LAPORAN TUGAS BESAR I
SISTEM PERSAMAAN LINIER, DETERMINAN, DAN APLIKASINYA

Disusun untuk memenuhi tugas mata kuliah IF2123 Aljabar Linear dan Geometri pada Semester 1 (satu) Tahun Akademik 2023/2024

Kelompok 7 (Rudal Sekeloa):

Razi Rachman Widyadhana	(13523004)
Nayaka Ghana Subrata	(13523090)
Muhammad Adha Ridwan	(13523098)

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

2024

DAFTAR ISI

BAB I
DESKRIPSI

1.1 Spesifikasi Tugas

Buatlah pustaka (library atau package) dalam Bahasa Java untuk menemukan solusi SPL dengan metode eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n peubah dan n persamaan), menghitung determinan matriks dengan reduksi baris dan dengan ekspansi kofaktor, dan menghitung balikan matriks.

Gunakan pustaka di atas untuk membuat program penyelesaian berbagai persoalan dalam bentuk SPL, menyelesaikan persoalan interpolasi dan regresi, menghitung matriks balikan, menghitung determinan matriks dengan berbagai metode (reduksi baris dan ekspansi kofaktor).

1.2 Spesifikasi Program

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

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

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

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

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

3. Untuk persoalan invers, metode yang digunakan ada 2 yaitu menggunakan OBE dan Matriks Adjoin.

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

```
8.0 2.0794
9.0 2.1972
9.5 2.2513
8.3
```

5. Untuk persoalan regresi, masukannya jika dari keyboard adalah n (jumlah peubah x), m (jumlah sampel), semua nilai-nilai x_{1i}, x_{2i}, …, x_{ni}, nilai y_i, dan nilai-nilai x_k yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.

6. Untuk persoalan SPL, luaran program adalah solusi SPL. Jika solusinya tunggal, tuliskan nilainya. Jika solusinya tidak ada, tuliskan solusi tidak ada, jika solusinya banyak, maka tuliskan solusinya dalam bentuk parametrik (misalnya $x_4 = -2$, $x_3 = 2s - t$, $x_2 = s$, dan $x_1 = t$).

7. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan. Contoh luaran untuk interpolasi adalah

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

dan untuk regresi adalah

$$f(x) = -9.5872 + 1.0732x_1, \qquad f(x_k) = \ldots$$

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

$x_1 = X$

$x_3 = X^2$

$x_5 = XY$

[Persamaan dan Solusi]

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

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

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

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

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

10. Bahasa program yang digunakan adalah Java. Anda bebas untuk menggunakan versi java apapun dengan catatan di atas java versi 8 (8/9/11/15/17/19/20/21).

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

    MENU

    1. Sistem Persamaan Linear
    2. Determinan
    3. Matriks Balikan
    4. Interpolasi Polinom
    5. Interpolasi Bicubic Spline
    6. Regresi Linier dan Kuadratik Berganda
    7. Interpolasi Gambar (Bonus)
    8. Keluar

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

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

BAB II
TEORI SINGKAT

2.1 Sistem Persamaan Linier (SPL)

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Anda sudah mempelajari berbagai metode untuk menyelesaikan SPL, termasuk menghitung determinan matriks. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan ($x = A^{-1}\ b$), dan kaidah *Cramer* (khusus untuk SPL dengan n

peubah dan n persamaan). Solusi sebuah SPL mungkin tidak ada, banyak (tidak berhingga), atau hanya satu (unik/tunggal).

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

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

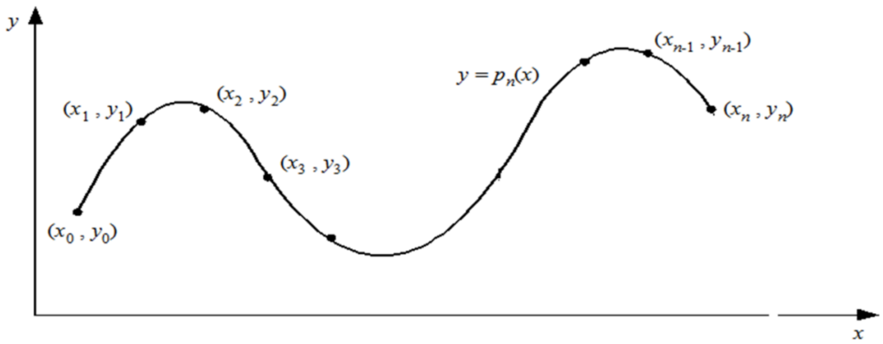
Di dalam Tugas Besar 1 ini, Anda diminta membuat satu atau lebih library aljabar linier dalam Bahasa Java. Library tersebut berisi fungsi-fungsi seperti eliminasi Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung determinan, kaidah Cramer (kaidah *Cramer* khusus untuk SPL dengan n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

Beberapa tulisan cara membuat library di Java:

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

1.2 Interpolasi Polinomial

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



Gambar 2. Ilustrasi beberapa titik yang diinterpolasi secara polinomial

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

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

$$a_0 + a_1x_0 + a_2\,{x_0}^2 + \ ... + a_n\,{x_0}^n = y_0$$
$$a_0 + a_1x_0 + a_2\,{x_0}^2 + \ ... + a_n\,{x_0}^n = y_1$$
$$\dots \qquad\qquad\qquad \dots.$$
$$a_0 + a_1x_0 + a_2\,{x_0}^2 + \ ... + a_n\,{x_0}^n = y_n$$

Solusi sistem persamaan lanjar ini, yaitu nilai $a_0,\ a_1,\ a_2,\ \dots,\ \ a_n$. diperoleh dengan menggunakan metode eliminasi Gauss yang sudah anda pelajari. Sebagai contoh, misalkan diberikan tiga buah titik yaitu $(8.0, 2.0794)$, $(9{,}0,\ \ 2.1972)$, $(9.5,\ 2.2513)$. Tentukan polinom interpolasi kuadratik lalu estimasi nilai fungsi pada $x = 9.2$. Polinum kuadratik berbentuk $p_2(x) = a_0 + a_1x + a_2x^2$. Dengan menyulihkan ketiga buah titik data ke dalam polinom tersebut, diperoleh sistem persamaan lanjar yang terbentuk adalah

$$a_0 + 8.0a_1 + 64.00a_2 = 2.0794$$
$$a_0 + 9.0a_1 + 81.00a_2 = 2.1972$$
$$a_0 + 9.5a_1 + 90.25a_2 = 2.2513$$

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

1.3 Regresi Berganda

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

1.3.1 Regresi Linier Berganda

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

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

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

$$\begin{array}{ccccccccc}
nb_0 & + b_1 \sum_{i=1}^{n} x_{1i} & + b_2 \sum_{i=1}^{n} x_{2i} & + \ \dots & + b_k \sum_{i=1}^{n} x_{ki} & = \sum_{i=1}^{n} y_i \\
b_0 \sum_{i=1}^{n} x_{1i} & + b_1 \sum_{i=1}^{n} x_{1i}{}^2 & + b_2 \sum_{i=1}^{n} x_{1i}\, x_{21} & + \ \dots & + b_k \sum_{i=1}^{n} x_{1i}\, x_{ki} & = \sum_{i=1}^{n} x_{1i}\, y_i \\
\vdots & \vdots & \vdots & & \vdots & \vdots \\
b_0 \sum_{i=1}^{n} x_{ki} & + b_1 \sum_{i=1}^{n} x_{ki}\, x_{1i} & + b_2 \sum_{i=1}^{n} x_{ki}\, x_{2i} & + \ \dots & + b_k \sum_{i=1}^{n} x_{ki}{}^2 & = \sum_{i=1}^{n} x_{ki}\, y_i
\end{array}$$

1.3.2 Regresi Kuadratik Berganda

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

a. Variabel Linier: Variabel dengan derajat satu seperti X, Y, dan Z

b. Variabel Kuadrat: Variabel dengan derajat dua seperti X^2

c. Variabel Interaksi: 2 Variabel dengan derajat satu yang dikalikan dengan satu sama lain seperti XY, YZ, dan XZ

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

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

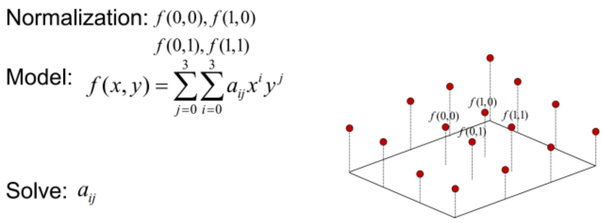
N menandakan jumlah peubah, terdapat 2 variabel linier, yaitu u_i dan v_i. 2 variabel kuadrat, yaitu $u_i{}^2$ dan $v_i{}^2$, dan 1 variabel interaksi, yaitu uv. Untuk setiap n-peubah, akan terdapat 1 konstan N (Terlihat di bagian atas kiri gambar), n variabel linier, n variable kuadrat, dan C_2^n variable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

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

1.4 Bicubic Spline Interpolation

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

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



Gambar 3. Pemodelan interpolasi *bicubic spine*

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

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

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

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

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

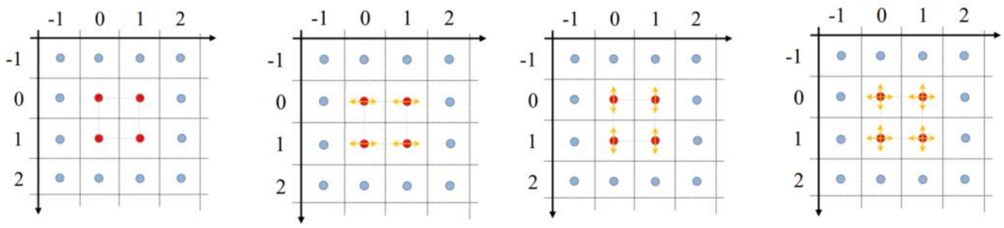
Dengan menggunakan nilai fungsi dan turunan berarah tersebut, dapat terbentuk sebuah matriks solusi X yang membentuk persamaan penyelesaian sebagai berikut.

$$y = Xa$$

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

Perlu diketahui bahwa elemen pada matriks X adalah nilai dari setiap komponen koefisien a_{ij} yang diperoleh dari persamaan fungsi maupun persamaan turunan yang telah dijelaskan sebelumnya. Sebagai contoh, elemen matriks X pada baris 8 kolom ke 2 adalah koefisien dari a_{10} pada ekspansi sigma untuk $f_x(1{,}1)$ sehingga diperoleh nilai konstanta $1 \times 1^{1-1} \times 1^0$, sesuai denga nisi matriks X.

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



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

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

1.5 *Image Resizing and Stretching*

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

$$f(x,y) = I(x,y)$$
$$f_x(x,y) = [I(x+1,y) - I(x-1,y)]\ /\ 2$$
$$f_y(x,y) = [I(x,y+1) - I(x,y-1)]\ /\ 2$$
$$f_{xy}(x,y) = [I(x+1,y+1) - I(x-1,y) - I(x,y-1) - I(x,y)]\ /\ 4$$

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

$$y = DI$$

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

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

Gambar 5. Sebuah citra gambar asal (kiri) dan hasil pemrosesan gambar dengan skala 1.5 pada width dan skala 2 pada height (kanan).

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

BAB III
IMPLEMENTASI PUSTAKA DAN PROGRAM JAVA

Pada pengerjaan Tugas Besar ini, kami membuat 2 *packages*, dengan nama *package* "ADTMatriks" yang melakukan *handle* pada tipe data matriks, dan *package* "Methods" yang melakukan operasi pada matriks untuk melakukan metode-metode untuk menyelesaikan permasalahan tertentu.

3.1 Class ADTMatrix

Pada *package* ini, kami membangun tipe data matriks dan juga beberapa primitif dan operasi primitif matriks. Terdapat 2 file berekstensi ".java" dan file ini terdiri dari total (…) fungsi, yang akan dijabarkan pada tabel di bawah.

3.1.1 Matrix.java

Fungsi/Prosedur	Deskripsi
Matrix (int mRow, int nCol)	Membuat ADT Matriks dengan ukuran baris "mRow, dan ukuran kolom "nRow".
boolean isValidRowInput(String[] rowElmts, int nCol)	Mengecek validasi per baris terhadap kolom dan juga input per elemen.
void readMatrix(int mRow, int nCol)	Membaca masukan elemen matriks dari user.
void writeMatrix()	Melakukan *output* dan *print* matriks.
void importMatrix(String filename)	Membaca masukan (input) matriks dari *file*.
void importMatrixWithEmpty(String filename, int nEmpty)	Membaca matriks dari file dengan elemen kosong sebanyak nEmpty.

void exportMatrix()	Menulis dan ekspor matriks ke dalam *file*.

3.1.2 Operation.java

Fungsi/Prosedur	**Deskripsi**
boolean isSquareMatrix(Matrix M)	Mengembalikan true jika matriks M adalah matriks persegi.
boolean isIdentityMatrix(Matrix M)	Mengembalikan true jika matriks M adalah matriks identitas.
boolean isMatrixEqual(Matrix M1, Matrix M2)	Mengembalikan true jika kedua dimensi matriks sama.
Matrix copyMatrix(Matrix M)	Mengembalikan salinan matriks M.
Matrix createIdentityMatrix(int n)	Membuat matriks identitas.
Matrix multiplyMatrix(Matrix M1, Matrix M2)	Melakukan perkalian matriks M1 dan M2.
Matrix multiplyMatrixByScalar(Matrix M, double scalar)	Melakukan perkalian matriks M dengan skalar.

Matrix transposeMatrix(Matrix M)	Melakukan transpose matriks M.
Matrix expandCol(Matrix M1, Matrix M2)	Menggabungkan dua matriks berdasarkan kolom.
Matrix getMinor(Matrix M, int delRow, int delCol)	Mengembalikan matriks minor dari matriks M dengan menghapus baris ke-delRow dan kolom ke-delCol.
Matrix takeLastRow(Matrix M)	Mengambil baris terakhir dari matriks M.
Matrix takeLastCol(Matrix M)	Mengambil kolom terakhir dari matriks M.
Matrix dropLastRow(Matrix M)	Menghapus baris terakhir dari matriks M.
Matrix dropLastCol(Matrix M)	Menghapus kolom terakhir dari matriks M.
void errorRounding(Matrix M)	Melakukan pembulatan galat pada matriks.

3.2 Class Methods

Pada *package* ini, kami membuat operasi-operasi yang digunakan untuk menyelesaikan permasalahan matriks yang ada sesuai spesifikasi, yakni Sistem Persamaan Linear (SPL), determinan, invers, interpolasi polinom, interpolasi bikubik, regresi linear berganda, regresi kuadratik berganda, dan *image resizing*. Terdapat 7 file berekstensi ".java" dan file ini terdiri dari total (…) fungsi, yang akan dijabarkan pada tabel di bawah.

3.2.1 SPL.java

Fungsi/Prosedur	**Deskripsi**

SPL()	Konstruktor dari Sistem Persamaan Linear.
void InverseSPL(Matrix M)	Prosedur untuk implementasi solusi inverse.
void InverseSPLFile(Matrix M)	Menampilkan hasil dari proses SPL inverse jika inputnya *file*.
Matrix cramerSwap(Matrix M1, Matrix M2, int col)	Matrix melakukan Cramer Swap.
void CramerSPL(Matrix M)	Menampilkan solusi dari kaidah Cramer.
void CramerSPLFile(Matrix M)	Menampilkan hasil dari proses SPL kaidah Cramer jika inputnya *file*.
Matrix shiftZero(Matrix M)	Memindahkan baris dengan semua 0, ke baris paling bawah.
Matrix subMultiplyRow(Matrix M, int rowDes, int rowMin, double k)	Fungsi untuk mengurangkan dan mengalikan satu baris tertentu dari matriks.
Matrix multiplyRowByConst(Matrix M, int row, double k)	Fungsi untuk mengalikan semua elemen dalam satu baris tertentu dari matriks dengan sebuah konstanta.
Matrix swapRow(Matrix M, int row1, int row2)	Fungsi untuk menukar letak baris.
Matrix gauss(Matrix M)	Fungsi untuk implementasi metode eliminasi Gauss.

Matrix gaussJordan(Matrix M)	Fungsi untuk implementasi metode eliminasi Gauss-Jordan.
void gaussSPL(Matrix M, Matrix temp)	Menerima matriks gauss/gauss jordan SPL.
void gaussSPLFile(Matrix M, Matrix temp)	Prosedur untuk membuat file dari solusi SPL.
boolean isAllZero(Matrix M)	Mengecek apakah ada baris yang semua elemennya bernilai 0.
Matrix removeRowAll0 (Matrix M)	Menghapus baris yang seluruh elemennya 0.
boolean isSolvable(Matrix M)	Mengecek apakah suatu SPL dapat diselesaikan.
double recursion(int toplimit, int bottomlimit, int row, int varCol, double arrayHasil[], String arrayString[], Matrix M)	Fungsi untuk melakukan perhitungan rekursif yang terkait dengan manipulasi matriks dan sistem persamaan linear.
int checkSolusi(Matrix M)	Mengecek Solusi dari SPL, apakah trivial, unik, banyak, atau tidak ada.
void solusiKosong(Matrix M)	Melakukan output parameter pada SPL yang memiliki solusi banyak.
void solusiKosongFile(Matrix M, Matrix temp, String filename)	Melakukan output dan menulis pada *file* parameter pada SPL yang memiliki solusi banyak.
int cariSatu(Matrix M, int row)	Mencari satu utama dari baris.
void solusiUnik(Matrix M)	Prosedur untuk mencari solusi unik dari persamaan gauss.
void solusiUnikFile(Matrix M, Matrix temp, String filename)	Prosedur untuk menulis file solusi unik dari persamaan gauss.

void solusiBanyak(Matrix M, Matrix temp)	Prosedur untuk mencari solusi banyak dari persamaan gauss.
void solusiBanyakFile(Matrix M, Matrix temp, String filename)	Prosedur untuk menulis file solusi banyak dari persamaan gauss.
void solusiNone(Matrix M)	Prosedur untuk menampilkan tidak ada solusi.
void solusiNoneFile(Matrix M, Matrix temp, String filename)	Prosedur untuk menulis file tidak ada solusi.

3.2.2 Determinant.java

Fungsi/Prosedur	**Deskripsi**
double determinantCofactor(Matrix M)	Menghitung determinan matriks dengan metode kofaktor.
double determinantOBE(Matrix M)	Menghitung determinan matriks dengan metode eliminasi Gauss.
Matrix getCofactor(Matrix M)	Menghitung matriks kofactor dari matrix M.
void exportDet(Matrix M, BigDecimal det)	Menulis dan melakukan ekspor matriks ke *file*.

3.2.3 Inverse.java

Fungsi/Prosedur	**Deskripsi**
boolean isZeroRow(double[] row)	Mengecek apakah ada baris dengan elemen bernilai semua 0.

boolean isZeroColumn(Matrix M)	Mengecek apakah ada kolom dengan elemen bernilai semua 0.
void multiplyRowInv(Matrix M, int row, double factor)	Mengalikan baris invers dengan suatu faktor.
void subtractRowInv(Matrix M, int rowToSubtract, int targetRow, double factor)	Mengurangi satu baris invers dengan baris lainnya.
void swapRow(Matrix M, int row1, int row2)	Menukar dua baris pada matriks invers.
Matrix inverseGJ(Matrix M)	Membuat invers dari suatu matriks menggunakan metode eliminasi Gauss-Jordan.
Matrix getAdj(Matrix M)	Membuat matriks adjoin dari matriks M.
Matrix inverseDet(Matrix M)	Membuat invers matriks dengan metode determinan.
void exportInverse(Matrix M, Matrix inverse)	Menulis dan melakukan ekspor matriks invers ke *file*.

3.2.4 PolynomialInterpolation.java

Fungsi/Prosedur	Deskripsi
Matrix localInput()	Melakukan input titik-titik interpolasi dari keyboard.
Matrix takeX(Matrix M)	Fungsi untuk mengambil nilai x dari matriks input.
double minimumX(Matrix M)	Fungsi untuk mendapatkan nilai minimum dari x.
double maximumX(Matrix M)	Fungsi untuk mendapatkan nilai maksimum dari x.
Matrix takeFx(Matrix M)	Fungsi untuk mengambil nilai y (f(x)) dari matriks input.
double takeXValue(Matrix M)	Fungsi untuk mengambil nilai x yang akan dicari interpolasinya. (masukan dari keyboard)
double takeXValueFile(Matrix M)	Fungsi untuk mengambil nilai x yang akan dicari interpolasinya. (masukan dari file)
Matrix xi(Matrix x)	Membuat matriks xi untuk polinom interpolasi.
Matrix calculateCoeff(Matrix xi, Matrix fx)	Menghitung koefisien polinom.

double calculateFx(Matrix f, double x)	Menghitung nilai interpolasi f(x)
String showPolynom(Matrix f)	Menampilkan polinom dalam bentuk string.
void interpolationFile(Matrix M)	Menyimpan hasil interpolasi ke dalam file.

3.2.5 BicubicInterpolation.java

Fungsi/Prosedur	**Deskripsi**
Matrix inputBicubic()	Menerima input bikubik dari keyboard.
public double getA(Matrix M)	Mengambil nilai a.
public double getB(Matrix M)	Mengambil nilai b.
Matrix getMatrixX()	*Generate* matriks X.
Matrix getMatrixF(Matrix m4x4)	Mengubah matriks 4x4 menjadi 16x1.
Matrix multiplyMatrixBik(Matrix m1, Matrix m2)	Perkalian matriks khusus bikubik.
Matrix getMatrixAij(Matrix m16x1)	Mengambil nilai koefisien Aij.
double getFabResult(Matrix Maij, double a, double b)	Mengambil nilai Fab dari input.
void exportBicubic(Matrix Maij, double a, double b)	Melakukan ekspor hasil bikubik ke *file*.

3.2.6 Regression.java

Fungsi/Prosedur	Deskripsi
Matrix inputLinearRegression()	Melakukan input regresi linear melalui keyboard.
Matrix inputQuadraticRegression()	Melakukan input regresi kuadratik melalui keyboard.
Matrix regressionGJ(Matrix augmented)	Melakukan eliminasi Gauss-Jordan regresi.
double calcRow(Matrix M, int s, int rowReg, int colReg)	Melakukan operasi terhadap baris (penjumlahan).
double calcFxLinear(Matrix a, Matrix m)	Menghitung nilai Fx pada regresi linear.
Matrix normalize(Matrix m)	Melakukan normalisasi dari matriks regresi.
void linearRegression(Matrix M)	Melakukan dan mengoutput hasil regresi linear.
void linearRegressionFile(Matrix M)	Melakukan dan mengoutput hasil regresi linear ke *file*.
String generateLinearEquation(Matrix a)	Menampilkan fungsi fx untuk regresi linear.
Matrix QuadraticRegression(Matrix M)	Melakukan dan mengoutput hasil regresi kuadratik.
void calcFxQuadratic(Matrix beta, Matrix y)	Menghitung nilai Fx pada regresi kuadratik.

void quadraticRegressionFile(Matrix M, Matrix y)	Menampilkan fungsi fx untuk regresi kuadratik.

3.2.7 ImageResizing.java

Fungsi/Prosedur	**Deskripsi**
Matrix getMatrixD()	Membuat matriks D.
Matrix getMatrixXinvD()	Matriks hasil perkalian X^-1.D untuk mendapatkan nilai a.
Matrix[] getSurroundingPoints(int i, int j, BufferedImage input)	Mendapatkan nilai turunan di sekitar titik.
double[] getColorRGB (int in	Fungsi untuk mendapatkan RGB Value.
int getColorValue(double x, double y, Matrix[] colorMatrices) {	Mendapatkan nilai dari warna.
int clampTo8Bit	
imageProccesing(File imgFile)	Memroses gambar dan melakukan image resizing.

3.3 Driver (Main)

Pada file "Main.java" ini, kami membuat operasi-operasi yang digunakan untuk menjalankan *packages* dan *class* yang sudah diimplementasikan, juga sekaligus menjadi driver dari program. Secara garis besar, file ini hanya menjadi controller untuk menjalankan class-class yang sudah didefinisikan.

BAB IV
EKSPERIMEN DAN STUDI KASUS

4.1 Eksperimen dan Studi Kasus Sistem Persamaan Linear (SPL)

Pada SPL, ada tiga Solusi, yakni solusi unik, banyak, dan tidak ada Solusi.

Contoh output :

```
Matriks Hasil:
┌  1,00000  1,00000  -1,00000  -1,00000   1,00000 ┐
│  0,00000  1,00000  -1,66667  -1,00000  -1,33333 │
│  0,00000  0,00000   1,00000  -1,00000   1,00000 │
└  0,00000  0,00000   0,00000   0,00000   1,00000 ┘

Solusi SPL:
Tidak ada.
```

```
Matriks Hasil:
┌  1,00000  0,00000  0,00000  0,00000  -1,00000   3,00000 ┐
│  0,00000  1,00000  0,00000  0,00000  -2,00000   0,00000 │
│  0,00000  0,00000  0,00000  1,00000  -1,00000  -1,00000 │
└  0,00000  0,00000  0,00000  0,00000   0,00000   0,00000 ┘

Solusi SPL:
x1 = 3+S,
x2 = 2S,
x3 = T,
x4 = -1+S,
x5 = S,
```

4.2 Eksperimen dan Studi Kasus Determinan

Pada determinan, program akan mengeluarkan output determinan, atau pesan error untuk kesalahan input :

```
Matriks memerlukan 2 persamaan untuk dapat diselesaikan
Simpan dalam bentuk file?
```

```
Mengambil file dari folder test.
Nama file: 1e

Determinannya adalah: -3
```

4.3 Eksperimen dan Studi Kasus Matriks Balikan (Invers)

Pada invers, program akan mengeluarkan output matriks invers, atau pesan error untuk kesalahan input:

```
Mengambil file dari folder test.
Nama file: 1e

Balikannya adalah:
┌  -0,66667  -1,33333   1,00000 ┐
│  -0,66667   3,66667  -2,00000 │
└   1,00000  -2,00000   1,00000 ┘
```

4.4 Eksperimen dan Studi Kasus Interpolasi Polinom

Pada interpolasi polinom, program akan mengeluarkan nilai fx dan nilai taksirannya :

```
│ 0,30000 │
│ 0,50000 │
│ 0,70000 │
│ 0,90000 │
│ 1,10000 │
└ 1,30000 ┘
┌ 0,00300 ┐
│ 0,06700 │
│ 0,14800 │
│ 0,24800 │
│ 0,37000 │
│ 0,51800 │
└ 0,69700 ┘
f(x) =  - 0.0229765625000066 + 0.2400000000001187x + 0.19739583333326544x^2 + 1.7481039538003605E-13x^3 + 0.026041666666442493x^4 + 1.3959667659383922E-13x^5 - 3.360838505161542E-
14x^6
Nilai f(0.2) = 0.032960937500000086
```

4.5 Eksperimen dan Studi Kasus Interpolasi Bikubik

Pada interpolasi bikubik, program akan mengeluarkan nilai f dan nilai taksirannya :

```
========================================
|                Masukan               |
========================================
| 1 | File                             |
| 2 | Keyboard                         |
| 3 | Kembali Ke Menu                  |
========================================

Pilih metode (1 - 3) >> 1

Mengambil file dari folder test.
Nama file: 7a
Hasil dari f(0.0,0.0): 21.0
```

4.6 Eksperimen dan Studi Kasus Regresi Berganda

4.6.1 Regresi Linier

Diberikan sekumpulan data sesuai pada tabel berikut ini.

Table 12.1: Data for Example 12.1

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

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

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

Dari data-data tersebut, apabila diterapkan pada program :

```
Persamaan regresi linear: f(x) = -3,5077781409 - 0,0026249907x1 + 0,0007989410x2 + 0,1541550302x3
Hasil taksiran f(x): 0.9384342262216645

Simpan dalam bentuk file?
1. Ya
2. Tidak
>> 
```

4.6.2 Regresi Kuadratik

Diberikan sekumpulan data sesuai pada tabel berikut ini.

Table 12.1: Data for Example 12.1

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

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

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

Dari data-data tersebut, apabila diterapkan pada program :

```
Mengambil file dari folder test.
Nama file: 6
Persamaan regresi linear berganda f(x):
Persamaan regresi kuadratik: f(x):
 - 1146.437213242 + 0.1839546065x1 + 0.8385564601x2 + 75.4502801758x3 - 0.0000014836x1^2 - 0.00023765x2^2 - 1.2403842282x3^2 + 0.0000170039x1x2 - 0.00
64128222x1x3 - 0.0272481185x2x3

Hasil taksiran f(50.0, 76.0, 29.3): 0.943905838
```

4.7 Eksperimen dan Studi Kasus Image Resizing

Inputan berupa suatu *image*, *image* ini memiliki format antara jpg,jpeg ataupun png serta input berupa *scale* x yang ingin digunakan
Contoh *image* dibawah.

Dari *image* tersebut dilakukan suatu *image scaling* dengan menggunakan metode *bicubic spline interpolation.* Pada contoh ini digunakan *scale* X : 2, Y: 2 dan hasilnya sebagai berikut. Kiri merupakan *image* sebelum di *rescaling* dan kanan merupakan *image* yang sudah dilakukan *rescaling.*

BAB V
KESIMPULAN

5.1 Kesimpulan

Kami telah berhasil membuat *package* dalam Bahasa Java. *Package* tersebut dapat menemukan solusi SPL dengan metode eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer. Pustaka tersebut juga dapat menghitung determinan matriks dengan Reduksi baris (Gauss-Jordan) dan dengan ekspansi kofaktor. *Package* tersebut juga dapat menghitung matriks balikan dengan metode matriks adjoin dan matriks identitas. *Package* tersebut juga dapat menghitung interpolasi polinom, Interpolasi Bikubik, Regresi Linier Berganda, Regresi Kuadratik berganda, dan juga *Image Resizing*.

5.2 Saran

Masih terdapat perhitungan yang kurang presisi, karena perhitungan yang kompleks, dan sebaiknya ditambahkan lagi fungsi untuk menghitung angka desimal besar dengan presisi.

Selanjutnya, untuk spesifikasi masih terdapat kesalahan pada referensi, sehingga kami harap untuk ke depannya, bisa dievaluasi lagi sumber referensinya.

5.3 Refleksi

Melalui tugas besar algeo ini, kami bersyukur bisa bekerja sama dengan baik, bisa lebih saling mengenal satu sama lain. Tugas ini menjadi Langkah awal kami untuk menjadi pribadi yang dapat bekerja dalam tim lebih baik. Terakhir, kami menjadi lebih dekat dengan Yang Maha Kuasa.

LAMPIRAN

Video Demo : https://linktr.ee/Zirach