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

Oleh:

Kelompok 33 - Jawa adalah Koentji

Ferdinand Gabe Tua Sinaga	13523051
Henry Filberto Shenelo	13523108
Muhammad Iqbal Haidar	13522111

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

2024

Daftar Isi

Daftar Isi....2
Daftar Tabel....3
Daftar Gambar....4
BAB I
DESKRIPSI MASALAH....5
I. Sistem Persamaan Linier (SPL)....5
II. Interpolasi Polinomial....5
III. Regresi Berganda....7
IV. Bicubic Spline Interpolation....8
V. Image Resizing and Stretching....10
BAB II
TEORI SINGKAT....12
2.1 Operasi Baris Elementer....12
2.2 Metode Eliminasi Gauss....12
2.3 Metode Eliminasi Gauss-Jordan....12
2.4 Determinan....12
2.5 Matriks Balikan....13
2.6 Matriks Kofaktor....14
2.7 Matriks Adjoin....14
2.8 Kaidah Cramer....15
2.9 Interpolasi Polinomial....15
2.10 Bicubic Spline Interpolation....16
2.11 Regresi Linear dan Kuadratik Berganda....16
BAB III
IMPLEMENTASI....18
3.1 Folder Library....18
a. Class Matrix....18
b. Class OperasiDasarMatrix....19
c. Class EliminasiGaus....22
d. Class gaussjordan....23
e. Class SPL....25
f. Class Determinan....26
g. Class MatriksBalikan....27
3.2 Folder AplikasiSPL....27
a. Class InterpolasiPolinomial....27
b. Class RegresiBerganda....29
c. Class BicubicInterpolation....30
d. Class Imageresizer....32
BAB IV
EKSPERIMEN....35
4.1 Tampilan Awal....35
4.2 Pilihan Masukan....35
4.3 Studi Kasus Sistem Persamaan Linier....36

4.4 Studi Kasus Determinan....39
4.5 Studi Kasus Matriks Balikan....40
4.6 Studi Kasus Interpolasi Polinomial....41
4.7 Studi Kasus Regresi Linear dan Kuadratik Berganda....44
4.8 Studi Kasus Interpolasi Bicubic Spline....45
BAB V
KESIMPULAN....48
5.1 Kesimpulan.... 48
5.2 Saran....48
5.3 Refleksi.... 48
LAMPIRAN.... 49

Daftar Tabel

Tabel 3.1.1.1 Atribut Class Matriks 19
Tabel 3.1.1.2 Konstruktor Class Matriks 19
Tabel 3.1.1.3 Metode Class Matriks 19
Tabel 3.1.2.1 Atribut Class OperasiDasarMatrix 20
Tabel 3.1.2.2 Konstruktor Class OperasiDasarMatriks 20
Tabel 3.1.2.3 Metode Class OperasiDasarMatriks 20
Tabel 3.1.3.1 Atribut Class EliminasiGaus 23
Tabel 3.1.3.2 Konstruktor Class EliminasiGaus 24
Tabel 3.1.3.3 Metode Class EliminasiGaus 24
Tabel 3.1.4.1 Atribut Class gaussjordan 24
Tabel 3.1.4.2 Konstruktor Class gaussjordan 25
Tabel 3.1.4.3 Metode Class gaussjordan 25
Tabel 3.1.5.1 Konstruktor Class SPL 26
Tabel 3.1.5.2 Metode Class SPL 26
Tabel 3.1.6.1 Atribut Class Determinan 27
Tabel 3.1.6.2 Konstruktor Class Determinan 27
Tabel 3.1.6.3 Metode Class Determinan 27
Tabel 3.1.7.1 Konstruktor Class MatriksBalikan 28
Tabel 3.1.7.2 Metode Class MatriksBalikan 28
Tabel 3.2.1.1 Atribut Class InterpolasiPolinomial 28
Tabel 3.2.1.2 Konstruktor Class InterpolasiPolinomial 29
Tabel 3.2.1.3 Metode Class InterpolasiPolinomial 29
Tabel 3.2.2.1 Konstruktor Class RegresiBerganda 30
Tabel 3.2.2.2 Metode Class RegresiBerganda 30
Tabel 3.2.3.1 Atribut Class BicubicInterpolation 31
Tabel 3.2.3.2 Konstruktor Class BicubicInterpolation 32
Tabel 3.2.3.3 Metode Class BicubicInterpolation 32
Tabel 3.2.4.1 Atribut Class Imageresizer 33
Tabel 3.2.4.2 Konstruktor Class Imageresizer 34
Tabel 3.2.4.3 Metode Class Imageresizer 34
Tabel 4.3.1 Studi Kasus Sistem Persamaan Linear Metode Gauss 37
Tabel 4.3.2 Studi Kasus Determinan Metode Kofaktor 41
Tabel 4.4 Studi Kasus Interpolasi Polinomial 43
Tabel 4.5 Studi Kasus Nilai f(x) Interpolasi Polinomial 44
Tabel 4.6 Contoh Hasil Bicubic Spline Interpolation 46
Tabel 4.7 Hasil Image Resizer 47

Daftar Gambar

Gambar 1. Eliminasi Gauss dan Gauss Jordan dengan Matriks Eselon Baris dan Tereduksi 6
Gambar 2. Ilustrasi beberapa titik yang diinterpolasi secara polinomial 7
Gambar 3. Persamaan Umum Regresi Linear. 8
Gambar 4. Normal Estimation Equation for Multiple Linear Regression. 8
Gambar 5. Matriks regresi kuadratik berganda. 9
Gambar 6. Pemodelan interpolasi *bicubic spline* 9
Gambar 7. Persamaan Turunan Disekitar Titik 10
Gambar 8. Matriks Koefisien Interpolasi. 10
Gambar 9. Nilai fungsi yang akan diinterpolasi pada titik merah, turunan berarah terhadap sumbu x, terhadap sumbu y, dan keduanya (kiri ke kanan) 11
Gambar 10. Turunan di sekitar titik Image. 11
Gambar 11. Matriks Koefisien ImageResize. 12
Gambar 12. Sebuah citra gambar asal (kiri) dan hasil pemrosesan gambar dengan skala 1.5 pada *width* dan skala 2 pada *height* (kanan). 12
Gambar 13. Ilustrasi interpolasi polinomial. 16

BAB I
DESKRIPSI MASALAH

I. Sistem Persamaan Linier (SPL)

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Andstrea 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).

$$\begin{bmatrix} 0 & \mathbf{2} & 1 & -1 \\ 0 & 0 & \mathbf{3} & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \qquad \begin{bmatrix} 0 & \mathbf{1} & 0 & -\frac{2}{3} \\ 0 & 0 & \mathbf{1} & \frac{1}{3} \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

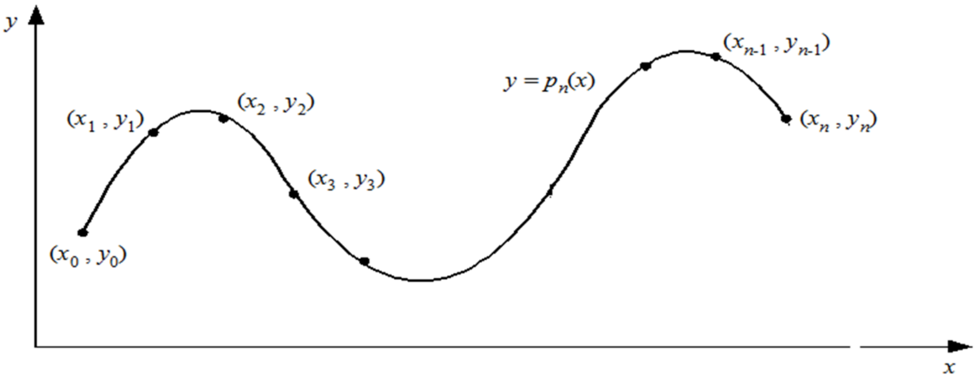
Gambar 1. Eliminasi Gauss dilakukan dengan matriks eselon baris dan eliminasi Gauss-Jordan dengan matriks eselon baris tereduksi.

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/

II. Interpolasi Polinomial

Persoalan interpolasi polinom adalah sebagai berikut: Diberikan n+1 buah titik berbeda, (x_0, y_0), (x_1, y_1), ..., (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 $i = 0, 1, 2, \ldots, n$.



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), ..., (x_n, y_n). adalah berbentuk $p_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$. 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 persaman 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 $p_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$ untuk $i = 0, 1, 2, \ldots, n$, akan diperoleh n buah sistem persamaan lanjar dalam $a_0, a_1, a_2, \ldots, a_n$,

$$\begin{aligned}
a_0 + a_1x_0 + a_2x_0^2 + \ldots + a_n x_0^n &= y_0 \\
a_0 + a_1x_1 + a_2x_1^2 + \ldots + a_n x_1^n &= y_1 \\
\ldots \qquad\qquad & \ldots \\
a_0 + a_1x_n + a_2x_n^2 + \ldots + a_n x_n^n &= y_n
\end{aligned}$$

Solusi sistem persamaan lanjar ini, yaitu nilai a_0, a_1, …, 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

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

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

III. Regresi Berganda

Regresi (akan dipelajari lebih lanjut di Probabilitas dan Statistika) merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Pada tugas besar ini, anda diminta untuk membuat 2 jenis regresi yaitu Regresi Linier Berganda dan Regresi Kuadratik Berganda.

1. Regresi Linier Berganda

Meskipun sudah ada persamaan jadi untuk menghitung regresi linear sederhana, terdapat persamaan umum dari regresi linear yang bisa digunakan untuk regresi linear berganda, yaitu.

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

Gambar 3. Persamaan Umum Regresi Linear.

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

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

Gambar 4. Normal Estimation Equation for Multiple Linear Regression.

2. Regresi Kuadratik Berganda

Dalam kasus ini, proses mengubah data-data dalam regresi kuadratik berganda cukup berbeda dengan Regresi Linier Berganda. Bentuk persamaan dari regresi kuadratik ada 3, yaitu:

a. Variabel Linier: Variabel dengan derajat satu seperti X, Y, dan Z
b. Variabel Kuadrat: Variabel dengan derajat dua seperti X^2
c. Variabel Interaksi: 2 Variabel dengan derajat satu yang dikalikan dengan satu sama lain seperti XY, YZ, dan XZ

Setiap n-peubah, jumlah variabel linier, kuadrat, dan interaksi akan berbeda-beda. Perhatikan contoh regresi kuadratik 2 variabel peubah sebagai berikut!

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

Gambar 5. Matriks regresi kuadratik berganda.

N menandakan jumlah peubah, terdapat 2 variabel linier yaitu u_i dan v_i, 2 variabel kuadrat yaitu u_i^2 dan v_i^2, dan 1 variabel interaksi yaitu uv. Untuk setiap n-peubah, akan terdapat 1 konstan N (Terlihat di bagian atas kiri gambar), n variabel linier, n variabel kuadrat, dan C_2^n variabel linier (dengan syarat n > 1). Tentu dengan bertambahnya peubah n, ukuran matriks akan bertumbuh lebih besar dibandingkan regresi linier berganda tetapi solusi tetap bisa didapat dengan menggunakan SPL.

Kedua model regresi yang dijadikan sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

IV. Bicubic Spline Interpolation

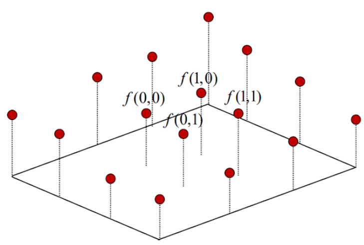
Bicubic spline interpolation adalah metode interpolasi yang digunakan untuk mengaproksimasi fungsi di antara titik-titik data yang diketahui. *Bicubic spline interpolation* melibatkan konsep *spline* dan konstruksi serangkaian polinomial kubik di dalam setiap sel segi empat dari data yang diberikan. 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 6. 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.

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

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

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

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

Gambar 7. Persamaan Turunan Disekitar Titik

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

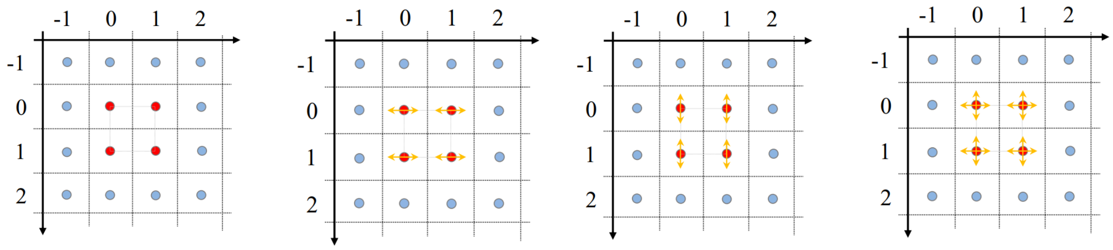
$$y = Xa$$

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

Gambar 8. Matriks Koefisien Interpolasi.

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 x 1^{1-1} x 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 x 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 9. Nilai fungsi yang akan diinterpolasi 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.

V. Image Resizing and Stretching

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.

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

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

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

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

Gambar 10. Turunan di sekitar titik Image.

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

$$
y = DI
$$

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

Gambar 11. Matriks Koefisien ImageResize.

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 12. Sebuah citra gambar asal (kiri) dan hasil pemrosesan gambar dengan skala 1.5 pada *width* dan skala 2 pada *height* (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 untuk *width* dan *height* pada gambar sesuai keinginan pengguna.

BAB II
TEORI SINGKAT

2.1 Operasi Baris Elementer

Metode Operasi Baris Elementer merupakan cara yang digunakan untuk mendapatkan solusi dari beberapa persamaan linear atau invers yang nantinya banyak dipakai untuk mencari determinan, interpolasi, regresi, dll. Operasi ini dilakukan terus menerus pada suatu matriks augmented hingga terbentuk matriks eselon baris atau matriks eselon baris tereduksi sesuai dengan solusi yang dihasilkan. Terdapat 3 tipe solusi yang dihasilkan jika matrix sudah berbentuk eselon yaitu, solusi unik (1 solusi), solusi banyak (lebih dari 1 solusi), dan tidak punya solusi. Beberapa OBE terhadap matrix yaitu:

a. Mengalikan sebuah baris dengan konstanta tidak nol.
b. Dua baris bertukar posisi.
c. Operasi tambah dan kurang suatu baris ke baris lainnya.

2.2 Metode Eliminasi Gauss

Metode Eliminasi Gaus adalah salah satu cara untuk menyelesaikan SPL (Sistem Persamaan Linear) yang memanfaatkan Operasi Baris Elementer (OBE) untuk merubah sebuah matriks menjadi matriks eselon baris. Setelah terbentuk matriks eselon baris, untuk menyelesaikan SPL tersebut diperlukan proses Back Substitution.

2.3 Metode Eliminasi Gauss-Jordan

Metode ini adalah pengembangan dari metode eliminasi Gauss. Operasi baris elementer (OBE) diterapkan pada matriks *augmented* sehingga menghasilkan matriks eselon baris tereduksi. Proses substitusi mundur tidak lagi diperlukan untuk mendapatkan nilai-nilai variabel karena nilai variabel langsung diperoleh dari matriks augmented akhir (jika solusinya tunggal).

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

Metode ini terdiri dari dua fase:

a. Fase maju (forward phase) atau fase eliminasi Gauss - Menghasilkan nilai-nilai 0 di bawah satu utama (*leading one)*
b. Fase mundur (backward phase) - Menghasilkan nilai-nilai 0 di atas satu utama (*leading one)*

2.4 Determinan

Determinan matriks A dapat dilambangkan dengan

$$\det(A) = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix}$$

Determinan berguna dalam mencari balikan matriks maupun menyelesaikan permasalahan SPL. Terdapat dua cara mencari determinan sebuah matriks:

a. Metode reduksi baris (OBE)

Pada metode ini, dilakukan operasi baris elementer untuk mendapatkan matriks segitiga atas atau bawah, yaitu matriks dengan semua elemen di atas (matriks segitiga atas) atau matriks di bawah (segitiga bawah) diagonal utama bernilai 0. Setelah itu, semua elemen pada diagonal utama dikalikan.

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

Terdapat aturan pada saat melakukan OBE, yaitu:

1. Jika mengalikan baris dengan k, maka hasil determinan harus dibagi k
2. Jika terdapat pertukaran baris, maka hasil determinan harus dikali $(-1)^n$ dengan n adalah banyak pertukaran baris.
3. Jika suatu baris ditambah k baris lain, hasil determinan tetap.

b. Metode ekspansi kofaktor

Metode kofaktor adalah salah satu cara untuk mencari determinan dari matriks berukuran (n×n) dengan cara mendekompose matriks secara rekursif menjadi submatriks yang lebih kecil. Pada metode ini determinan dihitung dengan memilih salah satu baris atau kolom sebagai acuan atau titik awal dari ekspansi. Setiap elemen pada baris atau kolom tersebut dikalikan dengan determinan dari submatriks yang terbentuk setelah baris dan kolom terkait dihapus. Selain itu, setiap elemen diberi tanda positif atau negatif secara bergantian sesuai dengan aturan $(-1)^{i+j}$. Proses ini diulang secara rekursif hingga mencapai kasus basis, yaitu matriks berukuran 1×1 atau 2×2 yang dapat dihitung langsung.

$$A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

$$\det(A) = a \cdot \det\begin{pmatrix} e & f \\ h & i \end{pmatrix} - b \cdot \det\begin{pmatrix} d & f \\ g & i \end{pmatrix} + c \cdot \det\begin{pmatrix} d & e \\ g & h \end{pmatrix}$$

2.5 Matriks Balikan

Khusus matriks persegi, matriks dapat memiliki matriks balikan/inversnya. Matriks yang tidak memiliki balikan disebut sebagai matriks singular. Dalam matriks, ketika suatu matriks dikalikan dengan matriks balikannya, akan menghasilkan matriks identitas, yaitu matriks persegi yang nilai diagonal utamanya adalah satu dan nilai lainnya adalah nol.

Misalkan, A adalah sebuah matriks berukuran n×n dan balikannya adalah A^{-1} , maka $AA^{-1} = A^{-1}A = I$ Ada dua metode untuk mencari balikan suatu matriks berukuran n×n:

a. Metode eliminasi Gauss-Jordan.

Pada metode ini akan dilakukan dengan cara melakukan metode eliminasi Gauss Jordan pada matriks A dan matriks identitas seperti:

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

akan diperoleh matriks balikan (A^{-1}) pada sisi kanan matriks.

b. Metode matriks adjoin.

Pada metode ini perlu dicari terlebih dahulu matriks adjoin dari matriks asal. Lalu balikan matriks asal diperoleh dengan mengalikan matriks adjoin dengan 1/det matriks asal.

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

2.6 Matriks Kofaktor

Kofaktor adalah nilai determinan sebuah upa-matriks yang dimana elemen-elemennya bukan berada pada baris i dan kolom j. Rumus kofaktor adalah

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

M_{ij} = minor entri a_{ij}
= determinan upa-matriks (*submatrix*) yang elemen-elemennya tidak berada pada baris i dan kolom j

Matriks kofaktor adalah matriks yang dibentuk dari kofaktor tiap-tiap elemen sebuah matriks. Posisi elemen pada matriks kofaktor bersesuaian dengan indeks kofaktornya.

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

2.7 Matriks Adjoin

Matriks adjoin adalah hasil transpose dari sebuah matriks kofaktor.

$$\mathrm{adj}(A) = (\mathrm{K})^T$$

Di mana:

- K adalah matriks kofaktor dari matriks A,

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

2.8 Kaidah Cramer

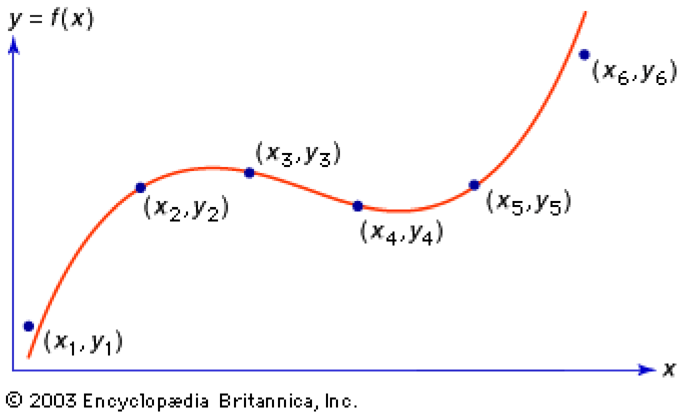
Jika diberikan suatu Sistem Persamaan Linear (SPL) Ax = **b** dengan n persamaan dan n peubah yang memiliki solusi unik sehingga det(A) ≠ 0. Maka solusi SPL dapat dicari dengan

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

dimana A□ adalah matriks A yang kolom ke-j nya diganti dengan entri dari matriks **b.**

2.9 Interpolasi Polinomial

Interpolasi polinomial adalah metode yang dipakai untuk menaksir nilai pada suatu titik dengan cara mengambil beberapa sebanyak (n+1) titik dan melakukan interpolasi sehingga diperoleh suatu persamaan polinomial $p_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$. Hal ini dapat dipakai untuk menaksir nilai y pada selang $[x_0, x_n]$



Gambar 13. Ilustrasi interpolasi polinomial.

Untuk mendapatkan koefisien dari setiap suku di polinomial $p_n(x)$, dapat dilakukan substitusi titik pada $p_n(x)$ sehingga diperoleh

$$\begin{aligned} a_0 + a_1x_0 + a_2x_0^2 + \ldots + a_n x_0^n &= y_0 \\ a_0 + a_1x_1 + a_2x_1^2 + \ldots + a_n x_1^n &= y_1 \\ \ldots \qquad\qquad & \ldots \\ a_0 + a_1x_n + a_2x_n^2 + \ldots + a_n x_n^n &= y_n \end{aligned}$$

dengan $(x_n y_n)$ berasal dari titik-titik yang diberikan. Dari persamaan ini, dapat diselesaikan seperti SPL pada umumnya dan diperoleh nilai setiap koefisien suku polinomial p_n . Nilai y untuk sembarang x di rentang $[x_0, x_n]$ dapat ditaksir dengan mensubstitusi x ke persamaan $p_n(x)$ yang telah didapat.

2.10 Bicubic Spline Interpolation

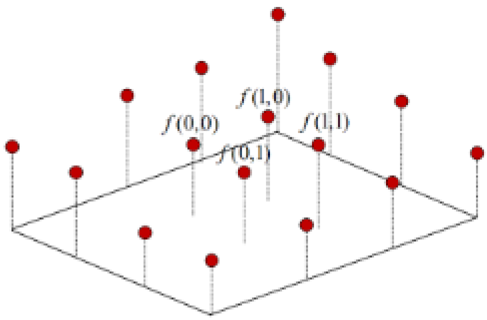
Bicubic Spline Interpolation adalah salah satu metode interpolasi 2D yang menggunakan 16 titik di sekitar area utama yang akan di interpolasi. Perbedaan utama antara Bicubic Spline Interpolation dan Bicubic interpolation biasa adalah digunakannya polinomial kubik dalam dua arah (x dan y), di mana koefisiennya ditentukan berdasarkan nilai titik dan turunannya.

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

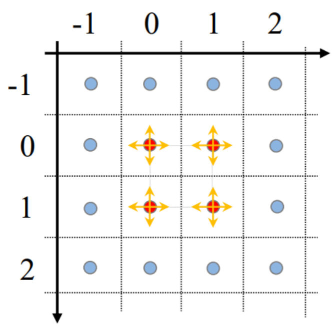
$$f_x(x,y) = \sum_{j=0}^{3}\sum_{i=1}^{3} a_{ij} i x^{i-1} y^j$$

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

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



Dengan menggunakan polinomial tersebut kita akan mendapatkan hasil interpolasi yang lebih halus disekitar titik interpolasi dengan range [0..1]. Dengan memanfaatkakan konsep ini kita bisa melakukan Image resizing pada sebuah gambar



2.11 Regresi Linear dan Kuadratik Berganda

Regresi digunakan untuk memahami hubungan antara satu variabel dependen (terikat) dan satu atau lebih variabel independen (bebas). Tujuannya untuk menemukan model atau persamaan yang bisa memprediksi nilai variabel dependen berdasarkan variabel independen. Apabila regresi melibatkan lebih dari satu variabel independen, maka regresi tersebut adalah regresi berganda.

Regresi linear mengasumsikan hubungan variabel dependen dengan variabel independen bersifat linear, artinya model atau persamaan digambarkan dengan garis lurus. Persamaan umum regresi linear adalah

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

Regresi kuadratik mengasumsikan hubungan variabel dependen dengan variabel independen bersifat non-linear, artinya model atau persamaan digambarkan dengan pola melengkung atau parabola.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_2^2 + \beta_5 X_1 X_2 + \epsilon$$

Contoh persamaan regresi kuadratik dua variabel

BAB III
IMPLEMENTASI

3.1 Folder Library

a. Class Matrix

Class Matrix digunakan untuk membuat dan mendefinisikan matrix

1. Atribut

Tabel 3.1.1.1 Atribut Class Matriks

Atribut	Deskripsi
ROW_CAP	Konstanta yang mendefinisikan jumlah baris maksimum (1000).
COL_CAP	Konstanta yang mendefinisikan jumlah kolom maksimum (1000).
rowEff	Integer yang menyimpan jumlah baris efektif dalam matriks.
colEff	Integer yang menyimpan jumlah kolom efektif dalam matriks.
m	Array dua dimensi yang menyimpan elemen-elemen matriks dengan ukuran maksimal.

2. Konstruktor

Tabel 3.1.1.2 Konstruktor Class Matriks

Konstruktor	Deskripsi
Matrix()	Konstruktor default untuk kelas

3. Metode

Tabel 3.1.1.3 Metode Class Matriks

Metode	Deskripsi
get_ROW_EFF()	Mengembalikan jumlah baris

	efektif yang disimpan dalam atribut rowEff.
get_COL_EFF()	Mengembalikan jumlah kolom efektif yang disimpan dalam atribut colEff
get_ELMT(int i, int j)	Mengembalikan elemen matriks di baris i dan kolom j
set_ROW_EFF(int nRow)	Mengubah nilai jumlah baris efektif menjadi nRow
set_COL_EFF(int nCol)	Mengubah nilai jumlah kolom efektif menjadi nCol
set_ELMT(int i, int j, double x)	Mengubah elemen pada baris i dan kolom j menjadi x

b. Class OperasiDasarMatrix

Class OperasiDasarMatrix membuat dan mengolah matrix yang dipakai untuk memecahkan berbagai permasalahan.

1. Atribut

Tabel 3.1.2.1 Atribut Class OperasiDasarMatrix

Atribut	Deskripsi
ROW_CAP	Konstanta yang mendefinisikan jumlah baris maksimum (1000).
COL_CAP	Konstanta yang mendefinisikan jumlah kolom maksimum (1000).

2. Konstruktor

Tabel 3.1.2.2 Konstruktor Class OperasiDasa Matriks

Konstruktor	Deskripsi
OperasiDasarMatrix()	Konstruktor default untuk kelas

3. Metode

Tabel 3.1.2.3 Metode Class OperasiDasarMatriks

Metode	Deskripsi

get_ROW_EFF()	Mengembalikan jumlah baris efektif yang disimpan dalam atribut rowEff.
createMatrix(Matrix m, int nRows, int nCols)	Mengatur jumlah baris dan kolom yang efektif untuk matriks yang diberikan.
inputvalid(String angka)	Memvalidasi apakah string yang diberikan merupakan angka yang valid (hanya digit).
readMatrix(Matrix m, int nRow, int nCol)	Membaca elemen matriks dari input keyboard.
inputfilebicubic(String filename, Matrix m)	Membaca file dan mengisi matriks sambil mengekstrak nilai x dan y
readMatrixFile(String filename, Matrix m)	Membaca matriks dari file dan mengisi objek matriks.
readMatrixFileInterpolate(String filename, Matrix m)	Membaca matriks untuk interpolasi dari file.
Matrix readSPL()	Membaca sistem persamaan linear (SPL) dari input standar.
Matrix readRegresi()	Membaca data regresi dari input keyboard.
Matrix readSPLAug()	Membaca SPL augmented dari input keyboard.
displayMatrix(Matrix m)	Menampilkan elemen-elemen dari matriks m ke konsol. Setiap elemen dipisahkan oleh spasi, dan setiap baris dipisahkan dengan baris baru.
displayMatrixtoFile(Matrix m, String filename)	Menyimpan elemen-elemen dari matriks m ke dalam file dengan nama yang ditentukan oleh filename. Elemen dipisahkan oleh spasi, dan setiap baris matriks dituliskan pada baris baru dalam file.
isMatrixIdxValid(int i, int j)	Memeriksa apakah indeks i dan j valid untuk matriks berdasarkan kapasitas yang ditentukan

	(ROW_CAP dan COL_CAP).
getLastIdxRow(Matrix m)	Mengembalikan indeks baris terakhir dari matriks m, yang dihitung sebagai jumlah baris efektif dikurangi satu.
getLastIdxCol(Matrix m)	Mengembalikan indeks kolom terakhir dari matriks m, yang dihitung sebagai jumlah kolom efektif dikurangi satu.
isIdxEff(Matrix m, int i, int j)	Memeriksa apakah indeks i dan j merupakan indeks efektif dalam matriks m, dengan memastikan bahwa keduanya berada dalam rentang indeks yang valid.
getElmtDiagonal(Matrix m, int i)	Mengembalikan elemen diagonal dari matriks m pada indeks i, yaitu elemen di posisi (i,i).
copyMatrix(Matrix mIn)	Membuat salinan dari matriks mIn dan mengembalikannya. Matriks baru akan memiliki ukuran yang sama dan elemen yang sama dengan mIn
addMatrix(Matrix m1, Matrix m2)	Menambahkan dua matriks m1 dan m2, kemudian mengembalikan matriks hasil penjumlahan. Ukuran kedua matriks harus sama
subtractMatrix(Matrix m1, Matrix m2)	Mengurangi matriks m2 dari matriks m1 dan mengembalikan hasilnya sebagai matriks baru. Ukuran kedua matriks harus sama.
multiplyMatrix(Matrix m1, Matrix m2)	Mengalikan matriks m1 dengan matriks m2 dan mengembalikan hasilnya sebagai matriks baru. Ukuran matriks harus sesuai untuk operasi perkalian.
multiplyMatrixWithMod(Matrix m1, Matrix m2, int mod)	Mengalikan matriks m1 dan m2, kemudian menerapkan modulus dengan nilai mod pada hasil perkalian, dan mengembalikan

	matriks hasil.
multiplyByConst(Matrix m, double x)	Mengalikan setiap elemen matriks m dengan konstanta x, kemudian mengembalikan matriks hasil.
isMatrixSizeEqual(Matrix m1, Matrix m2)	Memeriksa apakah ukuran matriks m1 dan m2 sama.
isMatrixEqual(Matrix m1, Matrix m2)	Memeriksa apakah dua matriks, m1 dan m2, sama baik dari ukuran maupun nilai elemennya.
countElmt(Matrix m)	Menghitung jumlah elemen dalam matriks m.
isSquare(Matrix m)	Memeriksa apakah matriks m memiliki baris dan kolom yang sama.
isSymmetric(Matrix m)	Memeriksa apakah matriks m simetris
isIdentity(Matrix m)	Memeriksa apakah matriks m adalah matriks identitas.
isSparse(Matrix m)	Menentukan apakah matriks m elemennya mayoritas 0.
negation(Matrix m)	Menghasilkan negasi dari matriks m.
mergeMatrix(Matrix m1, Matrix m2)	Menggabungkan dua matriks m1 dan m2 menjadi satu matriks baru

c. **Class EliminasiGaus**

Class EliminasiGaus merupakan implementasi Gauss untuk menyelesaikan SPL dengan mengubahnya ke matriks eselon baris dan melakukan backsubstitution untuk menyelsaikan SPL itu.

1. Atribut

Tabel 3.1.3.1 Atribut Class EliminasiGaus

Atribut	Deskripsi
OperasiDasarMatrix	Objek yang digunakan untuk

	melakukan operasi dasar pada matriks, seperti penambahan, pengurangan, dan pengalian.

2. Konstruktor

Tabel 3.1.3.2 Konstruktor Class EliminasiGaus

EliminasiGaus()	Konstruktor default untuk kelas EliminasiGaus

3. Metode

Tabel 3.1.3.3 Metode Class EliminasiGaus

searchindexnonzero(Matrix matriks, int index, int rownow)	Mencari baris non-nol pertama di suatu kolom mulai dari baris tertentu.
backsubsperfected(Matrix m)	Melakukan metode back substitution dan menangani kasus solusi parametris dan solusi unik. Sekaligus melakukan print terhadap hasilnya
backsubsV2(Matrix matriks)	Melakukan substitusi mundur untuk mencari solusi dari sistem persamaan linier dan memasukkan hasilnya ke sebuah array
GausMethod(Matrix matriks)	Melakukan proses eliminasi Gauss untuk mengubah matriks menjadi bentuk eselon baris, termasuk pivoting dan penukaran baris.

d. Class gaussjordan

Class gaussjordan merupakan implementasi Gauss-Jordan untuk menyelesaikan SPL dengan mengubahnya ke matriks eselon baris tereduksi. Kelas ini juga menghitung balikan dan determinan matriks memanfaatkan metode Gauss-Jordan.

1. Atribut

Tabel 3.1.4.1 Atribut Class gaussjordan

Atribut	Deskripsi

OperasiDasarMatrix	Objek yang digunakan untuk melakukan operasi dasar pada matriks, seperti penambahan, pengurangan, dan pengalian.

2. Konstruktor

Tabel 3.1.4.2 Konstruktor Class gaussjordan

Konstruktor	Deskripsi
gaussjordan()	Konstruktor dari kelas gaussjordan. Menginisialisasi objek OperasiDasarMatrix untuk digunakan dalam operasi matriks

3. Metode

Tabel 3.1.4.3 Metode Class gaussjordan

Metode	Deskripsi
SubtractRow(Matrix m, double factor, int pivotRow, int targetRow)	Mengurangi targetRow dari pivotRow dengan faktor tertentu.
MultiplyRow(Matrix m, double x, int i)	Mengalikan seluruh elemen pada baris i dengan bilangan x
DivideRow(Matrix m, double x, int i)	Membagi seluruh elemen pada baris i dengan bilangan x
SwapRow(Matrix m, int row1, int row2)	Menukar dua baris row1 dan row2 dalam matriks m
ToEchelon(Matrix m)	Mengubah matriks m ke bentuk echelon menggunakan metode Gauss.
ToEchelonRed(Matrix m)	Mengubah matriks m ke bentuk echelon yang tereduksi.
findPivot(Matrix m, int row)	Mencari kolom pivot pada baris tertentu row. Mengembalikan indeks kolom pivot jika ditemukan, atau -1 jika tidak ada
solveSPL(Matrix m)	Menyelesaikan sistem persamaan linear (SPL) menggunakan metode Gauss-Jordan. Mengembalikan array string yang berisi solusi dari SPL.

solveSPL2(Matrix m)	Menyelesaikan sistem persamaan linear (SPL) menggunakan metode Gauss-Jordan. Mengembalikan array double yang berisi solusi unik dari SPL.
DeterminantOBE(Matrix m)	Menghitung determinan matriks m menggunakan metode operasi baris elementer.
MatriksBalikan(Matrix m)	Menghitung matriks invers dari matriks m menggunakan metode Gauss-Jordan.

e. Class SPL

Kelas SPL merupakan implementasi penyelesaian SPL, metode utamanya yang tersedia adalah invers matriks dan kaidah cramer.

1. Atribut

   Tidak ada atribut

2. Konstruktor

   Tabel 3.1.5.1 Konstruktor Class SPL

Konstruktor	Deskripsi
SPL()	Konstruktor default untuk kelas

3. Metode

   Tabel 3.1.5.2 Metode Class SPL

Metode	Deskripsi
SPLCramer(Matrix matriksAug)	Menerima masukkan SPL berbentuk matriks augmented, mengembalikan solusi dalam bentuk array satu dimensi yang dihitung dengan kaidah cramer
SPLInverse(Matrix matriksAug)	Menerima masukkan SPL berbentuk matriks augmented, mengembalikan solusi dalam bentuk array satu dimensi yang dihitung dengan metode invers
displaySPLCramer(double[]	Menerima solusi SPL,

solusi)	menampilkan solusi ke layar dengan format yang ditentukan
writeSPL(double[] solusi)	Menerima solusi SPL, menyimpan solusi kedalam file yang ditentukan

f. Class Determinan

Class Determinan adalah kelas tempat implementasinya fungsi determinan dengan ekspansi kofaktor

1. Atribut

Tabel 3.1.6.1 Atribut Class Determinan

OperasiDasarMatrix ODM	Digunakan untuk melakukan operasi dasar pada objek matriks seperti pembuatan matriks baru.

2. Konstruktor

Tabel 3.1.6.2 Konstruktor Class Determinan

Determinan()	Konstruktor default untuk kelas Determinan

3. Metode

Tabel 3.1.6.3 Metode Class Determinan

kofaktor(Matrix matriks)	Digunakan untuk menghitung determinan matriks dengan memecahnya menjadi kofaktor hingga ukuran matriks cukup kecil (1x1 atau 2x2). Jika matriks adalah 2x2 atau 1x1, maka hasil determinan dihitung langsung
createSubMatrix(Matrix matriks, int colExcluded)	Metode ini membuat submatriks dengan menghapus baris pertama dan kolom tertentu. Sehingga dihasilkan Matrix dengan ukuran submatriks adalah (n-1) x (n-1)

g. Class MatriksBalikan

Kelas MatriksBalikan merupakan implementasi penghitungan balikan, metode utamanya yang tersedia adalah invers dengan adjoin.

1. Atribut

   Tidak ada atribut

2. Konstruktor

   Tabel 3.1.7.1 Konstruktor Class MatriksBalikan

Konstruktor	Deskripsi
MatriksBalikan()	Konstruktor default untuk kelas

3. Metode

   Tabel 3.1.7.2 Metode Class MatriksBalikan

Metode	Deskripsi
matriksKofaktor(Matrix matriks)	Menerima sebuah matriks, mengembalikan matriks kofaktornya
transpose(Matrix matriks)	Menerima sebuah matriks, mengembalikan transpose dari matriks
inverseWithAdj(Matrix matriks)	Menerima sebuah matriks, mengembalikan balikan matriks yang dihitung dengan metode adjoin

3.2 Folder AplikasiSPL

a. Class InterpolasiPolinomial

Class InterpolasiPolinomial mengubah (n+1) titik menjadi matriks *augmented* dan menyelesaikannya seperti pada SPL untuk memperoleh setiap koefisien suku polinom yang kemudian disubstitusi untuk menaksir nilai y.

1. Atribut

   Tabel 3.2.1.1 Atribut Class InterpolasiPolinomial

Atribut	Deskripsi
Matrix m	*Instance* dari kelas Matrix, yang kemungkinan mewakili struktur

	matriks yang digunakan dalam interpolasi polinomial.
OperasiDasarMatrix operasi	*Instance* dari kelas OperasiDasarMatrix, digunakan untuk operasi dasar matriks seperti pembuatan dan manipulasi.
gaussjordan gj	*Instance* dari kelas gaussjordan, yang diperkirakan digunakan untuk melakukan eliminasi Gauss-Jordan dalam menyelesaikan sistem persamaan linier.

2. Konstruktor

Tabel 3.2.1.2 Konstruktor Class InterpolasiPolinomial

Konstruktor	Deskripsi
InterpolasiPolinomial()	Konstruktor default untuk kelas

3. Metode

Tabel 3.2.1.3 Metode Class InterpolasiPolinomial

Metode	Deskripsi
readPoint(Matrix m)	Meminta pengguna untuk memasukkan jumlah titik dan koordinatnya. Membuat dan mengembalikan matriks yang berisi titik-titik input.
ToMatrixInterpolasi(Matrix mIn)	Mengonversi matriks input dari titik-titik menjadi matriks yang sesuai untuk interpolasi polinomial.
SolveInterpolasi(Matrix m, boolean inputfile)	Menghitung nilai fungsi yang diinterpolasi pada nilai x yang ditentukan.

b. Class RegresiBerganda

Kelas RegresiBerganda menghitung persamaan regresi dengan lebih dari satu variabel independen terhadap variabel dependen. Pada kelas ini terdapat dua metode utama yakni regresi linear dan regresi berganda.

1. Atribut

   Tidak ada atribut

2. Konstruktor

   Tabel 3.2.2.1 Konstruktor Class RegresiBerganda

Konstruktor	Deskripsi
RegresiBerganda()	Konstruktor default untuk kelas

3. Metode

   Tabel 3.2.2.2 Metode Class RegresiBerganda

Metode	Deskripsi
RegresiLinear(Matrix matriksAug)	Menerima data sampel dalam bentuk matriks augmented, menjalankan keseluruhan prosedur regresi linear
RegresiKuadratik(Matrix matriksAug)	Menerima data sampel dalam bentuk matriks augmented, menjalankan keseluruhan prosedur regresi kuadratik
solveRegresiLinear(Matrix matriksAug)	Menerima data sampel dalam bentuk matriks augmented, mengembalikan koefisien persamaan regresi linear dalam bentuk array satu dimensi
displayRegresiLinear(double[] koefRegresi)	Menerima array koefisien regresi linear, menampilkan persamaan regresi linear ke layar
calculateYLinear(double[] koefRegresi)	Menerima array koefisien regresi linear, mengembalikan taksiran variabel dependen berdasarkan masukkan nilai variabel independen dari user
writeRegresiLinear(double[] koefRegresi, double taksiran)	Menerima array koefisien regresi linear & nilai taksiran, menyimpan kedalam file yang

	ditentukan
solveRegresiKuadratik(Matrix matriksAug)	Menerima data sampel dalam bentuk matriks augmented, mengembalikan koefisien persamaan regresi kuadratik dalam bentuk array satu dimensi
displayRegresiKuadratik(double[] coefficients, int n)	Menerima array koefisien regresi kuadratik & jumlah variabel independen, menampilkan persamaan regresi kuadratik ke layar
calculateYKuadratik(double[] coefficients, int n)	Menerima array koefisien regresi kuadratik & jumlah variabel independen, mengembalikan taksiran variabel dependen berdasarkan masukkan nilai variabel independen dari user
writeRegresiKuadratik(double[] koefRegresi, int n, double taksiran)	Menerima array koefisien regresi kuadratik & nilai taksiran & jumlah variabel independen, menyimpan kedalam file yang ditentukan

c. **Class BicubicInterpolation**

Class BicubicInterpolation adalah implementasi dari metode interpolasi bicubic, yang digunakan untuk memperkirakan nilai di antara titik-titik data pada grid dua dimensi dengan mempertimbangkan turunan disekitarnya

1. Atribut

Tabel 3.2.3.1 Atribut Class BicubicInterpolation

ODM	Instance untuk operasi dasar matriks seperti pembuatan dan perkalian matriks.
matriks	Matriks 4x4 untuk menyimpan input manual dari pengguna.
matriksCoeff	Matriks koefisien yang digunakan dalam perhitungan interpolasi.
arrayOfTitik	Array yang menyimpan titik-titik

	refrensi utama (x, y) di sekitar titik interpolasi.
inversematriks	Instance untuk melakukan inversi matriks.
Xmatriks	Matriks 16x16 untuk menyimpan perhitungan koefisien.
XmatrixDoneProces	Flag untuk memastikan Matriks koefisien dihitung sekali saja.

2. Konstruktor

Tabel 3.2.3.2 Konstruktor Class BicubicInterpolation

BicubicInterpolation()	Konstruktor default yang menginisialisasi objek.
Titik(double x, double y)	Konstruktor kelas internal Titik untuk membuat titik dengan koordinat (x, y).

3. Metode

Tabel 3.2.3.3 Metode Class BicubicInterpolation

getArrayOfTitik()	Mengembalikan array titik-titik yang digunakan dalam interpolasi.
pangkat(double base, int eksponen)	Menghitung pangkat dengan basis dan eksponen tertentu.
inputManualF()	Meminta input matriks 4x4 dari pengguna melalui terminal.
makeCoeff()	Menghitung koefisien interpolasi dan menyimpan hasilnya di matriks Xmatriks.
copyToXMatriks(Matrix matriksCoeff, Matrix Xmatriks, int row)	Menyalin koefisien dari matriksCoeff ke Xmatriks.
convert2DTo1Column(double[][] matriks)	Mengubah matriks 2D 4x4 menjadi vektor kolom 16x1.
convert1Column_To_2d_4x4(Matrix result)	Mengubah vektor kolom 16x1 menjadi matriks 2D 4x4.
isInRange(double x, double y)	Memeriksa apakah nilai (x, y)

	berada dalam rentang [0, 1].
getAll_a_value(Matrix matriksfile)	Menghitung nilai koefisien akhir dengan mengalikannya dengan matriks dari user dan mengubahnya menjadi bentuk 4x4.
SolutionBicubic(ArrayList<Double> xy, Matrix matriks)	Melakukan interpolasi dan menampilkan hasil interpolasi untuk titik yang diberikan

d. Class Imageresizer

Class Imageresizer adalah tempat implementasi fungsi untuk melakukan stretching dan resize gambar dengan memanfaatkan matriks koefisien dari bicubic interpolation untuk menghasilkan gambar yang halus.

1. Atribut

Tabel 3.2.4.1 Atribut Class Imageresizer

image	Menyimpan gambar yang akan diubah ukurannya.
bicubic	Instance untuk melakukan interpolasi bicubic.
ODM	Instance untuk operasi dasar matriks.
titikF	Array titik referensi untuk interpolasi.
titikI	Array titik yang digunakan untuk interpolasi di sekitar titik target.
Dmatriks	Matriks koefisien turunan untuk interpolasi.
MatrixforImage	Matrikskoefisien untuk gambar. Matriks ini hasil perkalian antara inverse Xmatriks dan DMatriks
MatrixforImageDoneProces	Flag apakah matriks koefisien gambar sudah pernah dihitung atau belum.
imagePixel	Matriks untuk menyimpan nilai piksel gambar (ARGB).

2. Konstruktor

Tabel 3.2.4.2 Konstruktor Class Imageresizer

Imageresizer()	Konstruktor default yang menginisialisasi objek dan atribut yang diperlukan.

3. Metode

Tabel 3.2.4.3 Metode Class Imageresizer

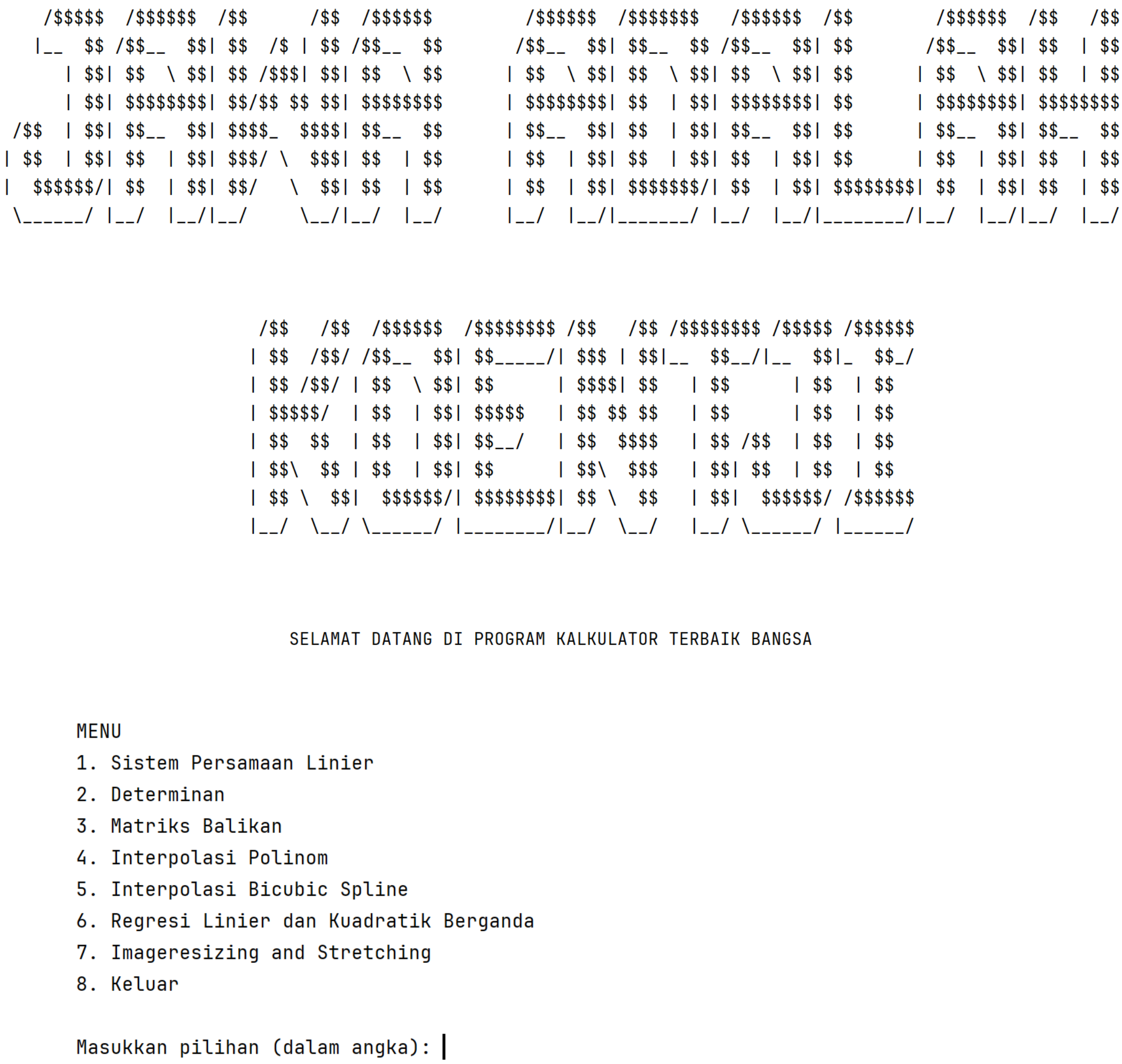
getImage()	Mendapatkan gambar yang sedang di proses
clamp(int value, int min, int max)	Memastikan nilai selalu diantara min dan max jika melebihi maka akan diambil min/max tergantung kedekatan nilainya.
initializeMatrixWithZero(Matrix matrix)	Mengisi semua elemen matriks dengan nilai nol.
inputImage(String imagePath)	Memuat gambar dari path yang diberikan dan membuat matriks piksel.
createimagePixel()	Membuat matriks piksel (ARGB) dari gambar yang diinput.
searchindexof(double x, double y)	Mencari indeks titik (x, y) di dalam array titikI.
get4x4(double[][] imageMatrix, int row, int col)	Mengambil blok 4x4 piksel di sekitar titik tertentu.
makeDMatrix()	Membuat matriks Dmatriks yang isinya koefisien turunan titik.
makeMatrixForImage()	Menghasilkan matriks untuk gambar dengan melakukan perkalian matriks interpolasi.
createVector16(double[][] block4x4)	Mengubah blok 4x4 piksel menjadi vektor kolom 16x1.
bicubicSplineInterpolate(double[][] coefficients, double deltaX, double deltaY)	Menghitung interpolasi bicubic menggunakan matriks koefisien 4x4.
resizer(double scalex, double	Mengubah ukuran gambar

scaley)	dengan skala tertentu menggunakan interpolasi bicubic.
mergeRGB(double[][][] resultImage, int width, int height)	Menggabungkan komponen ARGB dari hasil interpolasi menjadi gambar utuh.
displayImage(String imagePath)	Menampilkan gambar hasil resize.

BAB IV
EKSPERIMEN

4.1 Tampilan Awal

Pada tampilan awal, terdapat display awal program dan menu yang dapat dipilih



4.2 Pilihan Masukan

Pada setiap menu (kecuali *Imageresizing* and *Stretching*), terdapat pilihan untuk jenis masukan input (dari *keyboard* atau dari file).

```
--------------------------------------------------
PILIHAN METODE MASUKAN/INPUT
1. Input dari keyboard
2. Input dari file

Masukkan pilihan (dalam angka): 1
```

Untuk tampilan input dari file sama untuk setiap fungsi yaitu

```
Masukkan nama file input: 1a.txt
```

4.3 Studi Kasus Sistem Persamaan Linier

Jika dipilih nomor 1 pada menu, akan muncul pilihan metode penyelesaian

```
--------------------------------------
PILIHAN METODE PENYELESAIAN SPL
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer

Masukkan pilihan (dalam angka):
```

a. Metode Eliminasi Gauss

Jika pengguna memilih input dari keyboard, akan diminta input matriks A dan B (bentuk Ax = b) .

Tabel 4.3.1 Studi Kasus Sistem Persamaan Linear Metode Gauss

<table>
<tr>
<td>Jika Solusi Unik</td>
<td><pre>Masukkan banyak baris matriks A: 3
Masukkan banyak kolom matriks A: 3
Masukkan elemen-elemen matriks A:
-5 5 2
5 1 1
2 -5 -1
Masukkan elemen-elemen matriks B:
-30 12 21

1.0 -1.0 -0.4 6.0
0.0 1.0 0.5 -3.0
0.0 0.0 1.0 0.0

Solusi SPL tersebut adalah:
X1 = 3.0
X2 = -3.0
X3 = 0.0</pre></td>
</tr>
</table>

Jika Solusi Parametrik	Masukkan elemen-elemen matriks A: 1 -1 0 0 1 1 1 0 -3 0 2 -1 0 1 -1 -1 2 0 -2 -1 Masukkan elemen-elemen matriks B: 3 6 5 -1 Hasil Gaus: 1.0 0.0 -0.0 0.0 1.0 3.0 0.0 1.0 -0.0 0.0 2.0 0.0 0.0 0.0 0.0 1.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 Solusi untuk SPL mu: X1 = 3.0 + X5 X2 = 2.0X5 X4 = -1.0 + X5 X3 = Bebas X5 = Bebas
Jika Solusi Tidak ada	Masukkan banyak baris matriks A: 3 Masukkan banyak kolom matriks A: 3 Masukkan elemen-elemen matriks A: -5 -4 5 1 0 -1 3 5 -3 Masukkan elemen-elemen matriks B: -1 4 -2 1.0 0.8 -1.0 0.2 0.0 1.0 0.0 -1.0 0.0 0.0 0.0 3.0 Tidak ada solusi.

b. Metode Eliminasi Gauss-Jordan

Jika pengguna memilih input dari keyboard, akan diminta input matriks A dan B (bentuk Ax = b)

```
Masukkan banyak baris matriks A: 3
Masukkan banyak kolom matriks A: 3
Masukkan elemen-elemen matriks A:
1 2 3
4 5 6
7 8 9
Masukkan elemen-elemen matriks B:
1
2
3
```

Terdapat 3 kemungkinan hasil:

1. Solusi unik

```
Masukkan elemen-elemen matriks A:
1 2 3
0 1 -1
2 1 1
Masukkan elemen-elemen matriks B:
9 1 8

Hasil matriks eselon baris tereduksi:
1.0 0.0 0.0 2.625
0.0 1.0 0.0 1.875
-0.0 -0.0 1.0 0.875

Solusi SPL adalah
X1 = 2.625
X2 = 1.875
X3 = 0.875
```

2. Solusi banyak

```
Masukkan elemen-elemen matriks A:
1 -1 2 -1
2 1 -2 -2
-1 2 -4 1
3 0 0 -3
Masukkan elemen-elemen matriks B:
-1 -2 1 -3

Hasil matriks eselon baris tereduksi:
1.0 0.0 0.0 -1.0 -1.0
0.0 1.0 -2.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0

Solusi SPL adalah
X1 = -1.0  + X4
X2 = 0.0  + 2.0X3
X3 = Bebas
X4 = Bebas
```

3. Tidak ada solusi (contoh dengan input file)

```
Masukkan nama file input: 1a.txt

Hasil matriks eselon baris tereduksi:
1.0 0.0 0.0 0.6666666666666667 0.0
0.0 1.0 0.0 -2.6666666666666667 0.0
-0.0 -0.0 1.0 -1.0 0.0
0.0 0.0 0.0 0.0 1.0

 SPL tidak memiliki solusi
```

c. Metode Matriks Balikan

Input matriks pada metode matriks balikan ini berupa matriks augmented [x1 x2 x3 … xn y]

1. Contoh input keyboard & solusi unik

```
PILIHAN METODE PENYELESAIAN SPL
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer

Masukkan pilihan (dalam angka): 3
Masukan Berupa Matriks Augmented
Masukkan banyak baris: 4
Masukkan banyak kolom: 5

Masukkan Matriks Augmented
8 1 3 2 0
2 9 -1 -2 1
1 3 2 -1 2
1 0 6 4 3
x1 = -0.22432432432432434
x2 = 0.18243243243243243
x3 = 0.7094594594594594
x4 = -0.258108108108108
```

2. Contoh input file & solusi tidak unik

```
PILIHAN METODE PENYELESAIAN SPL
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer

Masukkan pilihan (dalam angka): 3
Masukkan nama file input: 1b.txt
Matriks tidak memiliki invers
```

d. Kaidah Cramer

Input matriks pada metode ini berupa matriks augmented [x1 x2 x3 … xn y] dengan n persamaan dan n peubah.

1. Contoh input file & solusi unik

```
PILIHAN METODE PENYELESAIAN SPL
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer

Masukkan pilihan (dalam angka): 4
Masukkan nama file input: 4.txt
x1 = 0.02030075187969925
x2 = -2.191729323308271
x3 = -1.1278195488721805
```

2. Contoh input keyboard & solusi tidak unik

```
PILIHAN METODE PENYELESAIAN SPL
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer

Masukkan pilihan (dalam angka): 4
Masukan Berupa Matriks Augmented
Masukkan banyak baris: 4
Masukkan banyak kolom: 5

Masukkan Matriks Augmented
1 1 -1 -1 1
2 5 -7 -5 -2
2 -1 1 3 4
5 2 -4 2 6
Determinan matriks sama dengan nol
```

4.4 Studi Kasus Determinan

Pada menu determinan, akan muncul tampilan pilihan metode penentuan determinan

```
------------------------------------
PILIHAN METODE PENENTUAN DETERMINAN
1. Metode reduksi baris
2. Metode ekspansi kofaktor

Masukkan pilihan (dalam angka):
```

a. Metode Reduksi Baris

Jika pengguna memilih input dari keyboard, akan diminta input matriks

```
Masukkan banyak baris matriks: 3
Masukkan banyak kolom matriks: 3
Masukkan elemen-elemen matriks:
1 2 3
4 5 6
7 8 9
```

Terdapat dua kemungkinan hasil:

1. Ada determinan

```
Masukkan elemen-elemen matriks:
1 2 3
4 5 6
7 8 9
Determinan matriks adalah -0.0
```

2. Tidak memiliki determinan

```
1 2 3
4 5 6
Matriks tidak memiliki determinan
```

b. Metode Ekspansi Kofaktor

Tabel 4.3.2 Studi Kasus Determinan Metode Kofaktor

Jika matriks memiliki Determinan	Jika tidak ada determinan
Masukkan pilihan (dalam angka): 2 Masukkan banyak baris matriks: 3 Masukkan banyak kolom matriks: 3 Masukkan elemen-elemen matriks: 4 1 3 7 5 8 1 2 1 Hasil Determinannya adalah -16.000000	Masukkan pilihan (dalam angka): 2 Masukkan banyak baris matriks: 2 Masukkan banyak kolom matriks: 3 Masukkan elemen-elemen matriks: 1 3 5 2 4 8 Matriks tidak memiliki determinan

4.5 Studi Kasus Matriks Balikan

Pada menu determinan, akan muncul tampilan pilihan metode penentuan determinan

```
----------------------------------------
PILIHAN METODE PENENTUAN BALIKAN/INVERSE
1. Metode eliminasi Gauss-Jordan
2. Metode adjoin

Masukkan pilihan (dalam angka):
```

a. Metode Eliminasi Gauss-Jordan

Pengguna akan diminta input matriks. Terdapat dua kemungkinan hasil:

1. Tidak memiliki balikan

```
Masukkan banyak baris matriks: 3
Masukkan banyak kolom matriks: 3
Masukkan elemen-elemen matriks:
1 2 3
4 5 6
7 8 9
Matriks tidak memiliki matriks balikan
```

2. Memiliki balikan

```
Masukkan elemen-elemen matriks:
2 1 -1
-3 -1 2
-2 1 2

Hasil matriks balikan:
4.0 3.0 -1.0
-2.0 -2.0 1.0
5.0 4.0 -1.0
```

b. Metode Adjoin

Input matriks pada metode adjoin merupakan matriks augmented [x1 x2 x3 … xn y]

1. Input keyboard & tidak ada balikan

```
PILIHAN METODE PENENTUAN BALIKAN/INVERSE
1. Metode eliminasi Gauss-Jordan
2. Metode adjoin

Masukkan pilihan (dalam angka): 2
Masukkan banyak baris matriks: 3
Masukkan banyak kolom matriks: 3
Masukkan elemen-elemen matriks:
1 2 3
4 5 6
7 8 9
Matriks tidak memiliki matriks balikan
```

2. Input file & ada balikkan

```
PILIHAN METODE PENENTUAN BALIKAN/INVERSE
1. Metode eliminasi Gauss-Jordan
2. Metode adjoin

Masukkan pilihan (dalam angka): 1
Masukkan nama file: 8b.txt

Hasil matriks balikan:
-3.961538461538462 1.9871794871794874 0.6025641025641025 -1.4871794871794874 0.37179487179487186
4.307692307692308 -2.435897435897436 -0.5128205128205128 1.4358974358974361 -0.35897435897435903
0.19230769230769218 -0.0641025641025641 0.012820512820512817 0.5641025641025641 -0.14102564102564102
-0.19230769230769218 0.0641025641025641 -0.012820512820512817 0.4358974358974359 0.14102564102564102
-0.8461538461538461 0.6153846153846154 0.07692307692307693 -0.6153846153846154 0.15384615384615385
```

4.6 Studi Kasus Interpolasi Polinomial

Pada menu ini, jika pengguna memilih input dari keyboard, akan muncul tampilan

```
Masukkan banyak titik: 4
Masukkan titik:
-1 -10
0 -4
1 -2
2 2
Masukkan nilai x yang akan ditaksir: 0,5
```

Untuk input dari file, jika file tidak sesuai, akan keluar “Matriks invalid”

```
Masukkan nama file: 5aii.txt
Matrix invalid
```

Isi file 5aii.txt:

```
0.1 0.003
0.3 0.067
0.5 0.148
0.7 0.248
0.9 0.370
1.1 0.518
1.3 0.697
0.55 0.2
```

Polinomial dan hasil taksiran y akan ditampilkan ke layar

```
Persamaan polinomial: - 0.02297656250000014 + 0.24000000000000246x + 0.1973958333333196(x^2) + 3.441176380306895E-14(x^3) + 0.026041666666624073(x^4) + 2.55013385
Nilai f(x) untuk x = 0.85 adalah 0.3372
```

Selain itu, terdapat beberapa contoh hasil dari data-data yang diberikan

a. Jumlah kasus baru

Tabel 4.4 Studi Kasus Interpolasi Polinomial

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

1. 16/07/2022
   diperoleh hasil `Nilai f(x) untuk x = 7.5161 adalah 53562.1875`

2. 10/08/2022
   diperoleh hasil `Nilai f(x) untuk x = 8.3226 adalah 36351.6641`

3. 05/09/2022
   diperoleh hasil `Nilai f(x) untuk x = 9.1667 adalah -665034.2734`
   Tidak akurat karena merupakan di luar *range* interpolasi

b. Mencari nilai f(x) dari tabel

Tabel 4.5 Studi Kasus Nilai f(x) Interpolasi Polinomial

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

$x = 0.2$ `f(x) untuk x = 0.2 adalah 0.033`

$x = 0.55$ `f(x) untuk x = 0.55 adalah 0.1711`

$x = 0.85$ `f(x) untuk x = 0.85 adalah 0.3372`

$x = 1.28$ `f(x) untuk x = 1.28 adalah 0.6775`

c. Mencari nilai f(x) dari persamaan fungsi

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

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

Untuk n = 5 dan x = 0.7, diperoleh `Nilai f(x) untuk x = 0.7 adalah 0.4885`

4.7 Studi Kasus Regresi Linear dan Kuadratik Berganda

Jika user memilih opsi Regresi, maka akan diberi pilihan regresi linear atau kuadratik berganda serta metode inputnya.

```
PILIHAN JENIS REGRESI
1. Regresi linier berganda
2. Regresi kuadratik berganda

Masukkan pilihan (dalam angka):
```

```
PILIHAN METODE MASUKAN/INPUT
1. Input dari keyboard
2. Input dari file

Masukkan pilihan (dalam angka):
```

Jika user memilih input dari keyboard, maka akan ditanya berapa jumlah sampel data dan jumlah variabel independennya. Input berupa matriks dengan format [x1 x2 x3 … xn y] dimana x variabel independen dan y variabel dependen. User juga akan diminta untuk memberikan nilai tiap-tiap variabel peubah untuk dicari taksirannya

1. Regresi linier & input keyboard

```
PILIHAN JENIS REGRESI
1. Regresi linier berganda
2. Regresi kuadratik berganda

Masukkan pilihan (dalam angka): 1
Masukkan banyak sampel data: 6
Masukkan banyak peubah: 2

Masukkan data sampel perbaris x11 x21 x31 ... xn1 y
1 18 29
2 25 25
2 50 21
3 68 18
4 75 15
6 75 11

y = 32.62585618276683 - 1.8483348721362216.x1 - 0.1398235028879389.x2

Masukkan nilai x1: 3
Masukkan nilai x2: 80
Hasil taksiran berdasarkan input user: 15.894971335323055
```

2. Regresi kuadratik & input file

```
PILIHAN JENIS REGRESI
1. Regresi linier berganda
2. Regresi kuadratik berganda

Masukkan pilihan (dalam angka): 2
Masukkan nama file input: 6.txt

y = -1148.5214402522552 + 0.1845550965393509.x1 + 0.8409533611514751.x2 + 75.58456809026093.x3 - 1.5030423633510938E-6.x1^2 - 2.3813400269790004E-4.x2^2 - 1.242545651883181.x3^2 + 1.6647546978232298E-5.x1x2 - 0.00643222867059

Masukkan nilai x1: 50
Masukkan nilai x2: 76
Masukkan nilai x3: 29
Hasil taksiran berdasarkan input user: 0.7201302267905945
```

untuk pada Studi Kasus Regresi Linear dan Kuadratik Berganda

Table 12.1: Data for Example 12.1

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

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

Linear:

```
y = -3.5082707399757345 - 0.002624258122013531.x1 + 7.896250721715248E-4.x2 + 0.15419483330781514.x3
```

Kuadratik:

```
y = -1148.521440252552 + 0.1845550965393509.x1 + 0.8409533611514751.x2 + 75.5845680902609З.x3 - 1.5030423633510938E-6.x1^2 - 2.381340026979000 4E-4.x2^2 - 1.242545651883181.x3^2 + 1.6647546978232298E-5.x1x2
- 0.006432228670520094.x1x3 - 0.027326465825779283.x2x3
```

Taksiran (x1 = 50, x2 = 76, x3 = 29,3)
Linier: 0.892178025335264
Kuadratik: 0.9438503515464092

4.8 Studi Kasus Interpolasi *Bicubic Spline*

Jika User memilih Bicubic Spline Interpolation ia akan diminta untuk memilih inputnya lewat file / keyboard

```
Masukkan pilihan (dalam angka): 5
--------------------------------------------------
PILIHAN METODE MASUKAN/INPUT
1. Input dari keyboard
2. Input dari file

Masukkan pilihan (dalam angka): 1
Masukkan Elemen Matriks anda (4x4):
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
```

Contoh Studi kasus untuk bicubic interpolation adalah
Diberikan matriks input dengan bentuk sebagai berikut..

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

Tentukan nilai:

$f(0, 0) = ?$
$f(0.5, 0.5) = ?$
$f(0.25, 0.75) = ?$
$f(0.1, 0.9) = ?$

Tabel 4.6 Contoh Hasil Bicubic Spline Interpolation

<table>
<tr><td>f(0,0)</td><td>f(0.5,0,5)</td></tr>
<tr><td>Masukkan Elemen Matriks anda (4x4):
21 98 125 153
51 101 161 59
0 42 72 210
16 12 81 96
Masukkan nilai (x,y) dalam rentang [0, 1]: 0 0
Nilai yang dimasukkan valid.
Hasil f(0.00, 0.00) adalah 21.00</td><td>Masukkan pilihan (dalam angka): 1
Masukkan Elemen Matriks anda (4x4):
21 98 125 153
51 101 161 59
0 42 72 210
15 12 81 96
Masukkan nilai (x,y) dalam rentang [0, 1]: 0.5 0.5
Nilai yang dimasukkan valid.
Hasil f(0.50, 0.50) adalah 87.78</td></tr>
<tr><td>f(0.25,0.75)</td><td>f(0.1,0.9)</td></tr>
<tr><td>Masukkan pilihan (dalam angka): 1
Masukkan Elemen Matriks anda (4x4):
21 98 125 153
51 101 161 59
0 42 72 210
16 12 81 96
Masukkan nilai (x,y) dalam rentang [0, 1]: 0.25 0.75
Nilai yang dimasukkan valid.
Hasil f(0.25, 0.75) adalah 117.73</td><td>Masukkan pilihan (dalam angka): 1
Masukkan Elemen Matriks anda (4x4):
21 98 125 153
51 101 161 59
0 42 72 210
16 12 81 96
Masukkan nilai (x,y) dalam rentang [0, 1]: 0.1 0.9
Nilai yang dimasukkan valid.
Hasil f(0.10, 0.90) adalah 128.58</td></tr>
</table>

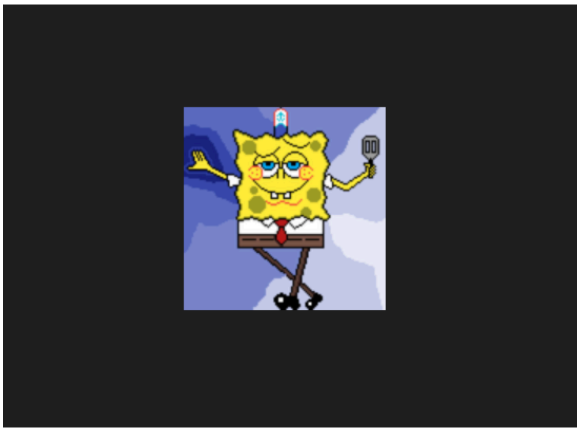
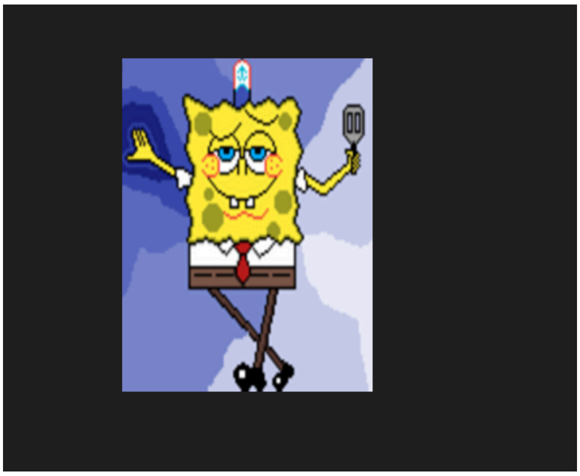
4.9 Studi Kasus Image Resizing dan Stretching

Jika User memilih Image resizer ia akan diminta untuk memasukkan path atau nama dari gambar yang ingin ia resize beserta skala panjang dan lebarnya.

```
Masukkan pilihan (dalam angka): 7
Silahkan tuliskan path dari gambar anda:
kungfupanda_Asli.jpg
Gambar Anda berhasil dimuat:
Matriks piksel berhasil dibuat.
Silahkan Input Skala Panjang:
0,5
Silahkan Input Skala Panjang dengan tepat
Silahkan Input Skala Panjang:
2
Silahkan Input Skala Tinggi:
2
Proses resize sedang berlangsung...
```

Contoh hasil resize

Tabel 4.7 Hasil Image Resizer

Sebelum	Sesudah
	

4.10 Menyimpan Jawaban

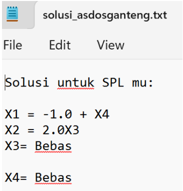
Setelah jawaban ditampilkan di layar, akan ada pilihan untuk menyimpan jawaban ke dalam file

```
Simpan jawaban dalam file?
1. Simpan
2. Tidak

Masukkan pilihan (dalam angka):
```

Jika disimpan, user akan diminta memasukkan nama file tempat penyimpanan

```
Masukkan nama file: asdosganteng
Jawaban berhasil disimpan di solusiasdosganteng.txt
```



BAB V
KESIMPULAN

5.1 Kesimpulan

Tugas besar ini merupakan implementasi dari materi perkuliahan IF2123 Aljabar Linier dan Geometri yang diterapkan dalam sebuah program yang ditulis menggunakan bahasa pemrograman Java. Program ini mengimplementasikan beberapa materi dasar yang berkaitan dengan penyelesaian sistem persamaan linier (SPL) [Gauss, Gauss-jordan, inverse, Cramer], perhitungan determinan [kofaktor, OBE], serta balikan matriks [OBE, adjoin], yang menjadi inti dari konsep aljabar linier.

Selain itu, fungsi-fungsi dasar tersebut dikembangkan lebih lanjut untuk membentuk aplikasi yang relevan dalam dunia nyata, seperti interpolasi polinom, regresi, interpolasi bicubic, dan *image resizer*, yang menunjukkan bagaimana teori dapat diterapkan dalam berbagai bidang. Secara keseluruhan, tugas ini merupakan kulminasi dari pembelajaran yang telah dilakukan selama perkuliahan, menggabungkan berbagai konsep teoritis dan teknik pemrograman menjadi satu kesatuan yang solid, sekaligus memberikan pemahaman yang lebih mendalam tentang cara penerapan materi kuliah dalam pemecahan masalah nyata.

5.2 Saran

Saran untuk pengembangan selanjutnya antara lain,

a. Pembuatan GUI agar program terlihat lebih menarik
b. Membuat implementasi fungsional lain dari Library yang telah dibuat
c. *Debugging* dan *testing* program dengan lebih menyeluruh

5.3 Refleksi

Tugas besar ini memberikan kami pengalaman berharga bekerja dalam tim. Kami dilatih untuk berkomunikasi efektif, mampu mengatasi tantangan, serta meningkatkan kedisiplinan diri. Melalui tugas besar ini, kami juga memperoleh pemahaman mendalam tentang bagaimana materi Aljabar Linear dan Geometri IF2123 dapat diimplementasikan di dunia nyata. Tidak hanya itu, kami juga menjadi lebih familiar dengan bahasa pemrograman Java dan cara penggunaannya.

LAMPIRAN

Referensi

R. Munir, "Determinan (Bagian 1)," IF2123Aljabar Linier dan Geometri, 2023. [Online]. Tersedia: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-08-Determinan-bagian1-2023.pdf. [Diakses: 9 Oktober 2024].

R. Munir, "Determinan (Bagian 2)," IF2123Aljabar Linier dan Geometri, 2023. [Online]. Tersedia: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-09-Determinan-bagian2-2023.pdf. [Diakses: 10 Oktober 2024].

R. Munir, "Matriks Eselon," IF2123 Aljabar Linier dan Geometri, 2023. [Online]. Tersedia: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-02-Matriks-Eselon-2023.pdf. [Diakses: 6 Oktober 2024].

R. Munir, "Sistem persamaan linier (Bagian 1: Metode eliminasi Gauss)," IF2123 Aljabar Linier dan Geometri, 2023. [Online]. Tersedia: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-03-Sistem-Persamaan-Linier-2023.pdf. [Diakses: 9 Oktober 2024].

R. Munir, "Sistem persamaan linier (Bagian 2: Tiga kemungkinan solusi sistem persamaan linier)," IF2123 Aljabar Linier dan Geometri, 2023. [Online]. Tersedia: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-04-Tiga-Kemungkinan-Solusi-SPL-2023.pdf. [Diakses: 7 Oktober 2024].

R. Munir, "Sistem persamaan linier (Bagian 3: Metode eliminasi GaussJordan)," IF2123 Aljabar Linier dan Geometri, 2023. [Online]. Tersedia: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-05-Sistem-Persamaan-Linier-2023.pdf. [Diakses: 10 Oktober 2024].

Repository

https://github.com/henry204xx/Algeo01-23051