LAPORAN TUGAS BESAR

Sistem Persamaan Linier, Determinan, dan Aplikasinya

IF2123 Aljabar Linier dan Geometri

Dosen pengajar: Dr. Ir. Rinaldi, M.T

Kelompok Oegla

Anggota Kelompok :

Husnia Munzayana – 13521077

Shelma Salsabila – 13521115

Althaaf Khasyi Atisomya – 13521130

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2022

DAFTAR ISI

DAFTAR TABEL

Bab 1
Deskripsi Masalah

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Anda 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 -1b), 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, anda diminta membuat satu atau lebih library aljabar linier dalam Bahasa Java. Library tersebut berisi fungsi-fungsi seperti eliminasi Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung determinan, kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n peubah dan n persamaan). Selanjutnya, gunakan library tersebut di dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi. Penjelasan tentang interpolasi dan regresi adalah seperti di bawah ini.

A. Buatlah pustaka dalam Bahasa Java untuk menemukan solusi SPL dengan metode eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n peubah dan n persamaan), menghitung determinan matriks dengan reduksi baris dan dengan ekspansi kofaktor, dan menghitung balikan matriks.
B. Gunakan pustaka di atas untuk membuat program penyelesaian berbagai persoalan dalam bentuk SPL, menyelesaikan persoalan interpolasi dan regresi linier, menghitung matriks balikan, menghitung determinan matriks dengan berbagai metode (reduksi baris dan ekspansi kofaktor).

Bab 2
Teori Singkat

2.1 Eliminasi Gauss

Eliminasi Gauss adalah metode eliminasi yang digunakan untuk memecahkan sistem persamaan linier dengan merepresentasikan (mengubah) menjadi bentuk matriks augmented kemudian diubah menjadi eselon baris melalui OBE (Operasi Baris Elementer).Langkah-langkah lebih jelas adalah sebagai berikut.

Misalkan terdapat persamaan linier:

$$2x + 3y - z = 5$$

$$4x + 4y - 3z = 3$$

$$-2x + 3y - z = 1$$

1. Nyatakan dalam bentuk matriks augmented

$$\begin{bmatrix} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{bmatrix}$$

2. Kemudian dijadikan matriks eselon baris dengan penerapan OBE.

$$\begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

3. Dari matriks tersebut diperoleh persamaan

$$x + 3/2y - 1/2z = 5/2$$

$$y + 1/2z = 7/2$$

$$z = 3$$

Dari persamaan di atas dapat ditentukan x = 1, y = 2, z = 3.

2.2 Eliminasi Gauss Jordan

Eliminasi Gauss Jordan adalah pengembangan dari eliminasi Gauss, perbedaannya pada eliminasi ini matriks yang dibentuk dengan menerapkan penerapan OBE adalah matriks eselon baris tereduksi. Setiap 1 utama di atas dan di bawahnya bernilai 0. Dari persamaan yang sama seperti eliminasi gauss diperoleh matriks eselon baris tereduksi sebagai berikut.

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

Dari persamaan di atas diperoleh x = 1, y = 2, dan z = 3.

2.3 Determinan Matriks

Determinan matriks hanya bisa dicari ketika matriks itu persegi. Untuk mencari determinan dalam tugas besar ini dapat diperoleh dengan du acari yaitu determinan dengan reduksi baris dan yang kedua dengan metode ekspansi kofaktor.

1. Menghitung determinan dengan reduksi baris
   1) Reduksi matriks hingga membentuk segitiga atas atau segitiga bawah.

   $$\begin{bmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & f \end{bmatrix} \text{atau} \begin{bmatrix} a & 0 & 0 \\ b & d & 0 \\ c & e & f \end{bmatrix}$$

   $$contoh\ segitiga\ atas\ dan\ bawah\ 3x3$$

   2) Kemudian determinan dapat dicari dengan mengalikan tiap elemen diagonalnya. Jika pada saat reduksi terjadi pertukaran baris jika jumlah pertukaran merupakan bilangan ganjil maka hasil determinan dikali -1.
2. Menghitung determinan dengan kofaktor

   Penghitungan ini dilakukan dengan cara menjumlahkan perkalian tiap elemennya dengan kofaktornya dalam satu baris atau kolom.

   Contohnya matriks M

   $$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

   Dengan memanfaatkan baris 1

   Det(M) = a*Ma – b*Mb + c*Mc

   Pengaplikasian determinan cukup banyak bisa digunakan untuk mencari SPL dengan menggunakan metode crammer kemudian mencari invers balikan juga bisa digunakan.

2.4 Matriks Balikan(Invers)

Misalkan suatu matriks B adalah invers dari A maka, AB = BA = I, ada dua metode yang dapat diterapkan untuk mencari matriks balikan ini.

1. Dengan metode eliminasi Gauss Jordan

   Misalnya diberikan suatu matriks sebagai berikut.

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 3 \\ 1 & 0 & 8 \end{bmatrix}$$

   Langkah pengerjaannya adalah sebagai berikut

   1) Tuliskan matriks di atas serta pinggirnya ditambah matriks identitas

$$\begin{pmatrix} 1 & 2 & 3 & 1 & 0 & 0 \\ 2 & 5 & 3 & 0 & 1 & 0 \\ 1 & 0 & 8 & 0 & 0 & 1 \end{pmatrix}$$

   2) Kemudian lakukan OBE sehingga matriks sebelah kiri menjadi sebuah matriks identitas.
   3) Sehingga didapat balikan matriksnya adalah

$$\begin{bmatrix} -40 & 16 & 9 \\ 13 & -5 & -3 \\ 5 & -2 & -1 \end{bmatrix}$$

2. Dengan adjoint

   Langkah-langkahnya sebagai berikut.

   1) Mencari determinan dari matriks di atas
   2) Setelah itu cari matriks adjoint dari matriks tersebut lalu transposkan. Hasil dari kedua operator tersebut menjadikan matriks sebagai berikut.

$$\begin{bmatrix} -2 & -2 & 4 \\ -1 & -71 & 16 \\ 4 & 32 & -8 \end{bmatrix}$$

   Matriks balikan dapat dicari dengan formula sebagai berikut.

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

$$A^{-1} = \frac{1}{28}\begin{bmatrix} -2 & -2 & 4 \\ -1 & -71 & 16 \\ 4 & 32 & -8 \end{bmatrix}$$

$$A^{-1} = \begin{bmatrix} -\frac{1}{14} & -\frac{1}{14} & \frac{1}{7} \\ -\frac{1}{28} & -\frac{71}{28} & \frac{4}{7} \\ \frac{1}{7} & \frac{8}{7} & -\frac{2}{7} \end{bmatrix}$$

Invers ini bisa digunakan pada banyak hal salah satu pengaplikasiannya adalah bicubic.

2.5 Matriks Kofaktor

Matriks ini berisi kofaktor dari setiap elemen di dalam matriks. Kofaktor setiap elemen dinyatakan sebagai

$$C_{ij} = (-1)^{i+j} M_{ij}$$

dengan

C_{ij} : kofaktor elemen matriks baris ke-i dan kolom ke-j

M_{ij} : Minor dari matriks M pada baris ke-i dan kolom ke-j

Untuk mencari minor dapat dicari dengan mencari determinan dari matriks baru yang menghilangkan elemen di M pada baris ke-i dan kolom ke-j.

Contoh pencarian kofaktor

$$\begin{bmatrix} 3 & 6 & 4 \\ -1 & 5 & 2 \\ 4 & 10 & 9 \end{bmatrix}$$

$$C_{11} = \begin{vmatrix} 5 & 2 \\ 10 & 9 \end{vmatrix} = 25 \quad C_{11} = (-1)^{1+1}.25 = 25$$

$$C_{12} = \begin{vmatrix} -1 & 2 \\ 4 & 9 \end{vmatrix} = -17 \quad C_{12} = (-1)^{1+2}.(-17) = 17$$

Sehingga diperoleh matriks kofaktor dari M adalah

$$\begin{bmatrix} 30 & 17 & -30 \\ -14 & 11 & -6 \\ -8 & -10 & 21 \end{bmatrix}$$

2.6 Kaidah Cramer

Kaidah cramer merupakan formula yang dipakai untuk menyelesaikan sistem persamaan linear dengan menggunakan determinan.

Misalkan persamaan sistem persamaan linier

$$Ax + By + Cz = D1$$

$$Ex + Fy + Gz = D2$$

$$Hx + Iy + Jz = D3$$

Ada beberapa determinan yang kita cari,

$$\text{Det} = \begin{bmatrix} A & B & C \\ E & F & G \\ H & I & J \end{bmatrix}$$

$$\text{Det x} = \begin{bmatrix} \text{D1} & B & C \\ D2 & F & G \\ D3 & I & J \end{bmatrix} \quad \text{Det y} = \begin{bmatrix} \text{A} & D1 & C \\ E & D2 & G \\ H & D3 & J \end{bmatrix} \quad \text{Det z} = \begin{bmatrix} \text{A} & B & D1 \\ E & F & D2 \\ H & I & D3 \end{bmatrix}$$

$$x = \frac{Det\ x}{Det} \qquad y = \frac{Det\ y}{Det} \qquad z = \frac{Det\ z}{Det}$$

2.7 Regresi Linier Berganda

Regresi linier berganda metode untuk memprediksi nilai fungsi dengan banyak peubah. Untuk menyelesaikan ini dilakukan dengan metode Normal Estimation Equation for Multiple Linier Regression membentuk sebuah persamaan seperti di bawah ini.

$$\begin{array}{lllll}
nb_0 + b_1 \sum_{i=1}^{n} x_{1i} & + b_2 \sum_{i=1}^{n} x_{2i} & + \cdots & + b_k \sum_{i=1}^{n} x_{ki} & = \sum_{i=1}^{n} y_i \\
b_0 \sum_{i=1}^{n} x_{1i} + b_1 \sum_{i=1}^{n} x_{1i}^2 & + b_2 \sum_{i=1}^{n} x_{1i}x_{2i} + & \cdots & + b_k \sum_{i=1}^{n} x_{1i}x_{ki} & = \sum_{i=1}^{n} x_{1i}y_i \\
\vdots \qquad \vdots & \vdots & & \vdots & \vdots \\
b_0 \sum_{i=1}^{n} x_{ki} + b_1 \sum_{i=1}^{n} x_{ki}x_{1i} + & b_2 \sum_{i=1}^{n} x_{ki}x_{2i} + & \cdots & + b_k \sum_{i=1}^{n} x_{ki}^2 & = \sum_{i=1}^{n} x_{ki}y_i
\end{array}$$

Setelah terbentuk itu maka gunakan gauss Jordan untuk menentukan koefisien-koefisien seperti b0,b1 dan seterusnya.

2.8 Interpolasi Bicubic

Bicubic interpolation merupakan teknik interpolasi pada data 2D umumnya digunakan dalam pembesaran citra yang merupakan pengembangan dari interpolasi linear dan cubic yang telah dipelajari pada kuliah metode numerik di aljabar geometri. Cara mencari permasamaan interpolasi

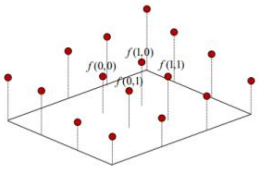
1. Gunakan pemodelan

Normalization: $f(0,0), f(1,0)$
$f(0,1), f(1,1)$

Model: $f(x,y) = \sum_{j=0}^{3}\sum_{i=0}^{3} a_{ij} x^i y^j$

$x = -1, 0, 1, 2$

Solve: a_{ij}



2. Melakukan substitusi nilai-nilai diketahui pada matriks 4 x 4 tersebut ke persamaan f(x,y) akan menghasilkan sebuah matriks persamaan

$$y = Xa$$

$$\begin{bmatrix} f(-1,-1) \\ f(0,-1) \\ f(1,-1) \\ f(2,-1) \\ f(-1,0) \\ f(0,0) \\ f(1,0) \\ f(2,0) \\ f(-1,1) \\ f(0,1) \\ f(1,1) \\ f(2,1) \\ f(-1,2) \\ f(0,2) \\ f(1,2) \\ f(2,2) \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 2 & 4 & 8 & -1 & -2 & -4 & -8 & 1 & 2 & 4 & 8 & -1 & -2 & -4 & -8 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 \\ 1 & -1 & 1 & -1 & 2 & -2 & 2 & -2 & 4 & -4 & 4 & -4 & 8 & -8 & 8 & -8 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 8 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 1 & 2 & 4 & 8 & 2 & 4 & 8 & 16 & 4 & 8 & 16 & 32 & 8 & 16 & 32 & 64 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30} \\ a_{01} \\ a_{11} \\ a_{21} \\ a_{31} \\ a_{02} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{03} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}$$

3. Elemen pada matrix X adalah koefisien a_{ij} yang diperoleh dari persamaan f(x,y) di atas. Sebagai contoh, elemen pada baris 4 kolom ke 10 adalah koefisien dari a_{12} dan diperoleh dari $2^1 * (-1)^2 = 2$, sesuai persamaan $x^i * y^j$. 3 Vektor a dapat dicari dari persamaan tersebut (menggunakan inverse), lalu vektor a digunakan sebagai nilai variabel dalam f(x,y). Sehingga terbentuk fungsi interpolasi bicubic sesuai model.

2.9 Interpolasi Polinom

Polinom interpolasi derajat n yang menginterpolasi titik-titik (x_0 , y_0), (x_1 , y_1), ..., (x_n , y_n) adalah berbentuk

$$p_n(x) = a_0 + a_1x + a_2x_2 + \ldots + a_nx_n \, .$$

Jika hanya ada dua titik, (x_0 , y_0) dan (x_1 , y_1), maka polinom yang menginterpolasi kedua titik tersebut adalah $p_1(x) = a_0 + a_1x$ yaitu berupa persamaan garis lurus. Jika tersedia tiga titik, (x_0 , y_0), (x_1 , y_1), dan (x_2 , y_2), maka polinom yang menginterpolasi ketiga titik tersebut adalah $p_2(x) = a_0 + a_1x + a_2x_2$ atau persaman kuadrat dan kurvanya berupa parabola. Jika tersedia empat titik, (x_0 , y_0), (x_1 , y_1), (x_2 , y_2), dan (x_3 , y_3), polinom yang menginterpolasi keempat titik tersebut adalah $p_3(x) = a_0 + a_1x + a_2x_2 + a_3x_3$, demikian seterusnya. Dengan cara yang sama kita dapat membuat polinom interpolasi berderajat n untuk n yang lebih tinggi asalkan tersedia (n+1) buah titik data. Dengan menyulihkan (x_i , y_i) ke dalam persamaan polinom

$$p_n(x) = a_0 + a_1x + a_2x_2 + ... + a_nx_n$$

untuk i = 0, 1, 2, .., n akan diperoleh n buah sistem persamaan lanjar dalam a_0 , a_1 , a_2 , .., a_n ,

$$\begin{aligned}
a_0 + a_1x_0 + a_2x_0^2 + \ldots + a_n x_0^n &= y_0 \\
a_0 + a_1x_1 + a_2x_1^2 + \ldots + a_n x_1^n &= y_1 \\
\ldots \qquad\qquad & \ldots \\
a_0 + a_1x_n + a_2x_n^2 + \ldots + a_n x_n^n &= y_n
\end{aligned}$$

Solusi sistem persamaan lanjar ini, yaitu nilai a_0 , a_1 , ..., a_n , diperoleh dengan menggunakan metode eliminasi Gauss.

Bab 3

Implementasi Program

3.1 Class Matrix

Nama	Tipe	Parameter	Deskripsi
createMatrix	public double[][]	Int row Int col	Membuat matriks kosong berukuran row dan col
nBaris	public int	double[][] m	Mengembalikan jumlah baris m
nKolom	public int	double[][] m	Mengembalikan jumlah kolom m
bacaMatrix	public void	-	Membaca matriks dengan elemen input user
tulisMatrix	public void	double[][] m	Menuliskan matriks ke layar
swapBaris	public double[][]	double[][] m	Menukarkan baris atas dan bawahnya
kaliBaris	public double[][]	double[][] m int idxRow double x	Mengalikan suatu baris dengan x
barisKurangNBaris	public double[][]	double[][] m int idxRow1 int idxRow2 double x	Mengurangkan suatu baris dengan baris yang lain yang menjadi acuan
isZero	public boolean	double x	Menghasilkan true ketika bernilai nol
isRowZero	public boolean	double[][] m int i	Mengembalikan true jika semua elemen pada suatu baris bernilai nol
AugmentedtoSquare	public double[][]	double[][] m	Mengubah bentuk matriks augmented jadi persegi
transpose	public double[][]	double[][] m	Mengembalikan matriks hasil transpose
kaliKonstanta	public double[][]	double[][] m double x	Mengembalikan matriks hasil kali dengan konstanta x

kaliMatriks	public double[][]	double[][] m1 double[][] m2	Mengembalikan matriks hasil kali dengan matriks lagi

Table 1 Class Matriks, penjelasan mengenai beberapa method yang ada pada Class Matrix

3.2 Class Menu

Nama	Tipe	Parameter	Deskripsi
displayMenu	public void	-	Untuk menampilkan tampilan menu
optionInput	public int	int min int max	Mengembalikan opsi dari user
displayMenuSPL	public void	-	Menampilkan submenu Determinan
displayMenuInverse	public void	-	Menampilkan submenu inverse
displayMenuInput	public void	-	Menampilkan submenu input
displayMenuSave	public void	-	Menampilkan submenu save

Table 2 Class Menu, penjelasan mengenai beberapa method yang ada pada Class Menu

3.3 Class Gauss

Nama	Tipe	Parameter	Deskripsi
eselonBaris	public double[][]	double[][] m	Method Untuk menghasilkan matriks eselon baris
idxLeadingOne	public int[]	double[][] m int[] pivot	Mengembalikan index elemen yang akan menjadi satu utama berikutnya

Table 3 Class Gauss, penjelasan mengenai beberapa method yang ada pada Class Gauss

3.4 Class Gauss_Jordan

Nama	Tipe	Parameter	Deskripsi
nextPivot	public int[]	double[][] m int[] pivot	Method mengeluarkan acuan yang nantinya digunakan untuk membuat matriks eselon baris tereduksi
eselonBarisTereduksi	public double[][]	double[][] m	Method Untuk menghasilkan matriks eselon baris tereduksi

Table 4 Class Gauss Jordan, penjelasan mengenai beberapa method yang ada pada Class Gauss Jordan

3.5 Class Determinan

Nama	Tipe	Parameter	Deskripsi
ElmtCofactor	public double[][]	double[][] m int row int col	Method untuk menghasilkan elemen matriks kofaktor
MatrixCof	public double[][]	double[][] m	Menghasilkan matriks kofaktor
determinan	public double	double[][] m	Menghasilkan determinan dengan penghitungan menggunakan kofaktor

Table 5 Class Determinan, penjelasan mengenai beberapa method yang ada pada Class Determinan

3.6 Class DeterminanGJ

Nama	Tipe	Parameter	Deskripsi
determiananGaussJordan	public double	double[][] m	Method untuk menghasilkan determinan dengan proses penyelesaian reduksi baris

Table 6 DeterminanGJ, penjelasan mengenai beberapa method yang ada pada Class DeterminanGJ

3.7 Class Invers

Nama	Tipe	Parameter	Deskripsi
InversCofactor	Public double[][]	double[][] m	Method untuk menghasilkan hasil invers matriks dengan proses penyelesaian dengan kofaktor

Table 7 Class Invers, penjelasan mengenai beberapa method yang ada pada Class Invers

3.8 Class InversGauss

Nama	Tipe	Parameter	Deskripsi
InversGauss	Public double[][]	double[][] m	Method untuk menghasilkan hasil invers matriks dengan proses penyelesaian dengan gauss Jordan

Table 8 Class InversGauss, penjelasan mengenai beberapa method yang ada pada Class InversGauss

3.9 Class Crammer

Nama	Tipe	Parameter	Deskripsi
crammerhasil	Public double[]	double[][] m2 double[][] m1	Method untuk menyelesaikan SPL namun menggunakan cara crammer memanfaatkan determinan

Table 9 Class Crammer, penjelasan mengenai beberapa method yang ada pada Class Crammer

3.10 Class SPLSolver

Nama	Tipe	Parameter	Deskripsi
whatSolution	Public int	double[][] m	Menghasilkan nilai untuk menentukan cara menyelesaikan SPL

splSolution	Public void	double[][] m	Method untuk memanggil method penyelesaian SPL yang tepat mengacu pada output method whatSolution
splUniqueSol	Public double[]	double[][] m	Mengembalikan array yang elemen elemennya merupakan penyelesaian dari SPL
displayUniqueSol	Public void	double[] result	Menampilkan hasil penyelesaian SPL dengan solusi unik
uniqueSol2Arr	Public String[]	double[] arr	Mengembalikan array yang berisi pengelesaian SPL dengan solusi unik
splInfiniteSol	Public double[]	double[][] m	Mengembalikan matrix yang merupakan penyelesaian dari SPL dengan solusi tak terhingga
displayInfiniteSol	Public void	String[] result	Menampilkan solusi SPL yang memiliki solusi tak hingga
infiniteSol2Arr	Public String[]	double[][] result	Mengubah matrix hasil penyelesaian SPL menjadi Array of string

Table 10 Class SPLSolver, penjelasan mengenai beberapa method yang ada pada Class SPLSolver

3.11 SolusiBalikan

Nama	Tipe	Parameter	Deskripsi
SolveBalikan	Public double[]	double[][] m	Method menghasilkan penyelesaian dari matriks balikan
turnToZero	Public double	double[] m	

Table 11 Class SolusiBalikan, penjelasan mengenai beberapa method yang ada pada Class SolusiBalikan

3.12 Bicubic

Nama	Tipe	Parameter	Deskripsi
bicubic	Public double	double[][] m double a double b	Mengembalikan nilai f(a,b)
bacaMatrixBicubic	Public double[][]	-	Input Matriks bicubic 4x4 yang dicari melalui keyboard
bacaABBicubic	Public double[]	-	Input nilai a,b yang dicari melalui keyboard
fileMatrixBicubic	Public double[][]	double[][] m	Mengembalikan matrix 4x4 dari inputan file
fileABBicubic	Public double[]	double[][] m	Mengembalikan nilai a dan b dari matriks yang merupakan inputan file

Table 12 Class Bicubic, penjelasan mengenai beberapa method yang ada pada Class Bicubic

3.13 InterpolasiPolinom

Nama	Tipe	Parameter	Deskripsi
inputInterpolasi	Public double[][]	int n	Melakukan input titik dari keyboard untuk Interpolasi Polinom sesuai format

InputtoAugmented	Public double[][]	double[][] mInput	Mengubah input interpolasi polinom menjadi matriks augmented
resultPolinom	Public double[]	double[][] mInterpolasi	Melakukan kalkulasi interpolasi polinom menggunakan metode Gauss
EstimasiFungsi	Public double	double[] a double x	Melakukan kalkulasi estimasi nilai fungsi dari persamaan yang duhasilkan dari interpolasi polinom
displayFx	Public void	double[] mRes	Menampilkan persamaan hasil interpolasi polinom sesuai format
inputInterpolasiKey	Public void		Melakukan input dari keyboard banyak titik untuk dilakukan kalkulasi interpolasi polinom
fileMatrixInterpolasi	Public void	double[][] m	Mengubah matriks hasil bacaan dari file menjadi matriks yang siap untuk di kalkulasi
findVal	public static double	double[][] m	Mengambil nilai x yang akan dicari estimasinya dari matriks hasil bacaan file

Table 13 Class InterpolasiPolinom, penjelasan mengenai beberapa method yang ada pada Class InterpolasiPolinom

3.14 RegMatrix

Nama	Tipe	Parameter	Deskripsi
HasilRegresi	Public void	double[][] m double[][] m2	Mencari persamaan regresi serta menghitung nilai hampiran suatu fungsi

Table 14 Class RegMatrix, penjelasan mengenai beberapa method yang ada pada Class RegMatrix

3.15 File

Nama	Tipe	Parameter	Deskripsi
inputFileName	public string	-	Method untuk Mengembalikan path file input yang telah divalidasi
fileRow	public int	String fileName	Menghitung baris dari matriks yang ada di file
fileCol	public int	String fileName	Mengembalikan kolom dari matrix yang ada di file
fileMatrix	public double[][]	String fileName	Mengubah file menjadi matrix
writeFile	public boolean	String dir double[][] m	Method untuk Menuliskan hasil operasi matriks ke file.
writeSPLSol	public boolean	String dir, String[] SPLsolved	Method untuk menuliskan hasil penyelesaian SPL ke file

writeDeterminan	public boolean	String dir double[][] m double det	Method untuk menuliskan hasil perhitungan determinan matriks ke file
writeInvers	public boolean	String dir double[][] m double[][] mInverse	Method untuk menuliskan hasil inverse matriks ke file
writeFailInverse	public boolean	String dir double[][] m	Method untuk menuliskan pesan error bahwa matriks tidak memiliki invers ketika nilai determinannya 0
writeInterpolasi	public boolean	String dir, double[][] a, double findValOf, double estimateVal	Menuliskan hasil perhitungan Interpolasi ke file

Table 15 Class File, penjelasan mengenai beberapa method yang ada pada Class File

Bab 4
Eksperimen

4.1 Eksperimen beberapa penyelesaian SPL

Penyelesaian SPL dengan Crammer & Solusi Balikan

Persoalan 1

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

Hasil Test Case

```
SPL Metode Matrix Inverse
Matrix memiliki determinan = 0
Matrix tidak dapat diselesaikan dengan metode ini
```

Persoalan 2

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

Hasil Test Case

```
SPL Metode Crammer
X1 = -0.22432432432432434
X2 = 0.18243243243243243
X3 = 0.7094594594594594
X4 = -0.2581081081081081
```

Penyelesaian SPL dengan Gauss

Persoalan
$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$$
Hasil Test Case
TERMINAL 0.0 0.0 0.0 1.0 -1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 SPL Memiliki Solusi Tak Hingga X1 = (1.0)t + 3.0 X2 = (2.0)t X3 = (1.0)s X4 = (1.0)t + -1.0 X5 = (1.0)t

Penyelesaian SPL dengan Gauss Jordan
Persoalan
$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$
Hasil Test Case
SPL Memiliki Solusi Tak Hingga X1 = (1.0)s X2 = (-1.0)u + 1.0 X3 = (1.0)t X4 = (-1.0)u + -2.0 X5 = (1.0)u + 1.0 X6 = (1.0)u

Persoalan 2

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

H adalah matriks Hilbert. Cobakan untuk n = 6 dan n = 10.

Hasil Test Case

Untuk n = 6

```
SPL Memiliki Solusi Unik
X1 = 36.00180458269252
X2 = -630.0486136258917
X3 = 3360.3171466206904
X4 = -7560.804286476033
X5 = 7560.871258942765
X6 = -2772.33836025545
```

Untuk n = 10

```
SPL Memiliki Solusi Unik
X1 = 106.59359787915218
X2 = -4711.36906512612
X3 = 61380.17902852587
X4 = -348702.26255321276
X5 = 976754.8012858032
X6 = -1278013.5072778307
X7 = 287097.5944128445
X8 = 1141114.4321210438
X9 = -1218331.5179490342
X10 = 383338.1048433947
```

Persoalan 3

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

Hasil Test Case

```
SPL Metode Gaus Jordan
1.0 0.0 0.0 -1.0 -1.0
0.0 1.0 -2.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
SPL Memiliki Solusi Tak Hingga
X1 = (1.0)t + -1.0
X2 = (2.0)s
X3 = (1.0)s
X4 = (1.0)t
```

Persoalan 4

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

Hasil Test Case

```
SPL Metode Gauss Jordan
1.0 0.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0 2.0
0.0 0.0 1.0 0.0 1.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
SPL Memiliki Solusi Unik
X1 = 0.0
X2 = 2.0
X3 = 1.0
X4 = 1.0
```

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

Hasil Test Case

```
SPL Metode Gauss Jordan
1.0 0.0 0.0 0.0 0.0 0.0 0.0 -8.881784197001252E-16 0.0 0.0
0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 1.0 0.0 0.0 -1.1102230246251565E-15 0.0 0.0
0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0
-0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 1.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
SPL Tidak Memiliki Solusi
```

Persoalan

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

Hasil Test Case

```
SPL Metode Gauss Jordan
1.0 0.0 0.0 0.6666666666666665 0.0
0.0 1.0 0.0 -2.6666666666666665 0.0
-0.0 -0.0 1.0 -1.0 0.0
0.0 0.0 0.0 0.0 1.0
SPL Tidak Memiliki Solusi
```

4.2 Eksperimen Determinan

Persoalan

$$\begin{pmatrix} 5 & 6 & 7 \\ 9 & 1 & 2 \\ 3 & 4 & 5 \end{pmatrix}$$
Hasil Test Case

```
Determinan Metode Reduksi Baris
-17.999999999999975
```

```
Determinan Ekspansi Kofaktor
-18.0
```

4.3 Eksperimen Invers

Persoalan

$$\begin{pmatrix} 5 & 6 & 7 \\ 9 & 1 & 2 \\ 3 & 4 & 5 \end{pmatrix}$$

Hasil Test Case

```
Matrix Invers Metode Gauss Jordan
-0.33333333333333335 0.0 0.33333333333333337
-0.10256410256410256 0.15384615384615383 0.02564102564102564
0.3076923076923077 -0.1282051282051282 -0.07692307692307693
```

```
Matrix Invers Metode Gauss Jordan
-0.33333333333333335 0.0 0.33333333333333337
-0.10256410256410256 0.15384615384615383 0.02564102564102564
0.3076923076923077 -0.1282051282051282 -0.07692307692307693
```

4.4 Eksperimen Regresi Linier Berganda

Persoalan

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.

Hasil

Persamaan Regresi

```
20.0 863.0999999999999 1530.4000000000003 587.8399999999999 19.42
863.0999999999999 54876.89 67000.09 25283.395 779.4769999999999
1530.4000000000003 67000.09 117912.32000000002 44976.866999999984 1483.4369999999997
587.8399999999999 25283.395 44976.866999999984 17278.508600000005 571.1219000000001
1.0 0.0 0.0 0.0 -3.5077781408831474
0.0 1.0 0.0 0.0 -0.0026249907458783875
0.0 0.0 1.0 0.0 7.989410472218425E-4
0.0 0.0 0.0 1.0 0.15415503019828913
persamaan regresi linier berganda adalah :
y = -3.507778 -0.002625 x1 + 0.000799 x2 + 0.154155 x3
```

Penyelesaian estimasi fungsi untuk soal

Dari persamaan di atas kita dapat mengestimasi nilai suatu fungsi, pada persoalan di atas kita diminta mengestimasi nilai Nitrous Oxide apabila Humidity bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30.

```
 Menaksir nilai fungsi
Masukkan 3 peubah 50
76
29.30
0.938434
```

4.5 Eksperimen Bikubik

Persoalan

Diberikan matriks input:

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

Tentukan nilai:

f(0,0) = ?

f(0.5, 0.5) = ?

f(0.25, 0.75) = ?

f(0.1, 0.9) = ?

Hasil Test Case

```
Bicubic Interpolation
Input Matrix 4x4
f(-1,-1): 153
f(-1,0): 59
f(-1,1): 210
f(-1,2): 96
f(0,-1): 125
f(0,0): 161
f(0,1): 72
f(0,2): 81
f(1,-1): 98
f(1,0): 101
f(1,1): 42
f(1,2): 12
f(2,-1): 21
f(2,0): 51
f(2,1): 0
f(2,2): 16
a: 0.25
b: 0.75

f(0.25,0.75) = 82.5020751953125
```

```
Bicubic Interpolation
Input Matrix 4x4
f(-1,-1): 153
f(-1,0): 59
f(-1,1): 210
f(-1,2): 96
f(0,-1): 125
f(0,0): 161
f(0,1): 72
f(0,2): 81
f(1,-1): 98
f(1,0): 101
f(1,1): 42
f(1,2): 12
f(2,-1): 21
f(2,0): 51
f(2,1): 0
f(2,2): 16
a: 0.5
b: 0.5

f(0.5,0.5) = 97.72656249999997
```

```
Bicubic Interpolation
Input Matrix 4x4
f(-1,-1): 153
f(-1,0): 59
f(-1,1): 210
f(-1,2): 96
f(0,-1): 125
f(0,0): 161
f(0,1): 72
f(0,2): 81
f(1,-1): 98
f(1,0): 101
f(1,1): 42
f(1,2): 12
f(2,-1): 21
f(2,0): 51
f(2,1): 0
f(2,2): 16
a: 0.1
b: 0.9

f(0.1,0.9) = 74.69611849999998
```

```
Bicubic Interpolation
Input Matrix 4x4
f(-1,-1): 153
f(-1,0): 59
f(-1,1): 210
f(-1,2): 96
f(0,-1): 125
f(0,0): 161
f(0,1): 72
f(0,2): 81
f(1,-1): 98
f(1,0): 101
f(1,1): 42
f(1,2): 12
f(2,-1): 21
f(2,0): 51
f(2,1): 0
f(2,2): 16
a: 0
b: 0

f(0.0,0.0) = 161.0
```

4.6 Eksperimen Polinom

Persoalan

4. **Studi Kasus Interpolasi**

a. 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	0.4	0.7	0.11	0.14	0.17	0.2	0.23
$f(x)$	0.043	0.005	0. 058	0.072	0.1	0.13	0.147

Lakukan pengujian pada nilai-nilai default berikut:

$x = 0.2$ f(x) = ?
$x = 0.55$ f(x) = ?

11

$x = 0.85$ f(x) = ?
$x = 1.28$ f(x) = ?

Hasil Test Case

Untuk kasus a

```
f(x) = -0.18455901915664086 + 10.27638398937188 x^1 + -163.91566261121488 x^2
 + 1220.854890647477 x^3 + -4346.313950916224 x^4 + 7102.3991626818315 x^5 +
-4212.434531895238 x^6
f(0.2) = 0.1299999999998896
```

Untuk kasus b

```
f(x) = -0.18455901915664086 + 10.27638398937188 x^1 + -163.91566261121488 x^2
 + 1220.854890647477 x^3 + -4346.313950916224 x^4 + 7102.3991626818315 x^5 +
-4212.434531895238 x^6
f(0.55) = 2.137571620904353
```

Untuk kasus c

```
f(x) = -0.18455901915664086 + 10.27638398937188 x^1 + -163.91566261121488 x^2
 + 1220.854890647477 x^3 + -4346.313950916224 x^4 + 7102.3991626818315 x^5 +
-4212.434531895238 x^6
f(0.85) = -66.26963931520572
```

Untuk kasus d

```
f(x) = -0.18455901915664086 + 10.27638398937188 x^1 + -163.91566261121488 x^2
 + 1220.854890647477 x^3 + -4346.313950916224 x^4 + 7102.3991626818315 x^5 +
-4212.434531895238 x^6
f(1.28) = -3485.1449016082825
```

Persoalan

b. 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 **polinom interpolasi** untuk melakukan prediksi jumlah kasus baru Covid-19 pada tanggal-tanggal berikut:

a. 16/07/2022
b. 10/08/2022
c. 05/09/2022
d. beserta masukan user lainnya berupa **tanggal (desimal) yang sudah diolah** dengan asumsi prediksi selalu dilakukan untuk tahun 2022.

Hasil Test Case

Untuk kasus a

```
f(x) = 7.1871179889419155E9 + -9.347057986136898E9 x^1 + 5.334238927153257E9
x^2 + -1.756821693899731E9 x^3 + 3.685531694209591E8 x^4 + -5.1132198648866E7
 x^5 + 4695835.438373981 x^6 + -275476.226807866 x^7 + 9372.90606280228 x^8 +
 -140.9945597747748 x^9
f(7.516) = 53.537702560424805
```

Untuk kasus b

```
f(x) = 7.1871179889419155E9 + -9.347057986136898E9 x^1 + 5.334238927153257E9
x^2 + -1.756821693899731E9 x^3 + 3.685531694209591E8 x^4 + -5.1132198648866E7
 x^5 + 4695835.438373981 x^6 + -275476.226807866 x^7 + 9372.90606280228 x^8 +
 -140.9945597747748 x^9
f(8.322) = 36.344032287597656
```

Untuk kasus c

```
f(x) = 7.1871179889419155E9 + -9.347057986136898E9 x^1 + 5.334238927153257E9
x^2 + -1.756821693899731E9 x^3 + 3.685531694209591E8 x^4 + -5.1132198648866E7
 x^5 + 4695835.438373981 x^6 + -275476.226807866 x^7 + 9372.90606280228 x^8 +
 -140.9945597747748 x^9
f(9.167) = -667.6967697143555
```

Untuk kasus d

```
f(x) = 7.1871179889419155E9 + -9.347057986136898E9 x^1 + 5.334238927153257E9
x^2 + -1.756821693899731E9 x^3 + 3.685531694209591E8 x^4 + -5.1132198648866E7
 x^5 + 4695835.438373981 x^6 + -275476.226807866 x^7 + 9372.90606280228 x^8 +
 -140.9945597747748 x^9
f(10.226) = -567112.2384338379
```

Persoalan

c. Sederhanakan fungsi

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

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 $h = (2 – 0)/5 = 0.4$.

```
f(x) = 2.0352572500000035 x^1 + -3.5526817708333502 x^2 + 3.2371145833333594 x^3
 + -1.4212662760416825 x^4 + 0.2362565104166701 x^5
f(2.0) = 0.5766520000000037
```

Bab 5
Kesimpulan, Saran, dan Refleksi

5.1 Kesimpulan

Program yang dibuat dapat digunakan untuk menghitung berbagai persoalan.

1. Menghitung solusi SPL dengan menggunakan beberapa metode seperti Gauss, Gauss Jordan, Matriks balikan, dan kaidah crammer.
2. Menghitung determinan matriks dengan ekspansi kofaktor dan metode reduksi baris.
3. Menghitung matriks balikan
4. Menyelesaikan persoalan interpolasi polinom, regresi linier berganda, serta bikubik.

5.2 Saran

Studi kasus yang diberikan masih sedikit sehingga kurang bisa meng-*cover* beberapa test yang ingin dilakukan.

5.3 Refleksi

Dari tugas besar yang telah dilakukan membuat kami banyak mengexplore berbagai hal mulai dari belajar bahasa pemrograman yang baru yaitu java serta hal-hal lainnya

Daftar Pustaka

B.Rowe, Daniel. BiLinear, Bicubic and In Between Spline Interpolation (https://www.mssc.mu.edu/~daniel/pubs/RoweTalkMSCS_BiCubic.pdf). Diakses pada 24 September 2022.

Multiple Linear Regression Model With Normal Equation. Diakses pada 27 September 2022 dari sumber https://www.geeksforgeeks.org/multiple-linear-regression-model-with-normal-equation/

Munir, R(2022). IF2123 Aljabar Linier dan Geometri – Semester I Tahun 2022/2023. Diakses pada 20 September 2022.

Syntax Bahasa Java Java Tutorial. Diakses pada 15 September 2022 dari sumber https://www.w3schools.com/java/default.asp.

Repositori

https://github.com/althaafka/Algeo01-21077.git