Tugas Besar 1 IF 2123 Aljabar Linier dan Geometri



Sistem Persamaan Linier, Determinan, dan Aplikasinya
Semester I Tahun 2020/2021

13519002 Steven Nataniel Kodyat

13519118 Cynthia Rusadi

13519154 Rafi Raihansyah Munandar

Bab I
Deskripsi Masalah

Sistem Persamaan Linier adalah sekumpulan persamaan linier yang terdiri dari beberapa variabel. Kata "sistem" di sini penting karena menunjukkan bahwa persamaan-persamaannya perlu dipertimbangkan bersamaan dan tidak berdiri sendiri. Dalam ilmu Matematika, teori sistem linear merupakan dasar aljabar linear. Aljabar linear sangat diperlukan dalam bidang Fisika, Kimia, Ilmu Komputer, dan Ekonomi.

Sistem Persamaan Linier memiliki dua bentuk umum berupa persamaan vektor dan persamaan matriks. Untuk Sistem Persamaan Linier *m* dan *n* dalam bentuk vektor dapat ditulis seperti:

$$x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} + \cdots + x_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Dalam bentuk matriks dapat ditulis seperti:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Beberapa metode dapat digunakan untuk menyelesaikan sistem persamaan linier, antara lain seperti metode eliminasi variabel, eliminasi Gauss, eliminasi Gauss-Jordan, dan kaidah Cramer. Metode-metode tersebut dapat digunakan untuk menyelesaikan SPL dalam bentuk matriks. Dengan menentukan determinan, matriks kofaktor, adjoin, dan matriks balikan, solusi SPL dapat ditentukan.

Pengaplikasian metode-metode tersebut memiliki langkah-langkah penyelesaian yang berulang dan berpola. Oleh karena itu, dapat dibuat suatu program untuk menyelesaikan SPL berbentuk matriks dengan berbagai metode dan prosedur. Dengan menggunakan bahasa pemrograman Java, program tersebut dapat direalisasikan.

SPL memiliki banyak aplikasi dalam bidang sains dan rekayasa, dua diantaranya diterapkan pada tugas besar ini, yaitu interpolasi polinom dan regresi linier.

Bab II

Teori Singkat

1. Metode Eliminasi Gauss

   Merupakan salah satu metode yang dapat digunakan untuk menyelesaikan sistem persamaan linier. Metode ini diberi nama Gauss untuk menghormati Carl Friedrich Gauss, yang menemukan Eliminasi Gauss. Metode Gauss ini menghasilkan matriks dengan bentuk eselon (*row echelon form*). Metode Eliminasi Gauss dikembangkan dari metode eliminasi, yang menghilangkan atau mengurangi jumlah variabel sehingga dapat diperoleh nilai dari suatu variabel yang bebas, dan metode eliminasi tersebut dapat dikatakan sebagai Operasi Baris Elementer (OBE).

$$\begin{bmatrix} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{bmatrix} \sim \overset{\text{OBE}}{\dots} \sim \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

2. Metode Eliminasi Gauss-Jordan

   Sedikit berbeda dari Metode Eliminasi Gauss, Metode Eliminasi Gauss-Jordan merupakan prosedur pemecahan sistem persamaan linear dengan mengubahnya menjadi bentuk matriks eselon baris tereduksi dengan Operasi Baris Elementer (OBE). Metode Eliminasi Gauss-Jordan terdiri atas 2 fase. Fase yang pertama adalah fase maju (*forward phase*) atau fase Eliminasi Gauss, yang menghasilkan nilai-nilai 0 di bawah 1 utama. Fase yang kedua adalah fase mundur (*backward phase*), yang menghasilkan nilai-nilai 0 di atas 1 utama.

$$\begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix} \overset{R1-(3/2)R2}{\sim} \begin{bmatrix} 1 & 0 & -5/4 & -11/4 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix} \underset{R2-(1/2)R3}{\overset{R1+(5/4)R3}{\sim}} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

3. Determinan

   Determinan adalah sebuah nilai yang dapat dihitung dari unsur suatu matriks persegi. Terdapat 2 metode untuk menghitung determinan, yaitu dengan reduksi baris dan ekspansi kofaktor. Dengan melakukan reduksi baris, determinan matriks dapat diperoleh dengan melakukan OBE sampai diperoleh matriks segitiga. Determinan matriks segitiga dibagi menjadi 2, yaitu matriks segitiga atas (*upper triangular*), semua elemen berada di bawah diagonal utama adalah nol, dan matriks segitiga bawah (*lower triangular*), semua elemen berada di atas diagonal utama adalah nol.

Matriks segitiga atas

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \longrightarrow \det(A) = a_{11}a_{22}a_{33}a_{44}$$

Matriks segitiga bawah

$$A = \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \longrightarrow \det(A) = a_{11}a_{22}\,a_{33}a_{44}$$

Jika menggunakan reduksi baris, maka determinannya dapat diperoleh dengan menggunakan salah satu metode di bawah. Matriks segitiga yang digunakan adalah matriks segitiga atas.

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

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

$$\text{maka } \det(A) = (-1)^p\, a'_{11}a'_{22} \dots a'_{nn}$$

p menyatakan banyaknya operasi pertukaran baris di dalam OBE

Untuk ekspansi kofaktor, jika $A_{n\times n} = [a_{ij}]$ maka kofaktor dari a_{ij} dapat melambangkan C_{ij} dan $C_{ij} = (-1)^{i+j}M_{ij}$, dengan M_{ij} menyatakan minor dari a_{ij} dan M_{ij} adalah determinan dari submatriks *A* yang diperoleh dengan mencoret semua entri pada baris ke-*i* dan semua entri pada kolom ke-*j*. Untuk mempermudah mengingat tanda positif dan negatif untuk C_{ij} adalah dengan menggunakan pola di bawah.

$$\begin{bmatrix} + & - & + & - & \dots \\ - & + & - & + & \dots \\ + & - & + & - & \dots \\ - & + & - & + & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

Salah satu contoh dari ekspansi kofaktor adalah seperti berikut.

$$A = \begin{bmatrix} 3 & -1 & 2 \\ 5 & 0 & 4 \\ 8 & 2 & -3 \end{bmatrix}$$

$$\det(A) = a_{12}C_{12} + a_{22}C_{22} + a_{32}C_{32}$$

$$\det(A) = -(-1)\begin{vmatrix} 5 & 4 \\ 8 & -3 \end{vmatrix} + 0\begin{vmatrix} 3 & 2 \\ 8 & -3 \end{vmatrix} - 2\begin{vmatrix} 3 & 2 \\ 5 & 4 \end{vmatrix}$$

$$\begin{aligned} &= \{(5)(-3) - (4)(8)\} + 0 - 2\{(3)(4) - (2)(5)\} \\ &= (-15 - 32) - 2\,(12 - 10) \\ &= (-47) - 2\,(2) \\ &= -\,51 \end{aligned}$$

4. Matriks Balikan (Invers Matriks)

   Jika A adalah sebuah matriks, maka matriks balikannya adalah A^{-1}. Sistem Ax= B dapat diselesaikan sebagai berikut:

$$Ax = B$$
$$A^{-1}Ax = A^{-1}B$$
$$Ix = A^{-1}B$$
$$x = A^{-1}B$$

(Dengan I adalah matriks indentitas)

   Sehingga dapat disimpulkan bahwa $AA^{-1} = A^{-1}A = I$. Salah satu metode untuk menghitung menghitung matriks balikan adalah dengan Metode Eliminasi Gauss-Jordan. Untuk matriks A yang berukuran n × n, matriks balikannya, yaitu A^{-1}, dapat dicari dengan cara $[A|I]$ ~(Gauss-Jordan)~ $[I|A^{-1}]$, yang dalam hal ini, I adalah matriks identitas yang berukuran n × n. Berikut adalah salah satu contohnya.

$$\left(\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 2 & 5 & 3 & 0 & 1 & 0 \\ 1 & 0 & 8 & 0 & 0 & 1 \end{array}\right) \underset{R3 - R1}{\overset{R2 - 2R1}{\sim}} \left(\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & -2 & 5 & -1 & 0 & 1 \end{array}\right) \overset{R3 + 2R2}{\sim} \left(\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & 0 & -1 & -5 & 2 & 1 \end{array}\right)$$

$$\overset{R3/(-1)}{\sim} \left(\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right) \overset{R1 - 2R2}{\sim} \left(\begin{array}{ccc|ccc} 1 & 0 & 9 & 5 & -2 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right) \underset{R2 + 3R3}{\overset{R1 - 9R3}{\sim}}$$

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & -40 & 16 & 9 \\ 0 & 1 & 0 & 13 & -5 & -3 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right) = (I|A^{-1})$$

   Jika saat A dikonversikan menjadi I dan terdapat baris yang seluruh elemennya adalah 0, dapat disimpulkan bahwa A tidak memiliki balikan.

5. Matriks Kofaktor

   Matriks kofaktor merupakan matriks yang terdiri dari kofaktor-kofaktor matriks itu sendiri. Susunan elemen-elemen dari matriks kofaktor tersebut juga mengikuti susunan (letak) dari kofaktor-kofaktornya. Misalkan A adalah matriks n × n dan C_{ij} adalah kofaktor entri dari a_{ij}, maka matriks kofaktor dari A adalah:

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

6. Matriks Adjoin

   Adjoin matriks merupakan transpose dari matriks kofaktor. Adjoin dapat disingkat menjadi adj. Misalkan matriks A, maka adjoin A ditulis adj(A). Adjoin matriks dapat digunakan untuk menentukan matriks balikan dari suatu matriks, dengan menggunakan rumus:

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

7. Kaidah Cramer

   Kaidah Cramer adalah salah satu metode untuk menyelesaikan persamaan matematika linier. Aturan Cramer ini berkaitan dengan penyelesaian sebuah matriks bujur sangkar (matriks yang baris dan kolomnya sama). Jika Ax = b adalah SPL yang terdiri dari n persamaan linier dengan n peubah (variabel) sedemikian sehingga det(A) ≠ 0, maka SPL tersebut memiliki solusi yang unik, yaitu:

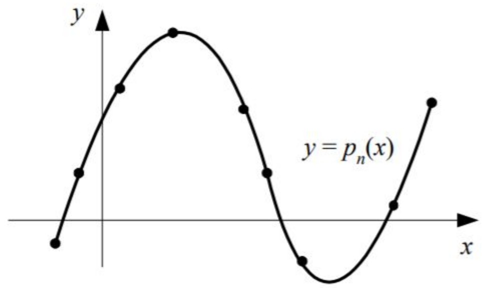
$$x_1 = \frac{\det(A_1)}{\det(A)}\ , \quad x_2 = \frac{\det(A_2)}{\det(A)}\ , \ \dots\ , \quad x_n = \frac{\det(A_n)}{\det(A)}$$

   yang dalam hal ini, A_j adalah matriks yang diperoleh dengan dengan mengganti entri pada kolom ke-j dari A dengan entri dari matriks

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

8. Interpolasi Polinom

   Jika data yang diketahui mempunyai ketelitian yang sangat tinggi, maka ada beberapa titik yang melewati kurva, seperti jika kurva fungsi yang sebenarnya dirajah melalui tiap titik tersebut. Dapat dikatakan bahwa titik-titik data dapat diinterpolasi dengan sebuah fungsi, seperti gambar di bawah. Jika fungsi yang digunakan berbentuk polinom, polinom tersebut dapat dinamakan sebagai polinom interpolasi.



9. Regresi Linier Berganda

   Regresi Linier Berganda adalah hubungan secara linier antara dua atau lebih variabel independen (X_1, X_2, …, X_n) dengan variabel dependen (Y). Analisis ini digunakan untuk mengetahui arah hubungan antara variabel independen dengan variabel dependen apakah masing-masing variabel independen berhubungan positif atau negatif, serta untuk memprediksi nilai dari variabel dependen apabila nilai variabel independen mengalami kenaikan atau penurunan. Data yang digunakan biasanya berskala interval atau rasio.

   Persamaan regresi linier berganda sebagai berikut:

$$Y' = a + b_1X_1 + b_2X_2 + ... + b_nX_n$$

Keterangan:

Y’	= Variabel dependen (nilai yang diprediksikan)
X_1, X_2, …, X_n	= Variabel independen
a	= Konstanta (nilai Y’ apabila X_1, X_2, …, X_n = 0)
b	= Koefisien regresi (nilai peningkatan ataupun penurunan)

Bab III

Implementasi Program

1. Class Matrix
   - Attribute

Nama	Tipe	Deskripsi
row	private int	Jumlah baris
col	private int	Jumlah kolom
data	private double[][]	Terisi dengan elemen-elemen matriks
hasChanged	private boolean	Digunakan untuk fungsi hasTrait
traits	private Set<MatrixTrait>	

   - Methods

Nama	Tipe	Parameter	Deskripsi
Matrix	public Constructor	int row int col double[][] data	Memindahkan elemen-elemen dari tipe Matriks ke tipe double[][]
getRowCount	public int		Mengembalikan integer, jumlah baris pada suatu matriks *augmented*
getColCount	piblic int		Mengembalikan integer, jumlah kolom pada suatu matriks *augmented*
getElement	public double	int row int col	Mengembalikan double, elemen dari suatu matriks dengan parameter baris dan kolom
setElement	public void	int row int col	Mengubah nilai suatu matriks dengan posisi baris dan kolom tertentu. Nilai hasChanged menjadi true
swapRow	public void	int rowA int rowB	Nilai hasChanged menjadi true.
swapCol	public void	int colA int colB	Nilai hasChanged menjadi true.

show	public void	boolean splFormat	Memperlihatkan matriks ke layar. Jika splFormat true, akan diperlihatkan dalam bentuk SPL, dan jika splFormat false, akan diperlihatkan dalam bentuk matriks
hasTrait	public boolean	MatrixTrait mt	Mengecek apakah matriks memiliki suatu sifat. Jika hasChanged false, langsung kembalikan hasil. Jika hasChanged true, gunakan prosedur updateTrait() terlebih dahulu.
updateTrait	public void		Prosedur untuk mengupdate sifat suatu matriks. Dipanggil di setiap fungsi yang mengubah isi suatu matriks.
rowOperation	public void	int row int otherRow MatrixOperation operation	Melakukan operasi pada setiap elemen di suatu baris dengan baris lain
rowOperation	public void	int row MatrixOperation operation	Melakukan operasi pada setiap elemen di suatu baris dengan konstanta
fixPrecision	public void		Prosedur untuk memperbaiki presisi nilai setiap elemen pada matriks
transpose	public void		Melakukan operasi transpose ke suatu matriks. Nilai hasChanged menjadi true.
multiply	public Matrix	Matrix otherMatrix	Melakukan operasi perkalian antara 2 matriks
findRow	public boolean	int row double value	Mencari jika apakah ada suatu nilai pada baris tertentu
findCol	public boolean	int col double value	Mencari jika apakah anda suatu nilai pada kolom tertentu

countXinCol	public int	int col double value	Mengecek frekuensi nilai pada kolom tertentu
countXinRow	public int	int row double value	Mengecek frekuensi nilai pada baris tertentu
mostXRow	public int	double value	Mengembalikan indeks baris yang memiliki suatu nilai terbanyak
mostXCol	public int	double value	Mengembalikan indeks kolom yang memiliki suatu nilai terbanyak
findFirstXinCol	public int	int col double value	Mencari indeks pertama yang bernilai X pada suatu kolom
findFirstXinRow	public int	int row double value	Mencari indeks pertama yang bernilai X pada suatu baris
isAllXinCol	public boolean	int col double value	Mengecek apakah suatu kolom semuanya bernilai X
isAllXinRow	public boolean	int row double value	Mengecek apakah suatu baris semuanya bernilai X
cofactor	public double	int row int col	Mengembalikan nilai kofaktor
determinant	public double		Mengembalikan nilai determinan dengan metode reduksi baris
determinantWithCofactor	public double		Mengembalikan nilai determinan dengan metode kofaktor
adjoint	public Matrix		Mengembalikan adjoin dari suatu matriks
inverse	public Matrix		Melakukan operasi perhitungan invers pada suatu matriks

2. Class AugMatrix
   - Attribute

Nama	Tipe	Deskripsi

left	private Matrix	Merupakan matriks *augmented* sebelah kiri
right	private Matrix	Merupakan matriks *augmented* sebelah kanan

- Methods

Nama	Tipe	Parameter	Deskripsi
AugMatrix	public Constructor	Matrix left Matrix right	Menginisiasi matriks *augmented* sebelah kiri dan sebelah kanan dengan parameternya 2 matriks
AugMatrix	public Constructor	Matrix matrix	Menginisiasi matriks *augmented* sebelah kiri dan sebelah kanan dengan parameternya 1 matriks
getLeft	public Matrix		Mengembalikan matriks dari matriks *augmented* sebelah kiri
getRight	public Matrix		Mengembalikan matriks dari matriks *augmented* sebelah kanan
show	public void	boolean splFormat	Memperlihatkan matriks ke layar. Jika splFormat true, akan diperlihatkan dalam bentuk SPL, dan jika splFormat false, akan diperlihatkan dalam bentuk matriks
swapRow	public void	int rowA int rowB	Menukarkan 2 baris matriks *augmented*
rowOperation	public void	int row int otherRow MatrixOperation operation	Melakukan operasi pada setiap elemen di suatu baris dengan baris lain
rowOperation	public void	int row MatrixOperation operation	Melakukan operasi pada setiap elemen di suatu baris dengan konstanta
fixPrecision	public void		Prosedur untuk memperbaiki presisi nilai setiap elemen pada matriks
divideBySingleElem	public void	int row	Untuk membagi nilai pada

ent		int col	suatu baris dengan nilai pada elemen suatu indeks baris dan kolom
findFirstXinCol	public int	int col double value	Mencari indeks pertama yang bernilai X pada suatu kolom
isAllXinCol	public boolean	int col double value	Mengecek apakah suatu kolom semuanya bernilai X
eliminateFromTop	public void	int row int col	Mengeliminasi suatu baris sebuah matriks dari atas sampai bawah
eliminateFromBottom	public void	int row int col	Mengeliminasi suatu baris sebuah matriks dari bawah sampai atas
countZeroinRowUntilX	public int	int row	Menghitung jumlah angka 0 sampai bukan angka 0 pada suatu baris
fixZeroRow	public boolean	int row	Menukar baris jika ada baris yang jumlah angka 0 lebih banyak dibandingkan baris-baris yang ada di bawahnya
getRowCount	public int		Mengembalikan integer, jumlah baris pada suatu matriks *augmented*
getColCount	public int		Mengembalikan integer, jumlah kolom pada suatu matriks *augmented*
isAllXinRow	public boolean	int row double value	Mengecek apakah suatu baris semuanya bernilai X

3. Class Crammer
   - Attribute

Nama	Tipe	Deskripsi
aug	private AugMatrix	Matriks *augmented* yang akan dipakai untuk operasi Cramer
result	private AugMatrix	Hasil dari operasi Cramer
hasSolution	private boolean	Untuk menandakan jika operasi Cramer dari

		matriks tersebut mempunyai solusi atau tidak
originalMatrix	private Matrix	Matriks mula-mula

- Methods

Nama	Tipe	Parameter	Deskripsi
Crammer	public Constructor	Matrix matrix	Menginisiasi matriks *augmented* dan mengecek apakah parameter matriksnya berbentuk persegi
operate	public void		Prosedur untuk melakukan operasi perhitungan Cramer
getInitialMatrix	public Matrix		Mengembalikan matriks mula-mula
getResult	public AugMatrix		Mengembalikan nilai hasil dari operasi Cramer jika ada solusinya. Jika tidak ada solusinya, akan mengembalikan null
hasSolution	public boolean		Mengecek apakah operasi Cramer tersebut mempunyai solusi atau tidak

4. Class Gauss
   - Attribute

Nama	Tipe	Deskripsi
result	privateAugMatrix	Hasil dari operasi Gauss
originalMatrix	privateMatrix	Matriks mula-mula
hasSolution	private boolean	Untuk menandakan jika operasi Gauss dari matriks tersebut mempunyai solusi atau tidak

- Methods

Nama	Tipe	Parameter	Deskripsi
Gauss	public Constructor	Matrix matrix	Menginisiasi matriks *augmented* untuk dioperasikan

operate	public void		Prosedur untuk melakukan operasi perhitungan Gauss
getInitialMatrix	public Matrix		Mengembalikan matriks mula-mula
getResult	public AugMatrix		Mengembalikan nilai hasil dari operasi Gauss jika ada solusinya. Jika tidak ada solusinya, akan mengembalikan null
hasSolution	public boolean		Mengecek apakah operasi Gauss tersebut mempunyai solusi atau tidak

5. Class GaussJordan
    - Attribute

Nama	Tipe	Deskripsi
result	privateAugMatrix	Hasil dari operasi Gauss Jordan
originalMatrix	privateMatrix	Matriks mula-mula
hasSolution	private boolean	Untuk menandakan jika operasi Gauss Jordan dari matriks tersebut mempunyai solusi atau tidak

- Methods

Nama	Tipe	Parameter	Deskripsi
GaussJordan	public Constructor	Matrix matrix	Menginisiasi matriks *augmented* untuk dioperasikan
operate	public void		Prosedur untuk melakukan operasi perhitungan Gauss Jordan
getInitialMatrix	public Matrix		Mengembalikan matriks mula-mula
getResult	public AugMatrix		Mengembalikan nilai hasil dari operasi Gauss Jordan jika ada solusinya. Jika tidak ada solusinya, akan mengembalikan null.
hasSolution	public boolean		Mengecek apakah operasi

			Gauss Jordan tersebut mempunyai solusi atau tidak

6. Class InverseGaussJordan
    - Attribute

Nama	Tipe	Deskripsi
hasSolution	private boolean	Untuk menandakan jika operasi Invers menggunakan metode Gauss Jordan dari matriks tersebut mempunyai solusi atau tidak

- Methods

Nama	Tipe	Parameter	Deskripsi
InverseGaussJordan	public Constructor	Matrix matrix	Menginisiasi hasil operasi dari Invers menggunakan metode Gauss Jordan
getInvers	public Matrix		Mengembalikan nilai hasil dari operasi Invers menggunakan operasi Gauss Jordan jika ada solusinya. Jika tidak ada solusinya, akan mengembalikan null.
hasSolution	public boolean		Mengecek apakah operasi Invers menggunakan metode Gauss Jordan mempunyai solusi atau tidak

Bab IV

Eksperimen

1. Solusi SPL Ax = b

a	Kaidah Crammer
$A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$	Hasil Matriks ini tidak memiliki solusi atau tidak bisa memakai Crammer. Silakan coba pakai cara lain. Rekomendasi: Gauss/GaussJordan

b	Metode Gauss-Jordan
$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$	X1 = b + 3 X2 = 2b X3 = a X4 = b -1 X5 = b

c	Metode Gauss
$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$	Hasil Tidak ada solusi untuk SPL ini.

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

	Metode Gauss
Jika n = 6	Hasil a = 15.22 b = -91.58 c = 125.125 d = -43 e = 79.25 f = -91

	Metode Gauss-Jordan
Jika n = 10	Hasil Tidak ada solusi untuk SPL ini.

2. SPL berbentuk matriks *augmented*

a	Metode Gauss
$\begin{bmatrix} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{bmatrix}.$	Hasil X1 = b -1 X2 = 2a X3 = a X4 = b

b	Metode Gauss-Jordan
$\begin{bmatrix} 2 & 0 & 8 & 0 & 8 \\ 0 & 1 & 0 & 4 & 6 \\ -4 & 0 & 6 & 0 & 6 \\ 0 & -2 & 0 & 3 & -1 \\ 2 & 0 & -4 & 0 & -4 \\ 0 & 1 & 0 & -2 & 0 \end{bmatrix}.$	Hasil Tidak ada solusi untuk SPL ini.

3. SPL Berbentuk

a	Metode Invers Balikan

<table>
<tr>
<td>$8x_1 + x_2 + 3x_3 + 2x_4 = 0$
$2x_1 + 9x_2 - x_3 - 2x_4 = 1$
$x_1 + 3x_2 + 2x_3 - x_4 = 2$
$x_1 + 6x_3 + 4x_4 = 3$</td>
<td>Hasil
Matriks ini tidak memiliki invers.</td>
</tr>
</table>

<table>
<tr><th>b</th><th>Metode Gauss-Jordan</th></tr>
<tr>
<td>$x_7 + x_8 + x_9 = 13.00$
$x_4 + x_5 + x_6 = 15.00$
$x_1 + x_2 + x_3 = 8.00$
$0.04289(x_3 + x_5 + x_7) + 0.75(x_6 + x_8) + 0.61396x_9 = 14.79$
$0.91421(x_3 + x_5 + x_7) + 0.25(x_2 + x_4 + x_6 + x_8) = 14.31$
$0.04289(x_3 + x_5 + x_7) + 0.75(x_2 + x_4) + 0.61396x_1 = 3.81$
$x_3 + x_6 + x_9 = 18.00$
$x_2 + x_5 + x_8 = 12.00$
$x_1 + x_4 + x_7 = 6.00$
$0.04289(x_1 + x_5 + x_9) + 0.75(x_2 + x_6) + 0.61396x_3 = 10.51$
$0.91421(x_1 + x_5 + x_9) + 0.25(x_2 + x_4 + x_6 + x_8) = 16.13$
$0.04289(x_1 + x_5 + x_9) + 0.75(x_4 + x_8) + 0.61396x_7 = 7.04$</td>
<td>Hasil
a = 0
b = 8.704
c = -0.703
d = 1
e = -5.63
f = -10.165
g = 0
h = -3.224
i = 13.946</td>
</tr>
</table>

4. Studi Kasus Rangkaian

<table>
<tr>
<td colspan="6">$i_{12} + i_{52} + i_{32} = 0$
$- i_{52} + i_{65} - i_{54} = 0$
$- i_{32} + i_{43} = 0$
$i_{54} - i_{43} = 0$
$i_{32} R_{32} \quad V_2 - V_3 = 0$
$i_{43} R_{43} + V_3 - V_4 = 0$
$i_{65} R_{65} + V_5 = V_6$
$i_{12} R_{12} + V_2 = V_1$
$i_{54} R_{54} + V_4 - V_5 = 0$
$i_{52} R_{52} + V_2 - V_5 = 0$

Tentukan nilai dari:

i_{12}, i_{52}, i_{32}, i_{65}, i_{54}, i_{13}, V_2, V_3, V_4, V_5

bila diketahui

R_{12} = 5 ohm, R_{52} = 10 ohm, R_{32} = 10 ohm
R_{65} = 20 ohm, R_{54} = 15 ohm, R_{14} = 5 ohm.
V_1 = 200 volt, V_6 = 0 volt.</td>
<td colspan="4">Metode Gauss-Jordan

a = 6.9565
b = -5.21843
c = -1.7392
d = -6.9565
e = -1.73921
f = -1.7392
g = 165.21699
h = 147.82495
i = 139.12947
j = 113.04325</td>
</tr>
<tr><th>i_{12}</th><th>i_{52}</th><th>i_{32}</th><th>i_{65}</th><th>i_{54}</th><th>i_{13}</th><th>V_2</th><th>V_3</th><th>V_4</th><th>V_5</th></tr>
<tr><td>6.9565</td><td>-5.21843</td><td>-1.7392</td><td>-6.9565</td><td>-1.73921</td><td>-1.7392</td><td>165.21
699</td><td>147.82
495</td><td>139.12
947</td><td>113.04
325</td></tr>
</table>

5. Interpolasi Titik

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 default berikut:

$x = 0.2$ f(x) = ?
$x = 0.55$ f(x) = ?
$x = 0.85$ f(x) = ?
$x = 1.28$ f(x) = ?

x	**0.2**	**0.55**	**0.85**	**1.28**
f(x)	F(0.2) = 0.032	F(0.55) = 0.168	F(0.85) = 0.335	F(1.28) = 0.677

6. Kasus Covid

Tanggal	Tanggal (desimal)	Jumlah Kasus
24/04/20	4,800	8.211
30/04/20	5,000	10.118
16/05/20	5,516	17.025
22/05/20	5,710	20.796
15/06/20	6,500	39.294
06/07/20	7,194	64.958
03/08/20	8,097	113.134
08/08/20	8,258	123.503
01/09/20	9,033	177.571
10/09/20	9,333	145.510

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

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

a. 25/05/20
b. 30/08/20
c. 15/09/20
d. beserta masukan user lainnya berupa **tanggal (desimal) yang sudah diolah** dengan asumsi prediksi selalu dilakukan untuk tahun 2020.

Tanggal	**25/05/20**	**30/08/20**	**15/09/20**
Prediksi	F(5.806) = 20889.707 20889 kasus	F(8.968) = 157208.422 157208 kasus	F(9.5) = 51745.371 51745 kasus

7. Penyederhanaan Fungsi

$f(x) = \dfrac{x^2 + \sqrt{x}}{e^x + x}$	**Derajat *n* dalam selang [0, 2]** **n = 5 (jarak *h* = 0,4)**
	$f(x) = 0.239x^5 - 1.43x^4 + 3.248x^3 - 3.533x^2$

	$+\ 2.034x$

8. Regresi Linier Berganda

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.

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

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

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

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

```
> Regresi Linear Berganda [15:58:44 02-10-2020]
Sistem Persamaan
20a + 863.1b + 1530.4c + 587.84d = 19.42
863.1a + 54876.89b + 67000.09c + 25283.395d = 779.477
1530.4a + 67000.09b + 117912.32c + 44976.867d = 1483.437
587.84a + 25283.395b + 44976.867c + 17278.5086d = 571.1219

Hasil
a = -3.50761
b = -0.00263
c = 0.0008
d = 0.15415
Masukkan X1: 50
Masukkan X2: 76
Masukkan X3: 29.3

Taksiran/Estimasi Nilai:
0.93826
```

Bab V

Kesimpulan, Saran, dan Refleksi

Sistem Persamaan Linier dapat diselesaikan dengan berbagai metode, salah satunya adalah dengan menggunakan metode-metode pada SPL berbentuk matriks. Metode yang dapat digunakan untuk menyelesaikan SPL antara lain adalah eliminasi Gauss, eliminasi Gauss-Jordan, kaidah Cramer, dan matriks invers. Keempat metode tersebut memiliki ketentuan dan aturan masing-masing, seperti kaidah Cramer dan metode matriks invers hanya dapat digunakan jika matriks *augmented* kiri berukuran $n \times n$.

Dengan menentukan determinan, matriks kofaktor, dan matriks adjoint, maka metode kaidah Cramer dan matriks invers dapat dihitung. Namun, hal tersebut memiliki ketentuan khusus, yaitu matriks harus berukuran $n \times n$. Jika ada SPL yang jumlah variabelnya berbeda dengan jumlah persamaannya, maka metode kaidah Cramer dan matriks invers tidak dapat digunakan. Alternatifnya, solusi SPL tersebut masih dapat dicari menggunakan metode eliminasi Gauss atau eliminasi Gauss-Jordan.

Solusi SPL memiliki 3 bentuk, yaitu *single-solution*, *multi-solution*, dan tidak ada solusi. Solusi unik (*single-solution)* terjadi saat nilai masing-masing variabel yang dicari bernilai unik atau hanya satu. Contohnya pada eksperimen kasus nomor 1 bagian d yang pertama, memiliki solusi:

$$x1 = 15.22,\ x2 = -91.58,\ x3 = 125.125,\ x4 = -43,\ x5 = 79,\ x6 = -91$$

Solusi banyak (*multi-solution*) terjadi saat persamaan variabelnya bergantung pada variabel lain, yang akan menghasilkan bentuk parametrik. Contohnya pada eksperimen kasus nomor 2 bagian a, memiliki solusi:

$$x1 = x4 - 1,\ x2 = 2 \times x3$$

$$Misal\ x3 = a,\ x4 = b$$

$$⁂\ x1 = b - 1,\ x2 = 2a,\ untuk\ a,\ b \in R$$

Tidak ada solusi terjadi ketika tidak ada nilai variabel apapun yang memenuhi persamaan. Contohnya pada eksperimen kasus nomor 1 bagian d yang kedua,

Aplikasi Sistem Persamaan Linier dapat digunakan pada berbagai hal, contohnya adalah interpolasi dan regresi linier berganda. Dengan memanfaatkan metode-metode penyelesaian SPL yang ada, solusi dari permasalahan-permasalahan tersebut dapat diselesaikan.

Interpolasi polinom dapat digunakan untuk memprediksi data berdasarkan kurva dari data-data yang telah didapat. Pada contoh eksperimen untuk kasus covid, prediksi jumlah kasus pada tanggal 25 Mei 2020 adalah sebanyak 20.889 kasus, pada tanggal 30 Agustus 2020 adalah sebanyak 15.7208 kasus, dan pada tanggal 15 September 2020 adalah sebanyak 51.745 kasus.

Regresi linier berganda dapat digunakan untuk mendapatkan suatu kesimpulan dari banyak data yang kemudian estimasi nilai suatu data dapat ditentukan dari hasil persamaan regresi linier. Contohnya untuk eksperimen pada kasus nomor 8, dari data-data yang ada dapat dibuat persamaan regresi liniernya, kemudian menaksirkan suatu nilai Nitrous Oxide sebesar 0,93826.

Permasalahan-permasalahan pada eksperimen dapat diselesaikan menggunakan program java yang telah dibuat.

Saran untuk pengembangan yang kami berikan berupa mengadakan *test-case* yang telah disertai solusi. Hal ini diperuntukkan untuk efisiensi waktu pengerjaan. Dalam mengecek beberapa matriks, khususnya yang berukuran besar, terkadang cukup memakan waktu. Potensi terjadinya error dan salah hitung juga bertambah. Dengan adanya *test-case* dan solusi di awal, maka akan mempermudah proses pengecekan dan menghemat waktu.

Untuk tugas ini, kami merasa dapat memperbaiki kinerja kami dari mulai pembagian tugas hingga tahap pengerjaan. Dengan pembagian tugas yang jelas, maka akan terbentuk *timeline* kerja yang lebih sistematis, sehingga pengontrolan masing-masing pekerjaan lebih mudah. Kemudian, pencicilan tugas juga dapat diperbaiki, yang idealnya dari hari pertama tugas besar dibagikan, selalu ada progress setiap harinya. Hal ini ditujukan agar tidak bekerja secara *deadliner*, yang berpotensi mengakibatkan hasil tugas tidak maksimal. Mengambil hari kosong tidak masalah, selama tidak terlalu banyak kosongnya.

Referensi

- https://docplayer.info/29888445-Metode-gauss-tujuan-dasar-teori-eliminasi-gauss-pembahasan-analisis.html (Diakses pada 28 September 2020, 17:22)
- https://docplayer.info/73571530-Metode-matriks-balikan.html (Diakses pada 29 September 2020, 00:11)
- http://duwiconsultant.blogspot.com/2011/11/analisis-regresi-linier-berganda.html (Diakses pada 29 September 2020, 02:09)
- http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-05-Sistem-Persamaan-Linier-2.pdf (Diakses pada 29 September 2020, 01:39)
- http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-08-Determinan-bagian1.pdf (Diakses pada 28 September 2020, 23:25)
- http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-09-Determinan-bagian2.pdf (Diakses pada 28 September 2020, 23:25)
- https://www.academia.edu/6817696/BAb_05_Interpolasi_Polinom (Diakses pada 29 September 2020, 02:55)
- https://www.madematika.net/2017/08/pengertian-minor-kofaktor-matriks.html (Diakses pada 29 September 2020, 00:44)
- https://www.profematika.com/eliminasi-gauss-dan-contoh-penerapannya/#:~:text=Eliminasi%20gauss%20ditemukan%20oleh%20Carl,Baris%20melalui%20Operasi%20Baris%20Elementer. (Diakses pada 27 September 2020, 23:27)
- https://www.profematika.com/eliminasi-gauss-jordan-beserta-contoh-penerapannya/#Eliminasi_Gauss-Jordan (Diakses pada 28 September 2020, 21:51)
- https://www.profematika.com/kelebihan-dan-kekurangan-metode-ekspansi-kofaktor/ (Diakses pada 28 September 2020, 23:28)