

LAPORAN TUGAS BESAR 01

IF2123 ALJABAR LINEAR GEOMETRI



“The Slebews”

Disusun oleh:

Salsabiila	13521062
Zahira Dina	13521085
Shulha	13521087

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2022

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).

I.1 Tujuan

1. 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)
2. Menghitung determinan matriks dengan reduksi baris dan ekspansi kofaktor
3. Menghitung invers matriks dengan metode adjoin dan matriks balikan
4. Menyelesaikan persoalan interpolasi polinom, interpolasi bicubic spline, dan regresi linear berganda.

I.2 Spesifikasi

1. Program dapat menerima masukan (*input*) baik dari *keyboard* maupun membaca masukan dari *file text*. Untuk SPL, masukan dari *keyboard* adalah m , n , koefisien a_{ij} , dan b_i . Masukan dari *file* berbentuk matriks *augmented* tanpa tanda kurung, setiap elemen matriks dipisah oleh spasi. Misalnya,

```
3 4.5 2.8 10 12
-3 7 8.3 11 -4
0.5 -10 -9 12 0
```

2. Untuk persoalan menghitung determinan dan matriks balikan, masukan dari *keyboard* adalah n dan koefisien a_{ij} . Masukan dari *file* berbentuk matriks, setiap elemen matriks dipisah oleh spasi. Misalnya,

```
3 4.5 2.8
```

-3 7 8.3

0.5 -10 -9

Luaran (*output*) disesuaikan dengan persoalan (determinan atau invers) dan penghitungan balikan/invers dilakukan dengan metode matriks balikan dan adjoin.

3. Untuk persoalan interpolasi, masukannya jika dari *keyboard* adalah n , (x_0, y_0) , (x_1, y_1) , ..., (x_n, y_n) , dan nilai x yang akan ditaksir nilai fungsinya. Jika masukannya dari *file*, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung. Masukan kemudian dilanjutkan dengan satu buah baris berisi satu buah nilai x yang akan ditaksir menggunakan fungsi interpolasi yang telah didefinisikan. Misalnya jika titik-titik datanya adalah $(8.0, 2.0794)$, $(9.0, 2.1972)$, dan $(9.5, 2.2513)$ dan akan mencari nilai y saat $x = 8.3$, maka di dalam *file text* ditulis sebagai berikut:

8.0 2.0794

9.0 2.1972

9.5 2.2513

8.3

4. Untuk persoalan regresi, masukannya jika dari *keyboard* adalah n (jumlah peubah x), m (jumlah sampel), semua nilai-nilai x_{1i} , x_{2i} , ..., x_{ni} , nilai y_i , dan nilai-nilai x_k yang akan ditaksir nilai fungsinya. Jika masukannya dari *file*, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.
5. Untuk persoalan SPL, luaran program adalah solusi SPL. Jika solusinya tunggal, tuliskan nilainya. Jika solusinya tidak ada, tuliskan solusi tidak ada, jika solusinya banyak, maka tuliskan solusinya dalam bentuk parametrik (misalnya $x_4 = -2$, $x_3 = 2s - t$, $x_2 = s$, dan $x_1 = t$).
6. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan. Contoh luaran untuk interpolasi adalah

$$f(x) = -0.0064x^2 + 0.2266x + 0.6762, \quad f(5) = \dots$$

dan untuk regresi adalah

$$f(x) = -9.5872 + 1.0732x_1, \quad f(x_k) = \dots$$

7. Untuk persoalan *bicubic spline interpolation*, masukan dari *file text* (.txt) yang berisi matriks berukuran 4 x 4 yang berisi konfigurasi nilai fungsi dan turunan berarah disekitarnya, diikuti dengan nilai a dan b untuk mencari nilai $f(a, b)$.

Misalnya jika nilai dari $f(0, 0), f(1, 0), f(0, 1), f(1, 1), f_x(0, 0), f_x(1, 0), f_x(0, 1), f_x(1, 1), f_y(0, 0), f_y(1, 0), f_y(0, 1), f_y(1, 1), f_{xy}(0, 0), f_{xy}(1, 0), f_{xy}(0, 1), f_{xy}(1, 1)$ berturut-turut adalah 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 serta nilai a dan b yang dicari berturut-turut adalah 0.5 dan 0.5 maka isi *file text* ditulis sebagai berikut:

```
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
0.5 0.5
```

Luaran yang dihasilkan adalah nilai dari $f(0.5, 0.5)$.

8. Luaran program harus dapat ditampilkan **pada layar komputer dan dapat disimpan ke dalam file**.
9. Bahasa program yang digunakan adalah Java. Anda bebas untuk menggunakan versi java apapun dengan catatan di atas java versi 8 (8/9/11/15/17/19/20).
10. Program **tidak harus** berbasis GUI, cukup *text-based* saja, namun boleh menggunakan GUI (memakai kaskas *Eclipse* misalnya).
11. Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan dirancang masing-masing. Misalnya, menu:

MENU

1. Sistem Persamaan Linier

2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic Spline
6. Regresi linier berganda
7. Keluar

Untuk pilihan menu nomor 1 ada sub-menu lagi yaitu pilihan metode:

1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer

Begitu juga untuk pilihan menu nomor 2 dan 3.

BAB II

TEORI SINGKAT

II.1 Operasi Baris Elementer

Operasi Baris Elementer merupakan suatu operasi pada matriks yang dapat digunakan untuk memperoleh invers suatu matriks atau memperoleh penyelesaian dari sebuah Sistem Persamaan Linier (SPL). Untuk mendapatkan solusi dari SPL, OBE dilakukan pada matriks *augmented* hingga terbentuk matriks eselon baris (matriks yang memiliki 1 utama pada setiap baris, kecuali baris yang seluruhnya nol) atau matriks eselon baris tereduksi (matriks eselon baris dengan sifat setiap kolom yang memiliki 1 utama memiliki nol di tempat lain). Adapun tiga OBE terhadap matriks augmented yakni:

1. Mengalikan persamaan dengan konstanta tak nol.
2. Menukar posisi dua persamaan.
3. Menambahkan kelipatan satu persamaan ke persamaan lainnya.

II.2 Metode Eliminasi Gauss

Metode eliminasi Gauss adalah suatu metode untuk mencari himpunan penyelesaian dari sistem persamaan linear dengan menggunakan OBE, sedemikian hingga matriksnya memiliki bentuk eselon baris. Selanjutnya, matriks tersebut diubah ke dalam bentuk sistem persamaan linear dan kemudian dilakukan substitusi balik mulai dari persamaan paling bawah.

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

II.3 Metode Eliminasi Gauss-Jordan

Metode eliminasi Gauss-Jordan adalah suatu metode untuk mencari himpunan penyelesaian dari sistem persamaan linear dengan menggunakan OBE, sedemikian hingga matriksnya memiliki bentuk eselon baris tereduksi. Selanjutnya, matriks tersebut diubah ke dalam bentuk sistem persamaan linear dan kemudian dilakukan substitusi balik mulai dari persamaan paling bawah.

Metode ini dapat dilakukan dengan dua fase. Fase maju untuk membuat matriks eselon baris dan fase mundur untuk menghasilkan nilai-nilai 0 di atas 1 utama.

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

II.4 Determinan

Determinan adalah nilai yang didapat dari operasi unsur-unsur suatu matriks persegi, yakni matriks yang memiliki jumlah baris dan kolom yang sama. Determinan suatu matriks dapat dihitung dengan berbagai metode, contohnya metode reduksi baris dan metode ekspansi kofaktor.

Dengan reduksi baris, Jika A adalah matriks segitiga atas atau segitiga bawah, atau diagonal dengan ukuran $n \times n$, maka $\det(A)$ adalah hasil kali elemen-elemen pada diagonal utamanya. Matriks segitiga atas/bawah dapat diperoleh dengan melakukan operasi baris elementer.

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

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

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

p menyatakan banyaknya operasi pertukaran baris di dalam OBE

Dengan A adalah matriks bujursangkar $n \times n$, maka determinan matriks A dapat dicari dengan mengalikan elemen-elemen pada sebarang baris (kolom) dengan kofaktornya dan menjumlahkannya.

(i) Ekspansi kofaktor sepanjang baris ke-i adalah

$$\det(A) = \sum_{j=1}^n a_{ij} C_{ij} = a_{i1} C_{i1} + a_{i2} C_{i2} + \dots + a_{in} C_{in}$$

(ii) Ekspansi kofaktor sepanjang kolom ke-j adalah

$$\det(A) = \sum_{j=1}^n a_{ij}C_{ij} = a_{1j}C_{1j} + a_{2j}C_{2j} + \cdots + a_{nj}C_{nj}$$

Dengan M_{ij} = minor entri a_{ij} atau determinan submatriks yang elemen-elemennya tidak berada pada baris i dan kolom j dan $C_{ij} = (-1)^{i+j}M_{ij}$ atau kofaktor entri a_{ij}

II.5 Matriks Balikan (Invers)

Matriks yang memiliki matriks balikan (invers) pasti merupakan matriks persegi dengan ukuran $n \times n$. Balikan matriks dapat diperoleh dengan metode adjoin maupun dengan eliminasi Gauss-Jordan.

Jika A adalah matriks bujur sangkar yang mempunyai invers, maka

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

dimana penjelasan tentang *adjoin* di poin selanjutnya.

Sementara dengan eliminasi Gauss-Jordan dengan prinsip bahwa matriks balikan dari matriks A merupakan $(A)^{-1}$ sedemikian sehingga $A(A)^{-1} = (A)^{-1}A = I$, dalam OBE jika ditemukan baris yang seluruh elemennya adalah 0 ketika dioperasikan dengan eliminasi Gauss-Jordan, matriks tersebut tidaklah memiliki balikan.

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

II.6 Matriks Kofaktor

Jika $A = a_{ij}$ adalah suatu matriks bujur sangkar, maka minor dari elemen a_{ij} , dinyatakan dengan M_{ij} , adalah determinan dari submatriks A dengan menghilangkan baris ke- i dan kolom ke- j . Sedangkan kofaktor dari elemen a_{ij} dinyatakan dengan $C_{ij} = (-1)^{i+j}M_{ij}$.

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

II.7 Matriks Adjoin

Adjoin matriks A merupakan transpose dari suatu matriks yang elemen-elemennya merupakan kofaktor dari elemen-elemen matriks A (matriks kofaktor A). Transpose dari matriks kofaktornya.

II.8 Kaidah Cramer

Kaidah Cramer adalah rumus yang dapat digunakan untuk menyelesaikan sistem persamaan linear dengan banyak persamaan sama dengan banyak variabel, dan berlaku ketika sistem tersebut memiliki solusi yang tunggal. Rumus ini menyatakan solusi dengan menggunakan determinan matriks koefisien (dari sistem persamaan) dan determinan matriks lain yang diperoleh dengan mengganti salah satu kolom matriks koefisien dengan vektor yang berada sebelah kanan persamaan. Metode ini dinamai dari matematikawan Swiss Gabriel Cramer (1704–1752), yang pada tahun 1750 menerbitkan kaidah ini untuk sebarang banyaknya variabel, walau Colin Maclaurin juga menerbitkan kasus khusus dari kaidah ini pada tahun 1748.

Pertimbangkan sistem n persamaan linear dengan n variabel, yang direpresentasikan dalam bentuk perkalian matriks sebagai:

$$A\mathbf{x} = \mathbf{b}$$

dengan matriks A berukuran $n \times n$ memiliki determinan bukan nol, Teorema menyatakan bahwa sistem memiliki solusi unik dalam keadaan ini, dengan nilai untuk setiap variabel diberikan oleh:

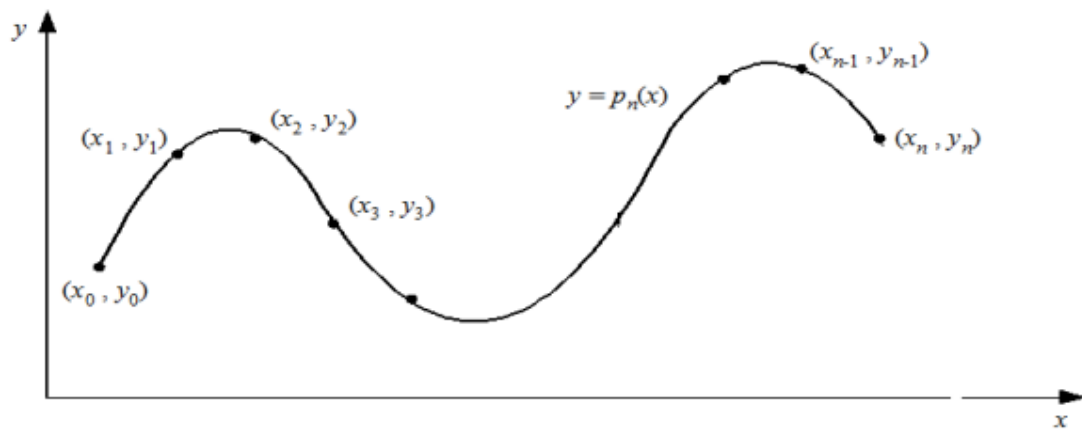
$$x_1 = \frac{\det(A_1)}{\det(A)}, \quad x_2 = \frac{\det(A_2)}{\det(A)}, \quad \dots, \quad x_n = \frac{\det(A_n)}{\det(A)}$$

dimana $A(i)$ adalah matriks yang dibentuk dengan mengganti kolom ke- i dari A dengan vektor kolom b .

II.9 Interpolasi Polinomial

Interpolasi polinomial merupakan teknik interpolasi dengan mengasumsikan pola data yang kita miliki mengikuti pola polinomial baik berderajat satu (linier) maupun berderajat tinggi. Interpolasi dengan metode ini dilakukan dengan terlebih dahulu membentuk persamaan polinomial. Persamaan polinomial yang terbentuk selanjutnya digunakan untuk melakukan interpolasi dari nilai yang diketahui atau ekstrapolasi (prediksi) dari nilai diluar rentang data yang diketahui.

Polinom interpolasi derajat n yang menginterpolasi titik-titik $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ adalah berbentuk $p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$. Sehingga dengan eliminasi gauss dapat diperoleh solusi nilai a_0, a_1, \dots, a_n , untuk menaksir nilai titik yang tidak diketahui.



II.10 Interpolasi Bicubic Spline

Bicubic interpolation merupakan teknik interpolasi pada data 2D umumnya digunakan yang merupakan pengembangan dari interpolasi linear dan cubic. Dalam pemrosesan menggunakan interpolasi *bicubic spline* digunakan 16 buah titik, 4 titik referensi utama di bagian pusat, dan 12 titik di sekitarnya sebagai aproksimasi turunan dari keempat titik referensi untuk membangun permukaan bikubik. Dengan memanfaatkan

model dasar dan turunan, dapat diperoleh nilai untuk persamaan yang dapat menaksir nilai titik dengan lebih halus.

II.11 Regresi Linear Berganda

Regresi linier berganda adalah salah satu metode untuk memprediksi nilai berdasarkan data yang dimiliki. Regresi linier berganda adalah model regresi linier yang melibatkan variabel independen yang lebih dari satu. Rumus umum dari regresi linier berganda adalah sebagai berikut.

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

dimana nilai β_i dapat diperoleh menggunakan *Normal Estimation Equation for Linear Regression* sebagai berikut.

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

BAB III

IMPLEMENTASI PUSTAKA DAN PROGRAM DALAM JAVA

III. 1. Class Matriks

III. 1. 1. Attribute

Nama	Tipe	Deskripsi
Matriks	public double[][]	Tempat menyimpan elemen matriks
nRows	public int	Jumlah baris
nCols	public int	Jumlah kolom

III. 1. 2. Method

Nama	Tipe	Parameter	Deskripsi
Matriks	public Constructor	int nRows, int nCols	Konstruktor dari Matriks, membuat matriks sesuai dengan nRows (baris) dan nCols (kolom) inputan
readMatriks	public void	int nRows, int nCols	Membaca matriks dari input user
displayMatriks	public void		Menampilkan matriks ke layar
displayMatriksDouble	public void		Menampilkan matriks ke layar dengan pembulatan
isMatrixIdxValid	public boolean	int i, int j	Mengembalikan true jika indeks matriks valid (kurang dari kapasitas maksimum)

			matriks 1000)
isMatrixIdxEff	public boolean	int i, int j	Mengembalikan true jika indeks matriks efektif (kurang dari ukuran matriks)
getLastIdxRow	Public int		Mengembalikan indeks baris terakhir matriks
getLastIdxCol	public int		Mengembalikan indeks kolom terakhir matriks
getElmtDiagonal	public double		Mengembalikan nilai dari indeks matriks yang diagonal
copyMatriks	public Matriks		Membuat salinan dari matriks, MCopy perlu dideclare terlebih dahulu untuk pemanggilan
addMatriks	public Matriks	Matriks M2	Mengembalikan matriks hasil penjumlahan matriks dengan matriks M2
subMatriks	public Matriks	Matriks M2	Mengembalikan matriks hasil pengurangan matriks dengan matriks M2
multiplyMatriks	public Matriks	Matriks M2	Mengembalikan matriks hasil perkalian matriks dengan

			matriks M2
multiplyMatriksWithMod	public Matriks	Matriks M2, int mod	Mengembalikan matriks hasil perkalian matriks dengan matriks M2 yang kemudian dimodulo kan
multiplyByConst	public Matriks	double k	Mengembalikan matriks hasil perkalian matriks dengan konstanta k
isMatrixSizeEqual	public boolean	Matriks M2	Mengembalikan true jika ukuran matriks sama
isMatrixEqual	public boolean	Matriks M2	Mengembalikan true jika isi matriks sama
isMatrixSquare	public boolean		Mengembalikan true jika matriks merupakan matriks persegi
countElmt	public int		Mengembalikan jumlah elemen matriks
isSymmetric	public boolean		Mengembalikan true jika matriks merupakan matriks simetris yaitu jika matriks[i][j] == matriks[j][i]
isIdentity	public boolean		Mengembalikan true jika

			matriks merupakan matriks identitas (diagonal nya berisi 1, sisanya 0)
transpose	public Matriks		Mengembalikan matriks hasil transpose matriks

III. 2. Class Operations

III. 2. 1. Attribute

Class Operations tidak memiliki attribute.

III. 2. 2. Method

Nama	Tipe	Parameter	Deskripsi
isEselonBaris	public boolean	Matriks M	Mengembalikan nilai true apabila matriks M sudah sesuai dengan karakteristik matriks eselon baris
convertOne	public void	Double x, int r, Matriks m	Membagi elemen matriks pada suatu baris dengan elemen yang sudah disimpan sebelumnya untuk menjadikan elemen setelah 0 pertama pada baris menjadi 1
convertZero	public void	Double x, int rx,	Mengurangi elemen suatu

		int ry Matriks m	baris ry dengan baris rx yang sudah dikalikan dengan x untuk menjadikan 0 elemen dibawah leading one
allRowZero	public boolean	Matriks m, int r	Mengembalikan true apabila baris r seluruhnya berelemen 0
allColZero	Public boolean	Matriks m, int c	Mengembalikan true apabila kolom k seluruhnya berelemen 0
allZeroUnder	public boolean	Matriks m, int r, int c	Mengembalikan true apabila elemen-elemen di bawah elemen pada baris r dan kolom k seluruhnya berelemen 0
allZeroBefore	public boolean	Matriks m, int r	Mengembalikan true apabila elemen-elemen sebelum elemen pada baris r berelemen 0
swapRow	public void	Matriks m, int r1, int r2	Menukarkan elemen-elemen pada baris r1 dengan elemen-elemen pada baris r2
leadingOneRow	public int	Matriks m, int c	Mengembalikan baris yang menyimpan satu pertama pada

			kolom c
leadingOneCol	public int	Matriks m, int r	Mengembalikan kolom yang menyimpan satu pertama pada baris r
firstNoZeroRow	public int	Matriks m, int r	Mengembalikan kolom yang menyimpan elemen pertama bukan nol pada baris r
firstNoZeroCol	public int	Matriks m, int c	Mengembalikan baris yang menyimpan elemen pertama bukan nol pada kolom c
indented	public boolean	Matriks m, int r, int c	Mengembalikan true apabila kolom c lebih besar indeks nya dibandingkan koLOM dengan angka bukan nol pertama pada baris r
inversible	public boolean	Matriks m	Mengembalikan true apabila matriks m memiliki invers yang dilihat dari determinannya
isLowerTriangular	public boolean	Matriks m	Mengembalikan true apabila seluruh elemen di bawah diagaonal nol
isUpperTriangular	public	Matriks m	Mengembalikan

	boolean		true apabila seluruh elemen di atas diagonal nol
noZeroInDiagonal	public boolean	Matriks m	Mengembalikan true apabila tidak ditemukan nol pada diagonal matriks
isProcessed	public boolean	Matriks m	Mengembalikan boolean jika matriks sudah diproses (sudah sesuai kaidah untuk dapat dicari determinannya)
oneSolution	public boolean	Matriks m	Mengembalikan true apabila matriks yang diberikan memenuhi kriteria matriks satu solusi
manySolution	public String[]	Matriks m, String[] j	Mengembalikan array of string yang berisikan solusi dengan parametrik
concatMatriksSPL	public Matriks	Matriks A, Matriks b	Mengembalikan matriks hasil konkat matriks A dan b
toSolve	public void	Matriks m	Melakukan operasi substitusi untuk mendapatkan nilai x

III. 3. Class SPL

III. 3. 1. Attribute

Class SPL tidak memiliki attribute.

III. 3. 2. Method

Nama	Tipe	Parameter	Deskripsi
toGauss	public void	Matriks m	Melakukan eliminasi Gauss pada matriks m
solveByGauss	public String[]	Matriks m	Mengembalikan solusi solusi SPL dengan metode eliminasi Gauss pada suatu array of string
solveByGaussResult	public Matriks	Matriks m	Mengembalikan solusi solusi SPL dengan metode eliminasi Gauss pada suatu matriks
toGaussJordan	public void	Matriks m	Melakukan eliminasi Gauss-Jordan pada matriks m
solveByGaussJordan	public String[]	Matriks m	Mengembalikan solusi solusi SPL dengan metode eliminasi Gauss-Jordan pada suatu array of string
solveByGaussJordan Result	public Matriks	Matriks m	Mengembalikan solusi solusi

			SPL dengan metode eliminasi Gauss-Jordan pada suatu matriks
solveByInverse	public String[]	Matriks m	Mengembalikan solusi solusi SPL dengan metode invers matriks dalam suatu array of string
Cramer	public String[]	Matriks M	Mengembalikan solusi SPL dengan metode Cramer matriks dalam suatu array of string
changeColwithb	public void	Matriks M, int i, Matriks b	Mengganti matriks M dengan kolom jawaban b

III. 4. Class Determinant

III. 4. 1. Attribute

Class Determinant tidak memiliki attribute.

III. 4. 2. Method

Nama	Tipe	Parameter	Deskripsi
cofactorAt	public double	Matriks M, int r, int k	Mengembalikan nilai double hasil kofaktor dari elemen a _{ij} . Selengkapnya lihat teori singkat II.6
determinantByCofac	public	Matriks M	Mengembalikan

tor	double		determinan dari matriks M yang didapatkan dengan menggunakan metode ekspansi kofaktor
determinantByRowReduction	public double	Matriks m	Mengembalikan determinan dari matriks M yang didapatkan dengan menggunakan metode reduksi baris

III. 5. Class Invers

III. 5. 1. Attribute

Class Invers tidak memiliki attribute.

III. 5. 2. Method

Nama	Tipe	Parameter	Deskripsi
matriksKofaktor	public Matriks	Matriks M	Mengembalikan matriks kofaktor dari matriks M
matriksAdjoin	public Matriks	Matriks M	Mengembalikan matriks adjoin dari matriks M
inversMatriks	public Matriks	Matriks M	Mengembalikan matriks invers dari matriks M
inversByGaussJordan	public Matriks	Matriks M	Mengembalikan matriks yang sudah diinvers dengan metode Gauss-Jordan

III. 6. Class Interpolasi

III. 6. 1. Attribute

Class Interpolasi tidak memiliki attribute.

III. 6. 2. Method

Nama	Tipe	Parameter	Deskripsi
solveByInterpolasi	public void	double[] x, double[] y, double I, Int nt	Melakukan interpolasi berdasar data yang didapatkan dari array x dan a

III. 7. Class Regresi

III. 7. 1. Attribute

Class Regresi tidak memiliki attribute.

III. 7. 2. Method

Nama	Tipe	Parameter	Deskripsi
multipleLinearRegression	public Matriks	int numOfVariable, int numOfData, Matriks dataMatrix	Mengembalikan matriks hasil multiple linear regression
mlrEquation	public void	Matriks solution	Menampilkan solusi dari matriks linear regression
mlrEstimation	public void	Matriks solution, Matriks untukDitaksir	Menampilkan hasil taksiran dari titik yang ingin ditaksir

III. 8. Class Bicubic

III. 8. 1. Attribute

Class Bicubic tidak memiliki attribute.

III. 8. 2. Method

Nama	Tipe	Parameter	Deskripsi
makeMatriksX	public Matriks		
convertMatriksBic	public Matriks	Matriks x4	
bicubicResult	public double	Matriks f, double xt, double yt	

III. 9. Class Util

III. 8. 1. Attribute

Class Util tidak memiliki attribute.

III. 8. 2. Method

Nama	Tipe	Deskripsi
welcome	public void	Menampilkan selamat datang ke program
bye	public void	Menampilkan pesan keluar dari program
displayMenu	public void	Menampilkan menu utama
displayMenuSPL	public void	Menampilkan menu SPL
displayMenuDet	public void	Menampilkan menu determinan
displayMenuInvers	public void	Menampilkan menu invets

BAB IV

Eksperimen

Program Flow

Program Kalkulator Matriks yang kami buat memiliki *flow kerja* seperti berikut

1. Menampilkan 'selamat datang' dan menu awal. Memilih operasi yang akan dilakukan.

```
-----KALKULATOR MATRIKS-----  
Selamat datang di program pemanfaatan operasi matriks!  
-----MENU-----  
1. Sistem Persamaan Linier  
2. Determinan  
3. Invers  
4. Interpolasi Polinom  
5. Interpolasi Bicubic Spline  
6. Regresi Linear Berganda  
7. Perbesaran Citra  
8. Keluar  
Operasi apa yang ingin kamu lakukan? (Masukkan pilihan menu dalam angka):
```

2. User dapat memilih untuk melakukan Input/Output

```
Pilih cara input? 1. Keyboard 2. File .txt: 2  
Pilih cara output? 1. Console 2. File .txt: 1
```

3. Jika user memilih melakukan SPL, Determinan, atau Invers. Pengguna dapat memilih metode yang dilakukan.

```
-----MENU SPL-----  
1. Metode Eliminasi Gauss  
2. Metode Eliminasi Gauss-Jordan  
3. Metode Matriks Balikan  
4. Kaidah Cramer
```

```
-----MENU DETERMINAN-----  
1. Metode Reduksi Baris  
2. Metode Ekspansi Kofaktor
```

```
-----MENU INVERS-----  
1. Metode Matriks Balikan  
2. Metode Adjoin
```

```
Operasi apa yang ingin kamu lakukan? (Masukkan pilihan menu dalam angka): 1
```

4. Jika pengguna memilih untuk memasukkan dari keyboard, maka akan ada berbagai instruksi agar sesuai, misal agar pengguna memasukkan matriks persegi, matriks dengan n peubah, dan sebagainya.

```
Masukkan SPL yang ingin kamu selesaikan dalam bentuk matriks A:  
Masukkan jumlah baris matriks: 3  
Masukkan jumlah kolom matriks: 3  
Masukkan nilai nilai matriks A:  
1 2 3  
1 2 3  
1 2 3  
Masukkan nilai nilai hasil dalam bentuk matriks b:  
8  
9  
7
```

Perlu diingat bahwa kaidah Cramer khusus untuk SPL dengan n peubah dan n persamaan!

```
Masukkan Matriks yang ingin kamu cari determinannya:  
Masukkan nilai n (jumlah baris dan kolom harus sama agar memiliki determinan): 3
```

```
Jumlah titik yang akan dimasukkan: 2  
Titik 1:  
--> x1, y1: 1 3  
Titik 2:  
--> x2, y2: 4 5  
Value x yang akan ditaksir: 0.2
```

```
Masukkan Matriks 4x4 sebagai titik dasar dan titik turunan yang diketahui:  
Masukkan nilai nilai matriks:  
1 2 3 4  
5 6 7 1  
2 3 4 5  
8 9 1 0  
Masukkan absis titik yang ingin kita cari: 1  
Masukkan ordinat titik yang ingin kita cari: 2
```

```
Masukkan jumlah peubah x: 3  
Masukkan jumlah data: 3  
1 2 3 4  
5 6 7 8  
1 3 4 5
```

```

Masukkan nilai variabel peubah 1: 1
Masukkan nilai variabel peubah 2: 2
Masukkan nilai variabel peubah 3: 3

```

5. Jika pengguna memilih untuk keluar, maka program akan selesai

```

8. Keluar
Operasi apa yang ingin kamu lakukan? (Masukkan pilihan menu dalam angka): 8
-----TERIMA KASIH-----

```

IV. 1. Solusi SPL $Ax = B$

Test Case	Hasil
$A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$	<p>Tidak memiliki matriks balikan!</p> <p>Tidak bisa dipecahkan dengan metode matriks balikan!</p> <p>Tidak ada solusi.</p>
$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$	<p>Tidak bisa dipecahkan dengan metode matriks balikan!</p> <p>x1 = 3.0 + r x2 = 2.0r x3 = s x4 = -1.0 + r x5 = r</p>
$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$	<p>Tidak bisa dipecahkan dengan metode matriks balikan!</p> <p>x1 = r x2 = 1.0 - s x3 = t x4 = -2.0 - s x5 = 1.0 + s x 6 = s</p>

IV. 2. SPL berbentuk matriks *augmented*

Test Case	Hasil
-----------	-------

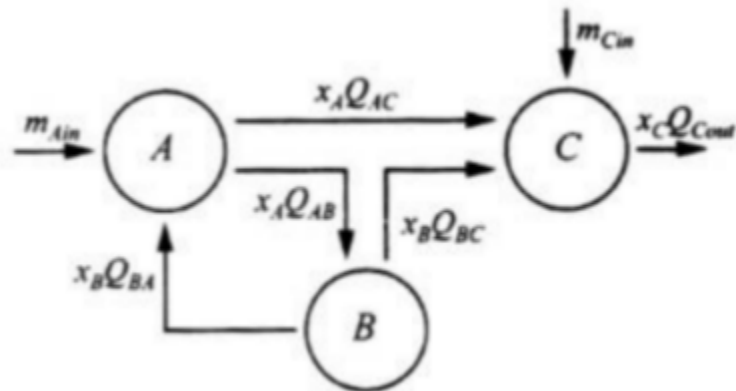
$\begin{bmatrix} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{bmatrix}$	<hr/> 1 $x_1 = -1.0 + r$ 2 $x_2 = 2.0s$ 3 $x_3 = s$ 4 $x_4 = r$ <hr/>
$\begin{bmatrix} 2 & 0 & 8 & 0 & 8 \\ 0 & 1 & 0 & 4 & 6 \\ -4 & 0 & 6 & 0 & 6 \\ 0 & -2 & 0 & 3 & -1 \\ 2 & 0 & -4 & 0 & -4 \\ 0 & 1 & 0 & -2 & 0 \end{bmatrix}$	<hr/> 1 $x_1 = 0.0$ 2 $x_2 = 2.0$ 3 $x_3 = 1.0$ 4 $x_4 = 1.0$ <hr/>

IV. 3. SPL berbentuk

Test Case	Hasil
$\begin{aligned} 8x_1 + x_2 + 3x_3 + 2x_4 &= 0 \\ 2x_1 + 9x_2 - x_3 - 2x_4 &= 1 \\ x_1 + 3x_2 + 2x_3 - x_4 &= 2 \\ x_1 + 6x_3 + 4x_4 &= 3 \end{aligned}$	<hr/> 1 $x_1 = -0.22432432432432436$ 2 $x_2 = 0.18243243243243246$ 3 $x_3 = 0.7094594594594594$ 4 $x_4 = -0.25810810810810797$ <hr/>
$\begin{aligned} x_7 + x_8 + x_9 &= 13.00 \\ x_4 + x_5 + x_6 &= 15.00 \\ x_1 + x_2 + x_3 &= 8.00 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_6 + x_8) + 0.61396x_9 &= 14.79 \\ 0.91421(x_3 + x_5 + x_7) + 0.25(x_2 + x_4 + x_6 + x_8) &= 14.31 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_2 + x_4) + 0.61396x_1 &= 3.81 \\ x_3 + x_6 + x_9 &= 18.00 \\ x_2 + x_5 + x_8 &= 12.00 \\ x_1 + x_4 + x_7 &= 6.00 \\ 0.04289(x_1 + x_5 + x_9) + 0.75(x_2 + x_6) + 0.61396x_3 &= 10.51 \\ 0.91421(x_1 + x_5 + x_9) + 0.25(x_2 + x_4 + x_6 + x_8) &= 16.13 \\ 0.04289(x_1 + x_5 + x_9) + 0.75(x_4 + x_8) + 0.61396x_7 &= 7.04 \end{aligned}$	$\begin{aligned} x_1 &= 0.0 \\ x_2 &= 0.0 \\ x_3 &= 0.0 \\ x_4 &= 0.0 \\ x_5 &= 0.0 \\ x_6 &= 0.0 \\ x_7 &= 0.0 \\ x_8 &= 0.0 \\ x_9 &= 0.0 \end{aligned}$ <p>solusi di samping saat disubstitusikan ke persamaan tidak menghasilkan hasil yang sama sehingga sebenarnya matriks ini</p>

	tidak memiliki solusi akan tetapi dituliskan sebagai berikut sesuai dengan karakteristik dari matriks dengan banyak solusi yaitu terdapat baris yang seluruhnya berelemen 0.
--	--

IV. 4. SPL dengan sistem reaktor pada gambar berikut



Dengan laju volume Q dalam m^3/s dan input massa min dalam mg/s . Konservasi massa pada tiap inti reaktor adalah sebagai berikut:

$$A: m_{Ain} + Q_{BA}x_B - Q_{AB}x_A - Q_{AC}x_A = 0$$

$$B: Q_{AB}x_A - Q_{BA}x_B - Q_{BC}x_B = 0$$

$$C: m_{Cin} + Q_{AC}x_A + Q_{BC}x_B - Q_{Cout}x_C = 0$$

Tentukan solusi x_A, x_B, x_C dengan menggunakan parameter berikut : $Q_{AB} = 40, Q_{AC} = 80, Q_{BA} = 60, Q_{BC} = 20$ dan $Q_{Cout} = 150 m^3/s$ dan $m_{Ain} = 1300$ dan $m_{Cin} = 200 mg/s$.

Hasil	
1	$x_1 = 0.1r$
2	$x_2 = 1.4444444444444444r$
3	$x_3 = 0.7222222222222222r$
4	$x_4 = r$

IV. 5. Studi Kasus Interpolasi

Menggunakan data berikut:

x	8	9	9.5
$f(x)$	2.0794	2.1972	2.2513

$x = 9.2$ $f(x) = \dots$

Hasil
<div>Persamaan polinom interpolasi berdasarkan data tersebut adalah sebagai berikut: $f(X) = 0.676 + 0.227 X^1 - 0.006 X^2$ Hasil: 2.219</div>

IV. 6. Studi Kasus Regresi Linear Berganda

Test Case

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.

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

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

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

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

Hasil

Persamaan regresi linear berganda berdasarkan data tersebut adalah sebagai berikut:
 $f(X) = -3.508 - 0.003 X_1 + 0.001 X_2 + 0.154 X_3$
 Estimasi nilai variabel terikat berdasarkan regresi linear berganda adalah:
 0.938

IV. 7. Studi Kasus Interpolasi *Bicubic Spline*

Test Case	Hasil
-----------	-------

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

Tentukan nilai:

$$f(0, 0) = ?$$

$$f(0.5, 0.5) = ?$$

$$f(0.25, 0.75) = ?$$

$$f(0.1, 0.9) = ?$$

1 Hasil dari petaan bicubic spline nya adalah 21.000

1 Hasil dari petaan bicubic spline nya adalah 148.719

1 Hasil dari petaan bicubic spline nya adalah 137.999

1 Hasil dari petaan bicubic spline nya adalah 107.103

Bab V

Penutup

V. 1. Kesimpulan

1. Solusi SPL dapat ditemukan 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). Metode tersebut dapat dengan lebih mudah dan lebih cepat jika diimplementasikan dengan bantuan program komputer.
2. Determinan matriks persegi juga dapat dihitung dengan reduksi baris dan ekspansi kofaktor. Matriks berukuran kecil mungkin dapat dengan mudah dihitung secara manual oleh manusia. Namun dengan program komputer yang dibuat ini, penghitungan matriks berukuran besar lebih mudah dilakukan.
3. Invers matriks persegi yang determinannya tidak nol dapat dilakukan dengan metode adjoin dan matriks balikan yang juga lebih efisien dilakukan dengan program komputer.
4. Berbagai metode yang melibatkan matriks dapat digunakan untuk menyelesaikan persoalan matematika yang lebih kompleks seperti interpolasi polinom, interpolasi bicubic spline, dan regresi linear berganda.

V. 2. Saran

1. Pengerjaan tugas besar seharusnya dilakukan dengan lebih bertahap sehingga lebih bisa melakukan eksplorasi
2. Nama variabel, fungsi, dan komentar ditulis lebih baik untuk meningkatkan keterbacaan

V. 3. Komentar dan Refleksi

Dalam pengerjaan tugas ini, kami banyak belajar dan bereksplorasi terutama terkait implementasi aljabar linear dalam bentuk matriks dan implementasinya dalam berbagai bentuk matematika. Banyaknya kekurangan kami masih dapat diperbaiki dengan terus belajar dan bereksplorasi. Semoga di kesempatan selanjutnya kami dapat melakukan tugas lebih baik lagi.

V. 4. Link Repository

Link repository kelompok kami untuk tugas besar 1 mata kuliah IF2123 Aljabar Linier dan Geometri adalah sebagai berikut:

<https://github.com/shulhajws/TubesAlgeoPertama>

Referensi

<https://www.tutorialspoint.com/get-the-current-working-directory-in-java>

https://p2k.stekom.ac.id/ensiklopedia/Kaidah_Cramer

https://bookdown.org/moh_rosidi2610/Metode_Numerik/interpolation.html