMat(int r, int c, double newelmnt)`	Prosedur untuk mengubah nilai elemen matriks pada baris r dan kolom c menjadi newelmnt
`public void setRow(int newelmnt)`	Prosedur untuk mengubah jumlah baris sebuah matriks
`public void setCol(int newelmnt)`	Prosedur untuk mengubah jumlah kolom sebuah matriks

- Atribut

Atribut	Deskripsi
`int row`	Jumlah baris yang digunakan untuk matriks bertipe integer

`int col`	Jumlah kolom yang digunakan untuk kolom bertipe integer
`double[][] elmnt`	Elemen-elemen dari matriks berupa double

- *Method*

Method	**Deskripsi**
`public void print()`	Prosedur untuk mencetak matriks
`public void read(Scanner sc)`	Prosedur untuk membaca matriks hasil input pengguna
`public void readFile()`	Prosedur untuk membaca input pengguna dari file
`public void solutionInverseCramer()`	Prosedur untuk mencetak solusi SPL dengan metode matriks balikan dan kaidah *Cramer* dalam bentuk x1=..., x2=..., dan seterusnya
`public Matrix mulMatrix(Matrix m)`	Fungsi mengembalikan hasil perkalian sebuah matriks dengan matriks m
`public Matrix mulDouble(double multiplier)`	Fungsi mengembalikan hasil perkalian sebuah matriks dengan konstanta multiplier
`public Matrix transpose()`	Fungsi mengembalikan hasil transpose sebuah matriks
`public Matrix cofactor(int p, int q)`	Fungsi mengembalikan matriks yang baris p dan kolom q nya dihilangkan
`public Matrix MatCof()`	Fungsi mengembalikan matriks kofaktor yang elemennya merupakan kofaktor entri aij
`public Matrix adjoin()`	Fungsi mengembalikan matriks adjoin
`public Matrix inverse()`	Fungsi mengembalikan matriks balikan dengan metode adjoin
`public Matrix idenMatrix(int r)`	Fungsi mengembalikan matriks identitas
`public Matrix matTanpaB()`	Fungsi mengembalikan matriks yang

	awalnya merupakan matriks *augmented* *Ax=b* menjadi matriks tanpa kolom b (*Ax* saja)
`public Matrix matB()`	Fungsi mengembalikan matriks 1×1 yang elemennya adalah b pada matriks *augmented Ax=b*
`public void swapRow(int row1, int row2)`	Prosedur untuk menukar baris row1 dengan baris row2
`public void divRow(int dRow, double divider)`	Prosedur untuk membagi semua seluruh elemen pada dRow dengan divider
`public void addRow(int aRow, int rowAdder, double multiplier)`	Prosedur untuk menambahkan seluruh elemen pada baris aRow dengan multiplier×rowAdder
`public void gaussElimination()`	Prosedur untuk melakukan metode eliminasi Gauss
`public void jordanElimination()`	Prosedur untuk melakukan metode eliminasi Gauss-Jordan
`public void matBalikan()`	Prosedur untuk melakukan metode matriks balikan hanya sampai didapatkan matriks balikan (hanya proses OBE saja)
`public Matrix metodeBalikan()`	Fungsi mengembalikan matriks 1×1 yang elemennya merupakan solusi SPL dari metode matriks balikan
`public Matrix kaidahCramer()`	Fungsi mengembalikan matriks 1×1 yang merupakan solusi SPL dengan kaidah *Cramer*
`public double determinanReduksiBaris()`	Fungsi mengembalikan nilai determinan sebuah matriks dengan metode reduksi baris
`public double determinanEkspansiKofaktor()`	Fungsi mengembalikan nilai determinan sebuah matriks dengan metode ekspansi kofaktor
`public void gaussSolution()`	Prosedur untuk menghasilkan solusi SPL dari metode eliminasi Gauss
`public void`	Prosedur untuk menghasilkan solusi SPL

`gaussJordanSolution()`	dari metode eliminasi Gauss Jordan
`public Matrix concat(Matrix X)`	Fungsi mengembalikan matriks hasil konkatenasi sebuah matriks dengan matriks x

3.2 SPL.java

- Konstruktor
  -
- Selektor
  -
- Atribut
  -
- *Method*

Method	Deskripsi
`public static void driver(Scanner sc)`	Prosedur untuk menyelesaikan soal SPL dengan menginput matriks dan memilih metode yang ingin digunakan untuk mendapatkan solusi

3.3 Determinant.java

- Konstruktor
  -
- Selektor
  -
- Atribut
  -
- *Method*

Method	Deskripsi
`public static void driver(Scanner sc)`	Prosedur untuk menghasilkan determinan sebuah matriks yang diinput oleh pengguna dengan dua metode

3.4 Invers.java

- Konstruktor
  -

- Selektor
  -
- Atribut
  -
- *Method*

Method	Deskripsi
`public static void driver(Scanner sc)`	Prosedur untuk menghasilkan matriks balikan dari matriks yang diinput oleh pengguna dengan dua metode

3.5 Interpolate.java

- Konstruktor
  -
- Selektor
  -
- Atribut
  -
- *Method*

Method	Deskripsi
`public static void driver(Scanner sc)`	Prosedur untuk menyelesaikan persoalan interpolasi polinom
`public static Matrix interpolasiLinier(Matrix X, Matrix XP, Matrix YV)`	Fungsi untuk mencari matriks interpolasi linier

3.6 Bicubic.java

- Konstruktor
  -
- Selektor
  -
- Atribut
  -
- *Method*

Method	Deskripsi
`public static void`	Prosedur untuk menyelesaikan persoalan

`driver(Scanner sc)`	interpolasi *bicubic spline*
`public static Matrix flatten(Matrix X)`	Fungsi untuk mengubah matriks 4x4 menjadi matriks 16x1
`public static Matrix muli()`	Fungsi untuk menghasilkan invers dari matriks X
`public static Matrix muliD()`	Fungsi untuk menghasilkan matriks D
`public static void bicubicInterpolation(Matrix f, double tx, double ty)`	Prosedur untuk memproses hasil input pengguna menjadi interpolasi *bicubic spline*

3.7 Regression.java

- Konstruktor
  -
- Selektor
  -
- Atribut
  -
- *Method*

Method	**Deskripsi**
`public static void driver(Scanner sc)`	Prosedur untuk menyelesaikan persoalan regresi linier dengan dua metode
`public static Matrix regresiLinier(Matrix X, Matrix Y)`	Fungsi untuk menghasilkan matriks regresi linier berganda
`public static Matrix regresiKuadratik(Matrix X, Matrix Y)`	Fungsi untuk menghasilkan matriks regresi kuadratik berganda

3.8 ImageProcessing.java

- Konstruktor
  -
- Selektor
  -

- Atribut

  -
- *Method*

Method	Deskripsi
`public static void driver(Scanner sc)`	Prosedur untuk memperbesar gambar dengan menggunakan interpolasi *bicubic spline*
`public static void imageProcessing(String sourceFile, String destFile, int w, int h)`	Prosedur untuk menghasilkan gambar yang sudah diperbesar dengan rasio tertentu
`public static double bicubic(Matrix f, double tx, double ty, Matrix a)`	Fungsi untuk menghasilkan nilai interpolasi *bicubic spline*
`public static int clamp(int value, int min, int max)`	Fungsi untuk memastikan nilai berada dalam batasan, yaitu 0 sampai 255

3.9 Main.java

- Konstruktor

  -
- Selektor

  -
- Atribut

  -
- *Method*

Method	Deskripsi
`public static void main(String[] args)`	Prosedur utama untuk menjalankan program

BAB 4
EKSPERIMEN

4.1 Menemukan solusi SPL Ax = b

a. Test case 1

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$$

- Metode Gauss

```
Tidak ada solusi
```

- Metode Gauss-Jordan

```
Tidak ada solusi
```

- Metode Matriks Balikan

```
Tidak dapat menggunakan metode matriks balikan.
```

- Kaidah *Cramer*

```
Tidak dapat menggunakan Kaidah Cramer
```

b. Test case 2

$$\boldsymbol{A} = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$$

- Metode Gauss

```
x1 = 3.0 + s
x2 = 0.0 + 2.0s
x3 = r
x4 = -1.0 + s
x5 = s
```

- Metode Gauss-Jordan

```
x1 = 3.0 + s
x2 = 0.0 + 2.0s
x3 = r
x4 = -1.0 + s
x5 = s
```

- Metode Matriks Balikan

```
Tidak dapat menggunakan metode matriks balikan.
```

- Kaidah *Cramer*

```
Tidak dapat menggunakan Kaidah Cramer
```

c. Test case 3

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$

- Metode Gauss

```
x1 = r
x2 = 1.0 - t
x3 = s
x4 = -2.0 - t
x5 = 1.0 + t
x6 = t
```

- Metode Gauss-Jordan

```
x1 = r
x2 = 1.0 - t
x3 = s
x4 = -2.0 - t
x5 = 1.0 + t
x6 = t
```

- Metode Matriks Balikan

```
Tidak dapat menggunakan metode matriks balikan.
```

- Kaidah *Cramer*

```
Tidak dapat menggunakan Kaidah Cramer
```

d. Test case 4

$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n+1} \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Kasus n=6

- Metode Gauss

```
x1 = 11.540412090797588
x2 = 46.61669408417674
x3 = -1130.9449694680602
x4 = 3969.618022772758
x5 = -5045.2501529345145
x6 = 2158.663954283237
```

- Metode Gauss-Jordan

```
x1 = 11.540412090797474
x2 = 46.61669408417765
x3 = -1130.9449694680616
x4 = 3969.618022772759
x5 = -5045.2501529345145
x6 = 2158.663954283237
```

- Metode Matriks Balikan

```
x1 = 11.540412090798554
x2 = 46.61669408428679
x3 = -1130.9449694682698
x4 = 3969.6180227729055
x5 = -5045.250152934459
x6 = 2158.663954283227
```

- Kaidah *Cramer*

```
x1 = 11.540412090794426
x2 = 46.61669408415071
x3 = -1130.9449694680939
x4 = 3969.6180227727646
x5 = -5045.2501529345145
x6 = 2158.663954283237
```

Kasus n=10

- Metode Gauss

```
x1 = 19.52562658603513
x2 = -159.54567014007645
x3 = 20.759112204421285
x4 = 1983.9407762087433
x5 = -5084.97673516848
x6 = 5660.14729468183
x7 = -4622.220507394422
x8 = 3257.960172080803
x9 = -623.8360119041495
x10 = -456.35528212233066
```

- Metode Gauss-Jordan

```
x1 = 19.52562658603562
x2 = -159.5456701400795
x3 = 20.759112204425946
x4 = 1983.9407762087394
x5 = -5084.976735168477
x6 = 5660.147294681828
x7 = -4622.2205073944215
x8 = 3257.960172080803
x9 = -623.8360119041495
x10 = -456.35528212233066
```

- Metode Matriks Balikan

```
x1 = 19.52562658603523
x2 = -159.5456701401581
x3 = 20.759112204987105
x4 = 1983.9407762076462
x5 = -5084.976735167751
x6 = 5660.147294681126
x7 = -4622.220507393635
x8 = 3257.9601720805804
x9 = -623.8360119042627
x10 = -456.3552821222984
```

- Kaidah *Cramer*

```
x1 = 19.52562658604407
x2 = -159.54567014014768
x3 = 20.759112204135043
x4 = 1983.9407762087258
x5 = -5084.976735168409
x6 = 5660.147294681841
x7 = -4622.220507394425
x8 = 3257.960172080804
x9 = -623.8360119041495
x10 = -456.35528212233066
```

4.2 SPL berbentuk matriks augmented

a. Test case 1

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

- Metode Gauss

```
x1 = -1.0 + s
x2 = 0.0 + 2.0r
x3 = r
x4 = s
```

- Metode Gauss-Jordan

```
x1 = -1.0 + s
x2 = 0.0 + 2.0r
x3 = r
x4 = s
```

- Metode Matriks Balikan

```
Tidak dapat menggunakan metode matriks balikan.
```

- Kaidah *Cramer*

```
Tidak dapat menggunakan Kaidah Cramer
```

b. Test case 2

$$\begin{bmatrix} 2 & 0 & 8 & 0 & 8 \\ 0 & 1 & 0 & 4 & 6 \\ -4 & 0 & 6 & 0 & 6 \\ 0 & -2 & 0 & 3 & -1 \\ 2 & 0 & -4 & 0 & -4 \\ 0 & 1 & 0 & -2 & 0 \end{bmatrix}$$

- Metode Gauss

```
x1 = 0.0
x2 = 2.0
x3 = 1.0
x4 = 1.0
```

- Metode Gauss-Jordan

```
x1 = 0.0
x2 = 2.0
x3 = 1.0
x4 = 1.0
```

- Metode Matriks Balikan

```
Tidak dapat menggunakan metode matriks balikan.
```

- Kaidah *Cramer*

```
Tidak dapat menggunakan Kaidah Cramer
```

4.3 SPL berbentuk

a. Test case 1

$$\begin{aligned} 8x_1 + x_2 + 3x_3 + 2x_4 &= 0 \\ 2x_1 + 9x_2 - x_3 - 2x_4 &= 1 \\ x_1 + 3x_2 + 2x_3 - x_4 &= 2 \\ x_1 + 6x_3 + 4x_4 &= 3 \end{aligned}$$

- Metode Gauss

```
x1 = -0.2243243243243243
x2 = 0.18243243243243246
x3 = 0.7094594594594594
x4 = -0.25810810810810797
```

- Metode Gauss-Jordan

```
x1 = -0.2243243243243243
x2 = 0.18243243243243246
x3 = 0.7094594594594594
x4 = -0.25810810810810797
```

- Metode Matriks Balikan

```
x0 = -0.22432432432432428
x1 = 0.1824324324324324
x2 = 0.7094594594594593
x3 = -0.25810810810810814
```

- Kaidah *Cramer*

```
x0 = -0.22432432432432411
x1 = 0.18243243243243237
x2 = 0.7094594594594594
x3 = -0.258108108108108
```

b. Test case 2

$$
\begin{aligned}
x_7 + x_8 + x_9 &= 13.00 \\
x_4 + x_5 + x_6 &= 15.00 \\
x_1 + x_2 + x_3 &= 8.00 \\
0.04289(x_3 + x_5 + x_7) + 0.75(x_6 + x_8) + 0.61396x_9 &= 14.79 \\
0.91421(x_3 + x_5 + x_7) + 0.25(x_2 + x_4 + x_6 + x_8) &= 14.31 \\
0.04289(x_3 + x_5 + x_7) + 0.75(x_2 + x_4) + 0.61396x_1 &= 3.81 \\
x_3 + x_6 + x_9 &= 18.00 \\
x_2 + x_5 + x_8 &= 12.00 \\
x_1 + x_4 + x_7 &= 6.00 \\
0.04289(x_1 + x_5 + x_9) + 0.75(x_2 + x_6) + 0.61396x_3 &= 10.51 \\
0.91421(x_1 + x_5 + x_9) + 0.25(x_2 + x_4 + x_6 + x_8) &= 16.13 \\
0.04289(x_1 + x_5 + x_9) + 0.75(x_4 + x_8) + 0.61396x_7 &= 7.04
\end{aligned}
$$

- Metode Gauss

```
Tidak ada solusi
```

- Metode Gauss-Jordan

```
Tidak ada solusi
```

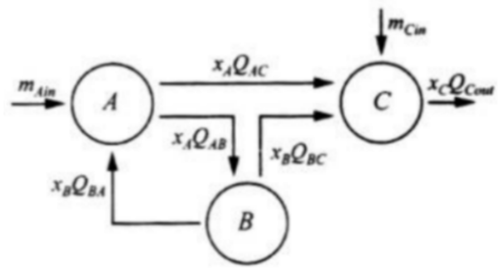
- Metode Matriks Balikan

```
Tidak dapat menggunakan metode matriks balikan
```

- Kaidah *Cramer*

```
Tidak dapat menggunakan Kaidah Cramer
```

4.4 Sistem reaktor



Dengan laju volume Q dalam m3/s dan input massa min dalam mg/s. Konservasi massa pada tiap inti reaktor adalah sebagai berikut:

$$\begin{aligned}
&\text{A:} \quad m_{A_{in}} + Q_{BA}x_B - Q_{AB}x_A - Q_{AC}x_A = 0 \\
&\text{B:} \quad Q_{AB}x_A - Q_{BA}x_B - Q_{BC}x_B = 0 \\
&\text{C:} \quad m_{C_{in}} + Q_{AC}x_A + Q_{BC}x_B - Q_{C_{out}}x_C = 0
\end{aligned}$$

Parameter: $Q_{AB} = 40$, $Q_{AC} = 80$, $Q_{BA} = 60$, $Q_{BC} = 20$ dan $Q_{Cout} = 150$ m^3/s dan $m_{Ain} = 1300$ dan $m_{Cin} = 200$ mg/s.

Dengan metode eliminasi Gauss, didapatkan X_A, X_B, dan X_C sebagai berikut ($X_1 = X_A$, $X_2 = X_B$, $X_3 = X_C$)

```
x1 = 14.444444444444446
x2 = 7.222222222222223
x3 = 6.148148148148149
```

4.5 Studi Kasus Interpolasi

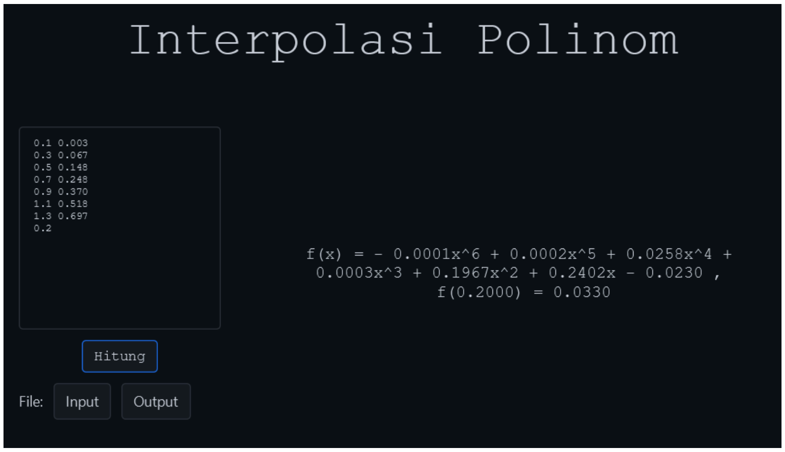
a. Test case 1

Gunakan tabel di bawah ini untuk mencari polinom interpolasi dari pasangan titik-titik yang terdapat dalam tabel. Program menerima masukan nilai *x* yang akan dicari nilai fungsi *f*(*x*).

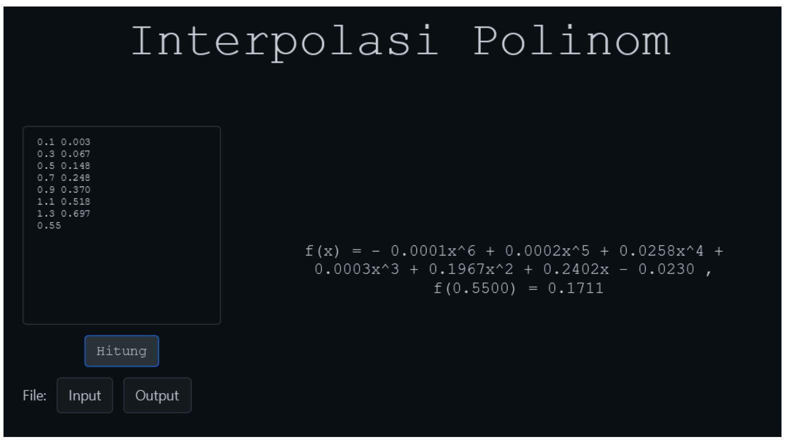
x	0.1	0.3	0.5	0.7	0.9	1.1	1.3
f(*x*)	0.003	0.067	0.148	0.248	0.370	0.518	0.697

Lakukan pengujian pada nilai-nilai berikut:

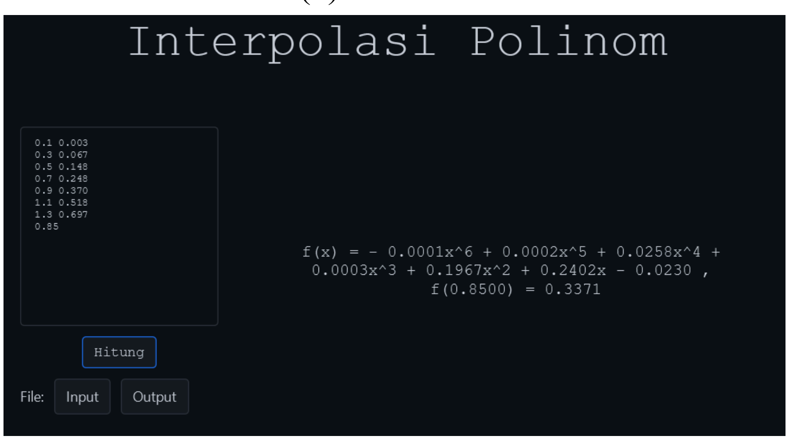
- x = 0.2 f(x) = 0.0330



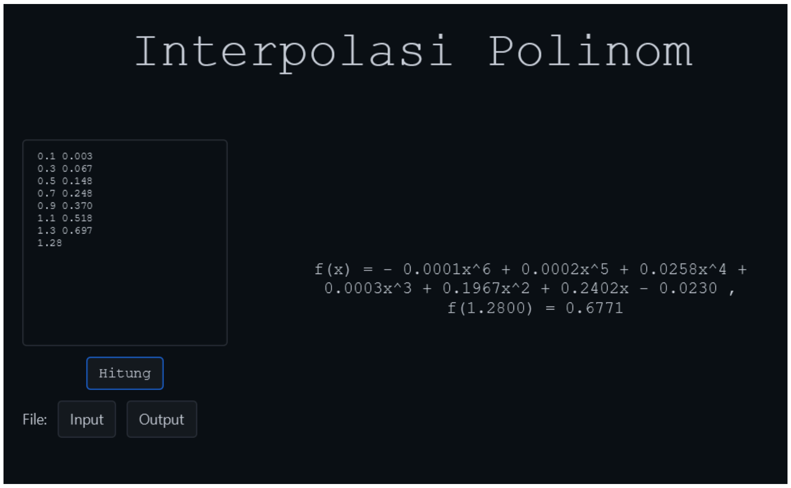
- $x = 0.55$ $f(x) = 0.1711$



- $x = 0.85$ $f(x) = 0.3371$



- $x = 1.28$ $f(x) = 0.6771$



b. Test case 2

Jumlah kasus positif baru Covid-19 di Indonesia semakin fluktuatif dari hari ke hari. Di bawah ini diperlihatkan jumlah kasus baru Covid-19 di Indonesia mulai dari tanggal 17 Juni 2022 hingga 31 Agustus 2022:

Tanggal	Tanggal (desimal)	Jumlah Kasus Baru
17/06/2022	6,567	12.624
30/06/2022	7	21.807
08/07/2022	7,258	38.391
14/07/2022	7,451	54.517
17/07/2022	7,548	51.952
26/07/2022	7,839	28.228
05/08/2022	8,161	35.764
15/08/2022	8,484	20.813
22/08/2022	8,709	12.408
31/08/2022	9	10.534

Tanggal (desimal) adalah tanggal yang sudah diolah ke dalam bentuk desimal 3 angka di belakang koma dengan memanfaatkan perhitungan sebagai berikut:

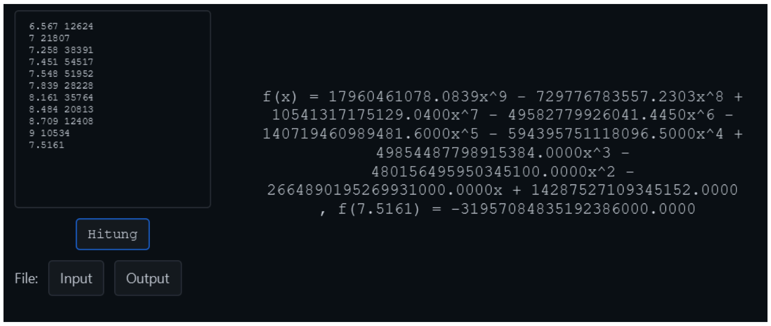
Tanggal (desimal) = bulan + (tanggal / jumlah hari pada bulan tersebut)

Sebagai contoh, untuk tanggal 17/06/2022 (dibaca: 17 Juni 2022) diperoleh tanggal(desimal) sebagai berikut:

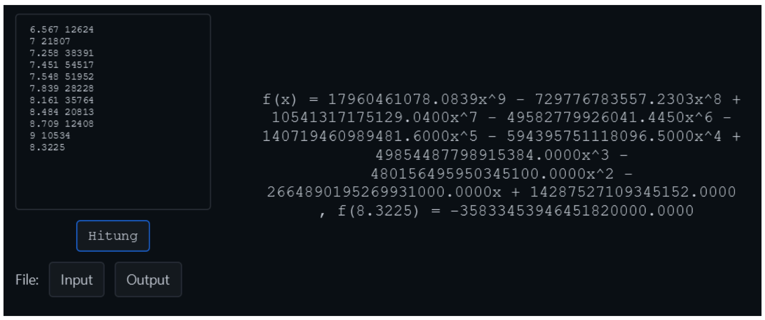
Tanggal (desimal) = 6 + (17/30) = 6,567

Gunakanlah data di atas dengan memanfaatkan **interpolasi polinomial** untuk melakukan prediksi jumlah kasus baru Covid-19 pada tanggal-tanggal berikut:

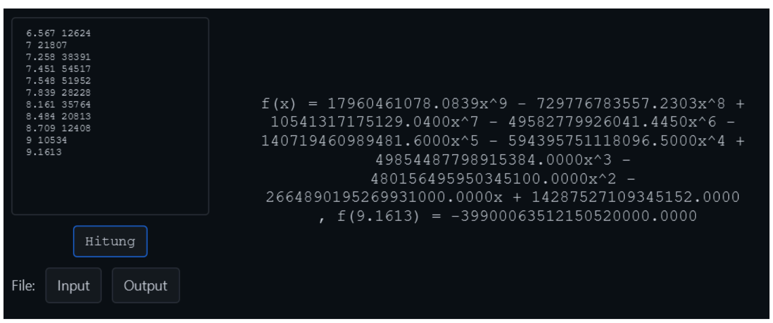
- 16/07/2022



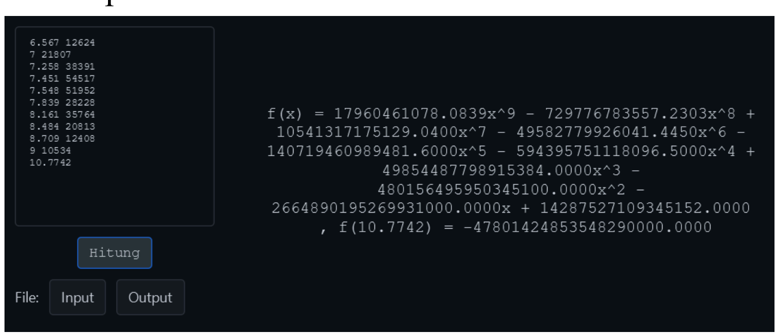
- 10/08/2022



- 05/09/2022



- Masukan user lainnya berupa tanggal (desimal) yang sudah diolah dengan asumsi prediksi selalu dilakukan untuk tahun 2022.



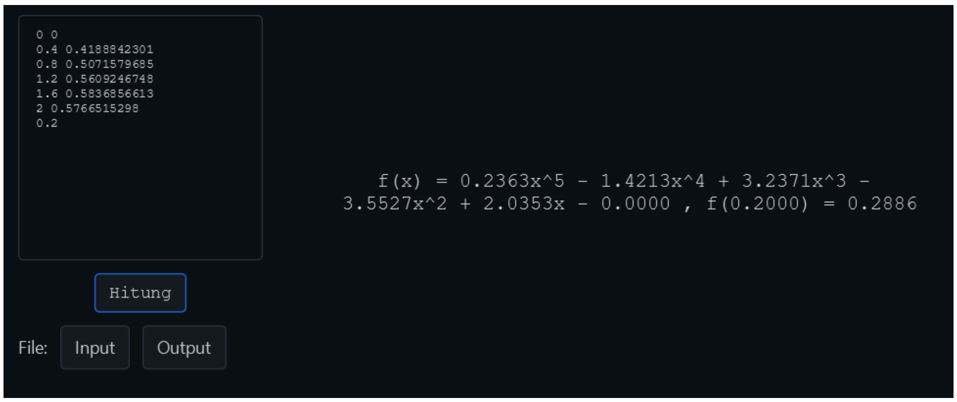
c. Test case 3

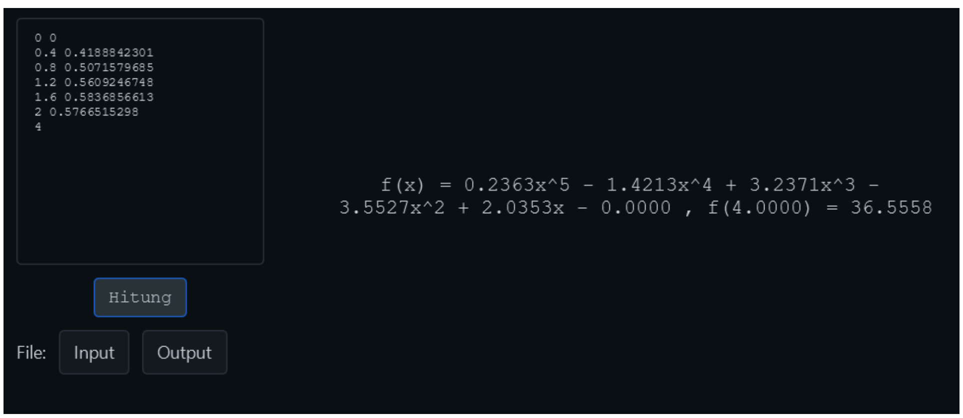
Sederhanakan fungsi $f(x)$ yang memenuhi kondisi

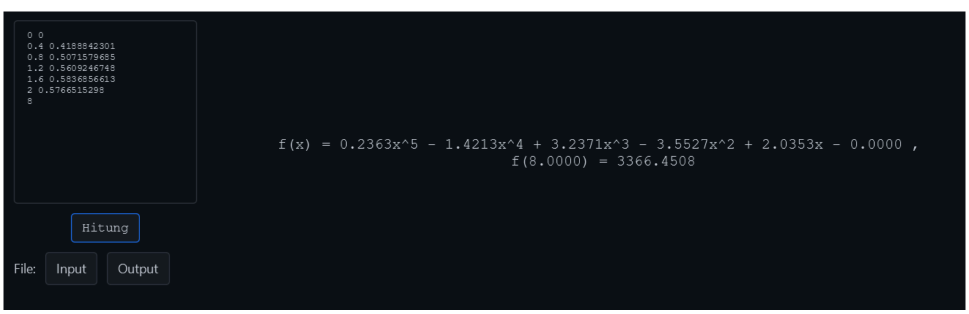
$$f(x) = \frac{x^2 + \sqrt{x}}{e^x + x}$$

dengan polinom interpolasi derajat n di dalam selang [0, 2].
Sebagai contoh, jika n = 5, maka titik-titik x yang diambil di dalam selang [0, 2] berjarak h = (2 – 0)/5 = 0.4.

Untuk n=5, titik-titik x yang diambil = {0.0, 0.4, 0.8, 1.2, 1.6, 2.0}







4.6 Studi Kasus Regresi Linear dan Kuadratik Berganda

Diberikan sekumpulan data sesuai pada tabel berikut ini.

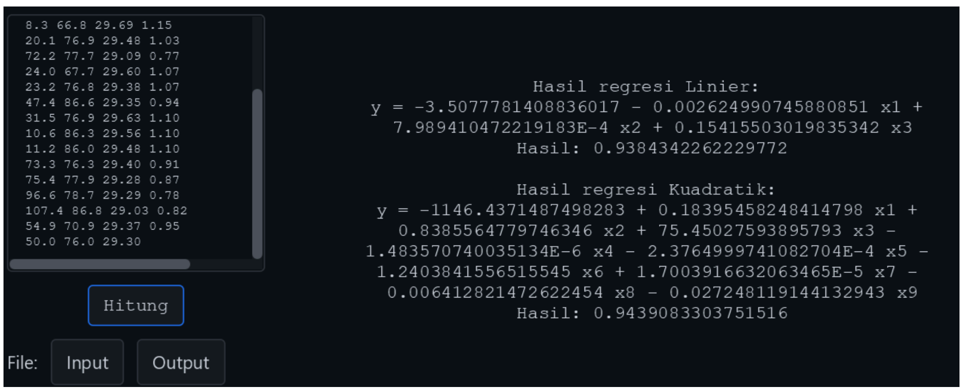
Table 12.1: Data for Example 12.1

Nitrous Oxide, y	Humidity, x_1	Temp., x_2	Pressure, x_3	Nitrous Oxide, y	Humidity, x_1	Temp., x_2	Pressure, x_3
0.90	72.4	76.3	29.18	1.07	23.2	76.8	29.38
0.91	41.6	70.3	29.35	0.94	47.4	86.6	29.35
0.96	34.3	77.1	29.24	1.10	31.5	76.9	29.63
0.89	35.1	68.0	29.27	1.10	10.6	86.3	29.56
1.00	10.7	79.0	29.78	1.10	11.2	86.0	29.48
1.10	12.9	67.4	29.39	0.91	73.3	76.3	29.40
1.15	8.3	66.8	29.69	0.87	75.4	77.9	29.28
1.03	20.1	76.9	29.48	0.78	96.6	78.7	29.29
0.77	72.2	77.7	29.09	0.82	107.4	86.8	29.03
1.07	24.0	67.7	29.60	0.95	54.9	70.9	29.37

Source: Charles T. Hare, "Light-Duty Diesel Emission Correction Factors for Ambient Conditions," EPA-600/2-77-116. U.S. Environmental Protection Agency.

Gunakan *Normal Estimation Equation for Multiple Linear Regression* untuk mendapatkan regresi linear berganda dari data pada tabel di atas, kemudian estimasi nilai Nitrous Oxide apabila Humidity bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30.

Regresi linear berganda:



Estimasi nilai Nitrous Oxide:

```
Hasil regresi linier:
0.9384342262229772
Hasil regresi kuadratik:
0.9439083303751516
```

4.7 Studi Kasus Interpolasi *Bicubic Spline*

$$\begin{pmatrix} 21 & 98 & 125 & 153 \\ 51 & 101 & 161 & 59 \\ 0 & 42 & 72 & 210 \\ 16 & 12 & 81 & 96 \end{pmatrix}$$

- f(0, 0)

```
f(0.0 , 0.0) = 21.0
```

- f(0.5, 0.5)

```
f(0.5 , 0.5) = 87.796875
```

- f(0.25, 0.75)

```
f(0.25 , 0.75) = 82.148193359375
```

- f(0.1, 0.9)

```
f(0.1 , 0.9) = 91.27126700000001
```

BAB 5
PENUTUP

5.1 Kesimpulan

Berbagai persoalan dapat dinyatakan dalam bentuk matriks. Mulai dari sistem persamaan linear sampai mengubah gambar bisa dinyatakan dalam bentuk matriks. Dalam tugas ini, kami mengimplementasikan berbagai metode untuk persoalan matriks, seperti menemukan solusi dari sebuah SPL, mencari determinan matriks, menemukan matriks balikan, mendapatkan persamaan polinom interpolasi, mendapatkan persamaan dan hasil regresi linear berganda, dan penyelesaian dari interpolasi *bicubic spline*.

Solusi SPL didapatkan dengan metode eliminasi Gauss, eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah *cramer*. Determinan dicari dengan dua cara, yaitu determinan reduksi baris dan determinan ekspansi kofaktor. Matriks balikan juga menggunakan dua metode, yaitu metode OBE dengan eliminasi Gauss-Jordan dan metode matriks adjoin. Semua metode ini sudah terimplementasi dalam program yang kami buat. Keseluruhan program dibuat dengan menggunakan bahasa Java dan javaFX untuk GUI.

Selain hal-hal di atas, terdapat implementasi dari GUI untuk membuat program menjadi lebih interaktif. Terdapat pula fitur *Image Resizing and Stretching* dan video penjelasan program.

5.2 Saran

Beberapa saran pengembangan untuk program ini adalah sebagai berikut.

1. Membagi isi *library* dengan lebih terstruktur dan tidak terpusat pada satu file saja.
2. Memberikan pilihan untuk melihat langkah penyelesaian SPL dan determinan.
3. Menambahkan parameter *alpha* atau *transparency value* pada *image processing* untuk memperbagus hasil *resizing image*.
4. Pemberian nama fungsi/prosedur dan komentar dengan lebih jelas untuk memudahkan pemahaman mengenai fungsi/prosedur.

5.3 Komentar

1. Bertha: “videonya bagus. Boleh ditonton”
2. Hima: “heehehhehehehhehehehehhehehehehehhehehhehhhehehheheeeheheheheskibidiheheh ehhehhehehehhehhehehehhehegyattttthehehhehehhehehhehehe”
3. Mik: “Bolehlah tubes pertama.”

5.4 Refleksi

Pengerjaan tubes ini membuat kami belajar banyak hal. Mulai dari belajar materi seperti regresi dan interpolasi sampai belajar untuk berekspresi. Komunikasi sangat penting dalam penyelesaian tubes ini. Tanpa adanya komunikasi, tugas ini tidak akan selesai.

Lampiran

Referensi

Howard Anton (2010). Elementary Linear Algebra, 10th edition, John Wiley and Sons, 2010

Mohammad Rosidi. (2019). Metode Numerik Menggunakan R Untuk Teknik Lingkungan. Diakses pada 19 Oktober 2024, dari https://www.google.com/url?q=https://bookdown.org/moh_rosidi2610/Metode_Numerik/interpolation.html%23:~:text%3DInterpolasi%2520polinomial%2520merupakan%2520teknik%2520interpolasi,terlebih%2520dahulu%2520membentuk%2520persamaan%2520polinomial&sa=D&source=docs&ust=1729700896533369&usg=AOvVaw0Lj0rAsZ1WrzWagzscjkOK.

Informatika.stei.itb.ac.id (2023). Sistem persamaan linier (Bagian 1: Metode eliminasi Gauss). Diakses pada 19 Oktober 2024, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-03-Sistem-Persamaan-Linier-2023.pdf.

Informatika.stei.itb.ac.id (2024). Sistem persamaan linier (Bagian 3: Metode eliminasi Gauss-Jordan). Diakses pada 19 Oktober 2024, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-05-Sistem-Persamaan-Linier-2-2023.pdf.

Informatika.stei.itb.ac.id (2023). Determinan (Bagian 1: menghitung determinan dengan reduksi baris). Diakses pada 19 Oktober 2024, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-08-Determinan-bagian1-2023.pdf.

Informatika.stei.itb.ac.id (2023). Determinan (Bagian 2: menghitung determinan dengan ekspansi kofaktor). Diakses pada 19 Oktober 2024, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-09-Determinan-bagian2-2023.pdf.

Tautan *Repository*

https://github.com/aibrahim185/Algeo01-23026

Tautan video

https://drive.google.com/file/d/1XQ0mfydK0DziHgmXaCZRL0dDF_xG3Rp4/view?usp=drive_link