

LAPORAN TUGAS BESAR 1

IF2123 ALJABAR LINEAR DAN GEOMETRI

Sistem Persamaan Linier, Determinan, dan Aplikasinya



Disusun oleh :

Alfian Hanif Fitria Yustanto	13523073
Jethro Jens Norbert Simatupang	13523081
Carlo Angkisan	13523091

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
JL. GANESA 10, BANDUNG 40132
2024

DAFTAR ISI

DAFTAR ISI.....	2
BAB I.....	4
1.1 Tujuan.....	4
1.2 Spesifikasi Program.....	4
BAB II.....	7
2.1 Metode Eliminasi Gauss.....	7
2.2 Metode Eliminasi Gauss-Jordan.....	7
2.3 Determinan.....	8
2.4 Matriks Balikan.....	9
2.5 Matriks Kofaktor.....	9
2.6 Matriks Adjoin.....	10
2.7 Kaidah Cramer.....	10
2.8 Interpolasi Polinom.....	10
2.9 Interpolasi Bicubic Spline.....	11
2.10 Regresi Linier dan Kuadratik Berganda.....	13
BAB III.....	15
3.1 Class Matrix.....	15
3.2 Class invers.....	17
3.3 Class determinan.....	18
3.4 Class SPL.....	19
3.5 Class RegresiBerganda.....	20
3.6 Class InterpolasiPolinomial.....	20
3.7 Class BicubicSplineInterpolation.....	21
3.8 Class ImageResizing.....	21
3.9 Class IO.....	22
3.10 Class Main.....	23
BAB IV.....	25
4.1 Solusi SPL $Ax = b$	25
4.2 SPL Berbentuk Matriks Augmented.....	27
4.3 Sistem Persamaan Linear.....	27
4.4 Sistem Reaktor.....	28
4.5 Studi Kasus Interpolasi.....	29
4.6 Studi Kasus Regresi Linear dan Kuadratik Berganda.....	31
4.7 Studi Kasus Interpolasi Bicubic Spline.....	32
4.8 Bonus : Image Resizing.....	33
BAB V.....	34
5.1 Kesimpulan.....	34
5.2 Saran.....	34

5.3 Refleksi.....	34
5.4 Komentar.....	35
LAMPIRAN.....	36

BAB I

DESKRIPSI MASALAH

1.1 Tujuan

Tugas besar ini dibuat untuk memenuhi tugas besar I mata kuliah IF2123 Aljabar Linear dan Geometri. Tujuan dari tugas besar ini adalah sebagai berikut.

1. Membuat satu atau lebih library yang diimplementasikan dalam Bahasa Java untuk :
 - Menemukan solusi SPL dengan metode eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer (khusus untuk SPL dengan n peubah dan n persamaan).
 - Menghitung determinan matriks dengan reduksi baris dan ekspansi kofaktor.
 - Menghitung balikan matriks dari sebuah matriks persegi.

1.2 Spesifikasi Program

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 invers, metode yang digunakan ada 2 yaitu menggunakan OBE dan Matriks Adjoin.
4. 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

5. Untuk persoalan regresi, masukannya jika dari *keyboard* adalah n (jumlah peubah x), m (jumlah sampel), semua nilai-nilai $x_{1i}, x_{2i}, \dots, x_{ni}$, 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.
6. 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 $x_1 = t$).
7. 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

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

untuk kasus regresi kuadratik, variabel boleh menggunakan x_1, x_2 , dan lain-lain tetapi perlu dijelaskan variabel tersebut merepresentasikan apa. Contoh

$$x_1 = X$$

.

$$x_3 = X^2$$

.

$$x_5 = XY$$

[Persamaan dan Solusi]

8. 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 $f(0.5, 0.5)$.

9. Luaran program harus dapat ditampilkan pada layar komputer dan dapat disimpan ke dalam *file*.
10. Bahasa program yang digunakan adalah Java. Anda bebas untuk menggunakan versi java apapun dengan catatan di atas java versi 8 (8/9/11/15/17/19/20/21).
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 dan kuadratik berganda
7. Interpolasi Gambar (Bonus)
8. 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, 3, dan 6

BAB II

TEORI SINGKAT

2.1 Metode Eliminasi Gauss

Metode Eliminasi Gauss merupakan metode yang dikembangkan dari metode eliminasi, yaitu menghilangkan atau mengurangi jumlah variabel sehingga dapat diperoleh nilai dari suatu variabel bebas. Untuk menyelesaikan sistem persamaan linear dengan Metode Eliminasi Gauss, dilakukan langkah-langkah sebagai berikut.

1. Membentuk matriks augmented

Matriks augmented merupakan suatu bentuk matriks yang diperoleh dengan menggabungkan matriks koefisien dari suatu sistem persamaan linear dengan vektor konstanta dari persamaan tersebut.

Contoh:

$$\begin{array}{rcl} x_1 + 3x_2 - 6x_3 & = & 9 \\ 2x_1 - 6x_2 + 4x_3 & = & 7 \\ 5x_1 + 2x_2 - 5x_3 & = & -2 \end{array} \quad \text{Matriks Augmented} \rightarrow \begin{bmatrix} 1 & 3 & -6 & 9 \\ 2 & -6 & 4 & 7 \\ 5 & 2 & -5 & -2 \end{bmatrix}$$

2. Melakukan Operasi Baris Elementer (OBE)

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

Operasi Baris Elementer (OBE) merupakan suatu operasi yang diterapkan pada baris suatu matriks. Metode ini digunakan untuk mengubah matriks koefisien dari sistem persamaan linear menjadi bentuk segitiga atau bentuk eselon baris sehingga mudah untuk menemukan solusi dari sistem tersebut. Berikut tiga operasi baris elementer terhadap matriks augmented:

1. Kalikan sebuah baris dengan konstanta tidak nol.
 2. Pertukarkan dua buah baris.
 3. Tambahkan sebuah baris dengan kelipatan baris lainnya.
3. Melakukan substitusi balik (*backward substitution*)

Setelah terbentuk matriks eselon baris, maka lakukan substitusi mundur dari baris terakhir dan bergerak ke atas hingga menemukan semua solusi sistem persamaan linear (SPL).

2.2 Metode Eliminasi Gauss-Jordan

Metode Gauss-Jordan merupakan pengembangan dari metode eliminasi Gauss dalam menyelesaikan sistem persamaan linear. Metode ini menggunakan operasi baris elementer

untuk mengubah matriks augmented menjadi bentuk eselon baris tereduksi (reduced row echelon form), di mana setiap baris non-nol memiliki elemen utama bernilai satu, dan elemen di atas dan di bawahnya bernilai nol. Dengan transformasi ini, solusi sistem persamaan linear dapat langsung diperoleh tanpa memerlukan substitusi mundur, yang umumnya diperlukan pada metode Gauss.

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

2.3 Determinan

Determinan adalah konsep matematika yang digunakan untuk mengukur sifat-sifat matriks, yang merupakan susunan bilangan dalam bentuk persegi, yaitu matriks yang memiliki jumlah baris dan kolom yang sama. Determinan suatu matriks dapat dihitung dengan berbagai metode, antara lain metode reduksi baris dan metode ekspansi kofaktor. Pada metode reduksi baris, diterapkan operasi baris elementer (OBE) pada matriks persegi sehingga menghasilkan matriks segitiga. Determinan dari matriks segitiga ini dapat dihitung dengan mudah sebagai hasil kali dari elemen-elemen pada diagonal utama.

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

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

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

p menyatakan banyaknya operasi pertukaran baris di dalam OBE dan k_1, k_2, \dots, k_m adalah perkalian baris-baris matriks dengan konstanta selama operasi OBE.

Untuk metode ekspansi kofaktor, misalkan A adalah matriks berukuran $n \times n$.

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

Serta didefinisikan :

M_{ij} adalah minor entri dari elemen a_{ij}

$C_{ij} = (-1)^{i+j} M_{ij}$ atau kofaktor entri a_{ij}

Dari kedua informasi diatas, untuk menghitung determinannya dapat digunakan salah satu dari persamaan-persamaan berikut.

$$\det(A) = a_{11}C_{11} + a_{12}C_{12} + \dots + a_{1n}C_{1n}$$

$$\det(A) = a_{21}C_{21} + a_{22}C_{22} + \dots + a_{2n}C_{2n}$$

\vdots

$$\det(A) = a_{n1}C_{n1} + a_{n2}C_{n2} + \dots + a_{nn}C_{nn}$$

Secara baris

$$\det(A) = a_{11}C_{11} + a_{21}C_{21} + \dots + a_{n1}C_{n1}$$

$$\det(A) = a_{12}C_{12} + a_{22}C_{22} + \dots + a_{n2}C_{n2}$$

\vdots

$$\det(A) = a_{1n}C_{1n} + a_{2n}C_{2n} + \dots + a_{nn}C_{nn}$$

Secara kolom

2.4 Matriks Balikan

Matriks Balikan dapat diartikan sebagai matriks persegi berukuran $n \times n$ yang memenuhi relasi $A(A)^{-1} = (A)^{-1}A = I$. Apabila suatu matriks memiliki determinan yang bernilai nol, maka matriks tersebut tidak memiliki invers. Terdapat dua metode untuk memperoleh matriks balikan, yaitu melalui metode Gauss-Jordan dan metode adjoin.

Dalam penerapan metode Gauss-Jordan, persamaan yang digunakan yaitu :

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

Ketika operasi dilakukan pada sisi kiri, sisi kanan juga akan mengalami perubahan, sehingga sisi kiri akan menjadi matriks identitas, dan sisi kanan akan menjadi invers dari sisi kiri.

Untuk metode adjoin dapat menerapkan persamaan sebagai berikut.

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

2.5 Matriks Kofaktor

Matriks kofaktor adalah suatu matriks yang terbentuk dari kofaktor masing-masing entri dalam suatu matriks. Kofaktor dari elemen a_{ij} pada matriks A dinyatakan dengan simbol C_{ij} dan dapat dihitung dengan rumus $C_{ij} = (-1)^{i+j}M_{ij}$, dengan M_{ij} adalah minor dari entri a_{ij} , yang merupakan determinan dari submatriks yang diperoleh dengan menghapus baris ke- i dan kolom ke- j dari matriks A .

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

$$M_{23} = \begin{vmatrix} 6 & -3 \\ 1 & 5 \end{vmatrix}$$

Dari minor-minor dengan i dan j yang berbeda, kemudian disusun matriks kofaktor sebagai berikut:

$$\begin{bmatrix} + \begin{vmatrix} 6 & 3 \\ -4 & 0 \end{vmatrix} & - \begin{vmatrix} 1 & 3 \\ 2 & 0 \end{vmatrix} & + \begin{vmatrix} 1 & 6 \\ 2 & -4 \end{vmatrix} \\ - \begin{vmatrix} 2 & -1 \\ -4 & 0 \end{vmatrix} & + \begin{vmatrix} 3 & -1 \\ 2 & 0 \end{vmatrix} & - \begin{vmatrix} 3 & 2 \\ 2 & -4 \end{vmatrix} \\ + \begin{vmatrix} 2 & -1 \\ 6 & 3 \end{vmatrix} & - \begin{vmatrix} 3 & -1 \\ 1 & 3 \end{vmatrix} & + \begin{vmatrix} 3 & 2 \\ 1 & 6 \end{vmatrix} \end{bmatrix} = \begin{bmatrix} 12 & 6 & -16 \\ 4 & 2 & 16 \\ 12 & -10 & 16 \end{bmatrix}$$

2.6 Matriks Adjoin

Matriks adjoin adalah transpose dari matriks kofaktor ($\text{adj}(A) = \text{transpose matriks kofaktor}$). Setiap elemen matriks adjoin merupakan kofaktor dari matriks asal. Matriks ini berperan penting dalam penghitungan invers matriks.

$$\text{matriks kofaktor: } \begin{bmatrix} 12 & 6 & -16 \\ 4 & 2 & 16 \\ 12 & -10 & 16 \end{bmatrix} \quad \text{adj}(A) = \begin{bmatrix} 12 & 4 & 12 \\ 6 & 2 & -10 \\ -16 & 16 & 16 \end{bmatrix}$$

2.7 Kaidah Cramer

Kaidah Cramer adalah metode untuk menyelesaikan sistem persamaan linear dengan menggunakan determinan. Jika $Ax = b$ adalah SPL yang terdiri dari n persamaan linier dengan n peubah (variable) sedemikian sehingga $\det(A) \neq 0$, maka SPL tersebut memiliki solusi yang unik yaitu :

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

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

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

2.8 Interpolasi Polinom

Interpolasi polinom adalah metode untuk mengestimasi nilai-nilai antara titik-titik data yang ada atau untuk memprediksi nilai di luar rentang data yang telah diberikan. Polinom interpolasi derajat n yang menginterpolasi titik-titik $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$. adalah berbentuk $p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$. Jika hanya ada dua titik, (x_0, y_0) dan (x_1, y_1) , maka polinom yang menginterpolasi kedua titik tersebut adalah $p_1(x) = a_0 + a_1x$ yaitu berupa persamaan garis lurus. Jika tersedia tiga titik, $(x_0, y_0), (x_1, y_1)$, dan (x_2, y_2) ,

maka polinom yang menginterpolasi ketiga titik tersebut adalah $p_2(x) = a_0 + a_1x + a_2x^2$ atau persamaan kuadrat dan kurvanya berupa parabola. Jika tersedia empat titik, (x_0, y_0) , (x_1, y_1) , (x_2, y_2) , dan (x_3, y_3) , polinom yang menginterpolasi keempat titik tersebut adalah $p_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3$, demikian seterusnya. Dengan cara yang sama kita dapat membuat polinom interpolasi berderajat n untuk n yang lebih tinggi asalkan tersedia $(n+1)$ buah titik data. Dengan menyulihkan (x_i, y_i) ke dalam persamaan polinom $p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ untuk $i = 0, 1, 2, \dots, n$, akan diperoleh n buah sistem persamaan linier dalam $a_0, a_1, a_2, \dots, a_n$,

$$\begin{aligned} a_0 + a_1x_0 + a_2x_0^2 + \dots + a_nx_0^n &= y_0 \\ a_0 + a_1x_1 + a_2x_1^2 + \dots + a_nx_1^n &= y_1 \\ &\dots \\ a_0 + a_1x_n + a_2x_n^2 + \dots + a_nx_n^n &= y_n \end{aligned}$$

Solusi sistem persamaan linier ini, yaitu nilai a_0, a_1, \dots, a_n , diperoleh dengan menggunakan metode eliminasi Gauss yang sudah anda pelajari. Sebagai contoh, misalkan diberikan tiga buah titik yaitu $(8.0, 2.0794)$, $(9.0, 2.1972)$, dan $(9.5, 2.2513)$. Tentukan polinom interpolasi kuadrat lalu estimasi nilai fungsi pada $x = 9.2$. Polinom kuadrat berbentuk $p_2(x) = a_0 + a_1x + a_2x^2$. Dengan menyulihkan ketiga buah titik data ke dalam polinom tersebut, diperoleh sistem persamaan linier yang terbentuk adalah

$$\begin{aligned} a_0 + 8.0a_1 + 64.00a_2 &= 2.0794 \\ a_0 + 9.0a_1 + 81.00a_2 &= 2.1972 \\ a_0 + 9.5a_1 + 90.25a_2 &= 2.2513 \end{aligned}$$

Penyelesaian sistem persamaan dengan metode eliminasi Gauss menghasilkan $a_0 = 0.6762$, $a_1 = 0.2266$, dan $a_2 = -0.0064$. Polinom interpolasi yang melalui ketiga buah titik tersebut adalah $p_2(x) = 0.6762 + 0.2266x - 0.0064x^2$. Dengan menggunakan polinom ini, maka nilai fungsi pada $x = 9.2$ dapat ditaksir sebagai berikut: $p_2(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(9.2)^2 = 2.2192$.

2.9 Interpolasi Bicubic Spline

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

Dalam pemrosesan menggunakan interpolasi *bicubic spline* digunakan 16 buah titik, 4 titik referensi utama di bagian pusat, dan 12 titik di sekitarnya sebagai aproksimasi

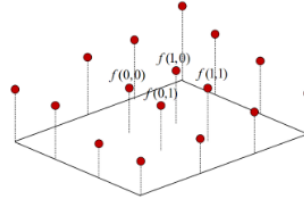
turunan dari keempat titik referensi untuk membagun permukaan bikubik. Bentuk pemodelannya adalah sebagai berikut.

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

$f(0,1), f(1,1)$

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

Solve: a_{ij}



Selain melibatkan model dasar, juga digunakan model turunan berarah dari kedua sumbu, baik terhadap sumbu x, sumbu y, maupun keduanya. Persamaan polinomial yang digunakan adalah sebagai berikut.

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

$$f_x(x,y) = \sum_{j=0}^3 \sum_{i=1}^3 a_{ij} i x^{i-1} y^j$$

$$f_y(x,y) = \sum_{j=1}^3 \sum_{i=0}^3 a_{ij} j x^i y^{j-1}$$

$$f_{xy}(x,y) = \sum_{j=0}^3 \sum_{i=0}^3 a_{ij} i j x^{i-1} y^{j-1}$$

Dengan menggunakan nilai fungsi dan turunan berarah tersebut, dapat terbentuk sebuah matriks solusi X yang membentuk persamaan penyelesaian sebagai berikut.

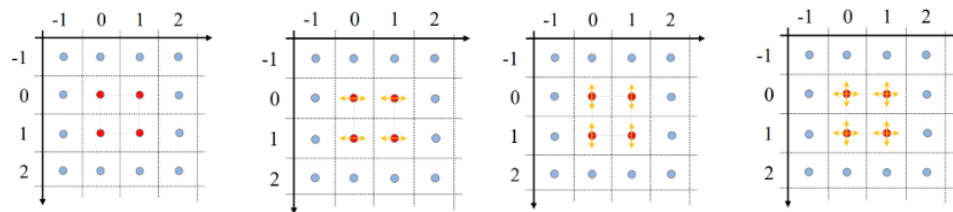
$$y = Xa$$

$$\begin{bmatrix} f(0,0) \\ f(1,0) \\ f(0,1) \\ f(1,1) \\ f_x(0,0) \\ f_x(1,0) \\ f_x(0,1) \\ f_x(1,1) \\ f_y(0,0) \\ f_y(1,0) \\ f_y(0,1) \\ f_y(1,1) \\ f_{xy}(0,0) \\ f_{xy}(1,0) \\ f_{xy}(0,1) \\ f_{xy}(1,1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 2 & 4 & 6 & 0 & 3 & 6 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30} \\ a_{01} \\ a_{11} \\ a_{21} \\ a_{31} \\ a_{02} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{03} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}$$

Perlu diketahui bahwa elemen pada matriks X adalah nilai dari setiap komponen koefisien a_{ij} yang diperoleh dari persamaan fungsi maupun persamaan turunan yang telah dijelaskan sebelumnya. Sebagai contoh, elemen matriks X pada baris 8 kolom ke 2

adalah koefisien dari a_{10} pada ekspansi sigma untuk $f_x(1, 1)$ sehingga diperoleh nilai konstanta $1 \times 11 - 1 \times 10 = 1$, sesuai dengan isi matriks X .

Nilai dari vektor a dapat dicari dari persamaan $y = Xa$, lalu vektor a tersebut digunakan sebagai nilai variabel dalam $f(x, y)$, sehingga terbentuk fungsi interpolasi bicubic sesuai model. Tugas Anda pada studi kasus ini adalah membangun persamaan $f(x, y)$ yang akan digunakan untuk melakukan interpolasi berdasarkan nilai $f(a, b)$ dari masukan matriks 4×4 . Nilai masukan a dan b berada dalam rentang $[0, 1]$. Nilai yang akan diinterpolasi dan turunan berarah disekitarnya dapat diilustrasikan pada titik berwarna merah pada gambar di bawah.



2.10 Regresi Linier dan Kuadratik Berganda

Regresi adalah metode statistik yang digunakan untuk memodelkan dan menganalisis hubungan antara sebuah variabel terikat dengan satu atau lebih variabel independen. Tujuan dari regresi adalah untuk memahami bagaimana perubahan pada variabel independen mempengaruhi variabel dependen, serta membuat prediksi berdasarkan model yang terbentuk. Terdapat beberapa jenis regresi, beberapa di antaranya adalah regresi linier berganda dan regresi kuadratik berganda.

1. Regresi Linier Berganda

Regresi linier berganda adalah jenis regresi yang memodelkan hubungan linier antara satu variabel dependen dengan beberapa variabel independen. Persamaan umum untuk regresi linier adalah sebagai berikut.

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

Untuk mendapatkan nilai dari setiap β_i dapat digunakan *Normal Estimation Equation for Multiple Linear Regression* sebagai berikut.

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

2. Regresi Kuadratik Berganda

Regresi kuadratik berganda adalah regresi yang memodelkan hubungan antara satu variabel dependen dengan beberapa variabel independen secara non-linier. Regresi kuadratik berganda disusun oleh 3 jenis variabel:

- (a) Variabel linier: Variabel dengan derajat satu, seperti X, Y, dan Z.
- (b) Variabel kuadrat: Variabel dengan derajat dua, seperti X^2 .
- (c) Variabel interaksi: Dua variabel derajat satu yang dikalikan, seperti XY, YZ, dan XZ.

Jumlah variabel linier, kuadrat, dan interaksi akan bervariasi tergantung pada jumlah n variabel independen. Misalnya pada regresi kuadratik 2 variabel

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

terdapat 2 variabel linier (u_i dan v_i), 2 variabel kuadrat (u_i^2 dan v_i^2), 1 variabel interaksi (uv), dan 1 konstan N. Untuk setiap n variabel independen, akan terdapat n variabel linier, n variabel kuadrat, C_2^n variabel interaksi, dan 1 konstan N. Dengan demikian, semakin banyak variabel independen yang digunakan, maka jumlah variabel dalam persamaan akan semakin banyak dan ukuran matriks persamaan juga akan bertambah besar, tetapi tetap dapat diselesaikan menggunakan metode eliminasi Gauss.

BAB III

IMPLEMENTASI PUSTAKA DAN PROGRAM

3.1 Class Matrix

- Attribute

Nama	Tipe Data	Deskripsi
matrix	double	Atribut yang digunakan untuk menyimpan elemen matrix.
rows	int	Atribut yang digunakan untuk menyimpan jumlah baris matrix.
cols	int	Atribut yang digunakan untuk menyimpan jumlah kolom matrix.

- Method

Nama	Tipe Data	Parameter	Deskripsi
Matrix	public Matrix	int _rows, int _cols	Fungsi konstruktor untuk membuat objek matriks dengan baris dan kolom tertentu.
getRowContent	public double[]	int row	Fungsi yang mengembalikan isi baris dari matriks berdasarkan indeks baris yang diberikan.
getColContent	public double[]	int col	Fungsi yang mengembalikan isi kolom dari matriks berdasarkan indeks kolom yang diberikan.
readMatrix	public void	-	Prosedur untuk membaca input matriks dari pengguna dan menangani kesalahan input.
printMatrix	public void	-	Prosedur untuk mencetak matriks dengan format dua angka di belakang koma.
getElmt	public double	int row, int col	Fungsi untuk mengambil elemen matriks pada posisi baris dan kolom tertentu.
getCol	public int	-	Fungsi yang mengembalikan jumlah kolom dari matriks.

getRow	public int	-	Fungsi yang mengembalikan jumlah baris dari matriks.
setEIMT	public void	int row, int col, double val	Prosedur untuk mengatur nilai elemen matriks pada posisi baris dan kolom tertentu.
swapRow	public void	int row1, int row2	Prosedur untuk menukar dua baris dalam matriks.
changeCol	public static Matrix	Matrix m, int nCol, Matrix newCol	Fungsi untuk mengubah kolom dalam matriks dengan kolom baru yang diberikan.
plusKRow	public void	int row, double k, int krow	Prosedur untuk menambahkan k kali baris lain ke baris yang diberikan.
divideRow	public void	int row, double k	Prosedur untuk membagi setiap elemen pada baris yang diberikan dengan konstanta k.
transpose	public void	-	Prosedur untuk melakukan transpose pada matriks.
rowLength	public int	int row	Fungsi untuk menghitung jumlah elemen nol dari awal sampai elemen bukan nol pada baris.
colLength	public int	int col	Fungsi untuk menghitung jumlah elemen nol dari awal sampai elemen bukan nol pada kolom.
sortRowByZero	public int	-	Fungsi untuk mengurutkan baris berdasarkan jumlah elemen nol dari sedikit ke banyak.
generateEselon	public void	-	Prosedur untuk menghasilkan bentuk eselon dari matriks.
generateEselon Reduksi	public void	-	Prosedur untuk menghasilkan bentuk eselon tereduksi dari matriks.
multiplyMatrix	public static Matrix	Matrix m1, Matrix m2	Fungsi untuk mengalikan dua matriks jika valid.

createAugmented	public static Matrix	Matrix m1, Matrix m2	Fungsi untuk membuat matriks augmented dari dua matriks.
disassembleAugmented	public static Matrix	Matrix m, boolean left	Fungsi untuk memisahkan matriks augmented menjadi matriks kiri atau kanan.
getHalfRight	public static Matrix	Matrix m	Fungsi untuk mengambil separuh kanan dari matriks.
getHalfLeft	public static Matrix	Matrix m	Fungsi untuk mengambil separuh kiri dari matriks.
createMatrixIdentity	public static Matrix	int nRowCol	Fungsi untuk membuat matriks identitas berukuran $n \times n$.
createMatrixKosong	public static Matrix	int nRowCol	Fungsi untuk membuat matriks kosong berukuran $n \times n$.
isSquare2x2	public boolean	-	Fungsi untuk memeriksa apakah matriks berukuran 2×2 dan persegi.
isSquare	public boolean	-	Fungsi untuk memeriksa apakah matriks berbentuk persegi.
haveInverse	public static boolean	Matrix m	Fungsi untuk memeriksa apakah matriks memiliki invers dengan menghitung determinan.
multiplyMatrixByConst	public static Matrix	Matrix m, double k	Fungsi untuk mengalikan setiap elemen matriks dengan konstanta.
copyMatrix	public static void	Matrix m1, Matrix m2	Prosedur untuk menyalin isi matriks m1 ke matriks m2.
createHilbert	public static Matrix	int n	Fungsi untuk membuat matriks Hilbert berukuran $n \times n$ dan mengembalikannya.
createHilbertSol	public static Matrix	int n	Fungsi untuk membuat matriks solusi Hilbert berukuran $n \times 1$ dan mengembalikannya.

3.2 Class invers

- Attribute

Class inverse tidak memiliki atribut.

- Method

Nama	Tipe Data	Parameter	Deskripsi
getInversOBE	public static Matrix	Matrix m	Fungsi untuk mengembalikan invers matriks menggunakan metode OBE (Operasi Baris Elementer), atau matriks kosong jika tidak memiliki invers.
showInversOBE	public static void	Matrix m	Prosedur untuk mencetak hasil invers dari matriks menggunakan metode OBE, atau pesan jika matriks tidak memiliki invers.
isInversValid	public static boolean	Matrix leftM	Fungsi untuk memeriksa apakah matriks sudah valid untuk invers setelah menggunakan metode OBE.
showInversAdjoin	public static void	Matrix m	Prosedur untuk mencetak hasil invers matriks menggunakan metode adjoin.
getInversAdjoin	public static Matrix	Matrix m	Fungsi untuk mengembalikan invers matriks menggunakan metode adjoin.

3.3 Class determinan

- Attribute
Class determinan tidak memiliki atribut.
- Method

Nama	Tipe Data	Parameter	Deskripsi
determinanReduksi	public static double	Matrix m	Fungsi untuk menghitung determinan matriks menggunakan metode eselon reduksi.
determinan2x2	public static double	Matrix m	Fungsi untuk menghitung determinan dari matriks berukuran 2x2.
getMinorEntri	public static Matrix	Matrix m, int row, int col	Fungsi untuk mengembalikan minor dari matriks dengan menghapus baris dan kolom tertentu.

getMatrixKofaktor	public static Matrix	Matrix m	Fungsi untuk menghasilkan matriks kofaktor dari matriks input.
getAdjoin	public static Matrix	Matrix m	Fungsi untuk menghasilkan matriks adjoin dari matriks input.
getDeterminanKofaktor	public static double	Matrix m	Fungsi untuk menghitung determinan matriks menggunakan ekspansi kofaktor secara rekursif.

3.4 Class SPL

- Attribute

Class SPL tidak memiliki atribut.

- Method

Nama	Tipe Data	Parameter	Deskripsi
getSolution	public static double[]	Matrix m	Fungsi untuk mendapatkan solusi unik dari sistem persamaan linier menggunakan metode eselon reduksi.
gaussJordanSolution	public static String[]	Matrix m	Fungsi untuk mencari solusi sistem persamaan linier menggunakan metode Gauss-Jordan. Menghasilkan solusi tunggal, banyak solusi, atau tidak ada solusi.
isUnique	public static boolean	Matrix m	Fungsi untuk memeriksa apakah sistem persamaan linier memiliki solusi unik.
isLotSolution	public static boolean	Matrix m	Fungsi untuk memeriksa apakah sistem persamaan linier memiliki banyak solusi.
isNoSolution	public static boolean	Matrix m	Fungsi untuk memeriksa apakah sistem persamaan linier tidak memiliki solusi.
gaussSolution	public static String[]	Matrix m	Fungsi untuk mencari solusi sistem persamaan linier menggunakan metode Gauss. Dapat menghasilkan solusi tunggal, banyak solusi, atau

			tidak ada solusi.
balikanSolution	public static String[]	Matrix A, Matrix B	Fungsi untuk mencari solusi sistem persamaan linier menggunakan metode invers matriks ($Ax = B$).
cramerSolution	public static String[]	Matrix A, Matrix B	Fungsi untuk mencari solusi sistem persamaan linier menggunakan metode Cramer.

3.5 Class RegresiBerganda

- Attribute

Class RegresiBerganda tidak memiliki atribut.

- Method

Nama	Tipe Data	Parameter	Deskripsi
RegresiLinierBerganda	public static String[]	int pilihan	Fungsi untuk melakukan regresi linier berganda berdasarkan input pengguna atau file.
RegresiKuadratikBerganda	public static String[]	int pilihan	Fungsi untuk melakukan regresi kuadratik berganda berdasarkan input pengguna atau file.
printCombinations	public static String	int n	Fungsi untuk menghasilkan kombinasi variabel linier, kuadrat, dan interaksi.
validasiInputInteger	public static int	Scanner scanner	Fungsi untuk memvalidasi input integer dari pengguna dan mengembalikan nilai yang valid.
validasiInputDouble	public static double	Scanner scanner, String message	Fungsi untuk memvalidasi input double dari pengguna dan mengembalikan nilai yang valid.
validasiInputDoubleArray	public static double[]	Scanner scanner, int expectedLength	Fungsi untuk memvalidasi input array double dari pengguna dan mengembalikan array yang valid.

3.6 Class InterpolasiPolinomial

- Attribute

Class InterpolasiPolinomial tidak memiliki atribut.

- Method

Nama	Tipe Data	Parameter	Deskripsi
main	public static String[]	int pilihan	Fungsi untuk mengambil input titik, melakukan regresi, dan menghasilkan hasil dalam bentuk polinomial.
validasiInputJumlahTitik	public static int	Scanner scanner	Fungsi untuk memvalidasi input jumlah titik dari pengguna dan mengembalikan nilai yang valid.
validasiInputDouble	public static double	Scanner scanner	Fungsi untuk memvalidasi input double dari pengguna dan mengembalikan nilai yang valid.

3.7 Class BicubicSplineInterpolation

- Attribute

Class BicubicSplineInterpolation tidak memiliki atribut.

- Method

Nama	Tipe Data	Parameter	Deskripsi
mainBicubicInterpolation	public static String[]	double[] temp	Program utama dari class BicubicSplineInterpolation.
constructX	public static void	Matrix X	Prosedur untuk mengisi elemen matriks X sesuai rumus bicubic spline interpolation.

3.8 Class ImageResizing

- Attribute

Class ImageResizing tidak memiliki atribut.

- Method

Nama	Tipe Data	Parameter	Deskripsi
mainImageResizing	public static void	-	Program utama dari class ImageResizing.
calculateRGB	public static double	Matrix multiplier, double x, double	Fungsi untuk mengembalikan nilai RGB pada titik (x,y) dalam gambar yang telah diinterpolasi

		y	menggunakan koefisien hasil perkalian matriks bicubic.
constructD	public static void	Matrix D	Prosedur untuk mengisi elemen matriks D sesuai rumus bicubic spline interpolation.

3.9 Class IO

- Attribute

Nama	Tipe Data	Deskripsi
fileName	String	Nama file.
fileScanner	public Scanner	Scanner untuk membaca file.
inputScanner	public static Scanner	Scanner untuk menerima input dari user.

- Method

Nama	Tipe Data	Parameter	Deskripsi
getFileName	public String	-	Fungsi untuk mengembalikan nama file yang telah ditetapkan saat objek IO dibuat.
openFile	public void	-	Prosedur untuk membuka file.
closeFile	public void	-	Prosedur untuk menutup file.
readFileName	public static String	-	Fungsi untuk meminta pengguna memasukkan nama file dan memvalidasi keberadaan file tersebut dalam direktori yang ditentukan, dan mengembalikan path file yang dimasukkan.
getRowCount	public int	-	Fungsi untuk mengembalikan banyak baris dalam file.
getColCount	public int	-	Fungsi untuk mengembalikan banyak kolom dalam file.
readMatrixFrom Keyboard	public static Matrix	-	Fungsi untuk membaca dan mengembalikan matriks dari masukan keyboard.
readMxNMatrix FromKeyboard	public static Matrix	int row, int col	Fungsi untuk membaca dan mengembalikan matriks m x n

			dari masukan keyboard.
readMatrixFrom File	public Matrix	-	Fungsi untuk membaca dan mengembalikan matriks dari file.
readPointsFrom File	public Matrix	-	Fungsi untuk membaca titik dari file.
readBicubicSplineDataFromKeyboard	public double[]	-	Fungsi untuk membaca dan mengembalikan data bicubic spline dari keyboard.
readBicubicSplineDataFromFile	public double[]	-	Fungsi untuk membaca dan mengembalikan data bicubic spline dari file.
writeStringArrayToFile	public static void	String[] content	Prosedur untuk menulis data array of string ke dalam file
saveFile	public static void	String[] stringArray	Prosedur untuk menyimpan file keluaran.
returnStringArr	public static String[]	String text	Fungsi untuk mengembalikan string dalam bentuk array of string.
matrixToStringArr	public static String[]	Matrix m	Fungsi untuk mengembalikan matriks dalam bentuk array of string.

3.10 Class Main

- Attribute

Nama	Tipe Data	Deskripsi
inputScanner	static Scanner	Scanner untuk input integer.

- Method

Nama	Tipe Data	Parameter	Deskripsi
main	public static void	String[] args	Program utama dari class Main.
loadingUI	public static void	-	Prosedur untuk menampilkan animasi loading.
clearScreen	public static void	-	Prosedur untuk menghapus isi terminal.

inputTypeUI	public static int	-	Fungsi untuk menampilkan menu pilihan tipe masukan dan mengembalikan pilihan masukan berupa integer.
splUI	public static void	-	Prosedur untuk menampilkan menu Sistem Persamaan Linear.
determinantUI	public static void	-	Prosedur untuk menampilkan menu Determinan.
inverseMatrixUI	public static void	-	Prosedur untuk menampilkan menu Matriks Balikan.
polynomialInterpolationUI	public static void	-	Prosedur untuk menampilkan menu Interpolasi Polinomial.
bicubicSplineInterpolationUI	public static void	-	Prosedur untuk menampilkan menu Interpolasi Bicubic Spline.
regressionUI	public static void	-	Prosedur untuk menampilkan menu Regresi.
imageInterpolationUI	public static void	-	Prosedur untuk menampilkan menu Interpolasi Gambar.
headerUI	public static void	-	Prosedur untuk menampilkan header menu utama.
menuUI	public static void	-	Prosedur untuk menampilkan menu utama.

BAB IV

EKSPERIMEN

4.1 Solusi SPL $Ax = b$

Temukan solusi SPL $Ax = b$, berikut

a.

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

Solusi:

```
Tidak ada solusi !
```

b.

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

Solusi:

```
Solusi :  
x1 = 3.0 + 1.0e  
x2 = 0.0 + 2.0e  
x3 = c  
x4 = -1.0 + 1.0e  
x5 = e
```

c.

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

Solusi:

```
Solusi :  
x1 = a  
x2 = 1.0 - 1.0f  
x3 = c  
x4 = -2.0 - 1.0f  
x5 = 1.0 + 1.0f
```

d.

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

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

Solusi:

$n = 6$

Solusi :

x1 = 36.0000

x2 = -630.0000

x3 = 3360.0000

x4 = -7560.0000

x5 = 7560.0000

x6 = -2772.0000

$n = 10$

Solusi :

x1 = 99.9971

x2 = -4949.7486

x3 = 79194.6474

x4 = -600551.3521

x5 = 2522287.9894

x6 = -6305662.2479

x7 = 9608553.6802

x8 = -8750588.8705

x9 = 4375269.1695

x10 = -923663.2626

4.2 SPL Berbentuk Matriks Augmented

a.

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

Solusi:

```
Solusi :  
x1 = -1.0 + 1.0d  
x2 = 0.0 + 2.0c  
x3 = c  
x4 = d
```

b.

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

Solusi:

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

4.3 Sistem Persamaan Linear

a. $8x_1 + x_2 + 3x_3 + 2x_4 = 0$

$$2x_1 + 9x_2 - x_3 - 2x_4 = 1$$

$$x_1 + 3x_2 + 2x_3 - x_4 = 2$$

$$x_1 + 6x_3 + 4x_4 = 3$$

Solusi:

```
Matrix Augmented A dan B :  
8.0000 1.0000 3.0000 2.0000 0.0000  
2.0000 9.0000 -1.0000 -2.0000 1.0000  
1.0000 3.0000 2.0000 -1.0000 2.0000  
1.0000 0.0000 6.0000 4.0000 3.0000
```

```
Solusi :  
x1 = -0.2243  
x2 = 0.1824  
x3 = 0.7095  
x4 = -0.2581
```

b.

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

Solusi:

Matrix Augmented A dan B :

```

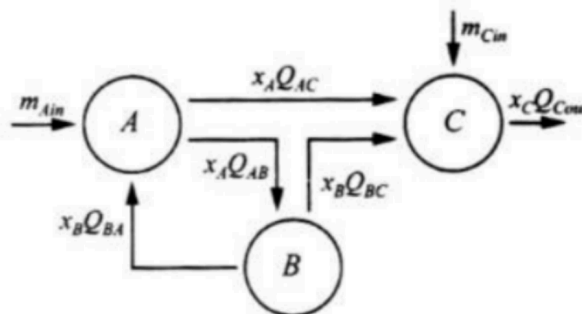
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 1.0000 1.0000 13.0000
0.0000 0.0000 0.0000 1.0000 1.0000 1.0000 0.0000 0.0000 0.0000 15.0000
1.0000 1.0000 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 8.0000
0.0000 0.0000 0.0429 0.0000 0.0429 0.7500 0.0429 0.7500 0.6140 14.7900
0.0000 0.2500 0.9142 0.2500 0.9142 0.2500 0.9142 0.2500 0.0000 14.3100
0.6140 0.7500 0.0429 0.7500 0.0429 0.0000 0.0429 0.0000 0.0000 3.8100
0.0000 0.0000 1.0000 0.0000 0.0000 1.0000 0.0000 0.0000 1.0000 18.0000
0.0000 1.0000 0.0000 0.0000 1.0000 0.0000 0.0000 1.0000 0.0000 12.0000
1.0000 0.0000 0.0000 1.0000 0.0000 0.0000 1.0000 0.0000 0.0000 6.0000
0.0429 0.7500 0.6140 0.0000 0.0429 0.7500 0.0000 0.0000 0.0429 10.5100
0.9142 0.2500 0.0000 0.2500 0.9142 0.2500 0.0000 0.2500 0.9142 16.1300
0.0429 0.0000 0.0000 0.7500 0.0429 0.0000 0.6140 0.7500 0.0429 7.0400

```

Tidak ada solusi !

4.4 Sistem Reaktor

Lihatlah 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:

$$\text{A: } m_{A_{in}} + Q_{BA}x_B - Q_{AB}x_A - Q_{AC}x_A = 0$$

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

$$\text{C: } m_{C_{in}} + Q_{AC}x_A + Q_{BC}x_B - Q_{C_{out}}x_C = 0$$

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 $Q_{C_{out}} = 150 m^3/s$ dan $m_{A_{in}} = 1300$ dan $m_{C_{in}} = 200 mg/s$.

Solusi:

```
Matrix Augmented A dan B :
-120.0000 60.0000 0.0000 -1300.0000
40.0000 -80.0000 0.0000 0.0000
80.0000 20.0000 -150.0000 200.0000

Solusi :
x1 = 14.4444
x2 = 7.2222
x3 = 7.3333
```

4.5 Studi Kasus Interpolasi

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

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

Lakukan pengujian pada nilai-nilai berikut:

$$x = 0.2 \quad f(x) = ?$$

$$x = 0.55 \quad f(x) = ?$$

$$x = 0.85 \quad f(x) = ?$$

$$x = 1.28 \quad f(x) = ?$$

Solusi:

$$f(x) = 0.0000x^6 - 0.0000x^5 + 0.0260x^4 + 0.0000x^3 + 0.1974x^2 + 0.2400x - 0.0230, f(0.2) = 0.0330$$

$$f(x) = 0.0000x^6 - 0.0000x^5 + 0.0260x^4 + 0.0000x^3 + 0.1974x^2 + 0.2400x - 0.0230, f(0.6) = 0.1711$$

$$f(x) = 0.0000x^6 - 0.0000x^5 + 0.0260x^4 + 0.0000x^3 + 0.1974x^2 + 0.2400x - 0.0230, f(0.9) = 0.3372$$

$$f(x) = 0.0000x^6 - 0.0000x^5 + 0.0260x^4 + 0.0000x^3 + 0.1974x^2 + 0.2400x - 0.0230, f(1.3) = 0.6775$$

- 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:

$$\text{Tanggal (desimal)} = \text{bulan} + (\text{tanggal} / \text{jumlah hari pada bulan tersebut})$$

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

$$\text{Tanggal (desimal)} = 6 + (17/30) = 6,567$$

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

(1) 16/07/2022

(2) 10/08/2022

(3) 05/09/2022

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

Solusi:

(1) 16/07/2022

$$f(x) = -141077.3438x^9 + 9378738.6741x^8 - 275658642.6959x^7 + 4699159288.7228x^6 - 51171084726.0325x^5 + 368856078309.8131x^4 - 1758392751973.9858x^3 + 5339470565331.3060x^2 - 9357207562921.2270x + 7195858178915.3400, f(7.5) = 53532.9258$$

(2) 10/08/2022

```
f(x) = -141077.3438x^9 + 9378738.6741x^8 - 275658642.6959x^7 + 4699159288.7228x^6 - 51171084726.0325x^5 + 368856078309.8131x^4 - 1758392751973.9858x^3 + 5339470565331.3060x^2 - 9357207562921.2270x + 7195858178915.3400, f(8.3) = 36305.0156
```

(3) 05/09/2022

```
f(x) = -141077.3438x^9 + 9378738.6741x^8 - 275658642.6959x^7 + 4699159288.7228x^6 - 51171084726.0325x^5 + 368856078309.8131x^4 - 1758392751973.9858x^3 + 5339470565331.3060x^2 - 9357207562921.2270x + 7195858178915.3400, f(9.2) = -664955.0781
```

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

21/06/2022

```
f(x) = -141077.3438x^9 + 9378738.6741x^8 - 275658642.6959x^7 + 4699159288.7228x^6 - 51171084726.0325x^5 + 368856078309.8131x^4 - 1758392751973.9858x^3 + 5339470565331.3060x^2 - 9357207562921.2270x + 7195858178915.3400, f(6.7) = 108669.4487
```

c. Sederhanakan fungsi $f(x)$ yang memenuhi kondisi

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

dengan polinom interpolasi derajat n di dalam selang $[0, 2]$.

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

Solusi:

$n = 2$

```
f(x) = -0.2496x^2 + 0.7874x, f(1.0) = 0.5379
```

$n = 4$

```
f(x) = -0.2008x^4 + 1.0069x^3 - 1.8648x^2 + 1.5966x, f(1.0) = 0.5379
```

4.6 Studi Kasus Regresi Linear dan Kuadratik Berganda

Diberikan sekumpulan data sesuai pada tabel berikut ini.

Table 12.1: Data for Example 12.1

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

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

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

Silahkan terapkan model-model ini pada *Multiple Quadratic Equation* juga dan bandingkan hasilnya.

Solusi :

Regresi linier berganda:

$$f(x) = 0.0001 - 0.0032 X_1 + 0.0007 X_2 + 0.0360 X_3, f(x_k) = 0.9456$$

Regresi kuadratik berganda:

$$f(x) = 0.0000 - 0.2242 X_1 + 1.1327 X_2 - 2.6963 X_3 + 0.0000 X_4 + 0.0000 X_5 + 0.0927 X_6 - 0.0001 X_7 + 0.0076 X_8 - 0.0384 X_9, f(x_k) = 0.9254$$

4.7 Studi Kasus Interpolasi Bicubic Spline

Diberikan matriks input dengan bentuk sebagai berikut. Format matriks masukan bukan mewakili nilai matriks, tetapi mengikuti format masukan pada bagian "Spesifikasi Tugas" nomor 7.

Tentukan nilai:

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

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

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

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

Solusi :

a. $f(0,0)$

$$f(0.0,0.0) = 21.0$$

b. $f(0.5,0.5)$

$$f(0.5,0.5) = 87.796875$$

c. $f(0.25,0.75)$

$$f(0.25,0.75) = 117.732177734375$$

d. $f(0.1,0.9)$

$$f(0.1,0.9) = 128.57518700000003$$

4.8 Bonus : Image Resizing



BAB V

PENUTUP

5.1 Kesimpulan

Dalam kuliah IF2123 Aljabar Linear dan Geometri, kami telah mempelajari berbagai konsep dan metode yang berkaitan dengan matriks. Kemudian, kami mengimplementasi ke dalam program Java untuk menyelesaikan berbagai persoalan, seperti menentukan solusi Sistem Persamaan Linear dengan metode eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer (khusus untuk SPL dengan n peubah dan n persamaan). Program ini juga dapat menentukan determinan matriks dengan reduksi baris dan ekspansi kofaktor, serta menghitung balikan matriks dari sebuah matriks persegi. Melalui pustaka yang telah dibuat, kami dapat menyelesaikan berbagai persoalan yang melibatkan matriks dan sistem persamaan linear, serta aplikasinya dalam kehidupan sehari-hari.

5.2 Saran

Dalam pelaksanaan tugas besar ini, terdapat beberapa saran yang dapat dipertimbangkan untuk pengembangan program di masa depan:

1. Sejak awal, sebaiknya dilakukan kesepakatan terkait penamaan kelas maupun variabel. Penamaan yang konsisten dan jelas akan meningkatkan keterbacaan kode dan memudahkan kolaborasi antar anggota tim.
2. Untuk meningkatkan pengalaman yang menyenangkan bagi pengguna dapat diimplementasikan *Graphical User Interface* (GUI).

5.3 Refleksi

Melalui tugas besar ini, kami menyadari betapa pentingnya koordinasi dan komunikasi yang efektif dalam pembagian tugas dan berbagai aspek lainnya. Kerjasama yang baik antar anggota tim menjadi kunci untuk mencapai tujuan bersama, sehingga setiap orang dapat berkontribusi secara maksimal sesuai dengan keahlian dan tanggung jawab yang telah ditentukan.

Kami juga memahami bahwa kemampuan manajemen waktu yang baik sangat krusial dalam menyelesaikan tugas besar ini, terutama mengingat jadwal akademik yang cukup padat. Dengan mengatur waktu secara efisien, kami dapat menghindari penundaan dalam menyelesaikan pekerjaan dan memastikan bahwa setiap tahap pengerjaan dapat dilakukan dengan optimal. Hal ini memungkinkan kami untuk menjalani proses belajar dengan lebih menyenangkan dan produktif, sehingga kami dapat menikmati setiap langkah dalam mengerjakan tugas besar ini.

5.4 Komentari

- Alfian : “ *Keos*, walau demikian tugas memberikan pemahaman lebih terhadap materi - materi yang telah diajarkan.”
- Jethro : “Cukup menantang, tetapi sangat bermanfaat karena meningkatkan keterampilan dalam mengaplikasikan materi-materi di kelas.”
- Carlo : “Seru dan jadi lebih paham dengan materi yang diajarkan di kelas.”

LAMPIRAN

1. Referensi

Slide Kuliah IF2123 2024/2025 :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2024-2025/algeo24-25.htm>

Slide BiLinear, Bicubic, and In Between Spline Interpolation Marquette University :

https://www.mssc.mu.edu/~daniel/pubs/RoweTalkMSCS_BiCubic.pdf

2. Tautan Repository

Berikut tautan repository github kelompok kami untuk Tugas Besar 1 IF2123 Aljabar Linear dan Geometri :

<https://github.com/carllix/Algeo01-23073>

3. Tautan Video

Berikut tautan video kelompok kami untuk Tugas Besar 1 IF2123 Aljabar Linear dan Geometri :

<https://youtu.be/VqY1v512FcU?si=2kaEoQtW-L71gOos>