IF2123 Aljabar Linear dan Geometri

IMPLEMENTASI KALKULATOR MATRIKS, SISTEM PERSAMAAN LINEAR, DAN APLIKASINYA DENGAN MENGGUNAKAN BAHASA PEMROGRAMAN JAVA

Laporan Tugas Besar 1

Disusun untuk memenuhi tugas mata kuliah Aljabar Linear dan Geometri
pada Semester 1 (satu) Tahun Akademik 2022/2023.

Oleh

Muhamad Aji Wibisono	**13521095**
Chiquita Ahsanunnisa	**13521129**
Alisha Listya Wardhani	**13521171**

Kelompok ー♡Xx_Bwasr3ngL0vers69_xX♡ー

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2022

DAFTAR ISI

BAB I
DESKRIPSI MASALAH

Matriks merupakan sebuah objek matematika yang tersusun dari angka-angka berdasarkan baris dan kolom. Teori-teori mengenai matriks memiliki banyak manfaat untuk menyelesaikan berbagai masalah di dunia matematika, sains, rekayasa, hingga ekonomi. Matriks digunakan sebagai representasi atau model matematika yang lebih ringkas dalam menyelesaikan berbagai permasalahan dalam bidangnya, misalnya dalam menemukan solusi masalah persamaan linear, transformasi linear.

Sistem persamaan linier (SPL) adalah persamaan-persamaan linier yang dikorelasikan untuk membentuk suatu sistem. Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. 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). Setiap metode penyelesaian SPL memiliki pola yang berulang. Oleh karena itu, dapat dibuat sebuah algoritma pada program komputer yang bisa menyelesaikan SPL dengan metode-metode tersebut.

Dalam Tugas Besar 1 Aljabar Linier dan Geometri ini, penulis membuat satu atau lebih pustaka (*library*) aljabar linier dalam bahasa Java. Library tersebut terdiri dari fungsi-fungsi penyelesaian SPL seperti eliminasi Gauss, eliminasi Gauss-Jordan, menentukan *inverse* matriks, menghitung determinan, dan kaidah Cramer. Library tersebut selanjutnya digunakan dalam sebuah program Implementasi Kalkulator Matriks yang dapat menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk Sistem Persamaan Linear, menyelesaikan persoalan interpolasi, dan persoalan regresi. Pustaka tersebut akan digunakan untuk membuat program penyelesaian berbagai persoalan dalam bentuk SPL, menyelesaikan persoalan interpolasi dan regresi linier, menghitung matriks balikan, menghitung determinan matriks dengan berbagai metode (reduksi baris dan ekspansi kofaktor).

Spesifikasi program yang dibuat adalah sebagai berikut:

1. Program dapat menerima masukan (input) baik dari keyboard maupun membaca masukan dari file text. Untuk SPL, masukan dari keyboard adalah m, n, koefisien

aij, dan bi. Masukan dari file berbentuk matriks augmented tanpa tanda kurung, setiap elemen matriks dipisah oleh spasi. Misalnya,

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

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

```
3 4.5 2.8
-3 7 8.3
0.5 -10 -9
```

3. Untuk persoalan interpolasi, masukannya jika dari keyboard adalah n, (x0, y0), (x1, y1), ..., (xn, yn), dan nilai x yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung. Misalnya jika titik-titik datanya adalah (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513), maka di dalam file text ditulis sebagai berikut:

```
8.0 2.0794
9.0 2.1972
9.5 2.2513
```

4. Untuk persoalan regresi, masukannya jika dari keyboard adalah n (jumlah peubah x), m (jumlah sampel), semua nilai-nilai x1i, x2i, ..., xni, nilai yi, dan nilai-nilai xk yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.

5. Untuk persoalan SPL, luaran (output) program adalah solusi SPL. Jika solusinya tunggal, tuliskan nilainya. Jika solusinya tidak ada, tuliskan solusi tidak ada, jika solusinya banyak, maka tuliskan solusinya dalam bentuk parametrik (misalnya x4 = -2, x3 = 2s – t, x2 = s, dan x1 = t.)

6. Untuk persoalan determinan dan matriks balikan, maka luarannya sesuai dengan persoalan masing-masing

```
-3 7 8.3 11 -4
3 4.5 2.8 10 12
0.5 -10 -9 12 0
0.1 0.2
```

7. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan. Contoh luaran untuk interpolasi adalah $f(x) = -0.0064x^2 + 0.2266x + 0.6762,\ f(5) =$ ... dan untuk regresi adalah $f(x) = -9.5872 + 1.0732x_1, f(x_k) =$ ...

8. Untuk persoalan interpolasi bicubic, masukan dari file text (.txt) yang berisi matriks berukuran 4x4 yang berisi nilai f(i,j) dengan i dan j adalah indeks matriks diikuti dengan nilai a dan b untuk mencari nilai f(a,b). misalnya jika nilai dari f(-1,-1), f(-1,0), f(-1,1), f(-1,2),f(0,-1), f(0,0), f(0,1), f(0,2), f(1,-1), f(1,0), f(1,1), f(1,2), f(2,-1), f(2,0), f(2,1), f(2,2) berturut-turut adalah 1, 2, 3, 4, 5, 6, 7, 8, 9 ,10, 11, 12, 13, 14, 15 ,16 serta nilai a dan b yang dicari berturut-turut adalah 0.5 dan 0.5 maka isi file text ditulis sebagai berikut:

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

   luaran yang dihasilkan adalah nilai dari f(0.5,0.5).
   masukannya adalah matriks 4 x 4, diikuti oleh nilai a dan b, maka luarannya adalah nilai f(a,b).

9. Luaran program harus dapat ditampilkan pada layar komputer dan dapat disimpan ke dalam file.

10. Bahasa program yang digunakan adalah Java.

11. Program tidak harus berbasis GUI, cukup text-based saja, namun boleh menggunakan GUI (memakai kakas Eclipse misalnya).

12. Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan dirancang masing-masing. Misalnya, menu:

```
MENU
1. Sistem Persamaaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic
6. Regresi linier berganda
7. Keluar
```

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

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

Begitu juga untuk pilihan menu nomor 2 dan 3.

BAB II
TEORI SINGKAT

Dalam implementasi pustaka Tugas Besar 1 ini terdapat lima bagian besar: penyelesaian SPL, interpolasi polinom, interpolasi bicubic, regresi linier berganda, dan *image scaling*. Berikut merupakan penjelasan tentang setiap bagian.

2.1 Sistem Persamaan Linier

Sistem persamaan linier (SPL) adalah persamaan-persamaan linear yang dikorelasikan untuk membentuk suatu sistem. Sistem tersebut dapat terdiri dari beberapa variabel, tetapi maksimal memiliki orde 1 (linear). Bentuk umum permasamaan linear m dengan n yang tidak diketahui dapat ditulis sebagai berikut.

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

2.1. Bentuk umum sistem persamaan linier (SPL)

Dengan $x_1, x_2, \cdots, x_n$ adalah variabel yang tidak diketahui, $a_{11}, a_{12}, \cdots, a_{mn}$ adalah koefisien dari setiap variabel, dan $b_1, b_2, \cdots, b_n$ adalah konstantanya. Matriks digunakan sebagai model atau representasi yang lebih ringkas dalam menuliskan SPL. Penulisan menggunakan matriks adalah sebagai $Ax = b$, dengan A adalah matriks $m \times n$, x adalah vektor kolom dengan entri n dan b adalah vektor kolom dengan entri n.

$$
A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}
$$

2.2. Bentuk sistem persamaan linier (SPL) dalam matriks

Sistem persamaan linier dapat diselesaikan dengan berbagai cara. Cara paling sederhana adalah dengan solusi eliminasi variabel. Namun eliminasi variabel tidak menggunakan matriks sebagai model penyelesaiannya. Oleh karena itu, berikut berbagai metode penyelesaian SPL dengan menggunakan matriks.

2.2 Metode Penyelesaian Eliminasi Gauss dan Gauss-Jordan

Metode pengurangan baris atau eliminasi Gauss menyelesaikan sistem persamaan linear dengan mewakilkan persamaan-persamaan yang ada dalam bentuk matriks. Matriks ini kemudian diubah dengan menukar posisi baris, menambahkan atau mengurangi satu baris dengan baris lain (atau disebut dengan Operasi Baris Elementer (OBE)), dengan tujuan untuk mencapai matriks eselon baris.

$$\begin{bmatrix} 1 & * & * \\ 0 & 1 & * \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & * & * & * \\ 0 & 1 & * & * \\ 0 & 0 & 1 & * \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & * & * \\ 0 & 1 & * \\ 0 & 0 & 1 \end{bmatrix}$$

2.3. Contoh bentuk matriks eselon baris

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_4 \end{bmatrix} \sim OBE \sim \begin{bmatrix} 1 & * & \cdots & * & * \\ 0 & 1 & \cdots & * & * \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & * \end{bmatrix}$$

2.4. Eliminasi Gauss

Dengan * adalah sembarang nilai. Matriks eselon (atau bentuk eselon baris) adalah matriks yang memiliki 1 utama pada setiap baris, kecuali baris yang seluruhnya nol. Setelah sudah dalam bentuk matriks eselon baris, persamaan pada matriks eselon baris dapat dipecahkan dengan teknik penyulihan mundur (*backward substitution*). Berikut merupakan contoh penyelesaian kasus dengan eliminasi Gauss.

$$\begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

diperoleh persamaan-persamaan linier sbb:

$x_1 + 3/2x_2 - 1/2x_3 = 5/2$ (i)

$x_2 + 1/2x_3 = 7/2$ (ii)

$x_3 = 3$ (iii)

Selesaikan dengan teknik penyulihan mundur sbb:

(iii) $x_3 = 3$

(ii) $x_2 + 1/2x_3 = 7/2 \rightarrow x_2 = 7/2 - 1/2(3) = 2$

(i) $x_1 + 3/2x_2 - 1/2x_3 = 5/2 \rightarrow x_1 = 5/2 - 3/2\,(2) - 1/2\,(3) = 1$

Solusi: $x_1 = 1$, $x_2 = 2$, $x_3 = 3$

2.5. Penyelesaian menggunakan Eliminasi Gauss

Metode Eliminasi Gauss-Jordan merupakan pengembangan dari metode eliminasi Gauss. Pada metode ini, hasil akhir dari OBE pada matriks augmented adalah eselon baris tereduksi.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2.6. Contoh bentuk matriks eselon baris tereduksi

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_4 \end{bmatrix} \sim OBE \sim \begin{bmatrix} 1 & 0 & \cdots & 0 & * \\ 0 & 1 & \cdots & 0 & * \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & * \end{bmatrix}$$

2.7. Eliminasi Gauss-Jordan

Pada metode ini, eliminasi terdiri menjadi dua fase, yaitu fase maju (eliminasi Gauss) untuk menghasilkan nilai 0 di bawah nilai 1 utama dan fase mundur untuk menghasilkan nilai 0 di atas semua 1 utama. Berikut merupakan contoh penyelesaian kasus dengan eliminasi Gauss-Jordan.'

$$\begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Diperoleh persamaan-persamaan berikut:

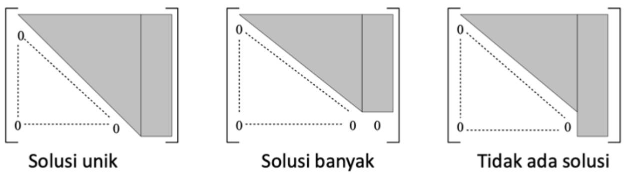
$x_1 - x_3 = 0 \rightarrow x_1 = x_3$

$x_2 + x_3 = 0 \rightarrow x_2 = -x_3$

$x_4 = 0$

2.8. Penyelesaian menggunakan Eliminasi Gauss

Dalam menyelesaikan sistem persamaan linear, terdapat tiga kemungkinan solusi yang bida didapatkan, yaitu solusi unik, solusi banyak, dan tidak ada dolusi. Untuk menentukan tipe solusi dari SPL, dapat dilihat hasil akhir matriks augmented setelah proses eliminasi.



$$\begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix} \begin{bmatrix} 1 & 3 & -2 & 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

2.9. Tiga Macam Solusi Persamaan Linier

2.3 Determinan Matriks

Determinan adalah nilai skalar yang merupakan fungsi dari entri sebuah matriks persegi. Hal ini memungkinkan karakterisasi dari properti yang dimiliki oleh matriks tersebut dan sebuah peta linier yang diwakili matriks. Nilai determinan sebuah matriks dapat menentukan apakah matriks tersebut memiliki matriks balikan atau tidak. Determinan matriks A dilambangkan dengan $det(A), det\ A,$ atau $|A|$.

Berikut merupakan teorema yang berkaitan dengan deteminan:

1. Jika A mengandung sebuah baris nol atau kolom nol, maka $\det(A) = 0$
2. Jika A^T adalah matriks transpos dari A, maka $\det(A^T) = \det(A)$
3. Jika $A = BC$ maka $\det(A) = \det(B)\det(C)$
4. Sebuah matriks hanya mempunyai balikan jika dan hanya jika $\det(A) \neq 0$
5. $det(A^{-1}) = {}^{1}/_{det(A)}$

Adapun terdapat beberapa cara dalam menghitung determinan sebuah matriks, pada makalah ini akan dibahas mengenai metode reduksi baris dan metode ekspansi kofaktor. Metode Reduksi Baris memanfaatkan Operasi Baris Elementer (OBE) pada matriks sehingga terbentuk matriks segitiga atas atau matriks segitiga bawah. Determinan matriks tersebut kemudian dapat dicari dengan mengkalikan semua elemen yang berada pada diagonal utamanya.

$$A_1 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \quad A_2 = \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{33} & a_{44} \end{bmatrix}$$

$$\det(A) = a_{11} \cdot a_{22} \cdot a_{33} \cdot a_{44}$$

3.1. Penyelesaian Determinan menggunakan Metode Reduksi Baris

Dalam menyelesaikan dengan metode reduksi baris, terdapat beberapa aturan yang perlu diperhatikan, antara lain:

$$A \xrightarrow{\text{Kalikan sebuah baris dengan k}} B\ , \quad \text{maka det(B) = k det(A)}$$

$$A \xrightarrow{\text{Pertukarkan dua baris}} B\ , \quad \text{maka det(B) = –det(A)}$$

$$A \xrightarrow[\text{dengan k kali baris yang lain}]{\text{Sebuah baris ditambahkan}} B\ , \quad \text{maka det(B) = det(A)}$$

3.2. Aturan dalam Penyelesaian Metode Reduksi Baris

Metode ekspansi kofaktor merupakan penyelesaian yang menggunakan matriks kofaktor dalam prosesnya. Untuk mencari nilai dari matriks kofaktor, terlebih dahulu harus dicari nilai monor setiap elemen matriks. Minor entri dari setiap elemen matriks merupakan determinan sub-matriks yang elemen-elemennya tidak berada pada baris i dan kolom j.

Setelah mendapatkan nilai minor dari masing-masing elemen matriks, dapat ditentukan nilai dari kofaktornya. Secara formal nilai kofaktor setiap elemen dituliskan dengan rumus $C_{ij} = (-1)^{i+j} M_{ij}$. Misalkan A adalah matriks persegi $n \times n$ dan C_{ij} adalah kodaktor entri dari a_{ij}, maka matriks kofaktor A adalah sebagai berikut.

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

3.3. Matriks Kofaktor

Setelah didapat matriks kofaktor, determinan matriks tersebut diperoleh dengan menjumlahkan elemen-elemen pada satu baris yang sama atau pada satu kolom yang sama. Maka diperoleh $det(A) = a_{n1}C_{n1} + a_{n2}C_{n2} + \cdots a_{nn}C_{nn}$ secara baris atau $det(A) = a_{1n}C_{1n} + a_{2n}C_{2n} + \cdots a_{nn}C_{nn}$ secara kolom. Perlu diperhatikan bahwa penyelesaian menggunakan matriks kofaktor memungkinkan adanya sifat rekursif karena melibatkan perhitungan determinan dari sub-matriksnya.

2.4 Metode Penyelesaian Matriks Balikan

Matriks balikan (*inverse*) adalah kebalikan dari sebuah matriks (A^{-1}). Apabila matriks tersebut dikalikan dengan inversnya, maka menghasilkan matriks identitas ($A^{-1}A = I$). Matriks hanya memiliki invers apabila matriks tersebut berbentuk persegi dan merupakan matriks non-singular atau memiliki determinan bukan nol. Matriks balikan dapat digunakan sebagai alat bantu dalam penyelesaian sistem persamaan linier.

Bentuk $Ax = b$ dapat diubah menjadi bentuk $x = bA^{-1}$ yang mana terjadi perkalian matriks, sehingga dapat menghasilkan nilai dari setiap variabel x. Adapun terdapat dua cara dalam mencari matriks balikan, yaitu: metode matriks identitas dan metode matriks kofaktor.

2.4.1 Metode Matriks Identitas

Secara sistematis, berikut pencarian matriks balikan menggunakan matriks identitas:

$$[\, A \mid I \,] = [\, I \mid A^{-1} \,]$$

4.1. Sifat Matriks Identitas

Pada matriks persegi A, matriks balikannya dapat dicari dengan augmentasi matriks A dengan matriks identitasnya, lalu dilakukan metode eliminasi Gauss-Jordan pada kedua matriks sehingga bagian kiri dari matriks tersebut akan berbentuk matriks identitas. Berikut merupakan contoh penyelesaian mencari matriks balikan.

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

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

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

Jadi, balikan matriks A adalah

$$A^{-1} = \begin{bmatrix} -40 & 16 & 9 \\ 13 & -5 & -3 \\ 5 & -2 & -1 \end{bmatrix}$$

4.2. Penyelesaian matriks balikan menggunakan matriks identitas

2.4.2 Metode Matriks Kofaktor

Matriks balikan dapat dihitung dengan memanfaatkan determinan dan matriks adjoin dengan rumus sebagai berikut.

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

4.3 Rumus determinan menggunakan matriks kofaktor

Matriks adjoin merupakan matriks yang dihasilkan dari matriks kofaktor yang telah ditranspos sebelumnya. Hal ini berarti terdapat penukaran elemen C_{ij} pada matriks kofaktor menjadi elemen C_{ji} pada matriks adjoinnya. Sebagai contoh berikut merupakan adjoin matriks yang didapat dari hasil transpos matriks kofaktor.

$$kof(A) = \begin{bmatrix} 3 & -2 & 5 \\ 16 & 0 & 1 \\ 4 & 6 & 9 \end{bmatrix} \; adj(A) = \begin{bmatrix} 3 & 16 & 4 \\ -2 & 0 & 6 \\ 5 & 1 & 9 \end{bmatrix}$$

4.4 Contoh Matriks Adjoin

2.5 Kaidah Cramer

Kaidah Cramer merupakan rumus pencarian penyelesaian sistem persamaan linier dengan menggunakan determinan suatu matriks dan matriks lain yang disusun dengan mengganti salah satu kolom dengan vektor yang terdiri dari angka di sebelah kanan persamaannya. Jika $Ax = b$ adalah SPL yang terdiri dari n persamaan linier dengan n peubah sedemikian sehingga $det(A) \neq 0$, maka SPL tersebut memiliki solusi unik yaitu:

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

5.1. Penyelesaian SPL menggunakan Kaidah Cramer

dalam kasus ini, A_j adalah matriks yang diperoleh dengan mengganti entri kolom ke-j dari A dengan entri dari matriks b.

2.6 Interpolasi Polinom

Interpolasi polinom merupakan teknik interpolasi untuk menemukan suatu polinomial yang mempunyai grafik yang melalui sejumlah titik dengan absis yang berbeda. Misal diberikan sebanyak (n + 1) titik (dengan absis yang berbeda) sebagai berikut:

$$(x_0, y_0), (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$

Akan ditentukan sebuah polinom $p(x)$ berderajat n:

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga $y_i = p(x_i)$ untuk $i = 0, 1, 2, \dots, n$. Karena p(x) menginterpolasi (melewati) semua titik-titik tersebut, titik-titik tersebut harus memenuhi sistem persamaan linear berikut.

$$a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_n x_0^n = y_0$$

$$a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_n x_1^n = y_1$$
$$\vdots$$
$$a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_n x_n^n = y_n$$

Sistem persamaan linear di atas dapat dinyatakan dalam persamaan matriks:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{a}$$

dengan

$$\boldsymbol{y} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \qquad \boldsymbol{X} = \begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix} \qquad \boldsymbol{a} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}$$

6.1. Bentuk umum persamaan Interpolasi Polinom

Sistem persaman linear dalam bentuk matriks di atas dapat diselesaikan (untuk mendapatkan matriks **a**) dengan metode-metode yang sudah dibahas di bagian sebelumnya. Setelah fungsi polinomial p(x) didapat, fungsi tersebut dapat digunakan untuk menghitung nilai p(c) untuk sembarang c yang berada di range data titik yang digunakan di awal.

2.7 Bicubic Interpolation

Bicubic Interpolation merupakan teknik interpolasi pada data 2D umumnya digunakan dalam pembesaran citra yang merupakan pengembangan dari interpolasi linear dan cubic yang telah dipelajari pada kuliah metode numerik di aljabar geometri.

Misalkan diberikan sebuah matriks M yang berisi nilai f(x,y) untuk $x, y = -1, 0, 1, 2$:

$$\begin{bmatrix} f(-1,-1) & f(0,-1) & f(1,-1) & f(2,-1) \\ f(-1,0) & f(0,0) & f(1,0) & f(2,0) \\ f(-1,1) & f(0,1) & f(1,1) & f(2,1) \\ f(-1,2) & f(0,2) & f(1,2) & f(2,2) \end{bmatrix}$$

7.1. Matriks Interpolasi Bikubik

Akan ditentukan suatu fungsi f(x,y) yang memiliki model:

$$f(x,y) = \sum_{j=0}^{3} \sum_{i=0}^{3} a_{ij} x^i y^j , \qquad x, y = -1, 0, 1, 2$$

dan menginterpolasi nilai-nilai pada matriks M. Karena f(x,y) menginterpolasi semua titik-titik tersebut, titik-titik tersebut harus memenuhi sistem persamaan linear berikut.

$$f(-1,-1) = a_{00}(-1)^0(-1)^0 + a_{10}(-1)^1(-1)^0 + \cdots + a_{33}(-1)^3(-1)^3$$

$$\vdots$$

$$f(1,2) = a_{00}(1)^0(2)^0 + a_{10}(1)^1(2)^0 + \cdots + a_{33}(1)^3(2)^3$$

$$f(2,2) = a_{00}(2)^0(2)^0 + a_{10}(2)^1(2)^0 + \cdots + a_{33}(2)^3(2)^3$$

Sistem persamaan linear di atas dapat dinyatakan dalam persamaan matriks:

$$\boldsymbol{y} = \boldsymbol{Xa}$$

dengan

$$\boldsymbol{y} = \begin{bmatrix} f(-1,-1) \\ f(0,-1) \\ f(1,-1) \\ f(2,-1) \\ f(-1,0) \\ f(0,0) \\ f(1,0) \\ f(2,0) \\ f(-1,1) \\ f(0,1) \\ f(1,1) \\ f(2,1) \\ f(-1,2) \\ f(0,2) \\ f(1,2) \\ f(2,2) \end{bmatrix} \quad \boldsymbol{X} = \begin{bmatrix} 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 2 & 4 & 8 & -1 & -2 & -4 & -8 & 1 & 2 & 4 & 8 & -1 & -2 & -4 & -8 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 \\ 1 & -1 & 1 & -1 & 2 & -2 & 2 & -2 & 4 & -4 & 4 & -4 & 8 & -8 & 8 & -8 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 8 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 1 & 2 & 4 & 8 & 2 & 4 & 8 & 16 & 4 & 8 & 16 & 32 & 8 & 16 & 32 & 64 \end{bmatrix} \quad \boldsymbol{a} = \begin{bmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30} \\ a_{01} \\ a_{11} \\ a_{21} \\ a_{31} \\ a_{02} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{03} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}$$

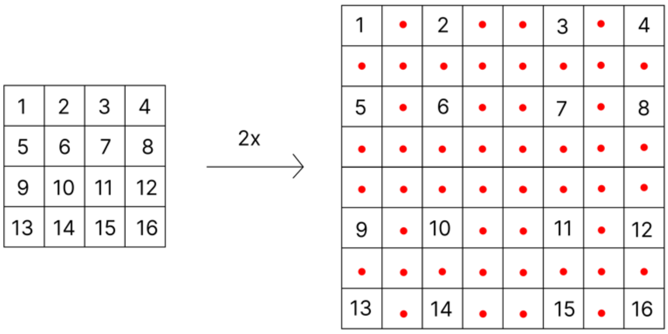
7.2. Penyelesaian Matriks dengan Interpolasi Bikubik

Sistem persaman linear dalam bentuk matriks di atas dapat diselesaikan (untuk mendapatkan matriks **a**) dengan metode-metode yang sudah dibahas di bagian sebelumnya. Setelah model interpolasi f(x,y) didapat, model tersebut dapat digunakan untuk menghitung nilai f(a,b) untuk sembarang a dan b yang ada di rentang [0, 1].

2.8 Image Scaling

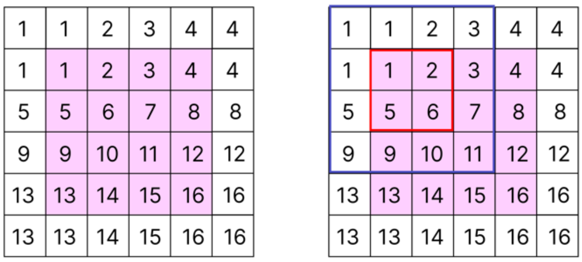
Dalam grafika komputer dan pencitraan digital, penskalaan gambar mengacu pada pengubahan ukuran pada gambar digital. Saat mengubah skala suatu gambar, primitif pada grafik tersebut dapat diskalakan menggunakan transformasi geometris. Tujuan utamanya adalah tidak kehilangan kualitas gambar. Terdapat beberapa algoritma yang digunakan, salah satunya adalah interpolasi bikubik (*bicubic interpolation*).

Bicubic interpolation menggunakan nilai warna dari setiap pixel, menghasilkan transisi berkelanjutan dalam pixel-pixel diantara pixel aslinya. Meskipun diinginkan hasil yang ideal, algoritma ini mengurangi kontras pada tepian suatu objek pada gambar sehingga mengurangi ketajaman gambar tersebut. Nilai yang diambil pada tugas besar 1 kami hanya terbatas pada interpolasi nilai *greyscale*. Nilai greyscale ini didapatkan dengan merata-ratakan semua nilai RGB. Hal ini dikarenakan interpolasi warna pada ketiga RGB akan memakan waktu dan computer cost yang lebih banyak.



8.1. Interpolasi bikubik pada gambar

Berikut merupakan *bicubic interpolation* dari nilai setiap pixel pada gambar. Titik merah merupakan pixel dimana nilainya harus diinterpolasi. Pemilihan letak nilai pixel awal (seperti 1,2, ...,16) bertujuan agar gradien yang dihasilkan pada gambar hasil mencapai hasil yang mendekati gambar awal. Masalah yang didapatkan dengan mengimplementasikan interpolasi tersebut adalah bahwa *bicubic interpolation* membutuhkan minimal 16 titik dengan range yang bisa diinterpolasi adalah [0,1]. Hal ini menyebabkan titik-titik pada ujung-ujung gambar tidak bisa diinterpolasi. Untuk menyelesaikan permasalahan ini, penulis memberikan padding pada ujung-ujung matriks.



8.2. Pemberian padding pada matriks

Matriks awal ditandai oleh warna pink. Padding merupakan duplikasi dari setiap entri matriks sehingga tidak akan merusak perkiraan hasil interpolasi. Padding tersebut memungkinkan untuk mencari interpolasi antara pixel 1 dan pixel 2 dengan matriks yang dipakai berwarna biru, dan range yang akan diinterpolasi ada dalam range [0,1] atau bewarna merah. Interpolasi dilakukan untuk setiap pixel pada gambar tersebut untuk menentukan nilai pixel dengan contoh sebagai berikut. Hasil matriks ini diubah menjadi nilai konstruktor RGB dari sebuah gambar sehingga dihasilkan gambar baru yang telah diperbesar dua kali lipat.

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

2x →

1	1.43	2	2.33	2.66	3	3.56	4
4.31	4.75	5.07	5.4	5.73	5.93	5.75	5.06
5	5.43	6	6.33	6.66	7	7.56	8
7.8	8.32	8.64	8.97	9.30	9.50	9.32	8.63
9.64	8.95	11.28	11.61	11.94	8.64	6.57	3.64
9	9.43	10	10.33	10.66	11	11.56	12
12.81	13.25	13.57	13.90	14.23	14.43	14.24	13.56
13	13.67	14	14.64	14.97	15	15.97	16

8.3. Hasil interpolasi bikubik pada matriks berukuran 4x4

2.9 Regresi Linier Berganda

Regresi linier berganda adalah model regresi yang mengandung lebih dari satu variabel regressor (predictor). Model regresi ini digunakan untuk menganalisis respons suatu variabel dependant y yang mungkin bergantung pada sejumlah n variabel regressor. Bentuk umum dari model regresi linier berganda dengan sejumlah n variabel regressor yaitu:

$$y = b_0 + b_1x_1 + b_2x_2 + \cdots + b_nx_n + \epsilon \quad (Persamaan\ 2.9.1)$$

Sebagai keterangan, $b_0,\ b_1, \ldots, b_n$ merupakan koefisien regresi dan ϵ adalah nilai error.

Salah satu metode yang digunakan untuk mengestimasi koefisien regresi di model ini adalah metode least squares. Misalkan dari hasil observasi, terdapat m sampel dengan m>n. Anggap x_{ji} mewakili nilai variabel x_j untuk observasi ke-i. Setiap sampel dapat dinyatakan dengan:

$$(x_{1i}, x_{2i}, \ldots, x_{ni}, y_i), \qquad i = 1, 2, \ldots, m, \qquad m > n$$

Setiap sampel $(x_{1i}, x_{2i}, \ldots, x_{ni}, y_i)$ memenuhi Persamaan 2.9.1, dapat dituliskan:

$$y_i = b_0 + b_1x_{1i} + b_2x_{2i} + \cdots + b_nx_{ni} + \epsilon_i \qquad (Persamaan\ 2.9.2)$$

$$y_i = b_0 + \sum_{j=1}^{n} b_j x_{ji} + \epsilon_i, \qquad i = 1, \dots, m$$

Dengan ide untuk meminimalkan nilai $\sum_{i=1}^{m} \epsilon_i^2$, didapat normal equation yang digunakan untuk mengestimasi koefisien regresi yaitu:

$$n\hat{b}_0 + \hat{b}_1 \sum_{i=1}^{m} x_{1i} + \hat{b}_2 \sum_{i=1}^{m} x_{2i} + \cdots + \hat{b}_n \sum_{i=1}^{m} x_{ni} = \sum_{i=1}^{m} y_i$$

$$\hat{b}_0 \sum_{i=1}^{m} x_{1i} + \hat{b}_1 \sum_{i=1}^{m} x_{1i}^2 + \hat{b}_2 \sum_{i=1}^{m} x_{1i}x_{2i} + \cdots + \hat{b}_n \sum_{i=1}^{m} x_{1i}x_{ni} = \sum_{i=1}^{m} x_{1i}y_i$$

$$\vdots$$

$$\hat{b}_0 \sum_{i=1}^{m} x_{ni} + \hat{b}_1 \sum_{i=1}^{m} x_{ni}x_{2i} + \hat{b}_2 \sum_{i=1}^{m} x_{ni}x_{2i} + \cdots + \hat{b}_n \sum_{i=1}^{m} x_{ni}^2 = \sum_{i=1}^{m} x_{ni}y_i$$

$$(Sistem\ Persamaan\ 2.9.3)$$

Sistem persamaan linear di atas dapat diselesaikan dengan menggunakan properti matriks.

Dalam bentuk matriks, persamaan 2.9.1 dapat dituliskan sebagai:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{b} + \boldsymbol{\epsilon}$$

dengan

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \qquad \boldsymbol{X} = \begin{bmatrix} 1 & x_{11} & x_{21} & \dots & x_{n1} \\ 1 & x_{12} & x_{22} & \dots & x_{n2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1m} & x_{2m} & \dots & x_{nm} \end{bmatrix} \qquad \boldsymbol{b} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix} \qquad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_m \end{bmatrix}$$

8.1. Bentuk umum regresi linier berganda pada matriks

maka sistem persamaan 2.9.3 (normal equations for least square estimation) dapat dituliskan dalam notasi matriks sebagai berikut.

$$\boldsymbol{X}^T\boldsymbol{X}\hat{\boldsymbol{b}} = \boldsymbol{X}^T\boldsymbol{y}$$

Sistem persaman linear dalam bentuk matriks di atas dapat diselesaikan (untuk mendapatkan matriks $\hat{\boldsymbol{b}}$) dengan metode-metode yang sudah dibahas di bagian sebelumnya. Setelah persamaan regresi didapat, persamaan tersebut dapat digunakan untuk menghitung nilai dari variabel dependant dari sembarang variabel regressor.

BAB III
IMPLEMENTASI PUSTAKA DAN PROGRAM DALAM BAHASA JAVA

Dalam Tugas Besar I, terdapat 4 folder utama: src, doc, bin, dan test. Folder src memuat 9 file java, diantaranya: Main.java, BicubicInterpolation.java, ImageUpsc.java, ImageUtil.java, InterpolasiPolinom.java, matriks.java, operasiMatriks.java, RegresiLinierberganda.java, dan SPL.java. Berikut merupakan rincian serta deskripsi (atribut dan metode) dari setiap file tersebut.

3.1 Class Matriks

Class ini berisi tipe data matriks yang digunakan dalam berbagai aplikasi program.

3.1.1 Atribut

`CAPACITY`	integer yang menyimpan jumlah elemen yang dapat ditampung oleh satu baris atau kolom matriks.
`jumlahKolom`	integer yang menyimpan jumlah kolom dalam matriks.
`jumlahBaris`	integer yang menyimpan jumlah baris dalam matriks.
`Mat`	array 2D bertipe double yang digunakan untuk menyimpan elemen matriks.

3.1.2 Metode

`void bacaMatriks(int m, int n)`
`{I.S. Matriks sembarang tidak terisi, m dan n > 0}`
`{F.S. Matriks terisi dengan jumlahBaris = m, dan jumlahKolom = n}`
	Metode untuk membaca matriks dari terminal dengan parameter m berisikan jumlah baris dan parameter n berisikan jumlah kolom.

`void bacaFileMatriks(String filename)`
`{I.S. Matriks sembarang tidak terisi}`
`{F.S. Matriks terisi dengan matriks yang ada di file ./test/filename}`
	Metode untuk membaca matriks dari sebuah file yang berada di folder test, parameter filename berisikan nama file.

`void bacaFileMatriksBolong(String filename, int nKosong)`
`{I.S. Matriks sembarang tidak terisi, file asal berisi matriks yang baris terbawahnya tidak lengkap}`
`{F.S. Matriks terisi dengan matriks yang ada di file ./test/filename, bagian bawah yang tidak lengkap diisi dengan -999.0}`
	Metode untuk membaca matriks dari sebuah file yang berada di folder test, yang memiliki parameter filename berisikan nama file dan parameter nKosong berisikan banyak elemen matriks pada baris terbawah yang kosong. Metode ini secara khusus digunakan untuk membaca matriks dari file yang akan diolah dengan interpolasi dan regresi.

`void writeMatrixFile(matrix m)`
`{I.S. Matriks terdefinisi}`
`{F.S. Terbuat file baru di ./test/ yang berisikan matriks m}`
	Metode untuk menuliskan matriks di sebuah file yang berada di folder test, nama file akan bergantung kepada input pengguna pada saat fungsi ini dijalankan.

`double getComponent(int m, int m)`
`{I.S. Matriks terdefinisi, 0<=m<=jumlahBaris-1, 0<=n<=jumlahKolom-1,}`
`{F.S. Mengembalikan komponen matriks.Mat[m][n]}`
	Metode primitif untuk mengambil komponen matriks pada baris ke m dan kolom ke n.

`Boolean penuhRow()`
`{I.S. Matriks terdefinisi}`
`{F.S. Mengembalikan true jika jumlahKolom = CAPACITY}`
	Metode primitif untuk memeriksa penuh atau tidaknya kolom matriks.

`Boolean penuhCol()`
`{I.S. Matriks terdefinisi}`
`{F.S. Mengembalikan true jika jumlahBaris = CAPACITY}`
	Metode primitif untuk memeriksa penuh atau tidaknya baris matriks.

`void tulisMatriks()`
`{I.S. Matriks terdefinisi}`
`{F.S. Menuliskan matriks di terminal}`
	Metode untuk menuliskan isi matriks pada terminal.

`void resetCap(int newCap)`
`{I.S. Matriks sembarang, newCap > 0}`
`{F.S. Mengosongkan matriks tersebut dan mengubah CAPACITY menjadi newCap}`
	Metode untuk mengosongkan matriks dan mengubah kapasitasnya menjadi lebih kecil atau lebih besar.

3.2 Class OperasiMatrix

Class ini berisikan metode operasi biner maupun uner yang dilakukan pada matriks.

3.2.1 Atribut

Class ini tidak memiliki atribut.

3.2.2 Metode

`Boolean isEqual(matriks M1, matriks M2)`
`{I.S. Matriks M1 dan M2 terdefinisi}`
`{F.S. Mengembalikan true jika M1 sama dengan M2}`
	Metode untuk menentukan apakah matriks M1 sama dengan matriks M2

`matriks cloneMatriks(matriks MIn)`
`{I.S. MIn terdefinisi}`
`{F.S. mengembalikan matriks baru yang sama dengan MIn}`
	Metode untuk menduplikasi matriks

`matriks pertambahanMatriks(matriks M1, matriks M2)`
`{I.S. Matriks M1 dan M2 terdefinisi dan berukuran sama}`
`{F.S. mengembalikan matriks baru yang sama dengan hasil M1 + M2}`
	Metode untuk menghasilkan pertambahan dari dua matriks

`matriks penguranganMatriks(matriks M1, matriks M2)`
`{I.S. Matriks M1 dan M2 terdefinisi dan berukuran sama}`
`{F.S. mengembalikan matriks baru yang sama dengan hasil M1 - M2}`
	Metode yang menghasilkan pengurangan dari dua matriks

`matriks perkalianMatriks(matriks M1, matriks M2)`	Metode yang menghasilkan perkalian dari dua matriks

{I.S. Matriks M1 dan M2 terdefinisi dan ukuran M1 aXb dan ukuran M2 bXc} {F.S. mengembalikan matriks baru yang sama dengan hasil M1 x M2, berukuran aXc}	
matriks transpose(matriks MIn) {I.S. Matriks Min terdefinisi} {F.S. mengembalikan matriks baru yang sama dengan hasil transpose MIn}	Metode yang menghasilkan transpose dari sebuah matriks
matriks swapBaris(matriks MIn, int b1, int b2) {I.S. Matriks M1 terdefinisi, 0<=b1<=MIn.jumlahBaris-1, 0<=b2<=MIn.jumlahBaris-1} {F.S. mengembalikan matriks baru dengan baris b1 ditukar dengan baris b2}	Metode untuk menukar baris pada matriks
matriks barisXkonstanta(matriks MIn, int baris, double konstanta) {I.S. Matriks MIn terdefinisi, 0<=baris<=MIn.jumlahBaris-1} {F.S. mengembalikan matriks baru dengan baris ke index baris dikalikan konstanta}	Metode yang menghasilkan matriks untuk mengalikan sebuah baris dengan sebuah konstanta.
matriks barisMinKaliBaris(matriks MIn, int barisTujuan, int barisPengurang, double konstanta) {I.S. Matriks MIn terdefinisi, 0<=barisTujuan<=MIn.jumlahBaris-1, 0<=barisPengurang<=MIn.jumlahBaris-1} {F.S. mengembalikan matriks baru dengan baris dengan index barisTujuan ditukar dengan baris dengan index barisPengurang dikalikan konstanta}	Metode yang menghasilkan matriks dimana salah satu barisnya dikurangi baris lain (yang dikali suatu konstanta).
matriks compact0(matriks MIn) {I.S. Matriks MIn terdefinisi} {F.S. mengembalikan matriks baru dengan baris yang mengandung 0 beruntun dari kiri dipadatkan ke bawah dengan nol paling banyak berada di paling bawah secara berurutan}	Metode untuk mengembalikan matriks baru dengan 0 beruntun dari kiri dipadatkan ke bawah.
matriks gauss(matriks MIn) {I.S. Matriks MIn terdefinisi} {F.S. mengembalikan matriks baru yang berbentuk matriks eselon baris dari matriks MIn}	Metode untuk mengembalikan matriks baru yang bebentuk matriks eselon baris.
matriks gaussJordan(matriks MIn) {I.S. Matriks MIn terdefinisi} {F.S. mengembalikan matriks baru yang berbentuk matriks eselon tereduksi dari matriks MIn}	Metode untuk mengembalikan matriks baru yang berbentuk matriks eselon baris tereduksi.
matriks invIdentitas(matriks MIn) {I.S. Matriks MIn terdefinisi dan determinannya tidak nol} {F.S. mengembalikan matriks baru yang berbentuk matriks inverse dari matriks MIn}	Metode untuk mengembalikan matriks baru yang berbentuk matriks invers dari matriks awal.
double detOBE(matriks MIn) {I.S. Matriks MIn terdefinisi dan berbentuk persegi} {F.S. mengembalikan nilai determinan dari matriks MIn}	Metode untuk mengembalikan nilai determinan dari sebuah matriks menggunakan metode OBE.

matriks sliceLastCol(matriks MIn) {I.S. Matriks MIn terdefinisi dan MIn.jumlahKolom > 1} {F.S. mengembalikan matriks baru yang berbentuk matriks MIn yang dihilangkan kolom paling kanannya}	Metode untuk mengembalikan matriks baru yang dihilangkan kolom paling kanannya.
matriks sliceLastRow(matriks MIn) {I.S. Matriks MIn terdefinisi dan MIn.jumlahBaris > 1} {F.S. mengembalikan matriks baru yang berbentuk matriks MIn yang dihilangkan baris paling bawahnya}	Metode untuk mengembalikan matriks baru yang dihilangkan baris paling bawahnya.
matriks takeLastCol(matriks MIn) {I.S. Matriks MIn terdefinisi} {F.S. mengembalikan matriks baru yang berbentuk matriks berukuran satu kolom yang berisikan kolom paling kanan dari MIn}	Metode untuk mengembalikan matriks baru yang merupakan kolom terakhir dari matriks awal.
matriks takeLastRow(matriks MIn) {I.S. Matriks MIn terdefinisi} {F.S. mengembalikan matriks baru yang berbentuk matriks berukuran satu baris yang berisikan baris paling bawah dari MIn}	Metode untuk mengembalikan matriks baru yang merupakan baris terakhir dari matriks awal.
matriks concatKolom(matriks m1, matriks m2) {I.S. Matriks m1 dan m2 terdefinisi, jumlah kolom m1 dan m2 sama} {F.S. mengembalikan matriks baru yang berbentuk matriks gabungan dari m1 dan m2 dengan m2 berposisi di bawah m1}	Metode untuk mengembalikan matriks baru yang merupakan gabungan dari dua matriks dengan matriks kedua berada dibawah matriks pertama.
matriks slice(matriks MIn, int i, int j) {I.S. Matriks MIn terdefinisi, MIn berukuran minimal 2x2} {F.S. mengembalikan matriks baru yang berbentuk matriks MIn dengan baris i dan kolom j dihilangkan}	Metode untuk mengembalikan matriks baru yang memiliki ukuran i x j.
double cof(matriks MIn, int i, int j) {I.S. Matriks MIn terdefinisi, MIn berbentuk persegi dan berukuran minimal 2x2} {F.S. mengembalikan nilai kofaktor dari matriks minor pada elemen di baris ke i dan kolom ke j}	Metode untuk mengembalikan nilai kofaktor dari matriks minor pada entri baris ke-i, kolom ke-j.
matriks matCof(matriks MIn) {I.S. Matriks MIn terdefinisi, MIn berbentuk persegi dan berukuran minimal 2x2} {F.S. mengembalikan matriks baru yang merupakan matriks kofaktor dari MIn}	Metode untuk mengembalikan matriks baru yang merupakan matriks kofaktor dari matriks awal.
double detExCofRow0(matriks MIn) {I.S. Matriks MIn terdefinisi dan berbentuk persegi} {F.S. mengembalikan nilai determinan dari matriks MIn}	Metode untuk mengembalikan determinan dari matriks awal dengan memanfaatkan baris paling atas.
Double detExCofRow(matriks MIn) {I.S. Matriks MIn terdefinisi dan berbentuk persegi} {F.S. mengembalikan nilai determinan dari matriks MIn}	Metode untuk mengembalikan determinan dari matriks awal dengan memanfaatkan kolom paling kiri.

`matriks invIdentitas(matriks MIn)`
`{I.S. Matriks MIn terdefinisi dan determinannya tidak nol}`
`{F.S. mengembalikan matriks baru yang berbentuk matriks inverse dari matriks MIn}`
	Metode untuk mengembalikan matriks baru yang merupakan matriks invers dari matriks awal menggunakan metode matriks identitas.

`matriks cramerSwap(matriks a, matriks fx, int col)`
`{I.S. Matriks a terdefinisi dan berbentuk persegi, matriks fx memilki ukuran kolom 1 dan baris sama dengan matriks a. 0<=col<=MIn.jumlahKolom-1}`
`{F.S. mengembalikan matriks baru yang merupakan matriks a dengan kolom dengan index col diganti dengan fx}`
	Metode untuk mengembalikan matriks baru yang merupakan matriks a dengan kolom index col diganti dengan konstanta pada SPL.

`void detFile(matriks MIn, double Det)`
`{I.S. MIn terdefinisi dan berbentuk persegi}`
`{F.S. menuliskan matriks dan determinannya pada file yang terletak di folder test. Nama file akan berasal dari input pengguna pada fungsi}`
	Metode untuk menuliskan matriks dan determinannya pada file yang terletak di folder test.

`matriks tidyup(matriks MIn)`
`{I.S. matriks MIn terdefinisi}`
`{F.S. Mengembalikan matriks MIn dengan elemen -0.0000001<nilai<0.0000001 dibulatkan menjadi nol dan 0.9999999<nilai<1.0000001 dibulatkan menjadi 1}`
	Metode yang digunakan untuk merapikan hasil operasi yang dilakukan pada suatu matriks. Metode ini diperlukan sebab tipe data double tidak bisa merepresentasikan pecahan dengan tepat dan sering melakukan pembulatan. Khusus untuk nilai yang amat sangat mendekati nilai 0 atau nilai 1 perlu disederhanakan agar memudahkan program menjalankan fungsi yang lain. Beberapa program akan mengalami malfungsi jika tidak dilakukan pembulatan menggunakan fungsi ini.

3.3 Class SPL

3.3.1 Atribut

Class ini tidak memiliki atribut.

3.3.2 Metode

`void solveSPL(matriks m)`
`{I.S. Matriks terdefinisi, berbentuk gauss atau gauss jordan}`
`{F.S. Menuliskan solusi dari SPL di terminal sesuai dengan jenis solusi matriks tersebut}`
	Metode untuk menuliskan solusi SPL di terminal sesuai dengan salah satu dari tiga jenis solusi matriks tersebut.

`void solveSPLFile(matriks m)`
`{I.S. Matriks terdefinisi}`
`{F.S. Menuliskan solusi dari SPL di file pada folder test sesuai dengan jenis solusi matriks`
	Metode untuk menuliskan solusi SPL di file sesuai dengan salah satu dari tiga jenis solusi matriks tersebut.

tersebut. Nama file akan berasal dari input pengguna pada fungsi}	
int checkSPL(matriks m) {I.S. Matriks terdefinisi, berbentuk gauss atau gauss jordan} {F.S. Mengembalikan integer bernilai 1..4 yang merupakan kode dari solusi yang diperlukan SPL tersebut, 0 berarti matriks kosong (berisi 0 semua), 1 berarti matriks memiliki solusi unik, 2 berarti matriks memiliki solusi banyak, dan 3 berarti matriks tidak memiliki solusi}	Metode untuk menghasilkan angka yang merepresentasikan jenis solusi yang dimiliki oleh SPL tersebut. Dengan rincian 0 berarti matriks kosong, 1 berarti SPL solusi unik 2. berarti SPL solusi banyak 3 berarti tidak punya solusi
void SolusiKosong(matriks MIn) {I.S. Matriks terdefinisi, berisikan semua 0} {F.S. Menuliskan solusi dari SPL di terminal}	Metode untuk menuliskan hasil solusi dari SPL di terminal jika solusi merupakan solusi kosong
void SolusiUnik(matriks MIn) {I.S. Matriks terdefinisi, berbentuk gauss atau gauss jordan, memiliki solusi unik secara matematis} {F.S. Menuliskan solusi dari SPL di terminal}	Metode untuk menuliskan hasil solusi dari SPL di terminal jika solusi merupakan solusi unik
double recursion(int toplimit, int bottom limit, int row, int varCol, double arrayHasil[], String arrayString[], matriks Min) {I.S. toplimit adalah indeks terakhir dari arrayhasil, bottom limit adalah batas bawah looping atau elemen, row adalah letak indeks baris elemen pada Min, var col adalah letak indeks kolom nilai variabel yang sedang dicari koefisiennya. arrayHasil terdefinisi, arrayString terdefinisi, matriks MIn berbentuk matriks eselon atau matriks eselon teraugmentasi} {F.S. mengembalikan nilai koefisien yang dibutuhkan untuk satu nilai variabel di persamaan parametrik matriks gauss atau gauss jordan}	Metode untuk mendapatkan nilai koefisien dari variabel tidak diketahui pada persamaan parametrik. Metode ini bersifat rekursif dan menjumlahkan nilai koefisien yang dikandung tiap elemen pada suatu baris.
void SolusiBanyak(matriks MIn) {I.S. Matriks terdefinisi, berbentuk gauss atau gauss jordan, memiliki solusi banyak secara matematis} {F.S. Menuliskan solusi dari SPL di terminal}	Metode untuk menuliskan hasil solusi dari SPL di terminal jika solusi merupakan solusi banyak
void SolusiNone(matriks MIn) {I.S. Matriks terdefinisi, berbentuk gauss atau gauss jordan, tidak memiliki solusi secara matematis} {F.S. Menuliskan solusi dari SPL di terminal}	Metode untuk menuliskan hasil solusi dari SPL di terminal jika SPL tidak mempunyai solusi.
void SolusiKosongFile(matriks MIn, String filename) {I.S. Matriks terdefinisi, berisikan semua 0} {F.S. Menuliskan solusi dari SPL di file pada folder test sesuai dengan jenis solusi matriks tersebut. Parameter filename berisikan adalah nama file}	Metode untuk menuliskan hasil solusi dari SPL di file pada folder test jika solusi merupakan solusi kosong

`void SolusiUnikFile(matriks MIn, String filename)`
`{I.S. Matriks terdefinisi, berbentuk gauss atau gauss jordan, memiliki solusi unik secara matematis}`
`{F.S. Menuliskan solusi dari SPL di file pada folder test sesuai dengan jenis solusi matriks tersebut. Parameter filename berisikan adalah nama file}`
	Metode untuk menuliskan hasil solusi dari SPL di file pada folder test jika solusi merupakan solusi unik

`void SolusiBanyakFile(matriks MIn, String filename)`
`{I.S. Matriks terdefinisi, berbentuk gauss atau gauss jordan, memiliki solusi banyak secara matematis}`
`{F.S. Menuliskan solusi dari SPL di file pada folder test sesuai dengan jenis solusi matriks tersebut. Parameter filename berisikan adalah nama file}`
	Metode untuk menuliskan hasil solusi dari SPL di file pada folder test jika solusi merupakan solusi banyak

`void SolusiNoneFile(matriks MIn, String filename)`
`{I.S. Matriks terdefinisi, berbentuk gauss atau gauss jordan, tidak memiliki solusi secara matematis}`
`{F.S. Menuliskan solusi dari SPL di file pada folder test sesuai dengan jenis solusi matriks tersebut. Parameter filename berisikan adalah nama file}`
	Metode untuk menuliskan hasil solusi dari SPL di file pada folder test jika SPL tidak mempunyai solusi.

`int cari1(matriks MIn, int baris)`
`{I.S. Matriks terdefinisi, berbentuk gauss atau gauss jordan}`
`{F.S. mengembalikan indeks kolom leading 1 pada baris ke indeks baris pada matriks MIn}`
	Metode untuk mencari indeks dari leading 1 pada suatu baris di matriks gauss jordan.

`matriks removebaris0(matriks MIn)`
`{I.S. Matriks terdefinisi, berbentuk gauss atau gauss Jordan }`
`{F.S. mengembalikan matriks MIn dengan baris yang berisi 0 saja dihilangkan}`
	Metode untuk menghilangkan baris 0 di paling bawah matriks.

`Boolean solvable(matriks MIn)`
`{I.S. Matriks terdefinisi, berbentuk gauss atau gauss jordan}`
`{F.S. mengembalikan true jika MIn dapat disolusikan secara matematis}`
	Metode untuk menentukan ada tidaknya solusi dari sebuah SPL, hal ini diambil dari bentuk yang ada di dasar teori.

`void SolveInverse(matriks M)`
`{I.S. Matriks terdefinisi, berukuran jumlah kolom = jumlah baris + 1, determinan matriks tanpa kolom ujung tidak sama dengan nol}`
`{F.S. Menuliskan solusi dari SPL di terminal}`
	Metode untuk menghasilkan solusi SPL menggunakan metode inverse lalu menuliskan solusinya di terminal.

`void SolveCramer(matriks M)`
`{I.S. Matriks terdefinisi, berukuran jumlah kolom = jumlah baris + 1, determinan matriks tanpa kolom ujung tidak sama dengan nol}`
`{F.S. Menuliskan solusi dari SPL di terminal}`
	Metode untuk menghasilkan solusi SPL menggunakan metode cramer lalu menuliskan solusinya di terminal.

`void SolveInverseFile(matriks M)`	Metode untuk menghasilkan solusi SPL menggunakan

`{I.S. Matriks terdefinisi, berukuran jumlah kolom = jumlah baris + 1, determinan matriks tanpa kolom ujung tidak sama dengan nol}` `{F.S. Menuliskan solusi dari SPL di file pada folder test sesuai dengan jenis solusi matriks tersebut. Nama file akan berasal dari input pengguna pada fungsi}`	metode inverse lalu menuliskan solusinya di sebuah file pada folder test.

`void SolveCramerFile(matriks M)`
`{I.S. Matriks terdefinisi, berukuran jumlah kolom = jumlah baris + 1, determinan matriks tanpa kolom ujung tidak sama dengan nol}`
`{F.S. Menuliskan solusi dari SPL di file pada folder test sesuai dengan jenis solusi matriks tersebut. Nama file akan berasal dari input pengguna pada fungsi}`
	Metode untuk menghasilkan solusi SPL menggunakan metode cramer lalu menuliskan solusinya di sebuah file pada folder test.

3.3 Class InterpolasiPolinom

Di class ini, digunakan beberapa istilah yang akan digunakan berulang-ulang:

Tipe Integer		
Istilah	Keterangan	
n	derajat polinomial yang akan diinterpolasi	
Tipe Double		
Istilah	Keterangan	
a	nilai x yang akan diinterpolasi	
Tipe Matriks		
Istilah	Keterangan	Format
$stdInput_{(n+2)\times 2}$	standar masukan pengguna, baris 0..n berisi titik masukan, baris n + 1 berisi nilai x yang akan diinterpolasi	$\begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \\ a & \end{bmatrix}$
$x_{(n+1)\times 1}$	kumpulan nilai absis dari titik masukan pengguna	$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$
$fx_{(n+1)\times 1}$	sama seperti matriks **y** (lihat Teori Singkat Interpolasi Polinom)	$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$
$xi_{(n+1)\times(n+1)}$	sama seperti matriks **X** (lihat Teori Singkat Interpolasi Polinom)	$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix}$
$ai_{(n+1)\times 1}$	sama seperti matriks **a** (lihat Teori Singkat Interpolasi Polinom)	$\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}$

3.3.1 Atribut

Class ini tidak memiliki atribut.

3.3.2 Metode

`matriks stdInputKeyboard()`
`{I.S. -}`
`{F.S. Mengembalikan matriks masukan (sesuai format matriks stdInput) dari pengguna yang akan diolah untuk interpolasi polinom}`
	Metode untuk menerima matriks masukan dari pengguna yang akan diolah untuk interpolasi polinom dan mengembalikan matriks masukan tersebut.

`matriks x(matriks stdInput)`
`{I.S. Matriks stdInput terdefinisi dan sudah sesuai sesuai formatnya}`
`{F.S. Mengembalikan matriks x (sudah sesuai format) yang diambil dari matriks stdInput}`
	Metode yang mengembalikan matriks x yang merupakan bagian dari matriks stdInput.

`matriks fx(matriks stdInput)`
`{I.S. Matriks stdInput terdefinisi dan sudah sesuai sesuai formatnya}`
`{F.S. Mengembalikan matriks fx (sudah sesuai format) yang diambil dari matriks stdInput}`
	Metode yang mengembalikan matriks fx yang merupakan bagian dari matriks stdInput.

`matriks a(matriks stdInput)`
`{I.S. Matriks stdInput terdefinisi dan sudah sesuai formatnya}`
`{F.S. Mengembalikan matriks a (sudah sesuai format) dari matriks stdInput}`
	Metode yang mengembalikan matriks a yang merupakan bagian dari matriks stdInput.

`matriks xi(matriks x)`
`{I.S. Matriks x terdefinisi dan sudah sesuai formatnya}`
`{F.S. Mengembalikan matriks xi (sudah sesuai format) yang dibuat dari matriks x}`
	Metode yang mengembalikan matriks xi yang dibuat dengan mengaplikasikan beberapa operasi pada matriks x.

`matriks ai(matriks xi, matriks fx)`
`{I.S. Matriks xi dan matriks fx terdefinisi dan sudah sesuai formatnya}`
`{F.S. Mengembalikan matriks ai (sudah sesuai format) yang dibuat dengan mengaplikasikan metode Gauss}`
	Metode yang mengembalikan matriks ai yang dibuat dengan mengaplikasikan metode Gauss pada matriks augmented xi|fx.

`double fa(matriks ai, double a)`
`{I.S. Matriks ai terdefinisi dan sudah sesuai formatnya, a terdefinisi}`
`{F.S. Mengembalikan nilai polinom jika disubstitusikan x = a}`
	Metode yang mengembalikan nilai polinom jika disubstitusikan x = a. Kembalian fungsi bernilai valid jika a berada dalam range data interpolasi.

`String fxString(matriks ai)`
`{I.S. Matriks ai terdefinisi dan sudah sesuai formatnya}`
`{F.S. Mengembalikan string berupa penjabaran polinom hasil interpolasi}`
	Metode yang mengembalikan string berupa penjabaran polinom hasil interpolasi.

`void IPFile(matriks ai, double a)`
`{I.S. Matriks ai terdefinisi dan sudah sesuai formatnya, a berada di range data interpolasi}`
`{F.S. Menuliskan hasil interpolasi ke dalam file .txt.}`
	Metode yang menuliskan hasil interpolasi (penjabaran fungsi beserta substitusinya) ke dalam file .txt.

3.4 Class BicubicInterpolation

Di class ini, digunakan beberapa istilah yang akan digunakan berulang-ulang:

Tipe Double	
Istilah	Keterangan

a, b	nilai x dan y yang akan diinterpolasi	
Tipe Matriks		
Istilah	Keterangan	Format
$stdInput_{5\times 4}$	bentuk standar masukan pengguna, baris 0..3 berisi nilai f(x,y), baris 4 berisi nilai x dan y yang akan diinterpolasi	$\begin{bmatrix} f(-1,-1) & f(-1,0) & f(-1,1) & f(-1,2) \\ f(0,-1) & f(0,0) & f(0,1) & f(0,2) \\ f(1,-1) & f(1,0) & f(1,1) & f(1,2) \\ f(2,-1) & f(2,0) & f(2,1) & f(2,2) \\ a & b & & \end{bmatrix}$
$fxy4x4_{4\times 4}$	sama seperti matriks **M** (lihat Teori Singkat Bicubic Interpolation)	$\begin{bmatrix} f(-1,-1) & f(0,-1) & f(1,-1) & f(2,-1) \\ f(-1,0) & f(0,0) & f(1,0) & f(2,0) \\ f(-1,1) & f(0,1) & f(1,1) & f(2,1) \\ f(-1,2) & f(0,2) & f(1,2) & f(2,2) \end{bmatrix}$
$fxy_{16\times 1}$	sama seperti matriks **y** (lihat Teori Singkat Bicubic Interpolation)	$\begin{bmatrix} f(-1,-1) \\ f(0,-1) \\ \vdots \\ f(1,2) \\ f(2,2) \end{bmatrix}$
$xiyj_{16\times 16}$	sama seperti matriks **X** (lihat Teori Singkat Bicubic Interpolation)	$\begin{bmatrix} 1 & -1 & \dots & -1 & 1 \\ 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & 1 & \cdots & 8 & 8 \\ 1 & 2 & \cdots & 32 & 64 \end{bmatrix}$
$aij_{16\times 1}$	sama seperti matriks **a** (lihat Teori Singkat Bicubic Interpolation)	$\begin{bmatrix} a_{00} \\ a_{10} \\ \vdots \\ a_{23} \\ a_{33} \end{bmatrix}$

3.4.1 Atribut

Class ini tidak memiliki atribut

3.4.2 Metode

`double a(matriks stdInput)`
`{I.S. Matriks stdInput terdefinisi dan sudah sesuai sesuai formatnya}`
`{F.S. Mengembalikan double a yang diambil dari matriks stdInput}`
	Metode yang mengembalikan double a yang merupakan bagian dari matriks stdInput.

`matriks fxy4x4(matriks stdInput)`
`{I.S. Matriks stdInput terdefinisi dan sudah sesuai sesuai formatnya}`
`{F.S. Mengembalikan matriks fxy4x4 (sudah sesuai format) yang dibuat dari matriks stdInput}`
	Metode yang mengembalikan matriks fxy4x4 yang merupakan bagian dari matriks stdInput.

`matriks fxy(matriks fxy4x4)`
`{I.S. Matriks fxy4x4 terdefinisi dan sudah sesuai sesuai formatnya}`
`{F.S. Mengembalikan matriks fxy (sudah sesuai format) yang dibuat dari matriks stdInput}`
	Metode yang mengembalikan matriks fxy yang dibuat dari matriks fxy4x4.

`matriks xiyj()` `{I.S. -}`	Metode yang mengembalikan matriks xiyj.

`{F.S. Mengembalikan matriks xiyj (sudah sesuai format)}`	

`matriks aij(matriks fxy)`
`{I.S. Matriks fxy terdefinisi dan sudah sesuai formatnya}`
`{F.S. Mengembalikan matriks aij yang dihitung dengan metode inverse}`
	Metode yang mengembalikan matriks aij yang dibuat dengan mengaplikasikan metode inverse: $aij = (xiyj)^{-1} fxy$

`double bicIntpol(matriks aij, double a, double b)`
`{I.S. Matriks aij terdefinisi dan sudah sesuai formatnya, a dan b terdefinisi}`
`{F.S. Mengembalikan nilai fungsi interpolasi jika disubstitusikan x = a dan y = b}`
	Metode yang mengembalikan nilai fungsi interpolasi jika disubstitusikan x = a dan y = b. Kembalian fungsi bernilai valid jika a dan b berada di range [0,1].

`void BIFile(matriks aij, double a, double b)`
`{I.S. Matriks aij terdefinisi dan sudah sesuai formatnya, a dan b berada di [0,1]}`
`{F.S. Menuliskan hasil interpolasi ke dalam file .txt.}`
	Metode yang menuliskan hasil interpolasi bikubik (substitusi x = a dan y = b) ke dalam file .txt.

3.5. Class RegresiLinierBerganda

Di class ini, digunakan beberapa istilah yang akan digunakan berulang-ulang:

Tipe Integer		
Istilah	Keterangan	
n	jumlah peubah x	
m	jumlah sampel	
Tipe Matriks		
Istilah	Keterangan	Format
$stdInput_{(m+1)\times(n+1)}$	standar masukan pengguna, baris 0..(m - 1) berisi titik sampel, baris m berisi data yang akan diregresi	$\begin{bmatrix} x_{11} & x_{21} & \cdots & x_{n1} & y_1 \\ x_{12} & x_{22} & \cdots & x_{n2} & y_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{1m} & x_{2m} & \cdots & x_{nm} & y_m \end{bmatrix}$
$xnm_{m\times(n+1)}$	sama seperti matriks **X** (lihat Teori Singkat Bicubic Interpolation)	$\begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{n1} \\ 1 & x_{12} & x_{22} & \cdots & x_{n2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1m} & x_{2m} & \cdots & x_{nm} \end{bmatrix}$
$ym_{m\times1}$	sama seperti matriks **y** (lihat Teori Singkat Bicubic Interpolation)	$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$
$xk_{1\times n}$	data nilai $x_1, x_2, \dots, x_n$ yang akan diregresi	$\begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}$
$b_{(n+1)\times1}$	sama seperti matriks $\hat{\boldsymbol{b}}$ (lihat Teori Singkat Regresi Linear Berganda)	$\begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}$

3.5.1 Atribut

Class ini tidak memiliki atribut

3.5.2 Metode

`matriks stdInputKeyboard()`
`{I.S. -}`
`{F.S. Mengembalikan matriks masukan (sesuai format matriks stdInput) dari pengguna yang akan diolah untuk regresi linear berganda}`
	Metode untuk menerima matriks masukan dari pengguna yang akan diolah untuk regresi linear berganda dan mengembalikan matriks masukan tersebut.

`matriks xnm(matriks stdInput)`
`{I.S. Matriks stdInput terdefinisi dan sudah sesuai sesuai formatnya}`
`{F.S. Mengembalikan matriks xnm (sudah sesuai format) yang diambil dari matriks stdInput}`
	Metode yang mengembalikan matriks xnm yang merupakan bagian dari matriks stdInput.

`matriks ym(matriks stdInput)`
`{I.S. Matriks stdInput terdefinisi dan sudah sesuai sesuai formatnya}`
`{F.S. Mengembalikan matriks ym (sudah sesuai format) yang diambil dari matriks stdInput}`
	Metode yang mengembalikan matriks ym yang merupakan bagian dari matriks stdInput.

`matriks xk(matriks stdInput)`
`{I.S. Matriks stdInput terdefinisi dan sudah sesuai sesuai formatnya}`
`{F.S. Mengembalikan matriks xk (sudah sesuai format) yang diambil dari matriks stdInput}`
	Metode yang mengembalikan matriks xk yang merupakan bagian dari matriks stdInput.

`matriks b(matriks xnm, matriks ym)`
`{I.S. Matriks xnm dan matriks ym terdefinisi dan sudah sesuai formatnya}`
`{F.S. Mengembalikan matriks b (sudah sesuai format) yang dibuat dengan mengaplikasikan metode Gauss}`
	Metode yang mengembalikan matriks b yang dibuat dengan mengaplikasikan metode Gauss pada matriks augmented $((xnm)^T \times xnm) \mid ((xnm)^T \times ym)$

`double fxk(matriks xk, matriks b)`
`{I.S. Matriks xk dan matriks b terdefinisi dan sudah sesuai formatnya}`
`{F.S. Mengembalikan nilai fungsi regresi jika disubstitusikan xk}`
	Metode yang mengembalikan nilai polinom jika disubstitusikan xk.

`String fxkString(matriks b)`
`{I.S. Matriks b terdefinisi dan sudah sesuai formatnya}`
`{F.S. Mengembalikan string berupa penjabaran fungsi regresi dan nilai f(xk)}`
	Metode yang mengembalikan string berupa penjabaran fungsi hasil regresi.

`void RLBFile(matriks xk, matriks b)`
`{I.S. Matriks xk dan matriks b terdefinisi dan sudah sesuai formatnya}`
`{F.S. Menuliskan hasil regresi linear berganda ke dalam file .txt.}`
	Metode yang menuliskan hasil perhitungan regresi (penjabaran fungsi beserta substitusinya) ke dalam file .txt.

3.6 Class ImageUtil

3.6.1 Atribut

Class ini tidak memiliki atribut.

3.6.2 Metode

`matriks loadImage(String filename)`
`{I.S. matriks sembarang, string berisi nama file .png/jpg/jpeg yang terletak di folder test}`
`{F.S. mengembalikan matriks yang berisikan value warna hitam putih dari gambar}`
	Metode yang digunakan untuk mengambil value hitam putih dari gambar dengan cara meratararatakan nilai merah, hijau, dan biru pada tiap pixel dan mengembalikannya dalam bentuk tipe data matriks

`void writeImage(String filename, matriks m)`
`{I.S. matriks bebas, string berisi nama file .png yang terletak di folder test}`
`{F.S. menghasilkan file .png di folder test, nama file adalah filename.png}`
	Metode yang digunakan untuk mengambil value hitam putih dari sebuah matriks lalu memasukkannya pada tiap pixel serta dimasukkan di sebuah file yang terletak di folder test.

3.7 Class ImageUpsc

3.7.1 Atribut

Class ini tidak memiliki atribut

3.7.2 Metode

`matriks zoning(matriks MIn, int sizex, int sizey, int startx, int starty)` `{I.S. matriks bebas, berukuran dengan jumlah kolom lebih besar dari startx + sizex dan jumlah baris lebih besar dari starty + sizey, startx dan starty lebih besar sama dengan nol dan tidak lebih besar dari indeks maksimum MIn, sizex + startx dan sizey + starty tidak melebihi indeks maksimum kolom MIn dan baris MIn}` `{F.S. Mengembalikan matriks berukuran sizey X sizex dari MIn dengan pojok kiri atas berasal elemen berindex startx, starty}`	Metode yang digunakan untuk mengambil submatriks dari matriks utama.

`void paddingAtas(matriks MIn)`
`{I.S. matriks terdefinisi, ukuran baris < capacity}`
`{F.S. matriks ditambah baris di atas yang bernilai sama dengan baris teratas sebelumnya}`
	Metode yang digunakan untuk menambahkan padding di atas matriks dengan nilai yang sama dengan baris yang paling atas. Padding dilakukan pada matriks yang sama menggunakan pointer dengan alasan meningkatkan performa.

`void paddingBawah(matriks MIn)`
`{I.S. matriks terdefinisi, ukuran baris < capacity}`
`{F.S. matriks ditambah baris di bawah yang bernilai sama dengan baris terbawah sebelumnya}`
	Metode yang digunakan untuk menambahkan padding di bawah matriks dengan nilai yang sama dengan baris yang paling bawah. Padding dilakukan pada matriks yang sama menggunakan pointer dengan alasan meningkatkan performa.

`void paddingKanan(matriks MIn)`
`{I.S. matriks terdefinisi, ukuran kolom < capacity}`
`{F.S. matriks ditambah kolom di kanan yang bernilai sama dengan kolom terkanan sebelumnya}`
	Metode yang digunakan untuk menambahkan padding di kanan matriks dengan nilai yang sama dengan baris yang paling kanan. Padding dilakukan pada matriks yang sama menggunakan pointer dengan alasan meningkatkan performa.

`void paddingKiri(matriks MIn)`
`{I.S. matriks terdefinisi, ukuran kolom < capacity}`
`{F.S. matriks ditambah kolom di kiri yang bernilai sama dengan kolom terkanan sebelumnya}`
	Metode yang digunakan untuk menambahkan padding di kiri matriks dengan nilai yang sama dengan baris yang paling kiri. Padding dilakukan pada matriks yang sama menggunakan pointer dengan alasan meningkatkan performa.

`void padding(matriks MIn)`
`{I.S. matriks terdefinisi, ukuran baris kolom < capacity}`
`{F.S. matriks ditambah kolom di kiri dan kanan, dan baris di atas dan bawah, yang bernilai sama dengan pojok sebelumnya}`
	Metode yang digunakan untuk menambahkan padding semua sisi matriks dengan nilai yang sama pada ujung ujung matriks. Padding dilakukan pada matriks yang sama menggunakan pointer dengan alasan meningkatkan performa.

`matriks interpolateStandar(matriks MIn, int startx, int starty, matriks inversedXiYj)` `{I.S. matriks MIn terdefinisi, berukuran dengan jumlah kolom lebih besar dari startx + sizex dan jumlah baris lebih besar dari starty + sizey, startx dan starty lebih besar sama dengan nol dan tidak lebih besar dari indeks maksimum MIn, sizex + startx dan sizey + starty tidak melebihi indeks maksimum kolom MIn dan baris MIn, inversedXiYj merupakan hasil inverse dari matriks XiYj interpolasi bicubic}` `{F.S. Mengembalikan matriks berukuran 3x3 dari hasil interpolasi zoning MIn dengan pojok kiri atas berasal elemen berindex startx, starty}`	Metode yang digunakan untuk menginterpolasi matriks 4x4 dan mengembalikan matriks 3x3 dari hasil interpolasi dengan nilai x = 0, 0.5, dan 1 dan y = 0, 0.5, dan 1. Digunakan untuk menginterpolasi gambar secara pada bagian umum.
`matriks interpolateTengahX(matriks MIn, int startx, int starty, matriks inversedXiYj)` `{I.S. matriks MIn terdefinisi, berukuran dengan jumlah kolom lebih besar dari startx + sizex dan jumlah baris lebih besar dari starty + sizey, startx dan starty lebih besar sama dengan nol dan tidak lebih besar dari indeks maksimum MIn, sizex + startx dan sizey + starty tidak melebihi indeks maksimum kolom MIn dan baris MIn, inversedXiYj merupakan hasil inverse dari matriks XiYj interpolasi bicubic}` `{F.S. Mengembalikan matriks berukuran 3x4 dari hasil interpolasi zoning MIn dengan pojok kiri atas berasal elemen berindex startx, starty}`	Metode yang digunakan untuk menginterpolasi matriks 4x4 dan mengembalikan matriks 3x4 dari hasil interpolasi dengan nilai x = 0, 0.33, 0.66, dan 1 dan y = 0, 0.5, dan 1. Digunakan untuk menginterpolasi gambar pada bagian tengah sumbu X.

`matriks interpolateTengahY(matriks MIn, int startx, int starty, matriks inversedXiYj)` `{I.S. matriks MIn terdefinisi, berukuran dengan jumlah kolom lebih besar dari startx + sizex dan jumlah baris lebih besar dari starty + sizey, startx dan starty lebih besar sama dengan nol dan tidak lebih besar dari indeks maksimum MIn, sizex + startx dan sizey + starty tidak melebihi indeks maksimum kolom MIn dan baris MIn, inversedXiYj merupakan hasil inverse dari matriks XiYj interpolasi bicubic}` `{F.S. Mengembalikan matriks berukuran 4x3 dari hasil interpolasi zoning MIn dengan pojok kiri atas berasal elemen berindex startx, starty}`	Metode yang digunakan untuk menginterpolasi matriks 4x4 dan mengembalikan matriks 4x3 dari hasil interpolasi dengan nilai x = 0, 0.5, dan 1 dan y = 0, 0.33, 0.66, dan 1. Digunakan untuk menginterpolasi gambar pada bagian tengah sumbu Y.
`matriks interpolateTengahXY(matriks MIn, int startx, int starty, matriks inversedXiYj)` `{I.S. matriks MIn terdefinisi, berukuran dengan jumlah kolom lebih besar dari startx + sizex dan jumlah baris lebih besar dari starty + sizey, startx dan starty lebih besar sama dengan nol dan tidak lebih besar dari indeks maksimum MIn, sizex + startx dan sizey + starty tidak melebihi indeks maksimum kolom MIn dan baris MIn, inversedXiYj merupakan hasil inverse dari matriks XiYj interpolasi bicubic}` `{F.S. Mengembalikan matriks berukuran 4x4 dari hasil interpolasi zoning MIn dengan pojok kiri atas berasal elemen berindex startx, starty}`	Metode yang digunakan untuk menginterpolasi matriks 4x4 dan mengembalikan matriks 4x3 dari hasil interpolasi dengan nilai x = 0, 0.33, 0.66, dan 1 dan y = 0, 0.33, 0.66, dan 1. Digunakan untuk menginterpolasi gambar pada bagian tengah sumbu X dan Y .

`matriks interpolate2x(matriks MIn)`
`{I.S. matriks MIn terdefinisi, berukuran minimal 2x2}`
`{F.S. Mengembalikan matriks berukuran dua kali lipat matriks MIn berdasarkan hasil interpolasi}`
	Metode yang digunakan untuk menginterpolasi matriks nilai suatu matriks menjadi dua kali lipat baris dan dua kali lipat kolom jika genap dan dua kali lipat baris-1 dan dua kali lipat kolom-1 jika ganjil. Interpolasi dilakukan secara bertahap dan menggunakan zoning 4x4 dan diinterpolasikan setiap zona sampai selesai.

`void assignLocation(matriks Main, matriks submatriks, int x, int y)`
`{I.S. matriks MIn terdefinisi dan minimal berukuran sama dengan submatriks, x dan y ditambah ukuran submatriks tidak melebihi ukuran matriks main}`
`{F.S. Mengembalikan matriks main dengan submatriks dimasukkan dengan elemen pojok`
	Metode yang digunakan untuk memasukkan submatriks pada suatu matriks utama. Digunakan untuk memasukkan hasil interpolasi.

`atas kiri submatriks berada di kolom indeks x dan baris berindeks y}`	

`matriks linearize(matriks MIn)`
`{I.S. matriks MIn terdefinisi}`
`{F.S. Mengembalikan matriks yang merupakan matriks kolom dari MIn}`
	Metode yang digunakan untuk menjadikan linear matriks yang berisi nilai akhir untuk keperluan interpolasi bicubic.

`matriks aijOptimized(matriks fxy, matriks InversedXiYj)`
`{I.S. matriks fxy merupakan matriks kolom dari matriks 4x4, matriks matriks inversedXiYj merupakan hasil inverse dari matriks XiYj interpolasi bicubic}`
`{F.S. Mengembalikan matriks yang merupakan matriks kolom dari koefisien yang digunakan pada persamaan interpolasi bicubic}`
	Metode yang digunakan untuk mencari koefisien yang digunakan pada persamaan interpolasi. Perbedaan dengan aij yang ada pada class BicubicInterpolation adalah parameter InversedXiYj yang merupakan input tambahan. Hal ini dilakukan untuk meningkatkan performa karena inverse membutuhkan waktu yang cukup lumayan. Inverse akan dilakukan pada fungsi interpolate2x lalu disimpan pada memori dan diakses secara langsung di fungsi ini sehingga tidak perlu melakukan inverse secara berkali kali setiap pemanggilan.

BAB IV
EKSPERIMEN

4.1. Solusi Sistem Persamaan Linear $Ax = b$

<table>
<tr><th colspan="2">Soal</th><th>Metode</th><th>Output</th></tr>
<tr><td rowspan="4">a)</td><td rowspan="4">$A = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix},$
$b = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$</td><td>Gauss</td><td>Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 1_A.txt

Persamaan linear tidak memiliki solusi.</td></tr>
<tr><td>Gauss-Jordan</td><td>Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 1_A.txt

Persamaan linear tidak memiliki solusi.</td></tr>
<tr><td>Matriks Balikan</td><td>Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 1_A.txt

Matriks tidak memiliki inverse sehingga tidak dapat disolusikan.</td></tr>
<tr><td>Cramer</td><td>Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 1_A.txt

Matriks tidak memiliki inverse sehingga tidak dapat disolusikan.</td></tr>
<tr><td colspan="4">Analisis: Solusi dari SPL sama untuk semua metode, yaitu persamaan linier tidak memiliki solusi. karena terdapat baris pada matriks augmented yang berisi angka 0 (metode Gauss dan Gauss-Jordan) dan tidak dapat disolusikan karena tidak memiliki matriks invers dan determinannya nol (metode matriks balikan dan cramer).</td></tr>
<tr><td rowspan="4">b)</td><td rowspan="4">$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}$
$, b = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$</td><td>Gauss</td><td>Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 1_B.txt

x1 = 3.0+1.00S,
x2 = +2.00S,
x3 = T,
x4 = -1.0+1.00S,
x5 = S,</td></tr>
<tr><td>Gauss-Jordan</td><td>Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 1_B.txt

x1 = 3.0+1.00S,
x2 = +2.00S,
x3 = T,
x4 = -1.0+1.00S,
x5 = S,</td></tr>
<tr><td>Matriks Balikan</td><td>Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 1_B.txt

Matriks memerlukan 5 persamaan untuk dapat disolusikan.</td></tr>
<tr><td>Cramer</td><td>Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 1_B.txt

Matriks memerlukan 5 persamaan untuk dapat disolusikan.</td></tr>
<tr><td colspan="4">Analisis: Solusi dari SPL sama untuk metode eliminasi Gauss dan Gauss-Jordan. Solusi menggunakan matriks balikan dan kaidah Cramer membutuhkan syarat khusus dimana matriks masukan harus berupa matriks persegi.</td></tr>
</table>

<table>
<tr>
<td rowspan="4">c)</td>
<td rowspan="4">$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix},$$
$$b = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$</td>
<td>Gauss</td>
<td>Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 1_C.txt

x1 = U,
x2 = 1.0-1.00S,
x3 = T,
x4 = -2.0-1.00S,
x5 = 1.0+1.00S,
x6 = S,</td>
</tr>
<tr>
<td>Gauss-Jordan</td>
<td>Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 1_C.txt

x1 = U,
x2 = 1.0-1.00S,
x3 = T,
x4 = -2.0-1.00S,
x5 = 1.0+1.00S,
x6 = S,</td>
</tr>
<tr>
<td>Matriks Balikan</td>
<td>Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 1_C.txt

Matriks memerlukan 6 persamaan untuk dapat disolusikan.</td>
</tr>
<tr>
<td>Cramer</td>
<td>Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 1_C.txt

Matriks memerlukan 6 persamaan untuk dapat disolusikan.</td>
</tr>
<tr>
<td colspan="4">Analisis: Solusi dari SPL sama untuk metode eliminasi Gauss dan Gauss-Jordan. Solusi menggunakan matriks balikan dan kaidah Cramer membutuhkan syarat khusus dimana matriks masukan harus berupa matriks persegi.</td>
</tr>
<tr>
<td rowspan="2">d)</td>
<td rowspan="2">$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n+2} \end{bmatrix},$$
$$b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$</td>
<td>Gauss</td>
<td>N = 6

Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 1_D_1.txt

x1 = 8.080835703347605,
x2 = -23.558849267122852,
x3 = -31.735397068465872,
x4 = 108.01793918179216,
x5 = -43.699409319623655,
x6 = -17.952854955151796,

N = 10

Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 1_D_2.txt

x1 = 6.239259018975206,
x2 = 8.751527031015044,
x3 = -157.5067810530248,
x4 = 243.84183754456302,
x5 = -69.71543733477822,
x6 = -3.140335015088084,
x7 = 88.84732450354912,
x8 = -135.63486531066155,
x9 = -76.96926612783591,
x10 = 91.34578884524092,</td>
</tr>
<tr>
<td>Gauss Jordan</td>
<td>N = 6</td>
</tr>
</table>

<table>
<tr>
<td rowspan="3"></td>
<td rowspan="3"></td>
<td></td>
<td>
<pre>
Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 1_D_1.txt

x1 = 8.080835703347597,
x2 = -23.558849267122838,
x3 = -31.73539706846589,
x4 = 108.01793918179217,
x5 = -43.699409319623655,
x6 = -17.952854955151796,
</pre>
N = 10
<pre>
Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 1_D_2.txt

x1 = 6.239259018975247,
x2 = 8.751527031015257,
x3 = -157.5067810530254,
x4 = 243.84183754456262,
x5 = -69.71543733477122,
x6 = -3.140335015093683,
x7 = 88.84732450354704,
x8 = -135.634865310664,
x9 = -76.96926612783591,
x10 = 91.34578884524092,
</pre>
</td>
</tr>
<tr>
<td>Matriks
Balikan</td>
<td>
N = 6
<pre>
Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 1_D_1.txt

x1 = NaN,
x2 = NaN,
x3 = NaN,
x4 = NaN,
x5 = NaN,
x6 = NaN,
</pre>
N = 10
--lama
</td>
</tr>
<tr>
<td>Cramer</td>
<td>
N = 6
<pre>
Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 1_D_1.txt

x1 = 8.080835703092752,
x2 = -23.558849265341948,
x3 = -31.735397070348398,
x4 = 108.01793918185999,
x5 = -43.69940932286645,
x6 = -17.95285495142196,
</pre>
N = 10
--lama
</td>
</tr>
<tr>
<td colspan="4">Analisis: Pada penyelesaian menggunakan Matriks balikan, diperoleh hasil NaN, hal ini dikarenakan determinan yang dihasilkan dari matriks sangat kecil (mendekati nol). Pada program, penulis menggunakan beberapa fungsi untuk pembulatan dan mengatasi error sehingga hasil kemungkinan terepresentasikan dengan nilai nol. Pada kasus N=10, program menghasilkan output dengan waktu yang lama dikarenakan terdapat proses rekursif yang mengharuskan untuk menjadi determinan dari sebuah matriks berukuran 10 x 10. Penulis memiliki hipotesis bahwa determinan juga akan menjadi sangat kecil (mendekati nol) untuk N=10.</td>
</tr>
</table>

4.2. Sistem Persamaan Linear berbentuk matriks *augmented*

Soal		Metode	Output
a)	$A = \begin{bmatrix} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{bmatrix}$	Gauss	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 2_A.txt x1 = -1.0+1.00T, x2 = +2.00S, x3 = S, x4 = T,
		Gauss Jordan	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 2_A.txt x1 = -1.0+1.00T, x2 = +2.00S, x3 = S, x4 = T,
		Matriks Balikan	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 2_A.txt Matriks tidak memiliki inverse sehingga tidak dapat disolusikan.
		Cramer	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 2_A.txt Matriks tidak memiliki inverse sehingga tidak dapat disolusikan.
Analisis: Solusi dari SPL sama untuk metode eliminasi Gauss dan Gauss-Jordan. Solusi menggunakan matriks balikan dan kaidah Cramer membutuhkan syarat khusus dimana matriks masukan harus berupa matriks persegi.			
b)	$A = \begin{bmatrix} 2 & 0 & 8 & 0 & 8 \\ 0 & 1 & 0 & 4 & 6 \\ -4 & 0 & 6 & 0 & 6 \\ 0 & -2 & 0 & 3 & -1 \\ 2 & 0 & -4 & 0 & -4 \\ 0 & 1 & 0 & -2 & 0 \end{bmatrix}$	Gauss	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 2_B.txt x1 = 0.0, x2 = 2.0, x3 = 1.0, x4 = 1.0,
		Gauss Jordan	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 2_B.txt x1 = 0.0, x2 = 2.0, x3 = 1.0, x4 = 1.0,
		Matriks Balikan	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 2_B.txt Matriks memerlukan 4 persamaan untuk dapat disolusikan.
		Cramer	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 2_B.txt Matriks memerlukan 4 persamaan untuk dapat disolusikan.
Analisis: Solusi dari SPL sama untuk metode eliminasi Gauss dan Gauss-Jordan. Solusi menggunakan matriks balikan dan kaidah Cramer membutuhkan syarat khusus dimana matriks masukan harus berupa matriks persegi.			

4.3. SPL berbentuk

Soal		Metode	Output
a)	$8x_1 + x_2 + 3x_3 + 2x_4 = 0$ $2x_1 + 9x_2 - x_3 - 2x_4 = 1$ $x_1 + 3x_2 + 2x_3 - x_4 = 2$ $x_1 + \quad 6x_3 + 4x_4 = 3$	Gauss	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 3_A.txt x1 = -0.2243243243243243, x2 = 0.18243243243243246, x3 = 0.7094594594594594, x4 = -0.258108108108108,
		Gauss Jordan	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 3_A.txt x1 = -0.22432432432432436, x2 = 0.18243243243243246, x3 = 0.7094594594594594, x4 = -0.258108108108108,
		Matriks Balikan	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 3_A.txt x1 = -0.22432432432432434, x2 = 0.18243243243243243, x3 = 0.7094594594594594, x4 = -0.25810810810810814,
		Cramer	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 3_A.txt x1 = -0.22432432432432434, x2 = 0.18243243243243243, x3 = 0.7094594594594594, x4 = -0.2581081081081081,
Analisis: Solusi dari SPL sama untuk semua metode. Adapun terdapat sedikit perbedaan pembulatam pada digit ke-16 dibelakang koma pada metode Gauss-Jordan. Namun hasil ini merupakan hasil yang tidak terlalu signifikan sehingga dapat diabaikan.			
b)	$x_7 + x_8 + x_9 = 13.00$ $x_4 + x_5 + x_6 = 15.00$ $x_1 + x_2 + x_3 = 8.00$ $0.04289(x_3 + x_5 + x_7) + 0.75(x_6 + x_8) + 0.61396x_1 = 14.79$ $0.91421(x_3 + x_5 + x_7) + 0.25(x_2 + x_4 + x_6 + x_8) = 14.31$ $0.04289(x_3 + x_5 + x_7) + 0.75(x_2 + x_4) + 0.61396x_1 = 3.81$ $x_3 + x_6 + x_9 = 18.00$ $x_2 + x_5 + x_8 = 12.00$ $x_1 + x_4 + x_7 = 6.00$ $0.04289(x_1 + x_5 + x_9) + 0.75(x_2 + x_6) + 0.61396x_3 = 10.51$ $0.91421(x_1 + x_5 + x_9) + 0.25(x_2 + x_4 + x_6 + x_8) = 16.13$ $0.04289(x_1 + x_5 + x_9) + 0.75(x_4 + x_8) + 0.61396x_7 = 7.04$	Gauss	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 3_B.txt Persamaan linear tidak memiliki solusi.
		Gauss Jordan	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 3_B.txt Persamaan linear tidak memiliki solusi.
		Matriks Balikan	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 3_B.txt Matriks memerlukan 9 persamaan untuk dapat disolusikan.
		Cramer	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 3_B.txt Matriks memerlukan 9 persamaan untuk dapat disolusikan.
Analisis: Solusi dari SPL sama untuk semua metode, yaitu persamaan linier tidak memiliki solusi. karena terdapat baris pada matriks augmented yang berisi angka 0 (metode Gauss dan Gauss-Jordan) dan tidak dapat disolusikan karena tidak memiliki matriks invers dan determinannya nol (metode matriks balikan dan cramer).			

4.4. Studi Kasus Interpolasi

a) 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	0.4	0.7	0.11	0.14	0.17	0.2	0.23
$f(x)$	0.043	0.005	0.058	0.072	0.1	0.13	0.147

Pengujian dalam pada nilai-nilai default berikut:

x	$f(x)$
0.2	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 4_A_1.txt Hasil Perhitungan Interpolasi Polinom Penjabaran f(x): f(x) = − 4212.434532224689x^6 + 7102.39916327766x^5 − 4346.313951325718x^4 + 1220.8548907861214x^3 − 163.91566263586265x^2 + 10.276383991569446x − 0.18455901923404383 Hasil substitusi dengan nilai x dari masukan: f(0.2) = 0.12999999999963252
0.55	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 4_A_2.txt Hasil Perhitungan Interpolasi Polinom Penjabaran f(x): f(x) = − 4212.434532224689x^6 + 7102.39916327766x^5 − 4346.313951325718x^4 + 1220.8548907861214x^3 − 163.91566263586265x^2 + 10.276383991569446x − 0.18455901923404383 Hasil substitusi dengan nilai x dari masukan: f(0.55) = 2.1375716210430085
0.85	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 4_A_3.txt Hasil Perhitungan Interpolasi Polinom Penjabaran f(x): f(x) = − 4212.434532224689x^6 + 7102.39916327766x^5 − 4346.313951325718x^4 + 1220.8548907861214x^3 − 163.91566263586265x^2 + 10.276383991569446x − 0.18455901923404383 Hasil substitusi dengan nilai x dari masukan: f(0.85) = −66.26963931971704
1.28	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 4_A_4.txt Hasil Perhitungan Interpolasi Polinom Penjabaran f(x): f(x) = − 4212.434532224689x^6 + 7102.39916327766x^5 − 4346.313951325718x^4 + 1220.8548907861214x^3 − 163.91566263586265x^2 + 10.276383991569446x − 0.18455901923404383 Hasil substitusi dengan nilai x dari masukan: f(1.28) = −3485.144901856087
Analisis: Perhatikan bahwa range absis interpolasi yaitu [0.11, 0.7]. Oleh karena itu, hasil kalkulasi di atas untuk f(0.2) dan f(0.55) bernilai valid. Namun, hasil kalkulasi untuk f(0.5) dan f(1.28) bernilai tidak valid karena nilai x yang dimasukkan di luar range absis interpolasi.	

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

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

a. 16/07/2022

```
Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 4_B_1.txt

Hasil Perhitungan Interpolasi Polinom
Penjabaran f(x):
f(x) = - 141014.5528100891x^9 + 9374316.584951006x^8 - 2.755204013376618E8x^7 + 4.696641467591478E9x^6 -
 5.114164166687559E10x^5 + 3.6862683133476794E11x^4 - 1.7572042930778328E12x^3 + 5.335514826369923E12x^2
 - 9.349536886388793E12x + 7.189255800788213E12
Hasil substitusi dengan nilai x dari masukan:
f(7.533) = 52775.66015625
```

b. 10/08/2022

```
Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 4_B_2.txt

Hasil Perhitungan Interpolasi Polinom
Penjabaran f(x):
f(x) = - 141014.5528100891x^9 + 9374316.584951006x^8 - 2.755204013376618E8x^7 + 4.696641467591478E9x^6 -
 5.114164166687559E10x^5 + 3.6862683133476794E11x^4 - 1.7572042930778328E12x^3 + 5.335514826369923E12x^2
 - 9.349536886388793E12x + 7.189255800788213E12
Hasil substitusi dengan nilai x dari masukan:
f(8.322) = 36343.8359375
```

c. 05/09/2022

```
Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 4_B_3.txt

Hasil Perhitungan Interpolasi Polinom
Penjabaran f(x):
f(x) = - 141014.5528100891x^9 + 9374316.584951006x^8 - 2.755204013376618E8x^7 + 4.696641467591478E9x^6 -
 5.114164166687559E10x^5 + 3.6862683133476794E11x^4 - 1.7572042930778328E12x^3 + 5.335514826369923E12x^2
 - 9.349536886388793E12x + 7.189255800788213E12
Hasil substitusi dengan nilai x dari masukan:
f(9.166) = -659043.8671875
```

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

Di sini contoh masukan user melalui file, yaitu tanggal 09/08/2022 yang ketika diubah menjadi desimal bernilai 8.290.

```
Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 4_B_4.txt

Hasil Perhitungan Interpolasi Polinom
Penjabaran f(x):
f(x) = − 141014.5528100891x^9 + 9374316.584951006x^8 − 2.755204013376618E8x^7 + 4.696641467591478E9x^6 −
 5.114164166687559E10x^5 + 3.6862683133476794E11x^4 − 1.7572042930778328E12x^3 + 5.335514826369923E12x^2
 − 9.349536886388793E12x + 7.189255800788213E12
Hasil substitusi dengan nilai x dari masukan:
f(8.29) = 37533.99609375
```

Analisis: Perhatikan bahwa range absis interpolasi yaitu [6.567, 9]. Oleh karena itu, hasil kalkulasi di atas untuk f(7.533), f(8.322), dan f(8.29) bernilai valid. Namun, hasil kalkulasi untuk f(9.166) bernilai tidak valid karena nilai x yang dimasukkan di luar range absis interpolasi (itulah alasannya angkanya tidak masuk akal karena jumlah kasus baru bernilai negatif).

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

n	Nilai fungsi (bentuk matriks)	Hasil interpolasi
4	0.000 0.000 0.500 0.445 1.000 0.538 1.500 0.581 2.000 0.577 1.00	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 4_C_1.txt Hasil Perhitungan Interpolasi Polinom Penjabaran f(x): f(x) = − 0.19933333333333342x^4 + 1.000666666666667x^3 − 1.856166666666667x^2 + 1.592833333333333x Hasil substitusi dengan nilai x dari masukan: f(1.0) = 0.538
5	0.000 0.000 0.400 0.419 0.800 0.507 1.200 0.561 1.600 0.584 2.000 0.577 1.00	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 4_C_2.txt Hasil Perhitungan Interpolasi Polinom Penjabaran f(x): f(x) = 0.23763020833333448x^5 − 1.4290364583333395x^4 + 3.252604166666678x^3 − 3.565104166666675x^2 + 2.038500000000002x Hasil substitusi dengan nilai x dari masukan: f(1.0) = 0.5345937500000003
6	0.000 0.000 0.333 0.398 0.667 0.482 1.000 0.538 1.333 0.572 1.667 0.584 2.000 0.577 1.000	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 4_C_3.txt Hasil Perhitungan Interpolasi Polinom Penjabaran f(x): f(x) = − 0.2697616840052167x^6 + 1.9061389717600785x^5 − 5.353641486105415x^4 + 7.601445186464517x^3 − 5.807817528895102x^2 + 2.4616365407811385x Hasil substitusi dengan nilai x dari masukan: f(1.0) = 0.5379999999999999

Sebagai catatan, untuk nilai x dan fungsi yang digunakan di dalam matriks menggunakan format tiga angka di belakang koma. Selain itu, di bagian bawah file matriks juga ada nilai “1.00” hanya untuk memenuhi bentuk standar matriks masukan untuk interpolasi polinomial.

4.5. Bicubic Interpolation

Diberikan matriks input:

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

Tentukan nilai f(0,0), f(0.5,0.5), f(0.25, 0.75), dan f(0.1, 0.9)!

Untuk soal ini, penulis menginterpretasikan matriks tersebut sebagai:

$$\begin{bmatrix} f(-1,-1) & f(0,-1) & f(1,-1) & f(2,-1) \\ f(-1,0) & f(0,0) & f(1,0) & f(2,0) \\ f(-1,1) & f(0,1) & f(1,1) & f(2,1) \\ f(-1,2) & f(0,2) & f(1,2) & f(2,2) \end{bmatrix}$$

Jika matriks input di atas disesuaikan dengan standar spesifikasi masukan bicubic interpolation, maka matriksnya akan berbentuk:

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

sehingga didapat:

f(x,y)	Hasil Interpolasi
f(0,0)	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 5_A.txt Hasil Bicubic Interpolation Hasil substitusi dengan nilai x dan y dari masukan: f(0.0,0.0) = 161.0
f(0.5,0.5)	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 5_B.txt Hasil Bicubic Interpolation Hasil substitusi dengan nilai x dan y dari masukan: f(0.5,0.5) = 97.72656249999999
f(0.25, 0.75)	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 5_C.txt Hasil Bicubic Interpolation Hasil substitusi dengan nilai x dan y dari masukan: f(0.25,0.75) = 105.5147705078125
f(0.1, 0.9)	Pastikan file masukan sudah dimasukkan ke folder test. Nama file (.txt): 5_D.txt Hasil Bicubic Interpolation Hasil substitusi dengan nilai x dan y dari masukan: f(0.1,0.9) = 104.22911850000004
Analisis: Perhatikan bahwa range x dan y untuk interpolasi bikubik yaitu [0, 1]. Oleh karena itu, semua hasil kalkulasi di atas bernilai valid.	

4.6. Regresi Linier Berganda

Table 12.1: Data for Example 12.1

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

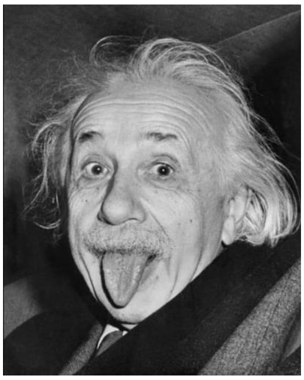
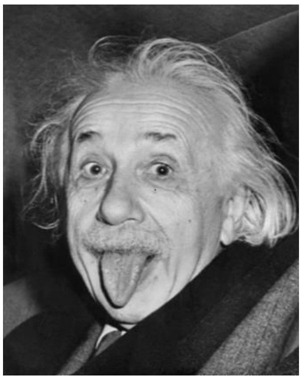
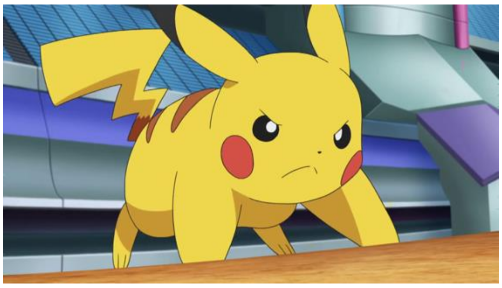
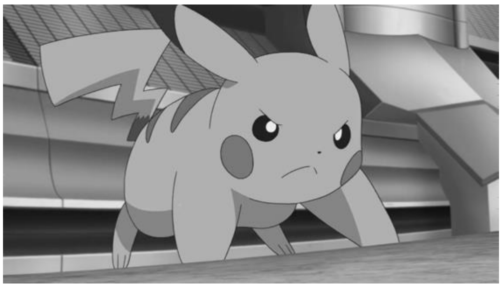
Source: Charles T. Hare, "Light-Duty Diesel Emission Correction Factors for Ambient Conditions," EPA-600/2-77-116. U.S. Environmental Protection Agency.

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.

```
Pastikan file masukan sudah dimasukkan ke folder test.
Nama file (.txt): 6.txt

Hasil Perhitungan Regresi Linear Berganda
Penjabaran f(xk):
f(xk) = - 3.507778140882833 - 0.002624990745878369x1 + 7.989410472204215E-4x2 + 0.1541550301982821x3
Hasil substitusi dengan nilai xk dari masukan:
f(xk) = 0.9384342262216663
```

4.7. Image Scaling

<table>
<tr><th>Citra Awal</th><th>Citra Hasil Upscale</th></tr>
<tr><td>
Ukuran Awal: 750 x 500 px</td><td>
Ukuran Akhir: 1500 x 1000 px</td></tr>
<tr><td colspan="2">Analisis: Pada pembesaran citra ini (dalam mode greyscale), terlihat bahwa warna terduplikasi secara sempurna. Diperoleh juga ukuran tepat dua kali lipat ukuran semula.</td></tr>
<tr><td>
Ukuran Awal: 1280 x 720 px</td><td>
Ukuran Akhir: 2560 x 1440 px</td></tr>
<tr><td colspan="2">Analisis: Pembesaran citra berhasil dengan ukuran tepat dua kali lipat semula. Karena pembesaran citra hanya terbatas pada greyscale, maka output yang dihasilkan juga dalam mode hitam-putih. Perlu diperhatikan bahwa p ada pembesaran citra dengan resolusi lebih tinggi, dihasilkan transisi antar pixel yang sangat smooth.</td></tr>
<tr><td>
Ukuran Awal: 360 x 338 px</td><td>
Ukuran Akhir: 720 x 676 px</td></tr>
<tr><td colspan="2">Analisis: Pembesaran citra berhasil dengan ukuran tepat dua kali lipat semula. Pada citra dengan resolusi rendah, resolusi terlihat lebih rendah dan image menjadi lebih choppy. Hal ini dikarenakan pixel yang diinterpolasi lebih sedikit dan menyebabkan terjadinya kesenjangan nilai antar pixel yang signifikan.</td></tr>
</table>

BAB V
KESIMPULAN, SARAN, DAN REFLEKSI

5.1 Kesimpulan

Pada Tugas Besar 1 IF2123 Aljabar Linier dan Geometri ini telah diimplementasikan sebuah objek matriks beserta fungsi-fungsi pengolahan matriks. Fungsi-fungsi tersebut mencakup fungsi metode eliminasi Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung determinan, kaidah Cramer, dan fungsi-fungsi penunjang lainnya. Implementasi tersebut kemudian berhasil digunakan dalam sebuah program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk Sistem Persamaan Linier (SPL). Penyelesaian SPL pada tugas besar ini dapat berupa satu dari tiga jenis solusi: solusi unik, solusi banyak, dan tidak mempunyai solusi.

Efisiensi metode penyelesaian SPL yang digunakan berdasarkan jenis matriks yang dimasukkan. Penyelesaian menggunakan kaidah Cramer dan matriks balikan memiliki syarat khusus: matriks masukan harus berupa matriks persegi. Sementara metode eliminasi Gauss dan Gauss-Jordan tidak membatasi matriks masukan dan memungkinkan untuk menghasilkan solusi banyak pada matriks. Hal ini dibuktikan dalam pengujian pada solusi matriks hilbert (dimana solusi dengan matriks cramer memakan waktu yang lama, sementara metode eliminasi Gauss dan Gauss-Jordan lebih cepat dan efisien). Solusi dengan metode Kaidah Cramer juga dibatasi apabila determinannya sangat mendekati nol.

Selanjutnya metode penyelesaian SPL ini digunakan sebagai dasar untuk menyelesaikan persoalan polinom, *bicubic interpolation*, perbesaran citra dan persoalan regresi. Penyelesaian-penyelesaian tersebut kemudian dirangkum dalam sebuah program utama dalam bahasa Java. Berikut merupakan fitur dari program implementasi utama kami:

1. Penyelesaian Sistem Persamaan Linier (SPL) menggunakan metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer
2. Penyelesaian determinan suatu matriks menggunakan metode OBE (Segitiga Atas), metode kofaktor baris, dan metode kofaktor kolom
3. Penyelesaian matriks balikan suatu matriks menggunakan metode identitas dan metode matriks adjoin
4. Penyelesaian interpolasi polinom
5. Penyelesaian interpolasi *bicubic*
6. Penyelesaian regresi linier berganda

7. Perbesaran gambar atau citra menggunakan implementasi interpolasi bicubic

Program implementasi utama kami juga sudah berhasil membaca input dari sebuah file dan menuliskan hasil pengolahan matriks ke sebuah file baru.

Dengan demikian, penulis menyimpulkan bahwa melalui Tugas Besar 1 IF2123 Aljabar Linier dan Geometri, dapat dibuat sebuah algoritma diimplementasikan dalam suatu program yang mengkomputasi penyelesaian dari sistem persamaan linier (SPL) menggunakan matriks dengan berbagai metode.

5.2 Saran

Tugas Besar 1 IF2123 Aljabar Linier dan Geometri Semester I Tahun 2022/2023 menjadi salah satu tugas yang memberikan banyak pelajaran bagi penulis. Berdasarkan pengalaman penulis dalam mengerjakan tugas ini, berikut merupakan saran untuk pembaca yang ingin melakukan atau mengerjakan hal yang serupa.

1. Spesifikasi program meminta penulis untuk menuliskan program dalam bahasa Java, yaitu bahasa yang belum pernah disentuh oleh sebagian besar mahasiswa IF2123 Semester I Tahun 2022/2023. Penulis merekomendasikan untuk menyediakan waktu yang cukup untuk mempelajari dan mengerjakan tugas ini. Hal ini juga dikarenakan fitur yang harus dimuat banyak dan rinci.
2. Keefektifan dalam kerja sama tim merupakan hal yang penting dalam mengerjakan tugas ini. Untuk pengerjaan secara sinkron, tugas ini sangat terbantu oleh pemakaian real-time collaboration app, seperti VSCode LiveShare (pembuatan program) dan kolaborasi langsung pada MS Word (pembuatan laporan). Selain itu, dalam pembuatan source code aplikasi tak jarang terjadi konflik antara dua atau lebih anggota. Pemakaian aplikasi pengelola version contol seperti Github sangat disarankan agar memudahkan untuk mengelola pekerjaan secara asinkron.
3. Mengingat bahwa program implementasi pada tugas ini sudah termasuk ke dalam pemrograman berorientasi objek, penggunaan Abstract Data Type dan Struktur Data yang tepat merupakan hal yang penting untuk diperhatikan. Pemakaian struktur data Matriks pada tugas ini sangat disarankan dan membuat fungsi menjadi lebih rapi dan dapat digunakan berulang kali oleh program.

5.3 Refleksi

Tugas Besar 1 IF2123 Aljabar Linier dan Geometri Semester I Tahun 2022/2023 merupakan salah satu tugas besar pertama yang penulis dapatkan pada semester ini. Proses pengerjaan tugas ini tentunya melalui berbagai rintangan. Dengan tugas ini, penulis merasakan apresiasi yang besar terhadap penggunaan matriks serta implementasinya dalam kehidupan, baik secara matematis ataupun dalam bidang teknologi. Keberasaan matriks serta atribut-atribut yang dimilikinya memberikan dampak tersendiri dalam menyelesaikan persoalan, terutama persoalan sistem persamaan linier.

Kegunaan matriks juga dapat dijumpai pada sebagian besar bidang Informatika, contohnya dalam pengolahan citra (image processing) ataupun kriptografi. Pada tugas ini, penulis diberi kesempatan untuk mengimplementasikan pengolahan citra pada matriks dan timbul rasa apresiasi yang tinggi dan kebanggaan dalam keberhasilan untuk memproduksi sebuah citra baru dari sekumpulan angka-angka berbentuk matriks. Dalam mengerjakan tugas ini, penulis juga merasakan euforia berhasil mempelajari bahasa pemrograman baru (Java) dan mengimplementasikannya dengan kurun waktu di bawah tiga minggu. Rintangan yang dihadapi untuk mempelajari dan menekuni matriks selama beberapa minggu ini dilengkap dengan sinergi dari setiap anggota kelompok yang saling membantu dan mengerjakan tugasnya dengan baik.

DAFTAR REFERENSI

Atul, K. &. (2018, December 29). *Image Processing - Bicubic Interpolation.* Retrieved from TheAILearner: https://theailearner.com/2018/12/29/image-processing-bicubic-interpolation/

How to Build Your On Java Library? (2015). Retrieved from Program Creek: https://www.programcreek.com/2011/07/build-a-java-library-for-yourself/

Howard Anton, Chris Rorres. (2013) Elementary Linear Algebra: Applications Version. 11th Edition, John Wiley & Sons Incorporated.

Montgomery, D.C. and Runger, G.C. (2003) Applied Statistics and Probability for Engineers. 3rd Edition, John Wiley & Son, Inc., Hoboken.

Munir, R. (2022). *Algeo #5 Tiga Kemungkinan Solusi SPL.* Retrieved from Homepage Rinaldi Munir: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-04-Tiga-Kemungkinan-Solusi-SPL.pdf

Munir, R. (2022). *Algeo #4 Metode Eliminasi Gauss.* Retrieved from Homepage Rinaldi Munir: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-03-Sistem-Persamaan-Linier.pdf

Munir, R. (2022). *Algeo #6 Metode Eliminasi Gauss-Jordan.* Retrieved from Homepage Rinaldi Munir: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-05-Sistem-Persamaan-Linier-2.pdf

Rowe, D. B. (2018, February 15). *BiLinear, Bicubic, and In Between Spline Interpolation.* Retrieved from MU MSCS Spring 2018: https://www.mssc.mu.edu/~daniel/pubs/RoweTalkMSCS_BiCubic.pdf

Sharma, A. (2021, Juni 11). *Resizing Images using Various Interpolation Techniques.* Retrieved from Medium : https://annmay10.medium.com/resizing-images-using-various-interpolation-techniques-4b99800999f2

Spesifikasi Tugas Besar 1 IF 2123 Aljabar Linier dan Geometri Sistem Persamaan Linier, Determinan, dan Aplikasinya Semester I Tahun 2022/2023. (2022). Program Studi Teknik Elektro dan Informatika Institut Teknologi Bandung.

LAMPIRAN

Berikut adalah link repository GitHub untuk program penulis.
https://github.com/alishalistyaa/Algeo01-21095