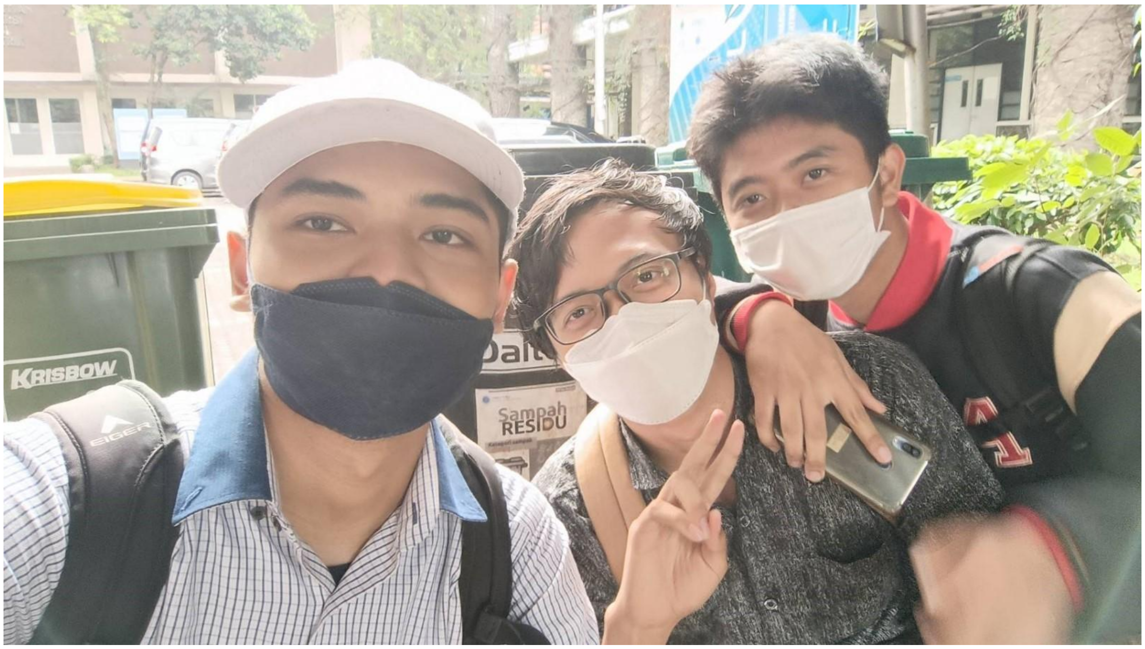
Laporan Tugas Besar 1
IF 2123 Aljabar Linier dan Geometri
“Sistem Persamaan Linier, Determinan, dan Aplikasinya”



Disusun oleh:
Kelompok 14 (Lele)

Anggota:

- Moh. Aghna Maysan Abyan (13521076)
- Haidar Hamda (13521105)
- Reza Pahlevi Ubaidillah (13521165)

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

SEMESTER I TAHUN AJARAN 2022/2023

DAFTAR ISI

BAB I
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 ($x = A^{-1}b$), 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, 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.

1.1 Interpolasi Polinom

Persoalan interpolasi polinom adalah sebagai berikut: Diberikan n+1 buah titik berbeda, (x_0, y_0), (x_1, y_1), ..., $(x\ , y\ )$. Tentukan polinom $p\ (x)$ yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga $y_i = p\ (x_i)$ untuk i = 0, 1, 2, …, n.

1.2 Bicubic Interpolation

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.

1.3 Regresi Linier Berganda

Regresi Linear (akan dipelajari lebih lanjut di Probabilitas dan Statistika) merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom.

1.4 Spesifikasi Tugas

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 II
TEORI SINGKAT

2.1 Metode Eliminasi Gauss

Metode Eliminasi Gauss adalah sebuah metode yang digunakan untuk menyelesaikan suatu sistem persamaan linier. Cara Metode Eliminasi Gauss bekerja adalah sistem persamaan linier yang ada dibentuk sebagai sebuah matriks *augmented*, setelah itu dilakukan salah satu Operasi Baris Elementer (OBE) berikut untuk setiap langkahnya:

- Menukar satu baris dengan baris lainnya,
- Melakukan operasi perkalian sebuah baris dengan bilangan selain nol, dan
- Menambah sebuah kelipatan salah satu baris dengan baris lainnya.

OBE terus dilakukan hingga terbentuk sebuah matriks segitiga atas (*upper triangular matrix*), yaitu sebuah matriks yang dimana semua angka dibawah diagonal utama bernilai 0. Hasil dari matriks segitiga atas tersebut dapat digunakan untuk menyelesaikan sistem persamaan linier yang ada.

2.2 Metode Eliminasi Gauss-Jordan

Metode Eliminasi Gauss-Jordan adalah metode lainnya yang dapat digunakan untuk menyelesaikan suatu sistem persamaan linier. Cara Metode Eliminasi Gauss-Jordan bekerja tidaklah jauh dari Metode Eliminasi Gauss. Langkah awalnya mirip dengan Metode Eliminasi Gauss, yaitu suatu sistem persamaan linier dibentuk sebagai sebuah matriks *augmented*, lalu dilakukan salah satu OBE untuk setiap langkahnya, terlampir pada poin 2.1.

Setelah terbentuknya sebuah matriks segitiga atas (*upper triangular matrix*), dimulailah perbedaan antara Metode Eliminasi Gauss dan Metode Eliminasi Gauss-Jordan. Pada Metode Eliminasi Gauss-Jordan, OBE diterapkan lagi, namun digunakan untuk membentuk sebuah matriks segitiga bawah (*lower triangular matrix*), yaitu sebuah matriks yang dimana semua angka diatas diagonal utama bernilai 0. Setelah didapatkan matriks segitiga bawah dari sistem persamaan linier tersebut, didapatkan sebuah matriks yang bersifat eselon tereduksi (*reduced echelon form*), yaitu sebuah matriks yang memiliki nilai 1 pada diagonal utamanya dan nilai 0 pada tempat-tempat lainnya di dalam matriks tersebut. Hasil dari matriks eselon tereduksi tersebut dapat digunakan untuk menyelesaikan sistem persamaan linier yang ada.

2.3 Determinan

Determinan merupakan sebuah nilai yang hanya bisa didapatkan pada sebuah matriks persegi. Misalkan, A adalah sebuah matriks berukuran 2x2:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

Maka, didapatkan determinannya sebagai det(A) = $a_{11}a_{22} - a_{12}a_{21}$.

Misalkan, B adalah sebuah matriks berukuran 3x3:

$$B = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Maka, didapatkan determinannya sebagai det(B) = $(a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32}) - (a_{31}a_{22}a_{13} + a_{32}a_{23}a_{11} + a_{33}a_{21}a_{12})$.

Semua matriks persegi berukuran n x n bisa didapatkan determinannya melalui sebuah OBE yang menghasilkan matriks segitiga atas ataupun matriks segitiga bawah.

Misalkan, C adalah sebuah matriks berukuran 4x4:

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

Melalui berbagai rangkaian OBE, diperoleh sebuah matriks segitiga atas sebagai berikut:

$$C = \begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & 13/2 & -3/2 & 9/2 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

Dari rangkaian OBE tersebut, dilakukan pertukaran baris sebanyak 0 kali, sehingga didapatkan determinan dari matriks C adalah:

$$det(C) = (-1)^0(2)(13/2)(1)(4) = 52$$

Sehingga, dapat diperoleh bahwa sebuah matriks berukuran n x n dapat ditemukan determinannya melalui Metode Eliminasi Gauss (yang memberikan hasil matriks segitiga atas/bawah) lalu kita kalikan tiap nilai pada diagonal utama, dan setelah itu kita kalikan hasilnya dengan nilai $(-1)^p$, yang dimana p tersebut sama dengan jumlah pergantian baris selama melakukan OBE.

Cara alternatif mencari determinan sebuah matriks adalah melalui ekspansi kofaktor, yang akan dibahas lebih lanjut pada poin 2.5.

Misalkan, B adalah sebuah matriks berukuran 3x3:

$$B = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Misalkan a(mn) adalah elemen matriks pada baris m dan kolom n, serta c(mn) adalah kofaktor dari elemen matriks pada baris m dan kolom n. Determinannya dapat dihitung melalui salah satu persamaan berikut:

- Dihitung secara baris
  det(B) = $a_{11}c_{11} + a_{12}c_{12} + a_{13}c_{13}$
  det(B) = $a_{21}c_{21} + a_{22}c_{22} + a_{23}c_{23}$
  det(B) = $a_{31}c_{31} + a_{32}c_{32} + a_{33}c_{33}$
- Dihitung secara kolom
  det(B) = $a_{11}c_{11} + a_{21}c_{21} + a_{31}c_{31}$
  det(B) = $a_{12}c_{12} + a_{22}c_{22} + a_{32}c_{32}$
  det(B) = $a_{13}c_{13} + a_{23}c_{23} + a_{33}c_{33}$

2.4 Matriks Balikan

Matriks balikan adalah sebuah matriks yang memiliki nilai sedemikian rupa sehingga jika dikalikan dengan matriks aslinya akan bernilai sebuah matriks identitas. Misalkan B adalah sebuah matriks berukuran 3x3, dan B^{-1} adalah invers/balikan dari matriks B, didapatkan:

$$BB^{-1} = B^{-1}B = I$$

Matriks balikan dapat dicari melalui 2 cara, yaitu metode eliminasi Gauss-Jordan dan matriks adjoin. (dibahas lebih lanjut pada 2.6.)

Pada metode eliminasi Gauss-Jordan, dilakukan sistem operasi sebagai berikut:

$$[B|I] \sim [I|B^{-1}]$$

Pada matriks adjoin, perlu dicari nilai matriks adjoin dan determinannya. Setelah itu, invers matriks dapat dicari melalui rumus:

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

2.5 Matriks Kofaktor

Matriks kofaktor adalah sebuah matriks yang mengandung nilai kofaktor dari tiap elemen. Rumus untuk mencari nilai kofaktor tiap elemen adalah sebagai berikut:

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

Dengan C_{ij} melambangkan kofaktor dari elemen yang terletak pada baris i dan kolom j dan M_{ij} melambangkan minor matriks elemen yang terletak pada baris i dan kolom j. Minor matriks dapat dicari dengan melakukan determinan dari sebuah matriks kecil yang berisi elemen-elemen yang tidak terletak pada baris i dan kolom j.

2.6 Matriks Adjoin

Sebelum memasuki definisi Matriks adjoin, perlu dipahami terlebih dahulu definisi dari *transpose* matriks. *Transpose* matriks adalah operasi pertukaran pada matriks yang dimana tiap elemen pada baris matriks berubah menjadi kolom matriks dan begitu juga sebaliknya.

Matriks adjoin adalah hasil *transpose* dari matriks kofaktor. Misalkan B adalah sebuah matriks kofaktor berukuran 3x3:

$$B = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Maka, matriks adjoinnya adalah adj(B):

$$adj(B) = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix}$$

2.7 Kaidah Cramer

Kaidah cramer adalah salah satu cara penyelesaian sistem persamaan linier. Misalkan, didapatkan sebuah sistem persamaan linier sebagai berikut:

$$
\begin{aligned}
a_1x + b_1y + c_1z &= d1 \\
a_2x + b_2y + c_2z &= d2 \\
a_3x + b_3y + c_3z &= d3
\end{aligned}
$$

Determinan matriks koefisiennya adalah:

$$
D = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}
$$

Setelah itu, cari determinan dimana salah satu kolom dari D diganti dengan konstanta d.

$$
D_x = \begin{vmatrix} d_1 & b_1 & c_1 \\ d_2 & b_2 & c_2 \\ d_3 & b_3 & c_3 \end{vmatrix}, D_y = \begin{vmatrix} a_1 & d_1 & c_1 \\ a_2 & d_2 & c_2 \\ a_3 & d_3 & c_3 \end{vmatrix}, D_z = \begin{vmatrix} a_1 & b_1 & d_1 \\ a_2 & b_2 & d_2 \\ a_3 & b_3 & d_3 \end{vmatrix}
$$

Didapatkan solusi dari sistem persamaan linear adalah:

$$
x = \frac{D_x}{D}, y = \frac{D_y}{D}, z = \frac{D_z}{D}
$$

2.8 Interpolasi Polinom

Interpolasi polinom adalah sebuah interpolasi yang menggunakan asumsi bahwa pola data yang digunakan mengikuti pola polinomial. Metode interpolasi polinom digunakan untuk menginterpolasi n buah titik yang dimana akan terbentuk sebuah fungsi berbentuk $p\ (x) = a_0 + a_1x + a_2x^2 + \ldots + a\ x^n$.

Dari fungsi tersebut, dapat ditemukan sistem persamaan linier sebanyak n yang akan diperoleh sebuah sistem persamaan linier sebagai berikut:

$$
\begin{aligned}
a_0 + a_1x_0 + \ldots + a_nx_0^n &= y_0 \\
a_0 + a_1x_1 + \ldots + a_nx_1^n &= y_1 \\
a_0 + a_1x_2 + \ldots + a_nx_2^n &= y_2 \\
&. \\
&. \\
&. \\
a_0 + a_1x_n + \ldots + a_nx_n^n &= y_n
\end{aligned}
$$

Setelah perhitungan fungsi diatas, akan diperoleh nilai $a_0, a_1, a_2, \ldots, a$. Nilai yang diperoleh, dapat di substitusikan kepada fungsi $p\ (x) = a_0 + a_1x + a_2x^2 + \ldots + a\ x^n$. Dan yang terakhir, dengan substitusi nilai x (yaitu titik yang sedang dicari) kedalam fungsi, akan ditemukan nilai y nya.

2.9 Interpolasi Bicubic

Interpolasi *bicubic* merupakan salah satu metode yang digunakan untuk mempertajam dan/atau memperbesar sebuah gambar. Metode interpolasi *bicubic* lebih sering dipilih dibandingkan interpolasi bilinier atau interpolasi *nearest neighbor* dikarenakan hasil citra dari metode interpolasi bicubic dapat menghasilkan gambar yang lebih jernih dibandingkan kedua metode lainnya. Dalam memproses sebuah gambar, interpolasi *bicubic* mengambil 16 pixel (ukuran 4x4), sehingga hasilnya lebih jernih dibandingkan metode interpolasi bilinier yang hanya mengambil 4 pixel (ukuran 2x2).

2.10 Regresi Linier Berganda

Regresi linier berganda merupakan salah satu bentuk analisis regresi linier (yaitu, metode yang digunakan untuk memprediksi nilai suatu fungsi) dengan banyak peubah/variabel bebas. Metode regresi linier berganda memerlukan rumus estimasi normal untuk regresi linier berganda, yaitu:

$$
\begin{array}{rcl}
n\hat{\beta}_0 + \hat{\beta}_1 \sum_{i=1}^{n} x_{i1} + \hat{\beta}_2 \sum_{i=1}^{n} x_{i2} + \cdots + \hat{\beta}_k \sum_{i=1}^{n} x_{ik} & = & \sum_{i=1}^{n} y_i \\
\hat{\beta}_0 \sum_{i=1}^{n} x_{i1} + \hat{\beta}_1 \sum_{i=1}^{n} x_{i1}^2 + \hat{\beta}_2 \sum_{i=1}^{n} x_{i1}x_{i2} + \cdots + \hat{\beta}_k \sum_{i=1}^{n} x_{i1}x_{ik} & = & \sum_{i=1}^{n} x_{i1}y_i \\
\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots & & \\
\hat{\beta}_0 \sum_{i=1}^{n} x_{ik} + \hat{\beta}_1 \sum_{i=1}^{n} x_{ik}x_{i1} + \hat{\beta}_2 \sum_{i=1}^{n} x_{ik}x_{i2} + \cdots + \hat{\beta}_k \sum_{i=1}^{n} x_{ik}^2 & = & \sum_{i=1}^{n} x_{ik}y_i
\end{array}
$$

Dari rumus diatas, akan diperoleh nilai untuk $_0$, $_1$, $_2$, …, . Substitusikan nilai-nilai serta nilai-nilai untuk x_0, x_1, x_2, …, x , maka nilai-nilai tersebut dapat memenuhi fungsi dibawah ini untuk mencari nilai y:

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

Dengan ϵ_i didefinisikan sebagai nilai eror fungsi/residual.

BAB III
IMPLEMENTASI PUSTAKA DAN PROGRAM DALAM JAVA

3.1 Class OBE

Atribut

Nama	Tipe	Deskripsi
-	-	-

Method

Nama	Tipe	Parameter	Deskripsi
changerows	void	int x int y double[][] m	Membuat kontainer matriks baru yang berisi elemen-elemen baris berjumlah x dan kolom berjumlah j, lalu dilakukan pertukaran satu baris dengan baris lainnya.
addsubrows	void	int x int y double[][] m boolean add	Melakukan penjumlahan sebuah baris dalam matriks tersebut dengan matriks lainnya.
multdivrows	void	int x double[][] m boolean mult double multiplydigit	Melakukan perkalian tiap elemen pada sebuah baris matriks dengan sebuah bilangan skalar multiplydigit.
findBaseVarIdx	int	int x double[][] m	Mencari indeks untuk elemen yang memiliki nilai 1 utama pada baris x.
findNonZero	int	double[][] m int y int a int b	Mencari indeks pertama yang berisi elemen selain nol. (Mencari berdasarkan kolom)
isColumnZero	boolean	double[][] m	“True” apabila satu kolom memiliki

		int y int a int b	elemen yang semuanya bernilai 0.
isRowZero	boolean	double[][] m int x int a int b	“True” apabila satu baris memiliki elemen yang semuanya bernilai 0.
toEchelon	void	double[][] m boolean reduced	Mengubah matriks menjadi bentuk eselon (Apabila “true”, terbentuk eselon tereduksi”)
getFirstNonZeroIdx	int	double[][] m	Mencari baris terakhir yang masih memiliki elemen tidak nol.

3.2 Class Matriks

Atribut

Nama	Tipe	Deskripsi
-	-	-

Method

Nama	Tipe	Parameter	Deskripsi
transpose	double[][]	double[][] m	Menukar elemen baris pada matriks m menjadi elemen kolom, dan sebaliknya.
kofaktor	double[][]	double[][] m int a int b	Mencari nilai kofaktor untuk tiap elemen pada matriks m.
adjoint	double[][]	double[][] m	Mendapatkan bentuk matriks adjoint dari matriks m (adjoint = transpose kofaktor).
inverse	double[][]	double[][] m	Mendapatkan bentuk matriks inverse dari matriks m.
fillNaN	void	double[] m	Mengisi semua elemen dengan “NaN” pada

			array 1 dimensi.
fillNaN	void	double[][] m	Mengisi semua elemen dengan “NaN” pada *array* 2 dimensi.
multiplyMatrix	double[][]	double[][] a double[][] b	Melakukan operasi perkalian antara matriks a dengan matriks b.
getLinear	double[]	double[][] m	Mendapatkan seluruh elemen baris i pada sebuah matriks.

3.3 Class determinant

Atribut

Nama	Tipe	Deskripsi
-	-	-

Method

Nama	Tipe	Parameter	Deskripsi
ekspansiKofaktor	double	double[][] m	Mencari nilai determinan matriks m menggunakan metode ekspansi kofaktor.
kofaktor	double[][]	double[][] m int a int b	Mengambil nilai kofaktor dari tiap elemen matriks m.
reduksiBaris	double	double[][] m	Mengubah matriks m menjadi bentuk segitiga atas, didapatkan nilai determinan melalui perkalian diagonal utama.
swap	void	double[][] m int i1 int i2	Menukar elemen pada baris i1 dengan elemen pada baris i2.

3.4 Class spl

Atribut

Nama	Tipe	Deskripsi
-	-	-

Method

Nama	Tipe	Parameter	Deskripsi
cramer	double[]	double[][] m	Mengembalikan solusi SPL matriks m dengan metode Cramer.
insertCol	void	double[][] a double[][] b int k	Menukar sebuah kolom pada matriks a dengan matriks b.
isSole	boolean	double[][] m int row	“True” apabila dalam satu baris, hanya terdapat 1 elemen.
hasNoSolution	boolean	double[][] m	“True” = SPL tidak memiliki solusi.
listDoubleToString	string[]	double[] m	Mengubah *array of double* menjadi *array of string*.
findNaN	int	double[] m	Mencari indeks bernilai “NaN” pertama.
isAllZero	boolean	double[] m	“True” apabila semua elemen pada sebuah baris bernilai 0.
addList	void	double[] m double[] n	Menjumlahkan *array* linier m dengan *array* linier n.
copyList	double[]	double[] m	Menyalin *array* linier m, lalu dipindahkan ke *array* linier yang baru.
multiplyList	double[]	double[] m double k	Mengalikan semua elemen pada *array* linier m dengan konstanta k.

resToParametric	String	double[] res int var double[] rawres	Mengubah bentuk “NaN” menjadi bentuk parametrik.
resListToParametric	String[]	double[] res double[][] m double[][] nm boolean parametric	Mengubah bentuk “NaN” menjadi bentuk parametrik, lalu parametrik tersebut dibuat *array*nya.
eliminasiGauss	double[]	double[][] m	Melakukan operasi eliminasi Gauss pada sebuah matriks, namun hanya bisa mengeluarkan hasil variabelnya dalam bentuk angka (dapat mengeluarkan hasil “NaN”).
eliminasiGauss	String[]	double[][] m boolean parametric	Melakukan operasi eliminasi Gauss pada sebuah matriks, mengeluarkan hasil variabelnya dalam bentuk angka atau parametrik.
eliminasiGaussJordan	double[]	double[][] m	Melakukan operasi eliminasi Gauss-Jordan pada sebuah matriks, namun hanya bisa mengeluarkan hasil variabelnya dalam bentuk angka (dapat mengeluarkan hasil “NaN”).
eliminasiGaussJordan	String[]	double[][] m boolean parametric	Melakukan operasi eliminasi Gauss-Jordan pada sebuah matriks, mengeluarkan hasil variabelnya dalam bentuk angka atau parametrik.
inverseMatrixMethod	double[]	double[][] m	Melakukan operasi invers pada matriks m untuk mendapatkan solusi SPL-nya.

3.5 Class txtscanner

Atribut

Nama	Tipe	Deskripsi

-	-	-

Method

Nama	Tipe	Parameter	Deskripsi
getMatrixFile	double[][]	String file	Mengembalikan matriks dari file.
getMatrixSize	int[]	String file	Mengembalikan ukuran matriks.

3.6 Class txtwriter

Atribut

Nama	Tipe	Deskripsi
-	-	-

Method

Nama	Tipe	Parameter	Deskripsi
writeMatrix	void	double[][] m String file	Menuliskan matriks m ke file.
writeRegresi	void	double[][] m double[] solusi double result double[] x String file	Menuliskan hasil regresi matriks m ke file.
writeInterpol	void	String file double[] xs double[] x double result	Menuliskan hasil interpolasi polinom matriks m ke file.
writeSolusi	void	String file double[] solusi	Menuliskan solusi SPL matriks m ke file.
writeSolusiParametrik	void	String file String[] solusi	Menuliskan solusi parametrik matriks m ke file.

3.7 Class utils

Atribut

Nama	Tipe	Deskripsi
-	-	-

Method

Nama	Tipe	Parameter	Deskripsi
readMatrix	void	double[][] m int n int ma	Menerima input dan meng-*assign* matriks.
makeMatrix	double[][]	-	Membuat matriks berdasarkan input.
printMatrix	void	double[][] m	Mencetak matriks.
forceCopyMatrix	void	double[][] min double[][] mout	Menyalin sebuah matriks tanpa menganalisis ukuran matriksnya (min ≠ mout).
copyMatrix	void	double[][] min double[][] mout	Menyalin sebuah matriks dengan menganalisis ukuran kedua matriksnya (min = mout).
fillZero	void	double[][] m	Isi semua elemen pada matriks m dengan 0.
max	int	int a int b	Mengembalikan nilai terbesar antara 2 elemen (a dan b).
abs	double	double a	Mengeluarkan nilai mutlak bilangan.
getMenu	void	-	Menampilkan dan menerima input pada menu utama.
splMenu	void	-	Menampilkan dan menerima input pada menu SPL.
determinantMenu	void	-	Menampilkan dan menerima input

			pada menu determinan.
isSameSize	boolean	double[][] m1 double[][] m2	“True” apabila m1 dan m2 memiliki ukuran yang sama.
printSolusi	void	double[] x	Mencetak hasil SPL dalam bentuk angka.
printSolusiPar	void	String[] x	Mencetak hasil SPL dalam bentuk angka dan parametrik.
isSquare	boolean	double[][] m	Mengembalikan “true” apabila matriks bersifat n x n.
augmentedtoMatrix	void	double[][] m double[][] a double[][] b	Mengubah bentuk *augmented* ke bentuk matriks.
matrixtoAugmented	double[][]	double[][] a double[][] b	Mengubah bentuk matriks ke bentuk *augmented.*
sumCol	double	double[][] m int j	Melakukan penjumlahan semua elemen di dalam sebuah kolom.
sumRow	double	double[][] m int i	Melakukan penjumlahan semua elemen di dalam sebuah baris.

3.8 Class Global

Atribut

Nama	Tipe	Deskripsi
isRun	boolean	“True” apabila program sedang jalan.

Method

Nama	Tipe	Parameter	Deskripsi
-	-	-	-

BAB IV
EKSPERIMEN

4.1 Menentukan solusi *Ax* = *b*

4.1.a

a.

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

```
input dari keyboard(1) atau file(2):
2
masukkan nama file beserta path nya:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\1a.txt

1.0 1.0 -1.0 -1.0 1.0
0.0 1.0 -1.6666666666666667 -1.0 -1.3333333333333333
-0.0 0.0 1.0 -1.0 0.9999999999999999
-0.0 -0.0 -0.0 -0.0 1.0
0.0 0.0 0.0 0.0 0.0

1.0 1.0 -1.0 -1.0 1.0
2.0 5.0 -7.0 -5.0 -2.0
2.0 -1.0 1.0 3.0 4.0
5.0 2.0 -4.0 2.0 6.0
tidak ada solusi
```

Matriks diatas tidak memiliki solusi, dikarenakan setelah melakukan OBE pada matriks A diatas, ditemukan sebuah baris yang tertulis $-0x_1 + -0x_2 + -0x_3 + -0x_4 = 1$. Tidak ada variabel yang dapat memenuhi solusi tersebut.

4.1.b

b.

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

```
1.0 -1.0 0.0 0.0 1.0 3.0
0.0 1.0 0.0 -1.5 -0.5 1.5
0.0 0.0 0.0 1.0 -1.0 -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

var: 3 i: 2
var: 1 i: 1
var: 0 i: 0
1.0 -1.0 0.0 0.0 1.0 3.0
1.0 1.0 0.0 -3.0 0.0 6.0
2.0 -1.0 0.0 1.0 -1.0 5.0
-1.0 2.0 0.0 -2.0 -1.0 -1.0
x1:x5 + 3.0
x2:2.0x5
x3:x3
x4:x5 - 1.0
x5:x5
```

Dari perhitungan matriks diatas, didapatkan sebuah persamaan parametrik, yaitu $x_1 = x_5 + 3; x_2 = 2x_5; x_3 = x_3; x_4 = x_5 - 1; x_5 = x_5$

4.1.c

c.

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

```
masukkan nama file beserta path nya:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\1c.txt

0.0 1.0 0.0 0.0 1.0 0.0 2.0
0.0 0.0 0.0 1.0 1.0 0.0 -1.0
-0.0 -0.0 -0.0 -0.0 1.0 -1.0 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 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0

var: 4 i: 2
var: 3 i: 1
var: 1 i: 0
0.0 1.0 0.0 0.0 1.0 0.0 2.0
0.0 0.0 0.0 1.0 1.0 0.0 -1.0
0.0 1.0 0.0 0.0 0.0 1.0 1.0
x1:x1
x2:-1.0x6 + 1.0
x3:x3
x4:-1.0x6 - 2.0
x5:x6 + 1.0
x6:x6
```

Dari perhitungan matriks diatas, didapatkan sebuah persamaan parametrik, yaitu $x_1 = x_1; x_2 = -x_6 + 1; x_3 = x_3; x_4 = -x_6 - 2; x_5 = x_6 + 1; x_6 = x_6$

4.1.d

d.

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

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

Untuk n = 6:

```
1.0 0.0 -0.0 0.0 -0.0 0.0 34.04545091596756
0.0 1.0 0.0 -0.0 0.0 -0.0 -538.0620764325583
0.0 0.0 1.0 0.0 -0.0 0.0 2667.7336889902977
0.0 0.0 0.0 1.0 0.0 -0.0 -5717.316309036693
0.0 0.0 0.0 0.0 1.0 0.0 5537.632596120627
0.0 0.0 0.0 0.0 0.0 1.0 -1988.6453783870872
0.0 0.0 0.0 0.0 0.0 0.0 0.0

1.0 0.5 0.33333 0.25 0.2 0.16667 1.0
0.5 0.33333 0.25 0.2 0.16667 0.14286 0.0
0.33333 0.25 0.2 0.16667 0.14286 0.125 0.0
0.25 0.2 0.16667 0.14286 0.125 0.11111 0.0
0.2 0.16667 0.14286 0.125 0.11111 0.1 0.0
0.16667 0.14286 0.125 0.11111 0.1 0.090909 0.0
x1:34.04545091596756
x2:-538.0620764325583
x3:2667.7336889902977
x4:-5717.316309036693
x5:5537.632596120627
x6:-1988.6453783870872
```

Untuk n = 10:

```
1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 26.63807601333056
-0.0 1.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -334.5787258402523
0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1156.0154703544192
-0.0 -0.0 -0.0 1.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -1227.9078353045031
0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 -83.29025657263035
0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 237.7822212933059
0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 -0.0 0.0 433.4410277417352
0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 -573.5698358671743
-0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 1.0 -0.0 1283.2896035272743
-0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 0.0 1.0 -924.9096609644951
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

1.0 0.5 0.33333 0.25 0.2 0.16667 0.14286 0.125 0.11111 0.1 1.0
0.5 0.33333 0.25 0.2 0.16667 0.14286 0.125 0.11111 0.1 0.09091 0.0
0.33333 0.25 0.2 0.16667 0.14286 0.125 0.11111 0.1 0.09091 0.08333 0.0
0.25 0.2 0.16667 0.14286 0.125 0.11111 0.1 0.09091 0.08333 0.07692 0.0
0.2 0.16667 0.14286 0.125 0.11111 0.1 0.09091 0.08333 0.07692 0.07143 0.0
0.16667 0.14286 0.125 0.11111 0.1 0.09091 0.08333 0.07692 0.07143 0.06667 0.0
0.14286 0.125 0.11111 0.1 0.09091 0.08333 0.07692 0.07143 0.06667 0.0625 0.0
0.125 0.11111 0.1 0.09091 0.08333 0.07692 0.07143 0.06667 0.0625 0.05882 0.0
0.11111 0.1 0.09091 0.08333 0.07692 0.07143 0.06667 0.0625 0.05882 0.055556 0.0
0.1 0.09091 0.08333 0.07692 0.07143 0.06667 0.0625 0.05882 0.055556 0.05263 0.0
x1:26.63807601333056
x2:-334.5787258402523
x3:1156.0154703544192
x4:-1227.9078353045031
x5:-83.29025657263035
x6:237.7822212933059
x7:433.4410277417352
x8:-573.5698358671743
x9:1283.2896035272743
x10:-924.9096609644951
```

4.2 Menentukan solusi SPL berbentuk matriks *augmented*

4.2.a

a.

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

```
input dari keyboard(1) atau file(2):
2
masukkan nama file beserta path nya:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\2a.txt

1.0 -1.0 2.0 -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
0.0 0.0 0.0 0.0 0.0

var: 1 i: 1
var: 0 i: 0
1.0 -1.0 2.0 -1.0 -1.0
2.0 1.0 -2.0 -2.0 -2.0
-1.0 2.0 -4.0 1.0 1.0
3.0 0.0 0.0 -3.0 -3.0
x1:x4 - 1.0
x2:2.0x3
x3:x3
x4:x4
```

Dari perhitungan matriks diatas, didapatkan sebuah persamaan parametrik, yaitu
$x_1 = x_4 - 1; x_2 = 2x_3; x_3 = x_3; x_4 = x_4$

4.2.b

b.

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

```
input dari keyboard(1) atau file(2):
2
masukkan nama file beserta path nya:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\2b.txt

1.0 0.0 4.0 0.0 4.0
0.0 1.0 0.0 4.0 6.0
-0.0 0.0 1.0 0.0 1.0
0.0 -0.0 0.0 1.0 1.0
0.0 -0.0 0.0 -0.0 0.0
-0.0 0.0 -0.0 0.0 0.0

var: 3 i: 3
var: 2 i: 2
var: 1 i: 1
var: 0 i: 0
2.0 0.0 8.0 0.0 8.0
0.0 1.0 0.0 4.0 6.0
-4.0 0.0 6.0 0.0 6.0
0.0 -2.0 0.0 3.0 -1.0
2.0 0.0 -4.0 0.0 -4.0
0.0 1.0 0.0 -2.0 0.0
x1:0.0
x2:2.0
x3:1.0
x4:1.0
```

Dari perhitungan matriks diatas, didapatkan nilai untuk tiap variabel, yaitu $x_1 = 0; x_2 = 2; x_3 = 1; x_4 = 1$

4.3 Menentukan solusi SPL berbentuk persamaan linier umum

4.3.a

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

```
input dari keyboard(1) atau file(2):
2
masukkan nama file beserta path nya:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\3a.txt

1.0 0.125 0.375 0.25 0.0
0.0 1.0 -0.2 -0.2857142857142857 0.11428571428571428
0.0 0.0 1.0 -0.19480519480519481 0.7597402597402596
0.0 -0.0 0.0 1.0 -0.2581081081081078
0.0 0.0 0.0 0.0 0.0

var: 3 i: 3
var: 2 i: 2
var: 1 i: 1
var: 0 i: 0
8.0 1.0 3.0 2.0 0.0
2.0 9.0 -1.0 -2.0 1.0
1.0 3.0 2.0 -1.0 2.0
1.0 0.0 6.0 4.0 3.0
x1:-0.22432432432432436
x2:0.1824324324324325
x3:0.7094594594594594
x4:-0.2581081081081078
```

Didapatkan nilai untuk variabelnya adalah:

$$x_1 = -0.22432432432432436$$
$$x_2 = 0.1824324324324325$$
$$x_3 = 0.7094594594594594$$
$$x_4 = -0.2581081081081078$$

4.3.b

b.

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

```
input dari keyboard(1) atau file(2):
2
masukkan nama file beserta path nya:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\3b.txt

1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 8.0
0.0 1.0 3.65684 1.0 3.65684 1.0 3.65684 1.0 0.0 57.24
0.0 0.0 1.0 0.0 1.0 17.48659361156447 10.0 17.48659361156447 14.31475868500816 344.8356260200513
0.0 0.0 0.0 1.0 1.0 1.0 0.0 0.0 0.0 15.0
0.0 0.0 -0.0 0.0 1.0 2984.2327185319405 1702.3092549750147 3086.3891245219556 2545.089099149842 58298.65163873439
0.0 0.0 -0.0 0.0 -0.0 1.0 0.5702338352881916 1.034085262597295 0.8530966713107108 19.533953907283664
0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.0 1.0 13.0
0.0 0.0 -0.0 -0.0 -0.0 -0.0 -0.0 1.0 2.387269515046585 19.968372992813848
-0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 16.168557536466775
-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

0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.0 1.0 13.0
0.0 0.0 0.0 1.0 1.0 1.0 0.0 0.0 0.0 15.0
1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 8.0
0.0 0.0 0.04289 0.0 0.04289 0.75 0.4289 0.75 0.61396 14.79
0.0 0.25 0.91421 0.25 0.91421 0.25 0.91421 0.25 0.0 14.31
0.61396 0.75 0.04289 0.75 0.0489 0.0 0.04289 0.0 0.0 3.81
0.0 0.0 1.0 0.0 0.0 1.0 0.0 0.0 1.0 18.0
0.0 1.0 0.0 0.0 1.0 0.0 0.0 1.0 0.0 12.0
1.0 0.0 0.0 1.0 0.0 0.0 1.0 0.0 0.0 6.0
0.04289 0.75 0.61396 0.0 0.04289 0.75 0.0 0.0 0.04289 10.51
0.91421 0.25 0.0 0.25 0.91421 0.25 0.0 0.25 0.91421 16.13
0.04289 0.0 0.0 0.75 0.04289 0.0 0.61396 0.75 0.04289 7.04
tidak ada solusi
```

Tidak ada solusi karena setelah dilakukan OBE segitiga atas, ditemukan sebuah baris yang memiliki persamaan $-0x_1 - 0x_2 - 0x_3 - 0x_4 - 0x_5 - 0x_6 - 0x_7 - 0x_8 - 0x_9 = 1$.

4.4 Studi Kasus Interpolasi Polinom

4.4.a

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) = ?

Untuk x = 0.2:

```
masukkan nama file beserta path nya:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\4a.txt
persamaan interpolasi polinom:
y=-0.18455901924208507x^0+10.276383991758758x^1-163.91566263816935x^2+1220.8548908084551x^3-4346.3139513733495x^4+7102.399163392868x^
5-4212.434532287046x^6
Masukkan nilai yang akan ditaksir: 0.2
hasil taksiran: 0.13000000007252066
nama dan path untuk menyimpan file:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\4asolusi1.txt
```

Untuk x = 0.55:

```
masukkan nama file beserta path nya:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\4a.txt
persamaan interpolasi polinom:
y=-0.18455901924208507x^0+10.276383991758758x^1-163.91566263816935x^2+1220.8548908084551x^3-4346.3139513733495x^4+7102.399163392868x^
5-4212.434532287046x^6
Masukkan nilai yang akan ditaksir: 0.55
hasil taksiran: 2.137571623870741
nama dan path untuk menyimpan file:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\4asolusi2.txt
```

Untuk x = 0.85:

```
masukkan nama file beserta path nya:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\4a.txt
persamaan interpolasi polinom:
y=-0.18455901924208507x^0+10.276383991758758x^1-163.91566263816935x^2+1220.8548908084551x^3-4346.3139513733495x^4+7102.399163392868x^
5-4212.434532287046x^6
Masukkan nilai yang akan ditaksir: 0.85
hasil taksiran: -66.26963930477791
nama dan path untuk menyimpan file:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\4asolusi3.txt
```

Untuk x = 1.28:

```
masukkan nama file beserta path nya:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\4a.txt
persamaan interpolasi polinom:
y=-0.18455901924208507x^0+10.276383991758758x^1-163.91566263816935x^2+1220.8548908084551x^3-4346.3139513733495x^4+7102.399163392868x^
5-4212.434532287046x^6
Masukkan nilai yang akan ditaksir: 1.28
hasil taksiran: -3485.1449018190506
nama dan path untuk menyimpan file:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\4asolusi4.txt
```

4.4.b

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.

Untuk tanggal 16/07/2022:

Bentuk desimal -> 7 + (16/31) ≈ 7.51613

```
masukkan nama file beserta path nya:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\4b.txt
persamaan interpolasi polinom:
y=5.842408558029621E8x^0-2.966757993205382E8x^1+5.7229381984038405E7x^2-5091412.969853585x^3+185374.78183215583x^4+4749.485038033828x
^5-3128.2304155671372x^6-625.0398723432754x^7+618.834334163294x^8-8.877259633537777x^9
Masukkan nilai yang akan ditaksir:
7.51613
hasil taksiran: 4.343316945466995E9
nama dan path untuk menyimpan file:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\4bsolusi1.txt
```

Untuk tanggal 10/08/2022:

Bentuk desimal -> 8 + (10/31) ≈ 8.32258

```
C:\Users\user\Documents\GitHub\Algeo01-21076\test\4b.txt
persamaan interpolasi polinom:
y=5.842408558029621E8x^0-2.966757993205382E8x^1+5.7229381984038405E7x^2-5091412.969853585x^3+185374.78183215583x^4+4749.485038033828x
^5-3128.2304155671372x^6-625.0398723432754x^7+618.834334163294x^8-8.877259633537777x^9
Masukkan nilai yang akan ditaksir: 8.32258
hasil taksiran: 9.998512502727901E9
nama dan path untuk menyimpan file:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\4bsolusi2.txt
```

Untuk tanggal 05/09/2022:

Bentuk desimal -> 9 + (5/31) ≈ 9.16

```
masukkan nama file beserta path nya:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\4b.txt
persamaan interpolasi polinom:
y=5.842408558029621E8x^0-2.966757993205382E8x^1+5.7229381984038405E7x^2-5091412.969853585x^3+185374.78183215583x^4+4749.485038033828x
^5-3128.2304155671372x^6-625.0398723432754x^7+618.834334163294x^8-8.877259633537777x^9
Masukkan nilai yang akan ditaksir: 9.16
hasil taksiran: 2.177829890189203E10
nama dan path untuk menyimpan file:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\4bsolusi3.txt
```

Untuk tanggal 01/08/2022 (contoh):

Bentuk desimal -> 8 + (1/31) ≈ 8.03226

```
masukkan nama file beserta path nya:
C:\Users\user\Documents\GitHub\Algeo01-21076\test\4b.txt
persamaan interpolasi polinom:
y=5.842408558029621E8x^0-2.966757993205382E8x^1+5.7229381984038405E7x^2-5091412.969853585x^3+185374.78183215583x^4+4749.485038033828x
^5-3128.2304155671372x^6-625.0398723432754x^7+618.834334163294x^8-8.877259633537777x^9
Masukkan nilai yang akan ditaksir: 8.03226
hasil taksiran: 7.483728599865899E9
```

4.4.c

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 $h = (2 - 0)/5 = 0.4$.

Dengan n = 5:

```
persamaan interpolasi polinom:
y=0.0x^0+2.035257250000014x^1-3.5526817708334204x^2+3.237114583333422x^3-1.4212662760416888x^4+0.23625651041670567x^5
```

Dapat ditemukan f(x) sebagai fungsi y, yaitu:

$$y = 0.23625651041670567x^5 - 1.4212662760416888x^4 + 3.2371145833333422x^3 - 3.5526817708334204x^2 + 2.035257250000014x$$

4.5 Studi Kasus Interpolasi *Bicubic*

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) = ?

Untuk f(0,0):

```
MENU
 1. Sistem Persamaaan Linier
 2. Determinan
 3. Matriks balikan
 4. Interpolasi Polinom
 5. Interpolasi Bicubic
 6. Regresi linier berganda
 7. Keluar

5
masukkan nama file beserta path nya:
D:\Kuliah\Semester 3\Algeo\Tubes i\Algeo01-21076\test\4bicubic.txt
Masukkan titik
0 0
20520.0
```

Untuk f(0.5, 0.5):

```
MENU
 1. Sistem Persamaaan Linier
 2. Determinan
 3. Matriks balikan
 4. Interpolasi Polinom
 5. Interpolasi Bicubic
 6. Regresi linier berganda
 7. Keluar

5
masukkan nama file beserta path nya:
D:\Kuliah\Semester 3\Algeo\Tubes i\Algeo01-21076\test\4bicubic.txt
Masukkan titik
0.5 0.5
2.10177894025879E7
```

Untuk f(0.25, 0.75):

```
MENU
 1. Sistem Persamaaan Linier
 2. Determinan
 3. Matriks balikan
 4. Interpolasi Polinom
 5. Interpolasi Bicubic
 6. Regresi linier berganda
 7. Keluar

5
masukkan nama file beserta path nya:
D:\Kuliah\Semester 3\Algeo\Tubes i\Algeo01-21076\test\4bicubic.txt
Masukkan titik
0.25 0.75
6995165.0702287685
```

Untuk f(0.1, 0.9):

```
MENU
 1. Sistem Persamaaan Linier
 2. Determinan
 3. Matriks balikan
 4. Interpolasi Polinom
 5. Interpolasi Bicubic
 6. Regresi linier berganda
 7. Keluar

5
masukkan nama file beserta path nya:
D:\Kuliah\Semester 3\Algeo\Tubes i\Algeo01-21076\test\4bicubic.txt
Masukkan titik
0.1 0.9
-2822767.042917425
```

4.6 Studi Kasus Regresi Linier 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.

Dari data-data tersebut, apabila diterapkan *Normal Estimation Equation for Multiple Linear Regression*, maka diperoleh sistem persamaan linear sebagai berikut.

$$20b_0 + 863.1b_1 + 1530.4b_2 + 587.84b_3 = 19.42$$

$$863.1b_0 + 54876.89b_1 + 67000.09b_2 + 25283.395b_3 = 779.477$$

$$1530.4b_0 + 67000.09b_1 + 117912.32b_2 + 44976.867b_3 = 1483.437$$

$$587.84b_0 + 25283.395b_1 + 44976.867b_2 + 17278.5086b_3 = 571.1219$$

Dari SPL tersebut, didapatkan hasil sebagai berikut:

```
matrix hasil normal estimation equation:
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
persamaan regresi linier berganda:
y=-3.50777814090711 -0.002624990745874451x1 + 7.989410472208533E-4x2 + 0.15415503019910123x3
masukkan 3 peubah yang akan ditaksir
50 76 29.30
hasil taksiran: 0.9384342262216183
```

Persamaan regresi linier bergandanya adalah:

$$y = -0.002624990745874451x_1 + 0.0007989410472208533x_2 + 0.15415503019910123x_3 - 3.50777814090711$$

Masukkan nilai estimasi Nitrous Oxide dengan *Humidity* = 50%, temperatur = 76°F, dan tekanan air = 29.30. Didapat hasil taksirannya adalah 0.9384342262216183.

BAB V
KESIMPULAN, SARAN, DAN REFLEKSI

5.1 Kesimpulan

Sistem Persamaan Linier (SPL) dapat dicari solusinya dengan mengubahnya ke dalam bentuk matriks, lalu menerapkan salah satu dari berbagai metode yang ada untuk mendapatkan nilai-nilai variabel SPL tersebut.

Meskipun begitu, ada beberapa kondisi dimana metode yang ada tidak bisa mengeluarkan nilai/hasil variabel. Misalkan, jika kita melakukan metode eliminasi Gauss atau eliminasi Gauss-Jordan, dan kita menemukan sebuah baris yang berbentuk $0x_1 + 0x_2 + \ldots + 0x_n = 1$, maka kita tidak bisa mendapatkan nilai dari tiap-tiap variabel *x*. Contoh lainnya adalah matriks balikan dan kaidah cramer, yang mana jika didapatkan determinan matriksnya sama dengan 0, maka kita tidak bisa menemukan matriksnya.

Determinan suatu matriks dapat ditemukan melalui 2 cara, yaitu metode eliminasi gauss (mengalikan tiap bilangan pada diagonal utama) dan metode ekspansi kofaktor (misal, kita mau mencari determinan matriks A, bisa dilakukan ekspansi kofaktor baris pertama $|A| = a_{11}c_{11} + a_{12}c_{12} + a_{13}c_{13}$, dengan a adalah nilai pada elemen baris i dan kolom j, serta c adalah minor matriksnya).

Penggunaan tiap fungsi matriks memiliki hubungan satu sama lain. Misalnya, jika kita memiliki matriks A berukuran n x n, dapat kita cari nilai minor matriksnya. Nilai minor matriks dapat dimanfaatkan untuk nilai kofaktor matriks A, dan setelah itu kita dapat melakukan transpose matriks A untuk mendapatkan adjoin matriks A (adj(A)). Ditambah dengan mencari nilai determinan matriks A (det(A)), dapat kita temukan invers dari matriks A, yang memiliki rumus:

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

SPL memiliki banyak penerapannya di dunia nyata, diantaranya adalah penggunaan interpolasi polinom, interpolasi *bicubic*, dan regresi linier berganda. Interpolasi polinom dapat digunakan untuk menaksir suatu nilai *x* yang belum diketahui berdasarkan nilai *x* lainnya yang telah diketahui berdasarkan fungsi *f*(*x*), penerapannya yang relevan di dunia nyata adalah interpolasi polinom digunakan untuk melakukan prediksi jumlah kasus COVID-19 pada tanggal-tanggal tertentu berdasarkan fungsi yang ada, dan juga interpolasi polinom bisa digunakan untuk menyederhanakan suatu fungsi jika diketahui interpolasi derajat dan selangnya. Untuk interpolasi *bicubic,* penerapannya adalah kita bisa menentukan nilai antara pada suatu fungsi yang penyajiannya adalah sebuah matriks. Yang terakhir, regresi linier berganda dapat digunakan pada kehidupan nyata seperti mengestimasi nilai Nitrous Oxide berdasarkan tabel yang disajikan.

5.2 Saran

Pada saat pengenalan bahasa JAVA, yang dijelaskan hanyalah pengenalan bahasa pemrograman JAVA, isinya sangat mendasar. Padahal, program yang digunakan untuk tugas besar sudah lebih tinggi daripada kata “mendasar”, ditambah lagi bahwa kami masih merasa asing dengan bahasa pemrograman yang berdasarkan OOP (*Object-Oriented Programming).* Saran dari kami, mungkin bisa diberikan PPT berisi program-program dalam bahasa JAVA yang lebih kompleks.

5.3 Refleksi

Dari pengalaman yang telah kami dapat selama mengerjakan tugas besar pertama kami di Teknik Informatika, kami mempelajari banyak hal tentang pemrograman bahasa JAVA. Kami juga mempelajari dan mulai memahami tentang *command* yang ada pada GitHub. Mungkin yang perlu kami evaluasi untuk tugas besar lainnya adalah untuk mengerjakannya seawal mungkin, supaya tidak merasa kepepet *deadline*.

REFERENSI

http://mlwiki.org/index.php/Normal_Equation#Normal_Equation

https://en.wikipedia.org/wiki/Bicubic_interpolation

https://www.youtube.com/watch?v=poY_nGzEEWM&ab_channel=Computerphile

https://cdn-edunex.itb.ac.id/29123-Geometric-and-Linear-Algebra/22545-module-2/1629846425731_Algeo-03-Sistem-Persamaan-Linier.pdf

https://cdn-edunex.itb.ac.id/29123-Geometric-and-Linear-Algebra/22545-module-2/1629846445897_Algeo-04-Tiga-Kemungkinan-Solusi-SPL.pdf

https://cdn-edunex.itb.ac.id/29123-Geometric-and-Linear-Algebra/22545-module-2/1629846460069_Algeo-05-Sistem-Persamaan-Linier-2.pdf

https://cdn-edunex.itb.ac.id/29123-Geometric-and-Linear-Algebra/22547-module-4/1629846583895_Algeo-08-Determinan-bagian1.pdf

https://cdn-edunex.itb.ac.id/29123-Geometric-and-Linear-Algebra/22547-module-4/1629846592878_Algeo-09-Determinan-bagian2.pdf

REPOSITORY GITHUB

https://github.com/obediqbal/Algeo01-21076