

Laporan Tugas Besar 1
IF2123 Aljabar Linier dan Geometri
Sistem Persamaan Linear, Determinan, dan Aplikasinya



Disusun oleh:
Maximilian Sulistiyo (13522061)
Marvel Pangondian (13522075)
Steven Tjhia (13522103)

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2023

Daftar Isi

Daftar Isi.....	2
Bab I Deskripsi Masalah.....	4
1.1 Abstraksi.....	4
1.2 Interpolasi Polinomial.....	5
1.3 Regresi Linier Berganda.....	6
1.4 Bicubic Spline Interpolation.....	6
Bab II Landasan Teori.....	10
2.1 Metode Eliminasi Gauss.....	10
2.2 Metode Eliminasi Gauss-Jordan.....	11
2.3 Determinan.....	11
2.4 Minor, Kofaktor, Matriks Kofaktor, dan Matriks Adjoin.....	12
2.5 Matriks Balikan.....	13
2.6 Kaidah Cramer.....	13
2.7 Interpolasi Polinom.....	13
2.8 Interpolasi Bicubic Spline.....	14
2.9 Regresi Linier Berganda.....	16
Bab III Implementasi Program.....	17
3.1 Class Matriks.....	17
3.1.1 Attribute.....	17
3.1.2 Method – Konstruktor.....	17
3.1.3 Method – Selektor.....	17
3.1.4 Method – Fungsi dan Prosedur.....	18
3.2 Class SPL.....	20
3.2.1 Attribute.....	20
3.2.2 Method.....	20
3.3 Class DisplaySolution.....	21
3.3.1 Attribute.....	21
3.3.2 Method.....	21
3.4 Class Polinom.....	21
3.4.1 Attribute.....	21
3.4.2 Method.....	22
3.5 Class Regresi.....	22
3.5.1 Attribute.....	22
3.5.2 Method.....	22
3.6 Class BicubicSpline.....	23
3.6.1 Attribute.....	23
3.6.2 Method.....	23
3.7 Class FileInputOutput.....	23
3.7.1 Attribute.....	23
3.7.2 Method.....	23
3.8 Class MenuIO.....	25
3.8.1 Attribute.....	25

3.8.2 Method.....	25
3.9 Program Utama – Class Main.....	26
3.9.1 Menu Utama.....	27
3.9.2 Menu SPL.....	27
3.9.2.1 Metode Elimintasi Gauss.....	27
3.9.2.2 Metode Eliminasi Gauss-Jordan.....	28
3.9.2.3 Metode Matriks Balikan.....	29
3.9.2.4 Kaidah Cramer.....	31
3.9.3 Menu Determinan.....	32
3.9.3.1 Metode Reduksi Baris.....	32
3.9.3.2 Metode Ekspansi Kofaktor.....	33
3.9.4 Menu Matriks Balikan.....	34
3.9.4.1 Metode Gauss Jordan.....	35
3.9.4.2 Metode Adjoint.....	36
3.9.5 Menu Interpolasi Polinom.....	37
3.9.6 Menu Interpolasi Bicubic Spline.....	38
3.9.7 Menu Regresi.....	40
Bab IV Eksperimen.....	42
Bab V Kesimpulan.....	69
5.1 Kesimpulan.....	69
5.2 Saran.....	69
5.3 Komentar.....	69
5.4 Refleksi.....	69
Daftar Referensi.....	70

Bab I

Deskripsi Masalah

1.1 Abstraksi

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).

$$\left[\begin{array}{cccc} 0 & 2 & 1 & -1 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 \end{array} \right] \cdot \left[\begin{array}{cccc} 0 & 1 & 0 & -\frac{2}{3} \\ 0 & 0 & 1 & \frac{1}{3} \\ 0 & 0 & 0 & 0 \end{array} \right].$$

Gambar 1. Eliminasi Gauss dilakukan dengan matriks eselon baris dan eliminasi Gauss-Jordan dengan matriks eselon baris tereduksi.

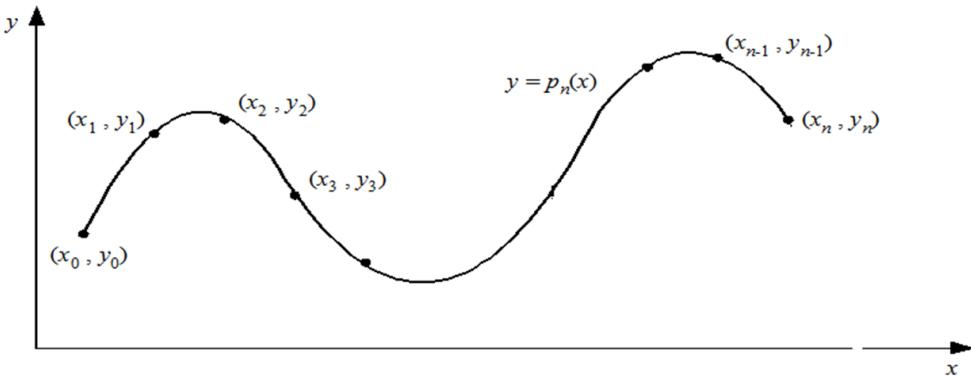
Di dalam Tugas Besar 1 ini, Anda diminta membuat satu atau lebih *library* aljabar linier dalam Bahasa Java. Library tersebut berisi fungsi-fungsi seperti eliminasi Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung determinan, kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n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

Beberapa tulisan cara membuat library di Java:

1. <https://www.programcreek.com/2011/07/build-a-java-library-for-yourself/>
2. <https://developer.ibm.com/tutorials/j-javalibrary/>
3. [https://stackoverflow.com/questions/3612567/how-to-create-my-own-java-library api](https://stackoverflow.com/questions/3612567/how-to-create-my-own-java-library-api)

1.2 Interpolasi Polinomial

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



Gambar 2. Ilustrasi beberapa titik yang diinterpolasi secara polinomial.

Setelah polinom interpolasi $p_n(x)$ ditemukan, $p_n(x)$ dapat digunakan untuk menghitung perkiraan nilai y di sembarang titik di dalam selang $[x_0, x_n]$.

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 lanjar dalam $a_0, a_1, a_2, \dots, a_n$,

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

Solusi sistem persamaan lanjar 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 kuadratik lalu estimasi nilai fungsi pada $x = 9.2$. Polinom kuadratik berbentuk $p_2(x) = a_0 + a_1x + a_2x^2$. Dengan menyulihkan ketiga buah titik data ke dalam polinom tersebut, diperoleh sistem persamaan lanjar yang terbentuk adalah

$$a_0 + 8.0a_1 + 64.00a_2 = 2.0794$$

$$a_0 + 9.0a_1 + 81.00a_2 = 2.1972$$

$$a_0 + 9.5a_1 + 90.25a_2 = 2.2513$$

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: $p_2(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(9.2)^2 = 2.2192$.

1.3 Regresi Linier Berganda

Regresi Linear (akan dipelajari lebih lanjut di Probabilitas dan Statistika) merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Meskipun sudah ada persamaan jadi untuk menghitung regresi linear sederhana, terdapat persamaan umum dari regresi linear yang bisa digunakan untuk regresi linear berganda, yaitu.

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

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

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

Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

1.4 Bicubic Spline Interpolation

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.

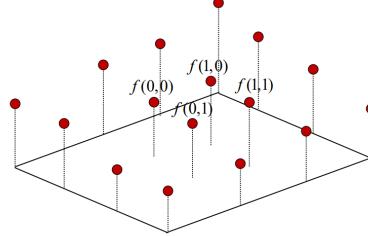
Dalam pemrosesan menggunakan interpolasi *bicubic spline* digunakan 16 buah titik, 4 titik referensi utama di bagian pusat, dan 12 titik di sekitarnya sebagai aproksimasi turunan dari keempat titik referensi untuk membagun permukaan bikubik. Bentuk pemodelannya adalah sebagai berikut.

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}



Gambar 3. Pemodelan interpolasi *bicubic spline*.

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.

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

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

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

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

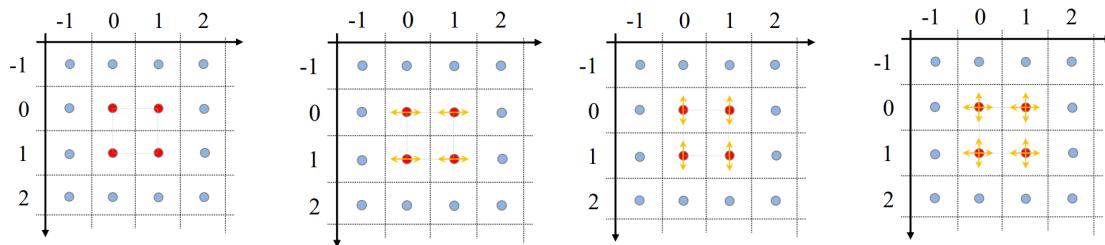
Dengan menggunakan nilai fungsi dan turunan berarah tersebut, dapat terbentuk sebuah matriks solusi X yang membentuk persamaan penyelesaian sebagai berikut.

$$y = Xa$$

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

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 $1 \times 1^{1-1} \times 1^0 = 1$, 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.



Gambar 4. Nilai fungsi yang akan di interpolasi pada titik merah, turunan berarah terhadap sumbu x , terhadap sumbu y , dan keduanya (kiri ke kanan).

Untuk studi kasus ini, buatlah matriks X menggunakan persamaan yang ada (tidak *hardcode*) serta carilah invers matriks X dengan *library* yang telah kalian buat dalam penyelesaian masalah. Berikut adalah [sebuah tautan](#) yang dapat dijadikan referensi.

Seperti yang telah dijelaskan sebelumnya bahwa interpolasi *bicubic spline* dapat digunakan untuk menciptakan permukaan yang halus pada gambar. Oleh karena itu, selain persamaan dasar $y = Xa$ yang telah dijabarkan, persamaan ini juga dapat menggunakan data sebuah citra untuk menciptakan kualitas gambar yang lebih baik. Misalkan $I(x, y)$ merupakan nilai dari suatu citra gambar pada posisi (x, y) , maka dapat digunakan persamaan nilai dan persamaan turunan berarah sebagai berikut.

$$f(x, y) = I(x, y)$$

$$f_x(x, y) = [I(x+1, y) - I(x-1, y)] / 2$$

$$f_y(x, y) = [I(x, y+1) - I(x, y-1)] / 2$$

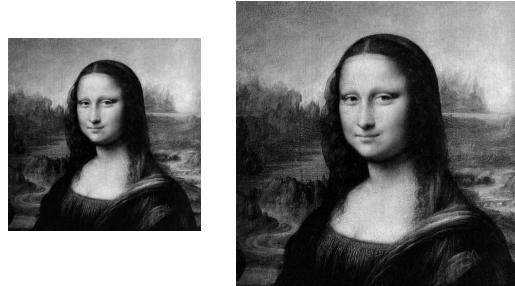
$$f_{xy}(x, y) = [I(x+1, y+1) - I(x-1, y) - I(x, y-1) - I(x, y)] / 4$$

Sistem persamaan tersebut dapat dipetakan menjadi sebuah matriks (dalam hal ini matriks D) dengan gambaran lengkap seperti yang tertera di bawah.

$$y = DI$$

$$\begin{bmatrix} f(0,0) \\ f(1,0) \\ f(0,1) \\ f(1,1) \\ f_x(0,0) \\ f_x(1,0) \\ f_x(0,1) \\ f_x(1,1) \\ f_y(0,0) \\ f_y(1,0) \\ f_y(0,1) \\ f_y(1,1) \\ f_{xy}(0,0) \\ f_{xy}(1,0) \\ f_{xy}(0,1) \\ f_{xy}(1,1) \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & 0 & 2 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I(-1,-1) \\ I(0,-1) \\ I(1,-1) \\ I(2,-1) \\ I(-1,0) \\ I(0,0) \\ I(1,0) \\ I(2,0) \\ I(-1,1) \\ I(0,1) \\ I(1,1) \\ I(2,1) \\ I(-1,2) \\ I(0,2) \\ I(1,2) \\ I(2,2) \end{bmatrix}$$

Dengan menggunakan kedua persamaan nilai y yang telah disebutkan dan dibahas sebelumnya, dapatkan nilai a yang lebih baik dan akurat dalam pemrosesan citra gambar, kemudian gunakan nilai dan persamaan $f(x, y)$ yang terbentuk untuk memperbaiki kualitas citra gambar monokrom pasca perbesaran dengan skala tertentu dengan melakukan interpolasi *bicubic spline*. Berikut adalah contohnya.



Gambar 5. Sebuah citra gambar asal (kiri) dan hasil perbesarannya dengan skala 1.5 (kanan).

Untuk bonus ini, buatlah matriks D menggunakan persamaan citra gambar yang ada (tidak *hardcode*) serta gunakan kembali persamaan y yang sebelumnya ($y = Xa$) dan korelasikan dengan persamaan $y = DI$ untuk mendapatkan nilai a yang lebih tepat untuk membangun persamaan $f(x, y)$. Tambahkan pula masukan berupa skala perbesaran gambar sesuai keinginan pengguna.

Bab II

Landasan Teori

2.1 Metode Eliminasi Gauss

Metode Eliminasi Gauss dikembangkan oleh Carl Friedrich Gauss. Ia merupakan matematikawan berkebangsaan Jerman yang mendalami bidang geometri, teori bilangan, fungsi, dan teori probabilitas. Metode eliminasi Gauss berasal dari operasi matematika pada baris matriks yang dilanjutkan sampai tersisa satu variabel.

Langkah-langkah melakukan metode eliminasi Gauss:

1. Persamaan linear diubah menjadi bentuk matriks yang teraugmentasi
2. Selanjutnya diubah ke bentuk Eselon Baris dengan Operasi Baris Elementer
3. Setelah menjadi matriks eselon baris, variabel-variabel bisa didapatkan dengan cara mensubstitusikannya.

$$\left[\begin{array}{cccccc} 1 & 2 & -5 & 3 & 6 & 14 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 2 \end{array} \right]$$

Matriks Eselon Baris

Pada penerapannya, proses eliminasi pada metode ini terdiri dari tiga operasi baris elementer:

- Pertukaran : posisi dua persamaan dapat ditukar karena tidak akan memengaruhi solusi akhir.
- Penskalaan : persamaan dapat dikalikan dengan konstanta bukan nol karena tidak akan memengaruhi solusi akhir.
- Penambahan : persamaan suatu baris ditambahkan dengan kelipatan tertentu baris lainnya.

Beberapa ciri-ciri metode eliminasi Gauss:

- Jika suatu baris tidak nol semua, maka bilangan pertama yang tidak nol adalah 1 (1 utama)
- Baris nol terletak paling bawah
- Dibawah 1 utama harus nol
- 1 utama berikutnya berada di kanan 1 utama baris diatasnya

2.2 Metode Eliminasi Gauss-Jordan

Metode eliminasi Gauss-Jordan merupakan pengembangan dari eliminasi Gauss yang lebih tersederhanakan lagi. Metode ini dimodifikasi oleh Wilhelm Jordan seorang insinyur Jerman pada tahun 1887.

Perbedaan metode eliminasi Gauss-Jordan dengan metode eliminasi Gauss adalah pada metode eliminasi Gauss-Jordan, kita melakukan operasi baris elementer pada matriks yang teraugmentasi sampai didapatkan eselon baris yang tereduksi, yaitu seluruh elemen matriks pada kolom yang terdapat satu utama (selain satu utamanya) harus bernilai nol. Dengan metode ini, selain dapat digunakan untuk menyelesaikan sistem persamaan linear juga dapat digunakan untuk mencari invers dari sebuah matriks.

$$\left[\begin{array}{cccccc} 1 & 2 & 0 & 3 & 0 & 7 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 2 \end{array} \right]$$

Matriks Eselon Baris Tereduksi

2.3 Determinan

Determinan merupakan sebuah nilai yang bisa dilakukan perhitungan berdasarkan bagian-bagian yang terdapat pada matriks persegi. Jadi, kita tidak dapat menghitung determinan pada matriks yang bukan matriks persegi. Determinan biasanya dinotasikan dengan $\det(A)$ dengan A adalah suatu matriks. Terdapat beberapa macam cara untuk mendapatkan nilai determinan, yaitu dengan Sarrus (hanya untuk matriks 3x3), ekspansi kofaktor, kaidah *Cramer*.

Teorema tentang determinan:

- Jika A mengandung sebuah baris nol atau kolom nol, maka $\det(A) = 0$
- Jika A^T adalah matriks transpose dari A , maka $\det(A^T) = \det(A)$
- Jika $A = BC$ maka $\det(A) = \det(B)\det(C)$
- Sebuah matriks hanya mempunyai balikan jika dan hanya jika $\det(A) \neq 0$
- $\det(A^{-1}) = \frac{1}{\det(A)}$

$$|A| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

Determinan Matriks 2x2

Beberapa aturan Determinan dengan B merupakan matriks hasil operasi terhadap matriks A:

Operasi terhadap matriks A	$\det(B)$
Mengalikan sebuah baris dengan k	$k \cdot \det(A)$
Menukar dua baris	$-\det(A)$
Sebuah baris ditambahkan dengan k kali baris yang lain	$\det(A)$

2.4 Minor, Kofaktor, Matriks Kofaktor, dan Matriks Adjoin

Minor suatu matriks A, dilambangkan dengan M_{ij} , merupakan determinan matriks bagian dari matriks A yang diperoleh dengan menghilangkan elemen-elemen pada baris ke- i dan kolom ke- j . Jumlah minor suatu matriks sama seperti jumlah elemen matriks tersebut.

$$A = \begin{bmatrix} -1 & 3 \\ 4 & -5 \end{bmatrix} \quad \begin{aligned} M_{11} &= -5 \\ M_{12} &= 4 \\ M_{21} &= 3 \\ M_{22} &= -1 \end{aligned}$$

Kofaktor adalah hasil perkalian minor dengan suatu angka yang besarnya menurut suatu aturan. Sama seperti minor, jumlah kofaktor suatu matriks mengikuti jumlah elemen matriks tersebut. Kofaktor suatu elemen baris ke- i dan kolom ke- j dari matriks A dilambangkan dengan C_{ij} .

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

$$A = \begin{bmatrix} -1 & 3 \\ 4 & -5 \end{bmatrix} \quad \begin{aligned} M_{11} &= -5 & C_{11} &= (-1)^{1+1}(-5) = -5 \\ M_{12} &= 4 & C_{12} &= (-1)^{1+2}(4) = -4 \\ M_{21} &= 3 & C_{21} &= (-1)^{2+1}(3) = -3 \\ M_{22} &= -1 & C_{22} &= (-1)^{2+2}(-1) = -1 \end{aligned}$$

Matriks kofaktor merupakan matriks yang terdiri dari kofaktor-kofaktor matriks itu sendiri. Susunan elemen matriks kofaktor mengikuti susunan kofaktornya.

$$\text{Matriks Kofaktor } A = \begin{bmatrix} -5 & -4 \\ -3 & -1 \end{bmatrix}$$

Matriks adjoint merupakan transpose dari matriks kofaktor. Biasanya dinotasikan dengan $\text{Adj}(A)$ dengan A adalah sebuah matriks, sedangkan transpose adalah pertukaran elemen pada baris menjadi kolom dan kolom menjadi baris. Matriks adjoint biasanya digunakan untuk menentukan invers matriks.

$$\text{Adj}(A) = \begin{bmatrix} -5 & -3 \\ -4 & -1 \end{bmatrix}$$

2.5 Matriks Balikan

Terdapat beberapa cara untuk mendapatkan matriks balikan, seperti menggunakan eliminasi Gauss-Jordan dan menggunakan matriks adjoint. Menghitung matriks balikan dengan eliminasi Gauss-Jordan menggunakan konsep dasar $AA^{-1} = A^{-1}A = I$.

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

Jika kita ingin mencari matriks balikan dengan menggunakan matriks adjoint, maka kita dapat menggunakan rumus berikut:

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

2.6 Kaidah Cramer

Metode *Cramer* ditemukan oleh Gabriel Cramer seorang matematikawan Swiss. Metode *Cramer* merupakan metode untuk menyelesaikan sistem persamaan dengan menggunakan determinan. Jika persamaan linear yang terdiri dari n persamaan dan n variabel yang tidak diketahui dan dinyatakan dalam $Ax = b$ dan $\det(A) \neq 0$, maka persamaan tersebut mempunyai penyelesaian:

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

2.7 Interpolasi Polinom

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 lanjar dalam $a_0, a_1, a_2, \dots, a_n$,

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

$$a_0 + a_1 x_n + a_2 x_n^2 + \dots + a_n x_n^n = y_n$$

Solusi sistem persamaan lanjar 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 kuadratik lalu estimasi nilai fungsi pada $x = 9.2$. Polinom kuadratik berbentuk $p_2(x) = a_0 + a_1 x + a_2 x^2$. Dengan menyulihkan ketiga buah titik data ke dalam polinom tersebut, diperoleh sistem persamaan lanjar yang terbentuk adalah

$$a_0 + 8.0a_1 + 64.00a_2 = 2.0794$$

$$a_0 + 9.0a_1 + 81.00a_2 = 2.1972$$

$$a_0 + 9.5a_1 + 90.25a_2 = 2.2513$$

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: $p_2(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(9.2)^2 = 2.2192$.

2.8 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.

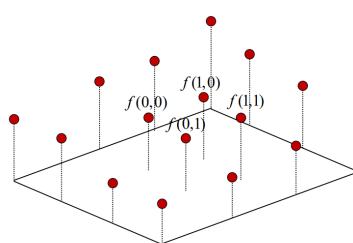
Dalam pemrosesan menggunakan interpolasi *bicubic spline* digunakan 16 buah titik, 4 titik referensi utama di bagian pusat, dan 12 titik di sekitarnya sebagai aproksimasi turunan dari keempat titik referensi untuk membagun permukaan bikubik. Bentuk pemodelannya adalah sebagai berikut.

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}



Gambar 3. Pemodelan interpolasi *bicubic spline*.

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.

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

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

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

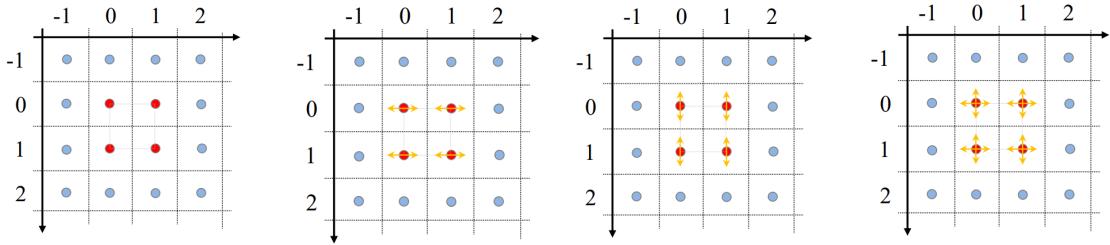
$$f_{xy}(x, y) = \sum_{j=0}^3 \sum_{i=0}^3 a_{ij} i j x^{i-1} y^{j-1}$$

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

$$\begin{bmatrix} f(0,0) \\ f(1,0) \\ f(0,1) \\ f(1,1) \\ f_x(0,0) \\ f_x(1,0) \\ f_x(0,1) \\ f_x(1,1) \\ f_y(0,0) \\ f_y(1,0) \\ f_y(0,1) \\ f_y(1,1) \\ f_{xy}(0,0) \\ f_{xy}(1,0) \\ f_{xy}(0,1) \\ f_{xy}(1,1) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 2 & 3 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 2 & 4 & 6 & 0 & 3 & 6 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30} \\ a_{01} \\ a_{11} \\ a_{21} \\ a_{31} \\ a_{02} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{03} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}$$

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 $1 \times 1^{1-1} \times 1^0 = 1$, 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.



Gambar 4. Nilai fungsi yang akan di interpolasi pada titik merah, turunan berarah terhadap sumbu x , terhadap sumbu y , dan keduanya (kiri ke kanan).

2.9 Regresi Linier Berganda

Regresi Linear (akan dipelajari lebih lanjut di Probabilitas dan Statistika) merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Meskipun sudah ada persamaan jadi untuk menghitung regresi linear sederhana, terdapat persamaan umum dari regresi linear yang bisa digunakan untuk regresi linear berganda, yaitu.

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

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

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

Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

Bab III

Implementasi Program

3.1 Class Matriks

3.1.1 Attribute

<code>int row, col;</code>	Menyimpan jumlah baris (row) dan jumlah kolom (col) suatu Matriks.
<code>double[][] content;</code>	Array of array of double untuk menyimpan elemen-elemen Matriks.
<code>int num_switch = 0;</code>	Informasi untuk menghitung determinan: jumlah operasi tukar baris yang dilakukan pada matriks.
<code>double K = 1;</code>	Informasi untuk menghitung determinan: konstanta perkalian untuk menentukan determinan apabila terdapat baris yang dikalikan suatu konstanta.
<code>boolean solvable = true;</code>	Informasi untuk menghitung determinan: nilai boolean yang menunjukkan apakah suatu Matriks SPL memiliki solusi penyelesaian.

3.1.2 Method – Konstruktor

<code>public Matrix(int r, int c)</code>	Konstruktor untuk membuat Matriks, belum termasuk nilai elemen pada Matriks.
<code>public void readMatrix()</code>	Prosedur untuk membaca nilai-nilai elemen pada Matriks.

3.1.3 Method – Selektor

<code>public int getRow()</code>	Fungsi yang mengembalikan jumlah baris suatu Matriks.
<code>public int getCol()</code>	Fungsi yang mengembalikan jumlah kolom suatu Matriks.
<code>public double getElmt(int r, int c)</code>	Fungsi yang mengembalikan nilai elemen pada baris ke r dan kolom ke c suatu Matriks.

<code>public int getLastIdxRow()</code>	Fungsi yang mengembalikan indeks terakhir baris suatu Matriks.
<code>public int getLastIdxCol()</code>	Fungsi yang mengembalikan indeks terakhir kolom suatu Matriks
<code>public void setElmt(int r, int c, double x)</code>	Prosedur untuk mengubah nilai elemen pada baris ke r dan kolom ke c suatu Matriks menjadi nilai x .

3.1.4 Method – Fungsi dan Prosedur

<code>public Matrix copyMatrix()</code>	Fungsi yang menghasilkan sebuah Matriks baru yang elemennya sama seperti Matriks awal.
<code>public Matrix copyMatrixCustom(int sRow, int sCol, int fRow, int fCol)</code>	Seperti fungsi copyMatrix, tetapi ini bisa memilih dimulai dari baris dan kolom tertentu ke suatu baris dan kolom tertentu.
<code>public void displayMatrix()</code>	Prosedur untuk menampilkan matriks.
<code>public void switchRow(int r1, int r2)</code>	Prosedur yang menukar elemen pada baris ke $r1$ dengan elemen pada baris ke $r2$ suatu Matriks.
<code>public static boolean isPersegi(Matrix mIn)</code>	Fungsi yang mengembalikan nilai True jika matriks mIn adalah matriks persegi.
<code>public int leadingOneRowIdx(int r)</code>	Fungsi yang mengembalikan indeks <i>leading one</i> baris ke r suatu Matriks.
<code>public static Matrix matMultiplyByConst(Matrix mIn, double k)</code>	Mengalikan setiap elemen pada matriks mIn dengan konstanta k .
<code>public static double determinanReduksi(Matrix mat)</code>	(Pre-kondisi : Matriks mat adalah matriks persegi). Fungsi yang mengembalikan hasil dari determinan yang didapatkan dengan Matriks Gauss.
<code>public static double determinanEkspansiKofaktor(Matrix x mat)</code>	(Pre-kondisi : Matriks mat adalah matriks persegi). Fungsi yang mengembalikan hasil dari determinan yang didapatkan dengan metode Ekspansi Kofaktor.

<pre>public static Matrix getMatrixConstant(Matrix mIn)</pre>	Fungsi yang mengeluarkan Matriks yang berisi elemen pada kolom terakhir Matriks mIn. (mIn adalah Matriks Augmented).
<pre>public static Matrix getMatrixCoefficient(Matrix mIn)</pre>	Fungsi yang mengeluarkan Matriks yang berisi berisi semua elemen Matriks mIn tetapi tanpa kolom terakhir. (mIn adalah Matriks Augmented).
<pre>public static Matrix subMatrix(Matrix mIn, int r, int c)</pre>	Fungsi yang menghasilkan Matriks Minor M_{rc} dari Matriks mIn.
<pre>public static Matrix changeColumn(Matrix mat, Matrix mat_col, int c)</pre>	(Pre-kondisi : Matriks mat_col adalah matriks yang hanya memiliki 1 kolom dan ukuran baris Matriks mat sama dengan ukuran baris m_col). Fungsi yang mengembalikan Matriks yang elemen pada kolom c diubah menjadi seperti elemen dari Matriks mat_col.
<pre>public static Matrix multiplyMatrix(Matrix m1, Matrix m2)</pre>	(Pre-kondisi : m1.row = m2.row dan m1.col = m2.col). Fungsi yang mengembalikan Matriks hasil perkalian Matriks m1 dengan m2.
<pre>public void identityMatrix()</pre>	(Pre-kondisi : Matriks yang akan dijadikan Matriks identitas adalah Matriks persegi). Prosedur yang mengubah suatu Matriks menjadi Matriks Identitas.
<pre>public static Matrix konkatMatrix(Matrix m1, Matrix m2)</pre>	(Pre-kondisi : m1.row = m2.row). Fungsi yang menghasilkan Matriks yang merupakan gabungan dari Matriks m1 dan m2 dengan m2 berada di sebelah kanan m1.
<pre>public static Matrix inversGaussJordan(Matrix mIn)</pre>	(Pre-kondisi : Matriks mIn adalah Matriks Persegi). Fungsi yang mengembalikan Matriks balikan dari Matriks mIn dengan metode Gauss-Jordan.
<pre>public static Matrix getMatrixCofactor(Matrix mIn)</pre>	Fungsi yang menghasilkan Matriks Kofaktor dari Matriks mIn.
<pre>public static Matrix getMatrixTranspose(Matrix mIn)</pre>	Fungsi yang menghasilkan Matriks Transpose dari Matriks mIn.

<code>public void turnToMark()</code>	Prosedur mengubah setiap elemen dari matriks menjadi mark .
<code>public static Matrix getMatrixInverseAdjoin(Matrix mIn)</code>	Fungsi yang menghasilkan Matriks balikan dari Matriks mIn dengan metode matriks balikan biasa dengan matriks adjoint.
<code>public void dfMat()</code>	Prosedur mengecek setiap elemen dari matriks, jika terdapat -0.0, maka elemen tersebut diubah menjadi 0.0.
<code>public void dfMat2()</code>	Prosedur membulatkan setiap elemen dari matriks menjadi 4 angka dibelakang desimal saja.
<code>public static boolean isMarkMat(Matrix mIn)</code>	Fungsi yang mengecek setiap elemen dari matriks, jika semua elemen matriks adalah mark, maka akan mengembalikan True.

3.2 Class SPL

3.2.1 Attribute

Tidak ada *attribute* khusus pada Class SPL.

3.2.2 Method

<code>public static Matrix gauss(Matrix matIn)</code>	Fungsi yang menghasilkan Matriks eselon baris dari Matriks matIn.
<code>public static Matrix cramerMethod(Matrix mIn)</code>	(Pre-kondisi : Matriks mIn adalah matriks <i>augmented</i> berukuran $n \times (n+1)$). Fungsi yang menghasilkan Matriks yang berisi penyelesaian dari SPL.
<code>public static Matrix gaussjordan(Matrix mIn)</code>	Fungsi yang mengembalikan Matriks eselon baris tereduksi dari Matriks mIn.
<code>public static Matrix splInverse(Matrix mIn)</code>	(Pre-kondisi : Matriks mIn adalah Matriks augmented). Fungsi yang menghasilkan penyelesaian SPL dengan metode Matriks Balikan dalam bentuk Matriks.

3.3 Class DisplaySolution

3.3.1 Attribute

Tidak ada *attribute* khusus pada Class DisplaySolution.

3.3.2 Method

<code>public static void displayGauss(Matrix mGauss)</code>	Prosedur untuk menampilkan hasil penyelesaian SPL dengan metode Gauss.
<code>public static void displayGaussJordan(Matrix mGaussJordan)</code>	Prosedur untuk menampilkan hasil penyelesaian SPL dengan metode Gauss-Jordan.
<code>public static void displayCramer(Matrix mat_Cramer)</code>	Prosedur untuk menampilkan hasil penyelesaian SPL dengan metode Cramer.
<code>public static void displaySPLbalikan(Matrix mat_SPL_balikan)</code>	Prosedur untuk menampilkan penyelesaian SPL dengan metode Matriks Balikan.
<code>public static void displayDeterminan(double det)</code>	Prosedur ini digunakan untuk menampilkan hasil determinan
<code>public static void displayMatriksInverse(Matrix matInv)</code>	Prosedur ini digunakan untuk menampilkan hasil matriks balikan
<code>public static boolean allZero(double[] arr)</code>	Prosedur ini digunakan untuk mengecek apakah terdapat baris 0 pada matriks
<code>public static double dformat(double num)</code>	Fungsi yang mengecek apakah nilai merupakan -0.0, jika iya, maka akan mengembalikan 0.0 (menangani kasus -0.0)
<code>public static double dformat2(double num)</code>	Fungsi yang menghasilkan nilai pembulatan <i>num</i> menjadi empat angka di belakang koma saja.

3.4 Class Polinom

3.4.1 Attribute

Tidak ada *attribute* khusus pada Class Polinom.

3.4.2 Method

<pre>public static Matrix createLinearEq(Matrix mPoint)</pre>	Fungsi ini menghasilkan sebuah matriks yang isinya merupakan linear equation yang dihasilkan dari point dari matriks point yang di input
<pre>public static Matrix getPolinomSol(Matrix mLinEq)</pre>	Fungsi ini menerima matriks linear equation dan mendapatkan solusi dari persamaan tersebut dalam bentuk matriks
<pre>public static String getPolinomEq(Matrix solMatrix)</pre>	Fungsi ini menghasilkan string persamaan polinom yang didapatkan dari solusi linear equation sebelumnya
<pre>public static double getTaksiran(Matrix solMatrix, double x)</pre>	Fungsi ini menerima x dan matriks solusi dan mengembalikan hasil taksiran dari nilai x pada persamaan polinom dari matriks solusi tersebut

3.5 Class Regresi

3.5.1 Attribute

Tidak ada *attribute* khusus pada Class Regresi.

3.5.2 Method

<pre>private static Matrix regresiBerganda(Matrix mIn)</pre>	(Pre-kondisi : matriks mIn merupakan matriks <i>augmented</i>). Fungsi ini menghasilkan matriks <i>Normal Estimation Equation for Multiple Linear Regression</i> .
<pre>public static Matrix solveRegresi(Matrix mIn)</pre>	Matriks mIn merupakan matriks <i>Normal Estimation Equation for Multiple Linear Regression</i> . Fungsi ini menghasilkan matriks hasil penyelesaian SPL dari matriks mIn.
<pre>public static String getRegresiEq(Matrix m_solved)</pre>	Fungsi ini menerima matriks solved matrix hasil solveRegresi dan menghasilkan string persamaan dari matriks tersebut
<pre>private static double getTaksiranRegresi(Matrix solMatrix, double[]arrVar)</pre>	Fungsi ini menerima matriks hasil penyelesaian SPL dan array yang berisi nilai-nilai variabel. Fungsi ini menghasilkan taksiran dari nilai-nilai variabel pada persamaan regresi

	tersebut.
<code>public static String getStrTaksiran(Matrix mSolve, double[]arrVar)</code>	Mengembalikan hasil taksiran dalam bentuk String.

3.6 Class BicubicSpline

3.6.1 Attribute

Tidak ada *attribute* khusus pada Class BicubicSpline.

3.6.2 Method

<code>private static Matrix toColumnMatrix(Matrix mIn)</code>	Fungsi yang menghasilkan matriks yang dijadikan hanya satu kolom saja dari matriks mIn..
<code>public static Matrix bicubicX()</code>	Fungsi yang menghasilkan matriks X untuk perhitungan <i>Bicubic Spline Interpolation</i> .
<code>public static Matrix solveBicubic(Matrix mIn)</code>	Fungsi yang menghasilkan konstanta hasil penyelesaian SPL <i>Bicubic Spline Interpolation</i> .
<code>public static double getTaksiranBicubic(Matrix solMatrix, double x, double y)</code>	Fungsi yang menghasilkan taksiran dari fungsi <i>Bicubic Spline Interpolation</i> pada titik (x,y).

3.7 Class FileInputOutput

3.7.1 Attribute

Tidak ada *attribute* khusus pada Class FileInputOutput.

3.7.2 Method

<code>public static Matrix readFileMatrix()</code>	Fungsi untuk membaca file .txt dan mengembalikan matriks pada file tersebut.
<code>public static int getRowCountFile(String path)</code>	Fungsi untuk menghitung berapa banyak baris pada sebuah matriks dalam file, mengembalikan banyaknya baris tersebut.
<code>public static int getColCountFile(String path)</code>	Fungsi untuk menghitung berapa banyak kolom pada sebuah matriks dalam file,

	mengembalikan banyaknya kolom tersebut
<code>public static Matrix getMatrixFromFile(String path)</code>	Menerima path file yang user input melalui fungsi readFileMatrix(), dan menggunakan path tersebut untuk membuat matrix sesuai dengan file .txt. Fungsi ini akan mengembalikan matriks dimana matriks tersebut akan diteruskan dan dikembalikan oleh readFileMatrix()
<code>public static void opsiSaveFile(String[] arr)</code>	Merupakan prosedur yang menerima array of string (hasil parametric matrix augmented) dan memberi opsi kepada user untuk save hasil tersebut ke dalam file .txt
<code>public static void opsiSaveFilePolinom(String[] arr)</code>	Merupakan prosedur yang menerima array of string (hasil polinom) dan memberi opsi kepada user untuk save hasil tersebut ke dalam file .txt
<code>public static void opsiSaveFile(double k)</code>	Merupakan prosedur yang menerima double number (determinan sebuah matrix) dan memberi opsi kepada user untuk menyimpan determinan tersebut ke dalam sebuah file .txt
<code>public static void opsiSaveFile(Matrix mInv)</code>	Merupakan prosedur yang menerima sebuah Matriks (hasil matriks inverse) dan memberi opsi kepada user untuk menyimpan hasil matriks tersebut ke dalam sebuah .txt file
<code>public static void writeSPL(String[] arr)</code>	Prosedur ini menulis hasil parametric matrix augmented yang diterima dari prosedur opsiSaveFile(String[] arr) ke dalam sebuah file .txt
<code>public static void writePolinom(String[] arr)</code>	Prosedur ini menulis hasil polinom yang diterima oleh opsiSaveFilePolinom(String[] arr) ke dalam sebuah file .txt
<code>public static void writeDeterminan(double k)</code>	Prosedur ini menulis hasil determinan yang diterima dari opsiSavefile(double k) ke dalam sebuah file .txt
<code>public static void writeMatrixInverse(Matrix mat)</code>	Prosedur ini menulis hasil matriks inverse yang diterima oleh opsiSaveFile(Matrix mInv) ke dalam sebuah file .txt

3.8 Class MenuIO

3.8.1 Attribute

Tidak ada *attribute* khusus pada Class MenuIO.

3.8.2 Method

<code>public static void mainMenu()</code>	Prosedur ini menampilkan ke layar opsi yang ada di main menu
<code>public static void subMenuSPL()</code>	Prosedur ini menampilkan ke layar opsi yang ada di submenu spl
<code>public static void subMenuDeterminan()</code>	Prosedur ini menampilkan ke layar opsi yang ada di submenu determinan
<code>public static void subMenuMatriksBalikan()</code>	Prosedur ini menampilkan ke layar opsi yang ada di submenu matriks balikan
<code>public static void inputType()</code>	Prosedur ini menampilkan ke layar opsi input yang dapat diterima oleh program
<code>public static void menuSPL()</code>	Prosedur ini mengembalikan solusi SPL sesuai dengan cara inputan dan cara mencari solusi spl sesuai yang diinginkan pengguna
<code>public static void menuMatriksBalikan()</code>	Prosedur ini mengembalikan matriks balikan sesuai dengan cara inputan dan cara mencari matriks balikan yang diinginkan pengguna
<code>public static void menuInterpolasiPolinom()</code>	Prosedur ini mengembalikan interpolasi sesuai dengan cara inputan yang diinginkan pengguna
<code>public static void menuDeterminan()</code>	Prosedur ini mengembalikan nilai sesuai dengan cara mencari determinan yang diinginkan pengguna
<code>public static void menuBicubic()</code>	Prosedur ini mengembalikan interpolasi sesuai dengan cara inputan yang diinginkan pengguna
<code>public static void menuRegresi()</code>	Prosedur ini mengembalikan interpolasi sesuai dengan cara inputan yang diinginkan pengguna

3.9 Program Utama – Class Main

*beberapa kode ada yang berbeda dengan program aslinya karena terdapat beberapa revisi pada program aslinya

Class Main

```
public class Main {
    public static void main(String[] args) throws IOException {
        Scanner scan = new Scanner(System.in);
        int menu_choice = 0;

        do {
            MenuIO.mainMenu();
            System.out.print("Masukkan pilihan : ");
            menu_choice = scan.nextInt();
            scan.nextLine();

            switch (menu_choice) {
                case 1 : // SPL
                    MenuIO.menuSPL();
                    break;
                case 2:
                    MenuIO.menuDeterminan();
                    break;
                case 3:
                    MenuIO.menuMatriksBalikan();
                    break;
                case 4:
                    MenuIO.menuInterpolasiPolinom();
                    break;
                case 5:
                    MenuIO.menuBicubic();
                    break;
                case 6:
                    MenuIO.menuRegresi();
                    break;
                default :
                    if (menu_choice != 7) {
                        System.out.println("Input tidak valid,
silakan ulangi");
                    }
                    else {
                        System.out.println("Terima kasih !");
                    }
            }
        }while(menu_choice != 7);
    }
}
```

3.9.1 Menu Utama

Class MenuIO

```
public static void mainMenu() {  
    System.out.println("===== Kalkulator Matrix =====");  
    System.out.println("===== Main Menu =====");  
    System.out.println("1.Sistem Persamaan Linier");  
    System.out.println("2.Determinan");  
    System.out.println("3.Matriks balikan");  
    System.out.println("4.Interpolasi Polinom");  
    System.out.println("5.Interpolasi Bicubic Spline");  
    System.out.println("6.Regresi linier berganda");  
    System.out.println("7.Keluar");  
}
```

Program akan meminta input berupa angka dari angka 1 sampai 7, jika dimasukkan angka lain maka akan muncul pesan bahwa input tidak valid dan diminta memasukkan ulang.

3.9.2 Menu SPL

Class MenuIO

```
public static void subMenuSPL() {  
    System.out.println("===== Sub-Menu =====");  
    System.out.println("1.Metode eliminasi Gauss");  
    System.out.println("2.Metode eliminasi Gauss-Jordan");  
    System.out.println("3.Metode matriks balikan");  
    System.out.println("4.Kaidah Cramer");  
}
```

Menerima input berupa angka 1 sampai 4 untuk memilih metode apa yang akan digunakan untuk menyelesaikan SPL.

3.9.2.1 Metode Elimintasi Gauss

Class MenuIO

```
public static void menuSPL() throws IOException {  
    . . .  
    case 1: // gauss  
        System.out.println("===== Metode Eliminasi Gauss =====");  
        inputType();  
        System.out.print("Masukkan pilihan : ");  
        input_choice = scan.nextInt();  
        scan.nextLine();  
        switch(input_choice) {  
            case 1://keyboard-gauss  
                int row,col;  
                System.out.print("Banyak baris : ");  
                row = scan.nextInt();  
                System.out.print("Banyak kolom : ");
```

```

        col = scan.nextInt();
        scan.nextLine();
        Matrix mat = new Matrix(row,col);
        System.out.println("Input matriks : ");
        mat.readMatrix();
        Matrix gauss = SPL.gauss(mat);
        DisplaySolution.displayGauss(gauss);
        break;
    case 2://file-gauss
        Matrix mat2 = FileInputOutput.readFileMatrix();
        Matrix gauss2 = SPL.gauss(mat2);
        DisplaySolution.displayGauss(gauss2);
        break;

    }
    break;
    . .
}

```

Akan diminta masukkan berupa angka 1 atau angka 2 untuk memasukkan cara memasukkan data.

- 1 : masukkan dari keyboard
→ Akan diminta memasukkan banyak baris dan banyak kolom berupa suatu angka integer. Kemudian diminta memasukkan matriks *augmented* yang tiap elemennya adalah angka riil. Kemudian akan ditampilkan solusi untuk SPL tersebut.
- 2 : masukkan dari file
→ Akan dipanggil prosedur `readFileMatrix()` dari Class `FileInputOutput` (penjelasan mengenai prosedur tersebut ada pada implementasi Class `FileInputOutput`). Kemudian akan ditampilkan solusi untuk SPL tersebut.

Setelah itu diminta memasukkan pilihan apakah ingin menyimpan hasil tersebut pada sebuah file atau tidak (penjelasan lebih lanjut ada pada implementasi Class `FileInputOutput`).

3.9.2.2 Metode Eliminasi Gauss-Jordan

Class MenuIO

```

public static void menuSPL() throws IOException {
    . .
    case 2 :
        System.out.println("===== Metode Eliminasi Gauss-jordan =====");
        inputType();
        System.out.print("Masukkan pilihan : ");
        input_choice = scan.nextInt();
        scan.nextLine();
        switch(input_choice) {
            case 1://keyboard-gauss-jordan
                int row,col;
                System.out.print("Banyak baris : ");
                row = scan.nextInt();
                System.out.print("Banyak kolom : ");
                col = scan.nextInt();
                scan.nextLine();
                Matrix mat = new Matrix(row,col);

```

```

        System.out.println("Input matriks : ");
        mat.readMatrix();
        Matrix gauss_jordan = SPL.gaussjordan(mat);
        DisplaySolution.displayGaussJordan(gauss_jordan);
        break;
    case 2://file-gauss-jordan
        Matrix mat2 = FileInputOutput.readFileMatrix();
        Matrix gaussJordan2 = SPL.gauss(mat2);
        DisplaySolution.displayGaussJordan(gaussJordan2);
        break;

    }
    break;
    .
    .

```

Akan diminta masukkan berupa angka 1 atau angka 2 untuk memasukkan cara memasukkan data.

- 1 : masukkan dari keyboard
→ Akan diminta memasukkan banyak baris dan banyak kolom berupa suatu angka integer. Kemudian diminta memasukkan matriks *augmented* yang tiap elemennya adalah angka riil. Kemudian akan ditampilkan solusi untuk SPL tersebut.
- 2 : masukkan dari file
→ Akan dipanggil prosedur `readFileMatrix()` dari Class `FileInputOutput` (penjelasan mengenai prosedur tersebut ada pada implementasi Class `FileInputOutput`). Kemudian akan ditampilkan solusi untuk SPL tersebut.

Setelah itu diminta memasukkan pilihan apakah ingin menyimpan hasil tersebut pada sebuah file atau tidak (penjelasan lebih lanjut ada pada implementasi Class `FileInputOutput`).

3.9.2.3 Metode Matriks Balikan

Class MenuIO

```

public static void menuSPL() throws IOException {
    .
    .

```

```

case 3: // metode matriks balikan
    System.out.println("===== Metode Matriks Balikan =====");
    inputType();
    System.out.print("Masukkan pilihan : ");
    input_choice = scan.nextInt();
    scan.nextLine();
    switch(input_choice) {
        case 1://keyboard-mat_balikan
            int row,col;
            System.out.print("Banyak baris : ");
            row = scan.nextInt();
            System.out.print("Banyak kolom : ");
            col = scan.nextInt();
            scan.nextLine();
            Matrix mat = new Matrix(row,col);
            System.out.println("Input matriks : ");
            mat.readMatrix();

            if (!Matrix.isPersegi(Matrix.getMatrixCoefficient(mat))) {
                System.out.println("Matrix tidak persegi !");
            }
            else if (
Double.compare(Matrix.determinanEkspansiKofaktor(Matrix.getMatrixCoefficient(mat)), 0.0) == 0 ) {
                System.out.println("Matriks tidak memiliki inverse (determinan kosong)");
            }
            else {
                Matrix SPLinv = SPL.splInverse(mat);
                DisplaySolution.displaySPLbalikan(SPLinv);
            }

            break;
        case 2://file-SPLbalikan
            Matrix mat2 = FileInputOutput.readFileMatrix();
            if (!Matrix.isPersegi(Matrix.getMatrixCoefficient(mat2))) {
                System.out.println("Matrix tidak persegi !");
            }
            else if (
Double.compare(Matrix.determinanEkspansiKofaktor(Matrix.getMatrixCoefficient(mat2)), 0.0) == 0 ) {
                System.out.println("Matriks tidak memiliki inverse (determinan kosong)");
            }
            else {
                Matrix SPLinv2 = SPL.splInverse(mat2);
                DisplaySolution.displaySPLbalikan(SPLinv2);
            }

            break;
    }
    Break;
    . .

```

Akan diminta masukkan berupa angka 1 atau angka 2 untuk memasukkan cara memasukkan data.

- 1 : masukkan dari keyboard
→ Akan diminta memasukkan banyak baris dan banyak kolom berupa suatu angka integer. Kemudian diminta memasukkan matriks *augmented* yang tiap elemennya

adalah angka riil. Kemudian akan ditampilkan solusi untuk SPL tersebut.

- 2 : masukkan dari file
→ Akan dipanggil prosedur readFileMatrix() dari Class FileInputOutput (penjelasan mengenai prosedur tersebut ada pada implementasi Class FileInputOutput). Kemudian akan ditampilkan solusi untuk SPL tersebut.

Setelah itu diminta memasukkan pilihan apakah ingin menyimpan hasil tersebut pada sebuah file atau tidak (penjelasan lebih lanjut ada pada implementasi Class FileInputOutput).

3.9.2.4 Kaidah Cramer

Class MenuIO

```
public static void menuSPL() throws IOException {
    . . .
    case 4:
        System.out.println("===== Metode Cramer =====");
        inputType();
        System.out.print("Masukkan pilihan : ");
        input_choice = scan.nextInt();
        scan.nextLine();
        switch(input_choice) {
            case 1://keyboard-cramer
                int row,col;
                System.out.print("Banyak baris dan kolom: ");
                row = scan.nextInt();
                col = row + 1;
                scan.nextLine();
                Matrix mat = new Matrix(row,col);
                System.out.println("Input matriks : ");
                mat.readMatrix();
                Matrix cramer = SPL.cramerMethod(mat);
                DisplaySolution.displayCramer(cramer);
                break;
            case 2://file-cramer
                Matrix mat2 = FileInputOutput.readFileMatrix();
                Matrix Cramer2 = SPL.cramerMethod(mat2);
                DisplaySolution.displayCramer(Cramer2);
                break;
        }
        break;
}
```

Akan diminta masukkan berupa angka 1 atau angka 2 untuk memasukkan cara memasukkan data.

- 1 : masukkan dari keyboard
→ Akan diminta memasukkan ukuran matriks persegi berupa suatu angka integer.
Kemudian diminta memasukkan matriks *augmented* yang tiap elemennya adalah angka riil. Kemudian akan ditampilkan solusi untuk SPL tersebut.
- 2 : masukkan dari file
→ Akan dipanggil prosedur `readFileMatrix()` dari Class `FileInputOutput` (penjelasan mengenai prosedur tersebut ada pada implementasi Class `FileInputOutput`). Kemudian akan ditampilkan solusi untuk SPL tersebut.
Setelah itu diminta memasukkan pilihan apakah ingin menyimpan hasil tersebut pada sebuah file atau tidak (penjelasan lebih lanjut ada pada implementasi Class `FileInputOutput`).

3.9.3 Menu Determinan

Class MenuIO

```
public static void menuDeterminan() throws IOException {
    System.out.println("===== Menu Determinan Matriks =====");
    int pilihan;
    System.out.println("Pilih metode metode mencari determinan matriks : ");
    do {

        System.out.println("1.Metode reduksi baris");
        System.out.println("2.Metode ekspansi kofaktor");
        System.out.print("Pilih metode :");
        pilihan = scan.nextInt();
        if (pilihan != 1 & pilihan != 2) {
            System.out.println("Pilihan anda salah, tolong pilih ulang!");
        }
    }while(pilihan != 1 && pilihan !=2);
    . .
}
```

Menerima input berupa angka 1 atau 2 untuk memilih metode apa yang akan digunakan untuk mendapatkan nilai determinan.

3.9.3.1 Metode Reduksi Baris

Class MenuIO

```
public static void menuDeterminan() throws IOException {
    .
    .
    if (pilihan == 1) {
        System.out.println("===== Metode reduksi baris =====");
        inputType();
        System.out.print("Masukkan pilihan :");
    }
}
```

```

int input_choice = scan.nextInt();
scan.nextLine();
switch (input_choice) {
    case 1 ://keyboard
        System.out.print("Masukkan N :");
        int n = scan.nextInt();
        Matrix mat = new Matrix(n,n);
        System.out.println("Input matrix :");
        mat.readMatrix();
        double det = Matrix.determinanReduksi(mat);
        DisplaySolution.displayDeterminan(det);
        break;
    case 2://input text
        Matrix mat2 = FileInputOutput.readFileMatrix();
        if (!Matrix.isPersegi(mat2)) {
            System.out.println("Matrix tidak persegi !, determinan matrix tidak dapat ditemukan");
        }
        else {
            double det2 = Matrix.determinanReduksi(mat2);
            DisplaySolution.displayDeterminan(det2);
        }
        break;
}
...

```

Akan diminta masukkan berupa angka 1 atau angka 2 untuk memasukkan cara memasukkan data.

- 1 : masukkan dari keyboard
→ Akan diminta memasukkan ukuran matriks persegi berupa suatu angka integer. Kemudian diminta memasukkan matriks persegi yang tiap elemennya adalah angka riil. Kemudian akan ditampilkan nilai determinan dari matriks tersebut.
- 2 : masukkan dari file
→ Akan dipanggil prosedur readFileMatrix() dari Class FileInputOutput (penjelasan mengenai prosedur tersebut ada pada implementasi Class FileInputOutput). Kemudian akan ditampilkan nilai determinan dari matriks tersebut.

Setelah itu diminta memasukkan pilihan apakah ingin menyimpan hasil tersebut pada sebuah file atau tidak (penjelasan lebih lanjut ada pada implementasi Class FileInputOutput).

3.9.3.2 Metode Ekspansi Kofaktor

Class MenuIO

```

public static void menuDeterminan() throws IOException {
    ...
    else {
        System.out.println("===== Metode Ekspansi Kofaktor =====");
        inputType();
        System.out.print("Masukkan pilihan :");
        int input_choice = scan.nextInt();
        scan.nextLine();
        switch (input_choice) {

```

```

        case 1 ://keyboard
        System.out.print("Masukkan N :");
        int n = scan.nextInt();
        Matrix mat = new Matrix(n,n);
        System.out.println("Input matrix : ");
        mat.readMatrix();
        double det = Matrix.determinanEkspansiKofaktor(mat);
        DisplaySolution.displayDeterminan(det);
        break;
    case 2://input text
        Matrix mat2 = FileInputOutput.readFileMatrix();
        if (!Matrix.isPersegi(mat2)) {
            System.out.println("Matrix tidak persegi !, determinan matrix tidak dapat ditemukan");
        }
        else {
            double det2 = Matrix.determinanEkspansiKofaktor(mat2);
            DisplaySolution.displayDeterminan(det2);
        }
        break;
    }
}
}

```

Akan diminta masukkan berupa angka 1 atau angka 2 untuk memasukkan cara memasukkan data.

- 1 : masukkan dari keyboard
→ Akan diminta memasukkan ukuran matriks persegi berupa suatu angka integer. Kemudian diminta memasukkan matriks persegi yang tiap elemennya adalah angka riil. Kemudian akan ditampilkan nilai determinan dari matriks tersebut.
- 2 : masukkan dari file
→ Akan dipanggil prosedur readFileMatrix() dari Class FileInputOutput (penjelasan mengenai prosedur tersebut ada pada implementasi Class FileInputOutput). Kemudian akan ditampilkan nilai determinan dari matriks tersebut.

Setelah itu diminta memasukkan pilihan apakah ingin menyimpan hasil tersebut pada sebuah file atau tidak (penjelasan lebih lanjut ada pada implementasi Class FileInputOutput).

3.9.4 Menu Matriks Balikan

Class MenuIO

```

public static void subMenuMatriksBalikan() {
    System.out.println("===== Sub-Menu =====");
    System.out.println("1.Metode Gauss Jordan");
    System.out.println("2.Metode Adjoint");
}

```

Menerima input berupa angka 1 atau 2 untuk memilih metode apa yang akan digunakan untuk mendapatkan nilai matriks balikan.

3.9.4.1 Metode Gauss Jordan

Class MenuIO

```
public static void menuMatriksBalikan() throws IOException{
    int sub_menu_choice, input_choice;
    subMenuMatriksBalikan();
    System.out.print("Masukkan pilihan : ");
    sub_menu_choice = scan.nextInt();
    scan.nextLine();
    switch(sub_menu_choice){
        case 1: // Gauss Jordan
            System.out.println("===== Metode Gauss Jordan =====");
            inputType();
            System.out.print("Masukkan pilihan : ");
            input_choice = scan.nextInt();
            scan.nextLine();
            switch(input_choice){
                case 1:
                    int row,col;
                    System.out.print("Banyak baris : ");
                    row = scan.nextInt();
                    System.out.print("Banyak kolom : ");
                    col = scan.nextInt();
                    scan.nextLine();
                    Matrix inverseMatrix = new Matrix(row,col);
                    System.out.println("Input matriks : ");
                    inverseMatrix.readMatrix();
                    inverseMatrix = Matrix.inversGaussJordan(inverseMatrix);
                    inverseMatrix.displayMatrix();
                    FileInputOutput.opsiSaveFile(inverseMatrix);
                    break;
                case 2:
                    Matrix inverseMatriks = FileInputOutput.readFileMatrix();
                    inverseMatriks = Matrix.inversGaussJordan(inverseMatriks);
                    inverseMatriks.displayMatrix();
                    FileInputOutput.opsiSaveFile(inverseMatriks);
                    break;
            }
            break;
    ...
}
```

Akan diminta masukkan berupa angka 1 atau angka 2 untuk memasukkan cara memasukkan data.

- 1 : masukkan dari keyboard
→ Akan diminta memasukkan ukuran matriks persegi berupa suatu angka integer.
Kemudian diminta memasukkan matriks persegi yang tiap elemennya adalah angka riil. Kemudian akan ditampilkan nilai determinan dari matriks tersebut.
- 2 : masukkan dari file

→ Akan dipanggil prosedur readFileMatrix() dari Class FileInputOutput (penjelasan mengenai prosedur tersebut ada pada implementasi Class FileInputOutput). Kemudian akan ditampilkan hasil matriks balikan dari matriks tersebut.

Setelah itu diminta memasukkan pilihan apakah ingin menyimpan hasil tersebut pada sebuah file atau tidak (penjelasan lebih lanjut ada pada implementasi Class FileInputOutput).

3.9.4.2 Metode Adjoint

Class MenuIO

```
public static void menuMatriksBalikan() throws IOException{
    . .
    case 2: // Adjoint
        System.out.println("===== Metode Adjoint =====");
        inputType();
        System.out.print("Masukkan pilihan : ");
        input_choice = scan.nextInt();
        scan.nextLine();
        switch(input_choice){
            case 1:
                int row,col;
                System.out.print("Banyak baris : ");
                row = scan.nextInt();
                System.out.print("Banyak kolom : ");
                col = scan.nextInt();
                scan.nextLine();
                Matrix inverseMatrix = new Matrix(row,col);
                System.out.println("Input matriks : ");
                inverseMatrix.readMatrix();
                inverseMatrix = Matrix.getMatrixInverseAdjoin(inverseMatrix);
                inverseMatrix.displayMatrix();
                FileInputOutput.opsiSaveFile(inverseMatrix);
                break;
            case 2:
                Matrix inverseMatriks = FileInputOutput.readFileMatrix();
                inverseMatriks = Matrix.getMatrixInverseAdjoin(inverseMatriks);
                inverseMatriks.displayMatrix();
                FileInputOutput.opsiSaveFile(inverseMatriks);
                break;
        }
        break;
}
```

Akan diminta masukkan berupa angka 1 atau angka 2 untuk memasukkan cara memasukkan data.

- 1 : masukkan dari keyboard
→ Akan diminta memasukkan ukuran matriks persegi berupa suatu angka integer.
Kemudian diminta memasukkan matriks persegi yang tiap elemennya adalah angka riil. Kemudian akan ditampilkan nilai determinan dari matriks tersebut.
- 2 : masukkan dari file
→ Akan dipanggil prosedur readFileMatrix() dari Class FileInputOutput
(penjelasan mengenai prosedur tersebut ada pada implementasi Class FileInputOutput). Kemudian akan ditampilkan hasil matriks balikan dari matriks tersebut.

Setelah itu diminta memasukkan pilihan apakah ingin menyimpan hasil tersebut pada sebuah file atau tidak (penjelasan lebih lanjut ada pada implementasi Class FileInputOutput).

3.9.5 Menu Interpolasi Polinom

Class MenuIO

```
public static void menuInterpolasiPolinom() throws IOException{
    int input_choice;
    System.out.println("===== MENU INTERPOLASI POLINOM =====");
    inputType();
    System.out.print("Masukkan pilihan : ");
    input_choice = scan.nextInt();
    scan.nextLine();
    switch(input_choice){
        case 1: // Input keyboard
            int n;
            System.out.print("Banyak Point : ");
            n = scan.nextInt();
            scan.nextLine();
            Matrix pointMatrix = new Matrix(n,2);
            pointMatrix.readMatrix();
            Matrix linEqMatrix = Polinom.createLinearEq(pointMatrix);
            Matrix solution = Polinom.getPolinomSol(linEqMatrix);
            String equation = Polinom.getPolinomEq(solution);
            String eqString = String.format("Persamaan interpolasi polinom tersebut adalah : %s", equation );
            System.out.println(eqString);
            System.out.println("Masukkan nilai x yang ingin ditaksir : ");
            double x = scan.nextDouble();
            double taksiran = Polinom.getTaksiran(solution, x);
            String strTaksiran = String.format("Taksiran f(%f) = %f", x, taksiran);
            System.out.println(strTaksiran);
            String[]arrstr = new String[]{eqString,strTaksiran};
            FileInputOutput.opeSaveFilePolinom(arrstr);
            break;
        case 2: // Input file
            Matrix matrixFile = FileInputOutput.readFileMatrix();
            Matrix poinMatrix = matrixFile.copyMatrix();
            poinMatrix.copyMatrixCustom(0,0,matrixFile.getRow()-1,matrixFile.getCol());
            double eks = matrixFile.getElmt(matrixFile.getRow()-1, 0);
            Matrix linEqMatriks = Polinom.createLinearEq(poinMatrix);
            Matrix sol = Polinom.getPolinomSol(linEqMatriks);
            String eq = Polinom.getPolinomEq(sol);
            String eqStr = String.format("Persamaan interpolasi polinom tersebut adalah : %s", eq);
```

```

        System.out.println(eqStr);
        double estimate = Polinom.getTaksiran(sol, eks);
        String strEstimate = String.format("Taksiran f(%f) = %f", eks, estimate);
        System.out.println(strEstimate);
        String[] arr = new String[]{eqStr,strEstimate};
        FileInputOutput.opsiSaveFilePolinom(arr);
        break;
    }
}

```

Akan diminta masukkan berupa angka 1 atau angka 2 untuk memasukkan cara memasukkan data.

- 1 : masukkan dari keyboard
→ Akan diminta memasukkan banyaknya titik koordinat yang ingin dimasukkan untuk dicari persamaan interpolasi polinomnya. Kemudian diminta memasukkan titik-titik koordinat yang ingin dimasukkan dalam bentuk matriks berukuran $n \times 2$. Kemudian akan ditampilkan persamaan interpolasi polinom. Setelah itu akan diminta memasukkan nilai X yang ingin ditaksir, kemudian akan ditampilkan hasil taksiran tersebut.
- 2 : masukkan dari file
→ Akan dipanggil prosedur readFileMatrix() dari Class FileInputOutput (penjelasan mengenai prosedur tersebut ada pada implementasi Class FileInputOutput). Kemudian akan ditampilkan persamaan interpolasi polinom dan nilai taksiran suatu X.

Setelah itu diminta memasukkan pilihan apakah ingin menyimpan hasil tersebut pada sebuah file atau tidak (penjelasan lebih lanjut ada pada implementasi Class FileInputOutput).

3.9.6 Menu Interpolasi *Bicubic Spline*

Class MenuIO

```

public static void menuBicubic() throws IOException {
    double x,y;
    int input_choice;
    System.out.println("===== Interpolasi Bicubic Spline =====");
    do {
        inputType();
        System.out.print("Masukkan pilihan : ");
        input_choice = scan.nextInt();
        scan.nextLine();
        switch (input_choice) {
            case 1://keyboard-bicubic
                int row, col;
                row = 4;
                col = 4;
                Matrix mat = new Matrix(row, col);
                System.out.println("Input matriks berukuran 4x4: ");
                mat.readMatrix();
                System.out.println("Input Titik (x,y) yang ingin ditaksir nilainya");

```

```

        System.out.print("x : ");
        x = scan.nextDouble();
        scan.nextLine();
        System.out.print("y : ");
        y = scan.nextDouble();
        scan.nextLine();
        Matrix msolved = new Matrix(16, 1);
        msolved = BicubicSpline.solveBicubic(mat);
        double taksiran = BicubicSpline.getTaksiranBicubic(msolved, x, y);
        System.out.printf("f(%f,%f) = %f\n", x, y, taksiran);
        FileInputOutput.opsiSaveFile(taksiran);
        break;

    case 2://file bicubic
        int i, j;
        Matrix m_input = FileInputOutput.readFileMatrix();

        Matrix mIn = new Matrix(m_input.getRow() - 1, m_input.getCol());
        for (i = 0; i < mIn.getRow(); i++) {
            for (j = 0; j < mIn.getCol(); j++) {
                mIn.setElmt(i, j, m_input.getElmt(i, j));
            }
        }
        x = m_input.getElmt(4, 0);
        y = m_input.getElmt(4, 1);
        Matrix mfile_solved = new Matrix(16,1);
        mfile_solved = BicubicSpline.solveBicubic(mIn);
        double file_taksiran = BicubicSpline.getTaksiranBicubic(mfile_solved, x, y);
        System.out.printf("f(%f,%f) = %f\n", x, y, file_taksiran);
        FileInputOutput.opsiSaveFile(file_taksiran);
        break;
    default:
        System.out.println("Input tidak valid, silakan ulangi");
    }
} while (input_choice != 1 && input_choice != 2);
}

```

Akan diminta masukkan berupa angka 1 atau angka 2 untuk memasukkan cara memasukkan data.

- 1 : masukkan dari keyboard
→ Akan diminta memasukkan matriks berukuran 4x4 dengan ketentuan format sebagai berikut:

$$\begin{array}{cccc}
 f(0, 0) & f(1, 0) & f(0, 1) & f(1, 1) \\
 f_x(0, 0) & f_x(1, 0) & f_x(0, 1) & f_x(1, 1) \\
 f_y(0, 0) & f_y(1, 0) & f_y(0, 1) & f_y(1, 1) \\
 f_{xy}(0, 0) & f_{xy}(1, 0) & f_{xy}(0, 1) & f_{xy}(1, 1)
 \end{array}$$

Kemudian akan diminta memasukkan nilai titik koordinat (x, y) yang ingin ditaksir, kemudian akan ditampilkan hasil taksiran tersebut.

- 2 : masukkan dari file
→ Akan dipanggil prosedur readFileMatrix() dari Class FileInputOutput (penjelasan mengenai prosedur tersebut ada pada implementasi Class FileInputOutput). Kemudian akan ditampilkan hasil taksiran dari titik koordinat (x, y) yang ada pada file input yang berada pada baris paling bawah.

Setelah itu diminta memasukkan pilihan apakah ingin menyimpan hasil tersebut pada sebuah file atau tidak (penjelasan lebih lanjut ada pada implementasi Class FileInputOutput).

3.9.7 Menu Regresi

Class MenuIO

```

public static void menuRegresi() throws IOException{
    int input_choice, nVar, nData, i, j;
    String strEq, strTaksiran;
    System.out.println("===== Regresi Linier Berganda =====");
    do {
        inputType();
        System.out.print("Masukkan pilihan : ");
        input_choice = scan.nextInt();
        scan.nextLine();
        switch (input_choice) {
            case 1://keyboard-regresi
                System.out.print("Masukkan banyaknya variabel: ");
                nVar = scan.nextInt();
                scan.nextLine();
                double[][] arrVarVal = new double[nVar][];
                System.out.print("Masukkan banyaknya data: ");
                nData = scan.nextInt();
                scan.nextLine();
                Matrix m_aug = new Matrix(nData, nVar+1);
                System.out.println("Input Matriks (dalam bentuk Matriks Augmented): ");
                m_aug.readMatrix();
                System.out.println("Masukkan nilai setiap variabel yang ingin ditaksir nilainya");
                for(i=0;i<nVar;i++){
                    System.out.printf("x%d : ",i+1);
                    arrVarVal[i] = scan.nextDouble();
                    scan.nextLine();
                }
                Matrix m_solved = new Matrix(m_aug.getRow(), 1);
                m_solved = Regresi.solveRegresi(m_aug);
                strEq = Regresi.getRegresiEq(m_solved);
                System.out.printf("Persamaan Regresi Linear Berganda:\n%s\n", strEq);
                strTaksiran = Regresi.getStrTaksiran(m_solved, arrVarVal);
                System.out.println(strTaksiran);
                String[] arrStrOut = new String[]{strEq,strTaksiran};
                FileInputOutput.opsiSaveFile(arrStrOut);
                break;

            case 2://file regresi
                Matrix m_input = FileInputOutput.readFileMatrix();
                Matrix mfile_aug = new Matrix((m_input.getRow()-1), m_input.getCol());
                for(i=0;i< mfile_aug.getRow();i++){
                    for(j=0;j< mfile_aug.getCol();j++){
                        mfile_aug.setElmt(i,j,m_input.getElmt(i,j));
                    }
                }
                double[][] arrVarValFile = new double[mfile_aug.getCol()-1];
                for(i=0;i<nVar;i++){
                    arrVarValFile[i] = m_input.getElmt(m_input.getRow()-1,i);
                }
                Matrix mfile_solved = new Matrix(mfile_aug.getRow(), 1);
                mfile_solved = Regresi.solveRegresi(mfile_aug);
                strEq = Regresi.getRegresiEq(mfile_solved);
                System.out.printf("Persamaan Regresi Linear Berganda:\n%s\n", strEq);
                strTaksiran = Regresi.getStrTaksiran(mfile_solved, arrVarValFile);
                System.out.println(strTaksiran);
                String[] arrStrOutFile = new String[]{strEq,strTaksiran};
                FileInputOutput.opsiSaveFile(arrStrOutFile);
                break;
        }
    }
}

```

```

        default:
            System.out.println("Input tidak valid, silakan ulangi");
        }
    } while (input_choice != 1 && input_choice != 2);
}

```

Akan diminta masukkan berupa angka 1 atau angka 2 untuk memasukkan cara memasukkan data.

- 1 : masukkan dari keyboard
→ Akan diminta memasukkan banyaknya variabel yang digunakan dan banyaknya data yang ingin dimasukkan. Kemudian diminta untuk memasukkan matriks dalam bentuk matriks *augmented* dengan ketentuan format sebagai berikut:

var1_1 var2_1 var3_1 ... vark_1 Y_1

.

.

var1_n var2_n var3_n ... vark_n Y_n

Kemudian akan diminta memasukkan nilai masing-masing variabel yang ingin ditaksir nilainya, kemudian akan ditampilkan persamaan regresi linear berganda dan hasil taksiran tersebut.

- 2 : masukkan dari file
→ Akan dipanggil prosedur readFileMatrix() dari Class FileInputOutput (penjelasan mengenai prosedur tersebut ada pada implementasi Class FileInputOutput). Kemudian akan ditampilkan persamaan regresi linear berganda dan hasil taksiran tersebut.

Setelah itu diminta memasukkan pilihan apakah ingin menyimpan hasil tersebut pada sebuah file atau tidak (penjelasan lebih lanjut ada pada implementasi Class FileInputOutput).

Bab IV

Eksperimen

1. Temukan solusi SPL $Ax = b$, berikut:

a.

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

Hasil metode Gauss :

```
-----  
===== Metode Eliminasi Gauss =====  
===== Metode Input =====  
1.Keyboard  
2.File txt  
Masukkan pilihan : 2  
=====Proses membaca file=====  
1.Nama file (pastikan file di folder test)  
2.Full path  
Pilihan (1/2) : 1  
Masukkan nama file (dengan extension) : 1_a.txt  
Matriks tidak ada solusi
```

Hasil metode Gauss Jordan :

```
===== Metode Eliminasi Gauss-jordan =====  
===== Metode Input =====  
1.Keyboard  
2.File txt  
Masukkan pilihan : 2  
=====Proses membaca file=====  
1.Nama file (pastikan file di folder test)  
2.Full path  
Pilihan (1/2) : 1  
Masukkan nama file (dengan extension) : 1_a.txt  
Matriks tidak ada solusi
```

Hasil metode Matriks Balikan :

```

===== Metode Matriks Balikan =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 1_a.txt
Matriks tidak memiliki inverse (determinan kosong)

```

Hasil metode Cramer :

```

===== Metode Cramer =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 1_a.txt
Metode cramer tidak dapat dilakukan

```

b.

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

Hasil Metode Gauss :

```

===== Metode Eliminasi Gauss =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
====Proses membaca file====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 1_b.txt
x1 = 3.0 + b
x2 = 2.0b
x3 = a
x4 = -1.0 + b
x5 = b
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak

```

Hasil Gauss Jordan :

```
===== Metode Eliminasi Gauss-jordan =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 1_b.txt
x1 = 3.0 + b
x2 = 2.0b
x3 = a
x4 = -1.0 + b
x5 = b
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
```

Hasil metode Matriks Balikan :

```
----- -
===== Metode Matriks Balikan =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
====Proses membaca file====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 1_b.txt
Matrix tidak persegi !
```

Hasil metode Cramer :

```
===== Metode Cramer =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
====Proses membaca file====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 1_b.txt
Matriks tidak persegi !
```

c.

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

Hasil metode Gauss :

```
===== Metode Eliminasi Gauss =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 1_c.txt
x1 = a
x2 = 1.0 - c
x3 = b
x4 = -2.0 - c
x5 = 1.0 + c
x6 = c
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
1
```

Hasil metode Gauss Jordan :

```
===== Metode Eliminasi Gauss-jordan =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 1_c.txt
x1 = a
x2 = 1.0 - c
x3 = b
x4 = -2.0 - c
x5 = 1.0 + c
x6 = c
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
1
Masukkan nama file bersama extension .txt : lcGaussJordan.txt
```

Hasil metode Matriks Balikan :

```
=====
Metode Matriks Balikan =====
=====
Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====
Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : l_c.txt
Matrix tidak persegi !
```

Hasil metode Cramer :

```
=====
Metode Cramer =====
=====
Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====
Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : l_c.txt
Matrix tidak persegi !
```

d.

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

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

- $n = 6$

Hasil metode Gauss :

```
===== Metode Eliminasi Gauss =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 1_d6.txt
x1 = 36.0
x2 = -630.0
x3 = 3360.0
x4 = -7560.0
x5 = 7560.0
x6 = -2772.0
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
1
Masukkan nama file bersama extension .txt : 1d6Gauss.txt
```

Hasil metode Gauss Jordan :

```
===== Metode Eliminasi Gauss-jordan =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 1_d6.txt
x1 = 36.0
x2 = -630.0
x3 = 3360.0
x4 = -7560.0
x5 = 7560.0
x6 = -2772.0
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
```

Hasil metode Matriks Balikan :

```
===== Metode Matriks Balikan =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 1_d6.txt
x1 = 36.0000
x2 = -630.0000
x3 = 3360.0000
x4 = -7560.0000
x5 = 7560.0000
x6 = -2772.0000
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
```

Hasil metode Cramer :

```
===== Metode Cramer =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 1_d6.txt
x1 = 36.0000
x2 = -629.9999
x3 = 3359.9997
x4 = -7559.9994
x5 = 7559.9995
x6 = -2771.9999
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
```

- $n = 10$

Hasil metode Gauss :

```
----- 
===== Metode Eliminasi Gauss =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 1_dl0.txt
x1 = 99.9904
x2 = -4949.1596
x3 = 79181.9853
x4 = -600435.405
x5 = 2521731.7416
x6 = -6304125.9094
x7 = 9606023.1512
x8 = -8748135.3476
x9 = 4373977.4702
x10 = -923378.5099
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
1
-----
```

Hasil metode Gauss Jordan :

```
===== Metode Eliminasi Gauss-jordan =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 1_dl0.txt
x1 = 99.9904
x2 = -4949.1596
x3 = 79181.9853
x4 = -600435.405
x5 = 2521731.7416
x6 = -6304125.9094
x7 = 9606023.1512
x8 = -8748135.3476
x9 = 4373977.4702
x10 = -923378.5099
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
```

Hasil metode Matriks Balikan :

```
===== Metode Matriks Balikan =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 1_d10.txt
x1 = 99.9904
x2 = -4949.1596
x3 = 79181.9853
x4 = -600435.4050
x5 = 2521731.7416
x6 = -6304125.9094
x7 = 9606023.1512
x8 = -8748135.3476
x9 = 4373977.4702
x10 = -923378.5099
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
```

Hasil metode Cramer :

```
1.Metode eliminasi Gauss
2.Metode eliminasi Gauss-Jordan
3.Metode matriks balikan
4.Kaidah Cramer
Masukkan pilihan : 4
===== Metode Cramer =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 1_d10.txt
x1 = 99.9917
x2 = -4949.2274
x3 = 79182.0702
x4 = -600435.1795
x5 = 2521731.7609
x6 = -6304125.9076
x7 = 9606023.1503
x8 = -8748135.3476
x9 = 4373977.4702
x10 = -923378.5099
```

2. SPL berbentuk matriks *augmented*

a.

$$\left[\begin{array}{ccccc} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{array} \right]$$

Hasil metode Gauss :

```
=====
Metode Eliminasi Gauss =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 2_a.txt
x1 = -1.0 + b
x2 = 2.0a
x3 = a
x4 = b
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
```

Hasil metode Gauss Jordan :

```
===== Metode Eliminasi Gauss-jordan =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 2_a.txt
x1 = -1.0 + b
x2 = 2.0a
x3 = a
x4 = b
```

Hasil metode Matriks Balikan :

```

=====
Metode Matriks Balikan =====
=====
Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====
Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 2_a.txt
Matriks tidak memiliki inverse (determinan kosong)

```

Hasil metode Cramer :

```

=====
Metode Cramer =====
=====
Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====
Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 2_a.txt
Metode cramer tidak dapat dilakukan

```

b.

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

Hasil metode Gauss :

```

=====
Metode Eliminasi Gauss =====
=====
Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====
Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 2_b.txt
x1 = 0.0
x2 = 2.0
x3 = 1.0
x4 = 1.0
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
1
Masukkan nama file bersama extenson .txt : 2bGauss.txt

```

Hasil metode Gauss Jordan :

```
===== Metode Eliminasi Gauss-jordan =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 1
Banyak baris : 6
Banyak kolom : 5
Input matriks :
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
x1 = 0.0
x2 = 2.0
x3 = 1.0
x4 = 1.0
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
```

Hasil metode Matriks Balikan :

```
===== Metode Matriks Balikan =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 2_b.txt
Matrix tidak persegi !
```

Hasil metode Cramer :

```
===== Metode Cramer =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 2_b.txt
Matriks tidak persegi !
```

3. SPL berbentuk

a.

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

Hasil metode Gauss :

```
=====
Metode Eliminasi Gauss =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 3_a.txt
x1 = -0.2243
x2 = 0.1824
x3 = 0.7095
x4 = -0.2581
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
1
Masukkan nama file bersama extension .txt : 3aGauss.txt
```

Hasil metode Gauss Jordan :

```
=====
Metode Eliminasi Gauss-jordan =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 3_a.txt
x1 = -0.2243
x2 = 0.1824
x3 = 0.7095
x4 = -0.2581
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
```

Hasil metode Matriks Balikan :

```
===== Metode Matriks Balikan =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 3_a.txt
x1 = -0.2243
x2 = 0.1824
x3 = 0.7095
x4 = -0.2581
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
```

Hasil metode Cramer :

```
===== Metode Cramer =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 3_a.txt
x1 = -0.2243
x2 = 0.1824
x3 = 0.7095
x4 = -0.2581
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
```

b.

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

Hasil Gauss :

```
Masukkan pilihan : 1
===== Metode Eliminasi Gauss =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 3_b.txt
Matriks tidak ada solusi
```

Hasil metode Gauss Jordan :

```
===== Metode Eliminasi Gauss-jordan =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 3_b.txt
Matriks tidak ada solusi
-----
```

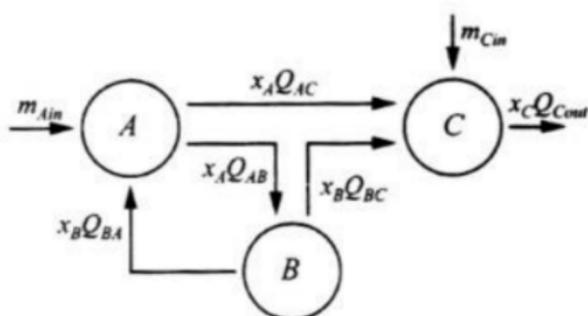
Hasil metode Matriks Balikan :

```
===== Metode Matriks Balikan =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 3_b.txt
Matriks tidak persegi !
.....
```

Hasil metode Cramer :

```
===== Metode Cramer =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 3_b.txt
Matriks tidak persegi !
```

4. 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:

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

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

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

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 $Q_{Cout} = 150 \text{ m}^3/\text{s}$ dan $m_{Ain} = 1300$ dan $m_{Cin} = 200 \text{ mg/s}$.

Input matrix :

```
-120 60 0 -1300
40 -80 0 0
80 20 -150 -200
```

Hasil Gauss :

```
===== Metode Eliminasi Gauss =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 4.txt
x1 = 14.4444
x2 = 7.2222
x3 = 10.0
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
```

Hasil metode Gauss Jordan :

```
===== Metode Eliminasi Gauss-jordan =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 4.txt
x1 = 14.4444
x2 = 7.2222
x3 = 10.0
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
```

Hasil metode Matriks Balikan :

```
=====
Metode Matriks Balikan =====
=====
Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====
Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 4.txt
x1 = 14.4444
x2 = 7.2222
x3 = 10.0000
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
```

Hasil metode Cramer :

```
=====
Metode Cramer =====
=====
Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====
Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 4.txt
x1 = 14.4444
x2 = 7.2222
x3 = 10.0000
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
```

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:

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

Hasil interpolasi di $x = 0.2$

```
===== MENU INTERPOLASI POLINOM =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 2
Masukkan full path dengan benar : /Users/maximilian/Documents/GitHub/Algeo01-22061/test/5_a1.txt
Persamaan interpolasi polinom tersebut adalah : f(x) = -0.022977 + 0.240000x^1 + 0.197396x^2 + 0.000000x^3 + 0.026042x^4 + 0.000000x^5 - 0.000000x^6
Taksiran f(0.200000) = 0.032961
```

b. $x = 0.55$ $f(x) = ?$

Hasil interpolasi di $x = 0.55$

```
===== MENU INTERPOLASI POLINOM =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 2
Masukkan full path dengan benar : /Users/maximilian/Documents/GitHub/Algeo01-22061/test/5_a2.txt
Persamaan interpolasi polinom tersebut adalah : f(x) = -0.022977 + 0.240000x^1 + 0.197396x^2 + 0.000000x^3 + 0.026042x^4 + 0.000000x^5 - 0.000000x^6
Taksiran f(0.550000) = 0.171119
```

c. $x = 0.85$ $f(x) = ?$

Hasil interpolasi di $x = 0.85$

```
===== MENU INTERPOLASI POLINOM =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 2
Masukkan full path dengan benar : /Users/maximilian/Documents/GitHub/Algeo01-22061/test/5_a3.txt
Persamaan interpolasi polinom tersebut adalah : f(x) = -0.022977 + 0.240000x^1 + 0.197396x^2 + 0.000000x^3 + 0.026042x^4 + 0.000000x^5 - 0.000000x^6
Taksiran f(0.850000) = 0.337236
```

d. $x = 1.28$ $f(x) = ?$

Hasil interpolasi di $x = 1.28$

```
===== MENU INTERPOLASI POLINOM =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 2
Masukkan full path dengan benar : /Users/maximilian/Documents/GitHub/Algeo01-22061/test/5_a4.txt
Persamaan interpolasi polinom tersebut adalah : f(x) = -0.022977 + 0.240000x^1 + 0.197396x^2 + 0.000000x^3 + 0.026042x^4 + 0.000000x^5 - 0.000000x^6
Taksiran f(1.280000) = 0.677542
```

- 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:

a. 16/07/2022

Hasil interpolasi di 16/07/2022, $x = 7.5161$

```
===== MENU INTERPOLASI POLINOM =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 2
Masukkan full path dengan benar : /Users/maximilian/Documents/GitHub/Algeo01-22061/test/5_ba.txt
Persamaan interpolasi polinom tersebut adalah : f(x) = 7189689299210.578000 - 9350047182614.938000x^1 + 5335781355210.227000x^2 - 1757285362019.246000x^3 + 368642656689.358150x^4 - 51143697738.618500x^5 + 4696819259.618698x^6 - 275530268.333486x^7 + 9374635.490914x^8 - 141019.126329x^9
Taksiran f(7.516100) = 53533.724609
```

b. 10/08/2022

Hasil interpolasi di 10/08/2022, $x = 8.3225$

```
===== MENU INTERPOLASI POLINOM =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 2
Masukkan full path dengan benar : /Users/maximilian/Documents/GitHub/Algeo01-22061/test/5_bb.txt
Persamaan interpolasi polinom tersebut adalah : f(x) = 7189689299210.578000 - 9350047182614.938000x^1 + 5335781355210.227000x^2 - 1757285362019.246000x^3 + 368642656689.358150x^4 - 51143697738.618500x^5 + 4696819259.618698x^6 - 275530268.333486x^7 + 9374635.490914x^8 - 141019.126329x^9
Taksiran f(8.322500) = 36319.832031
```

c. 05/09/2022

Hasil interpolasi di 05/09/2022, $x = 9.16$

```
===== MENU INTERPOLASI POLINOM =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 2
Masukkan full path dengan benar : /Users/maximilian/Documents/GitHub/Algeo01-22061/test/5_bc.txt
Persamaan interpolasi polinom tersebut adalah : f(x) = 7189689299210.578000 - 9350047182614.938000x^1 + 5335781355210.227000x^2 - 1757285362019.246000x^3 + 368642656689.358150x^4 - 51143697738.618500x^5 + 4696819259.618698x^6 - 275530268.333486x^7 + 9374635.490914x^8 - 141019.126329x^9
Taksiran f(9.160000) = -608667.062500
```

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

Hasil interpolasi di 17/08/2022, $x = 9.16$

```
===== MENU INTERPOLASI POLINOM =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 2
Masukkan full path dengan benar : /Users/maximilian/Documents/GitHub/Algeo01-22061/test/5_bd.txt
Persamaan interpolasi polinom tersebut adalah : f(x) = 7189689299210.578000 - 9350047182614.938000x^1 + 5335781355210.227000x^2 - 1757285362019.246000x^3 + 368642656689.358150x^4 - 51143697738.618500x^5 + 4696819259.618698x^6 - 275530268.333486x^7 + 9374635.490914x^8 - 141019.126329x^9
Taksiran f(8.548300) = 13390.582031
```

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

Hasil interpolasi dengan 5 titik dan di $x = 1.3$

```
===== MENU INTERPOLASI POLINOM =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 2
Masukkan full path dengan benar : /Users/maximilian/Documents/GitHub/Algeo01-22061/test/5_c.txt
Persamaan interpolasi polinom tersebut adalah : f(x) = 0.290600 + 0.376125x^1 - 0.147292x^2 + 0.021875x^3 - 0.003255x^4
Taksiran f(1.30000) = 0.569402
```

6. Studi Kasus Regresi Linear Berganda

Diberikan sekumpulan data sesuai pada tabel berikut ini.

Table 12.1: Data for Example 12.1

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

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

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

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

$$\begin{aligned}
 20b_0 + 863.1b_1 + 1530.4b_2 + 587.84b_3 &= 19.42 \\
 863.1b_0 + 54876.89b_1 + 67000.09b_2 + 25283.395b_3 &= 779.477 \\
 1530.4b_0 + 67000.09b_1 + 117912.32b_2 + 44976.867b_3 &= 1483.437 \\
 587.84b_0 + 25283.395b_1 + 44976.867b_2 + 17278.5086b_3 &= 571.1219
 \end{aligned}$$

Isi “6.txt”

```
6.txt - Notepad
File Edit Format View Help
72.4 76.3 29.18 0.90
41.6 70.3 29.35 0.91
34.4 77.1 29.24 0.96
35.1 68.0 29.27 0.89
10.7 79.0 29.78 1.00
12.9 67.4 29.39 1.10
8.3 66.8 29.69 1.15
20.1 76.9 29.48 1.03
72.2 77.7 29.09 0.77
24.0 67.7 29.60 1.07
23.2 76.8 29.38 1.07
47.4 86.6 29.35 0.94
31.5 76.9 29.63 1.10
10.6 86.3 29.56 1.10
11.2 86.0 29.48 1.10
73.3 76.3 29.40 0.91
75.4 77.9 29.28 0.87
96.6 78.7 29.29 0.78
107.4 86.8 29.03 0.82
54.9 70.9 29.37 0.95
50 76 29.30
```

Output

```
===== Kalkulator Matrix =====
===== Main Menu =====
1.Sistem Persamaan Linier
2.Determinan
3.Matriks balikan
4.Interpolasi Polinom
5.Interpolasi Bicubic Spline
6.Regresi linier berganda
7.Keluar
Masukkan pilihan : 6
===== Regresi Linier Berganda =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 6.txt
Persamaan Regresi Linear Berganda:
f(x) = -3.50235 - 0.00263x1 + 0.00080x2 + 0.15397x3
f(50.0000,76.0000,29.3000) = 0.9385
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
2
```

7. Studi Kasus Interpolasi *Bicubic Spline*

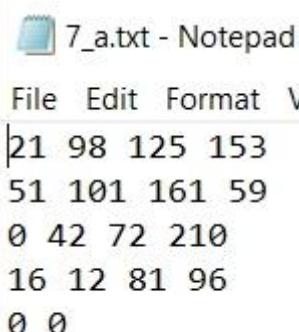
Diberikan matriks input dengan bentuk sebagai berikut. Format matriks masukan bukan mewakili nilai matriks, tetapi mengikuti format masukan pada bagian “Spesifikasi Tugas” nomor 7.

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

Tentukan nilai:

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

Isi “7_a.txt”



```
7_a.txt - Notepad
File Edit Format View
21 98 125 153
51 101 161 59
0 42 72 210
16 12 81 96
0 0
```

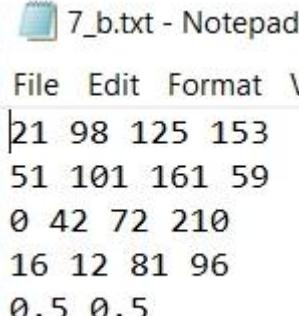
Output



```
===== Kalkulator Matrix =====
===== Main Menu =====
1.Sistem Persamaan Linier
2.Determinan
3.Matriks balikan
4.Interpolasi Polinom
5.Interpolasi Bicubic Spline
6.Regresi linier berganda
7.Keluar
Masukkan pilihan : 5
===== Interpolasi Bicubic Spline =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1>Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 7_a.txt
f(0.000000,0.000000) = 21.000000
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
2
```

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

Isi “7_b.txt”



```
7_b.txt - Notepad
File Edit Format View
21 98 125 153
51 101 161 59
0 42 72 210
16 12 81 96
0.5 0.5
```

Output

```
===== Kalkulator Matrix =====
===== Main Menu =====
1.Sistem Persamaan Linier
2.Determinan
3.Matriks balikan
4.Interpolasi Polinom
5.Interpolasi Bicubic Spline
6.Regresi linier berganda
7.Keluar
Masukkan pilihan : 5
===== Interpolasi Bicubic Spline =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 7_b.txt
f(0.500000,0.500000) = 87.796875
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
|
```

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

Isi “7_c.txt”

7_c.txt - Notepac

File Edit Format

```
21 98 125 153
51 101 161 59
0 42 72 210
16 12 81 96
0.25 0.75
```

Output

```
===== Kalkulator Matrix =====
===== Main Menu =====
1.Sistem Persamaan Linier
2.Determinan
3.Matriks balikan
4.Interpolasi Polinom
5.Interpolasi Bicubic Spline
6.Regresi linier berganda
7.Keluar
Masukkan pilihan : 5
===== Interpolasi Bicubic Spline =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 7_c.txt
f(0.250000,0.750000) = 117.732178
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
```

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

Isi “7_d.txt”

7_d.txt - Notepad

File Edit Format V

```
21 98 125 153
51 101 161 59
0 42 72 210
16 12 81 96
0.1 0.9
```

Output

```
===== Kalkulator Matrix =====
===== Main Menu =====
1.Sistem Persamaan Linier
2.Determinan
3.Matriks balikan
4.Interpolasi Polinom
5.Interpolasi Bicubic Spline
6.Regresi linier berganda
7.Keluar
Masukkan pilihan : 5
===== Interpolasi Bicubic Spline =====
===== Metode Input =====
1.Keyboard
2.File txt
Masukkan pilihan : 2
=====Proses membaca file=====
1.Nama file (pastikan file di folder test)
2.Full path
Pilihan (1/2) : 1
Masukkan nama file (dengan extension) : 7_d.txt
f(0.100000,0.900000) = 128.575187
Apakah ingin simpan hasil ke dalam file ?
1.Iya
2.Tidak
```

Bab V

Kesimpulan

5.1 Kesimpulan

Melalui Tugas Besar 1 IF2123 Aljabar Linier dan Geometri Sistem Persamaan Linear, Determinan, dan Aplikasinya, kami dapat mengetahui bahwa Sistem Persamaan Linear dapat diselesaikan dengan berbagai metode, seperti metode Gauss, Gauss-Jordan, matriks balikan, dan *Cramer*. Terdapat tiga kemungkinan dari hasil penyelesaian Sistem Persamaan Linear, yaitu solusi tunggal, solusi banyak/solusi parametrik, atau tidak memiliki solusi. Selain itu, kamu juga mengetahui dan dapat menerapkan metode penyelesaian Sistem Persamaan Linear dapat untuk menyelesaikan persmasalahan interpolasi polinomial, regresi linear berganda, dan *Bicubic Spline Interpolation*.

5.2 Saran

- 5.2.1 Agar tampilan lebih menarik, sebaiknya menggunakan GUI.
- 5.2.2 Mungkin pada program dapat digunakan semacam *clear screen* supaya hasil yang telah keluar tidak terlalu menumpuk.

5.3 Komentar

- 5.3.1 Pada spesifikasi bagian bonus sebenarnya spesifikasinya lumayan cukup akan tetapi akan lebih baik jika spesifikasinya lebih lengkap lagi.

5.4 Refleksi

Secara keseluruhan, tugas besar pertama ini membantu kami untuk belajar mengatur waktu dan energi. Ditengah gempuran kuis, praktikum, dan pekerjaan lain, kami diajarkan untuk mengatur waktu dan strategi yang lebih baik lagi agar kami dapat melewati dan menyelesaikan semuanya dengan semaksimal mungkin. Selain itu, kami juga belajar untuk melakukan eksplorasi secara mandiri dan melatih kemampuan dalam mencari suatu informasi dari segala sumber yang ada. Tentunya kami juga belajar untuk berkomunikasi dan berkoordinasi dengan anggota kelompok kami.

Daftar Referensi

Belajar Java [Dasar] - 67 - Project CRUD (part 7) - Finalisasi dengan Multi File. 2019. Diakses pada 22 September 2023 dari <https://youtu.be/tnzKIYoke1U?si=ebXkwmyzzotfFOZt>

Cubic interpolation and resampling | Image processing. 2021. Diakses pada 5 Oktober 2023 dari <https://www.youtube.com/watch?v=syH8ASkotFg>

Fajri, Dwi Latifatul. 2023. Memahami Metode Eliminasi Gauss dan Pembahasan Soal. Diakses pada 21 September 2023 dari <https://katadata.co.id/intan/lifestyle/63f8b44a67a6f/memahami-metode-eliminasi-gauss-dan-pembahasan-soal>

Metode Eliminasi Gauss, Gauss-Jordan, Sarrus dan Cramer. 2018. Diakses pada 21 September 2023 dari https://ameliadyn27.blogspot.com/2018/04/metode-eliminasi-gauss-gauss-jordan_20.html

Pengertian Minor, Kofaktor, Matriks Kofaktor, dan Adjoin Matriks. 2023. Diakses pada 21 September 2023 dari <https://www.madematika.id/2017/08/pengertian-minor-kofaktor-matriks.html>